



PRESENTED BY

Qualcomm

---

*FIRST® RISE™ powered by Star Wars: Force for Change*

2019-2020 *FIRST® Tech Challenge*

# Blocks Programming Guide



## Sponsor Thank You

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

**FIRST® Tech Challenge  
Season Presenting Sponsor**

# Qualcomm

**FIRST® Tech Challenge  
Official Program Sponsor**



## Collins Aerospace

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



## Volunteer Thank You

Thank you for taking the time to volunteer for a *FIRST*® Tech Challenge event. *FIRST*® and *FIRST*® Tech Challenge rely heavily on volunteers to ensure events run smoothly and are a fun experience for teams and their families, which could not happen without people like you. With over 5,500 teams competing yearly, your dedication and commitment are essential to the success of each event and the *FIRST* Tech Challenge program. Thank you for your time and effort in supporting the mission of *FIRST*!

Revision History		
Revision	Date	Description
1.0	11/3/2016	Initial Release
2.0	06/09/2017	Updated with support for REV Expansion Hub. New tabular format.
2.1	06/16/2017	Some of the screenshots were missing negation operator.
3.0	06/05/2019	Updated with Rev Control Hub content and Microsoft Manual of Style

## Contents

1	Introduction .....	6
	<b>Autonomous vs. Driver-Controlled</b> .....	6
1.1	Point-to-Point Control System.....	6
1.2	REV Robotics Expansion Hub .....	7
1.3	REV Robotics Control Hub .....	7
2	Required Materials .....	8
3	Setting Up Your Smartphones.....	11
3.1	What Needs to be Configured for My Control System .....	11
3.2	Renaming Your Smartphones.....	12
3.3	Installing the <i>FIRST</i> Tech Challenge Apps.....	16
3.4	Placing Phones into Airplane Mode with Wi-Fi On.....	20
3.5	Pairing the Driver Station to the Robot Controller .....	21
4	Connecting Devices to an Expansion Hub.....	30
4.1	Connecting 12V Power to the Hub .....	30
4.2	Connecting a Motor to the Hub .....	32
4.3	Connecting a Servo to the Hub.....	33
4.4	Connecting a Color-Distance Sensor to the Hub .....	34
4.5	Connecting a Touch Sensor to the Hub .....	35
5	Configuring Your Hardware on the Robot Controller .....	36
5.1	Connecting an Android Smartphone to an Expansion Hub .....	36
5.2	Creating a Configuration File Using the Driver Station .....	38
5.3	Configuring the DC Motor .....	41
5.4	Configuring a Servo .....	43

5.5	Configuring a Color Distance Sensor .....	44
5.6	Configuring a Digital Touch Sensor.....	47
5.7	Saving the Configuration Information .....	49
6	Writing an Op Mode.....	51
6.1	What's an Op Mode? .....	51
6.2	The FTC Blocks Programming Tool .....	51
6.3	Installing a Javascript-Enabled Browser.....	53
6.4	Connecting Your Laptop to the Program and Manage Network .....	54
6.5	Troubleshooting Your Wireless Connection .....	57
6.6	Creating Your First Op Mode .....	58
6.7	Examining the Structure of Your Op Mode.....	60
6.8	Modifying the Op Mode to Control a DC Motor .....	61
6.9	Inserting Telemetry Statements .....	66
6.10	Saving Your Op Mode.....	69
6.11	Exiting Programming Mode.....	69
7	Running Your Op Mode.....	70
8	Controlling a Servo Motor with an Op Mode.....	73
8.1	What is a Servo Motor?.....	73
8.2	Modifying Your Op Mode to Control a Servo .....	73
9	Using Sensors .....	82
9.1	Color-Distance Sensor .....	82
9.2	Touch Sensor .....	86
10	Troubleshooting .....	89
10.1	Manually Connecting to the Blocks Programming Mode Wi-Fi Network .....	89
10.2	Troubleshooting Tips.....	91
10.2.1	Cannot See the Blocks Programming Mode Wireless Network .....	91
10.2.2	"Save project failed. Error code 0." .....	92
10.2.3	Op Mode Blocks Are Missing.....	92
10.2.4	Driver Station Appears Unresponsive .....	94
10.2.5	Warning: problem communicating... .....	94
10.2.6	Additional Disconnect Incidents with Motorola E4, G5 and G5 Plus Phones.....	95
10.2.7	Wi-Fi Blocker at Venue (Listed 01/2019).....	97

## 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).

### **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](#).

Learn more about the roles of volunteers on our [Volunteer Resources page](#), “Volunteer Role Description”.

## 1 Introduction

This document shows how to set up, configure, and program the control system used for the *FIRST* Tech Challenge competition. The basic examples in this document use the REV Robotics Expansion Hub and the REV Robotics Control Hub as the input/output module. For detailed information about the REV Robotics Expansion Hub, refer to the *REV Robotics Expansion Hub Guide* which is available from the REV Robotics website (<http://www.revrobotics.com/>).

**Important Note:** In the 2019-2020 *FIRST* Tech Challenge season, the Control Hub will only be allowed for use in competition by *FIRST* Tech Challenge teams based in test pilot regions and will be illegal for competition use outside of test pilot regions. The only exception to this rule is in-school classes who purchased Class Pack robot kits and wish to extend that experience, create a *FIRST* Tech Challenge team, and compete. Those teams may use the Control Hubs provided in the Class Pack robot kits in competition in any region but should not expect specific FTA support should there be issues specific to the Control Hub. The Control Hub is being piloted in competition for the 2019-2020 season and is expected to become available to all teams next season.

## 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.

### 1.1 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.



Figure 1 – The control system has a *Driver Station* connected wirelessly to a *Robot Controller*.

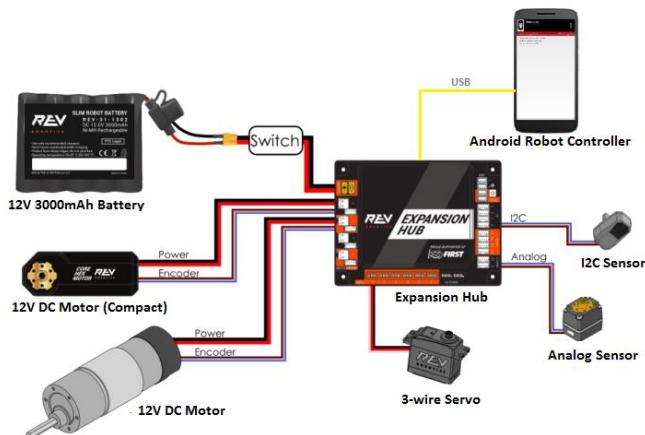
The *Robot Controller* acts as the “brains” of the robot and is mounted on the frame of the robot. There are two hardware options currently being used: REV Robotics Expansion Hub or the REV Robotics Control Hub.

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.

## 1.2 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.

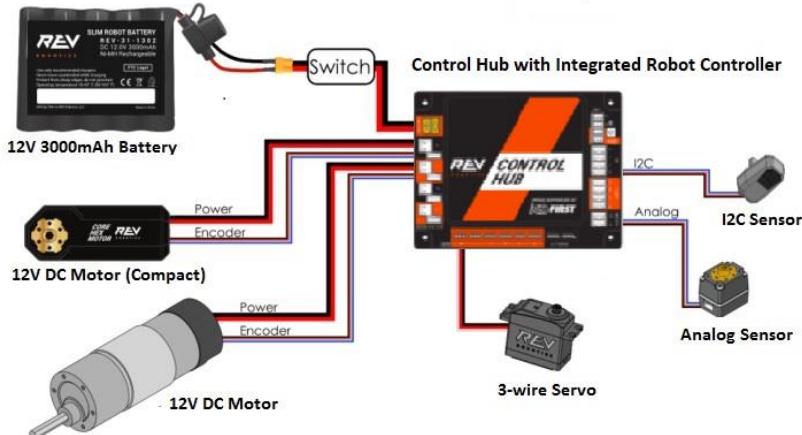


## 1.3 REV Robotics Control Hub

For the 2019-2020 season, teams in select test pilot regions will have the option of using the 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.



## 2 Required Materials

To follow along with the examples in this document, you will need the following items:

Two (2) FIRST-approved Android smartphones<sup>1</sup>



Or

One (1) Control Hub and one (1) FIRST-approved\* Android smartphone.



<sup>1</sup> Consult the official FIRST Tech Challenge Game Manual, Part 1 for a list of approved devices.

A Google account (available for free from Google) to access the Google Play store



Wireless Internet access

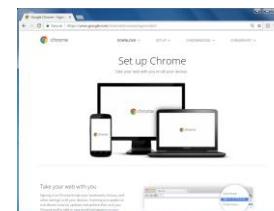


Laptop with Microsoft Windows 7, 8 or 10 and Wi-Fi capability.

NOTE: Your laptop should have the most current service packs and system updates from Microsoft.



Javascript-enabled web browser (Google Chrome is the recommended browser).



REV Robotics Expansion Hub (REV-31-1153), except Control Hub users.



REV Robotics Switch, Cable, and Bracket (REV-31-1387)



REV Robotics Tamiya to XT30 Adapter (REV-31-1382)



*FIRST*-approved 12V Battery (such as Tetrix W39057 or REV Robotics REV-31-1302)



*FIRST*-approved 12V DC Motor (such as Tetrix W39530, with power cable W41352)<sup>2</sup>



REV Robotics Anderson to JST VH Cable (REV-31-1381)



180-Degree Standard Scale Servo (such as Hitec HS-485HB)



REV Robotics Color Sensor with 4-Pin JST PH Cable (REV-31-1154)

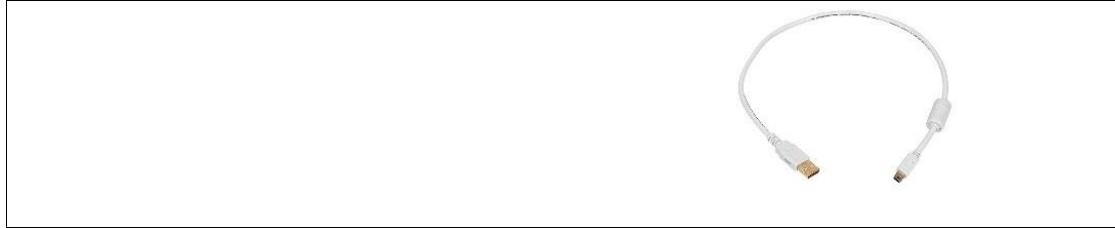


REV Robotics Touch sensor with 4-Pin JST PH Cable (REV-31-1425)



USB Type A male to type mini-B male cable

<sup>2</sup> Note that for the examples listed in this document, it is recommended that the user builds a simple structure using a compatible build kit (such as TETRIX Max) to properly secure the DC motor and prevent it from rolling about uncontrollably while running the sample op modes.



Two (2) micro USB OTG adapters



Logitech F310 USB Gamepad



### 3 Setting Up Your Smartphones

#### 3.1 What Needs to be Configured for My Control System

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>.

## 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.



### 3.2 Renaming Your Smartphones

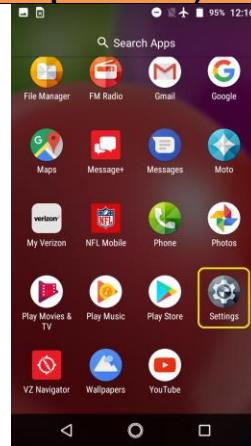
The official rules of the FIRST Tech Challenge (see <RS01>) require that you change the Wi-Fi name of your smartphones to include your team number and “-RC” if the phone is a Robot Controller or “-DS” if it is a Driver Station. A team can insert an additional dash and a letter (“A”, “B”, “C”, etc.) if 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.

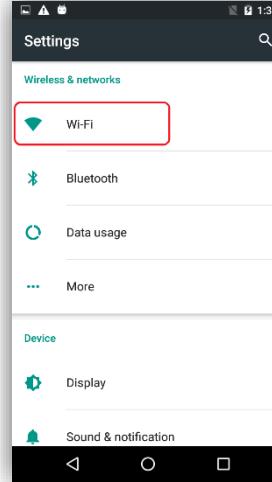
## Renaming Your Smartphone (Time Needed to Complete Task: 5 Minutes per Phone)

**Step 1:**

Select **Settings** icon to display the Android Settings screen.

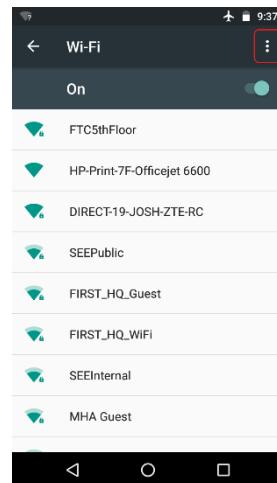
**Step 2:**

Select **Wi-Fi** to launch the Wi-Fi screen.

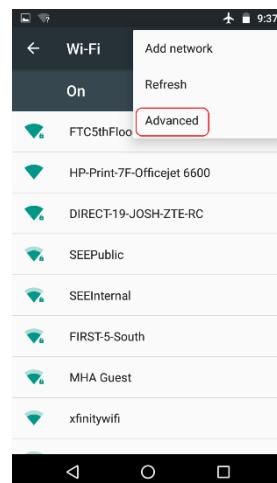


**Step 3:**

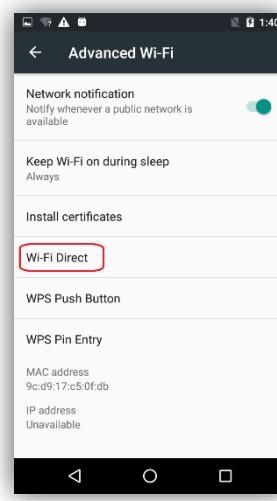
Touch the three vertical dots in upper right-hand corner to display a pop-up menu.

**Step 4:**

Select **Advanced** from the pop-up menu.

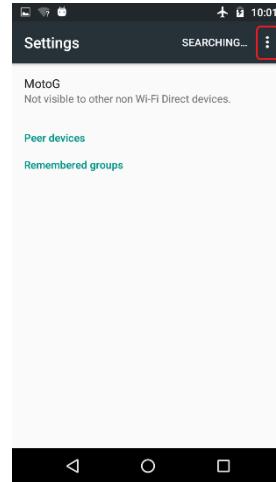
**Step 5:**

Select **Wi-Fi Direct** from the Advanced WiFi screen.

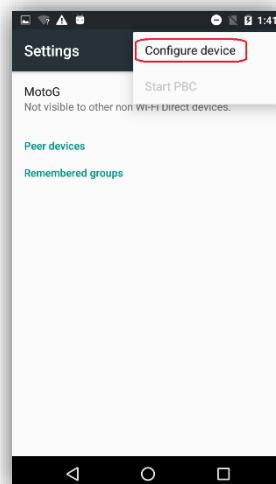


**Step 6:**

Touch the three vertical dots to display the pop-up menu.

**Step 7:**

Select **Configure Device** from the pop-up menu.



**Step 8:**

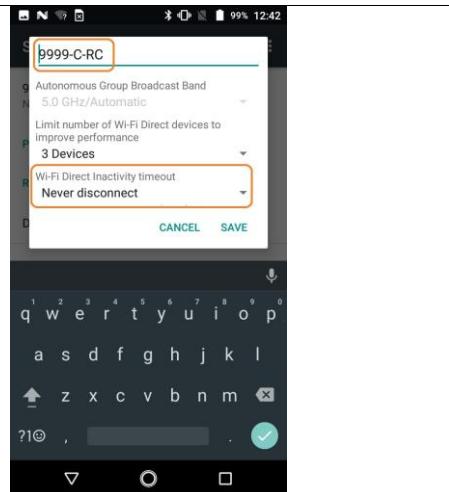
Use touch pad to enter the new name of device.  
 If the device will be a Robot Controller, specify your team number and "-RC".  
 If the device will be a Driver Station, specify your team number and "-DS".

You can also set the Wi-Fi Direct inactivity timeout to "Never disconnect" and then hit the  
**SAVE** button.

**NOTE:** 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.

**Step 9:**

After renaming the phone, power cycle the device.



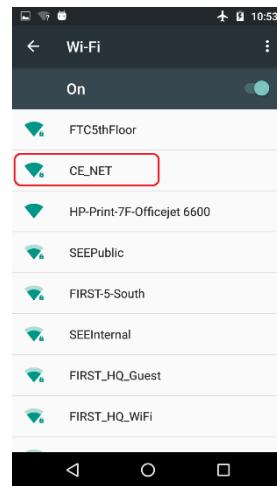
### 3.3 Installing the FIRST Tech Challenge Apps

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.

#### Installing the FIRST Tech Challenge Apps (Time Needed to Complete Task: 7.5 Minutes per Phone)

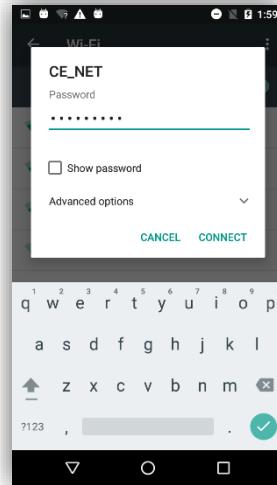
**Step 1:**

From the Android Wi-Fi screen, select the name of your wireless network ("CE\_NET" in this example) to log in.

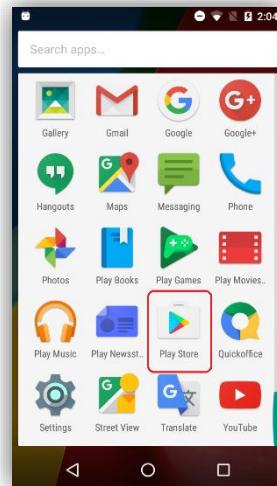


**Step 2:**

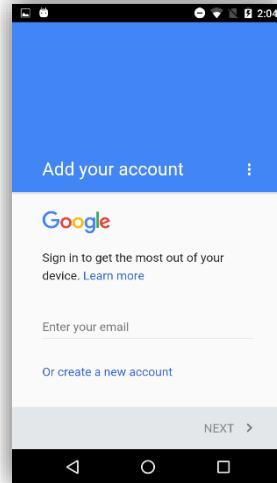
Specify the password using the touch keypad and hit **CONNECT** to connect to this wireless network.

**Step 3:**

Select the Google Play Store icon to launch the app.

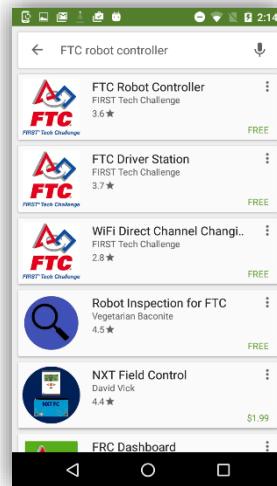
**Step 4:**

If you haven't signed into your Google account yet, follow the onscreen instructions to log into your Google account or create a new account.



**Step 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.

**Step 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.

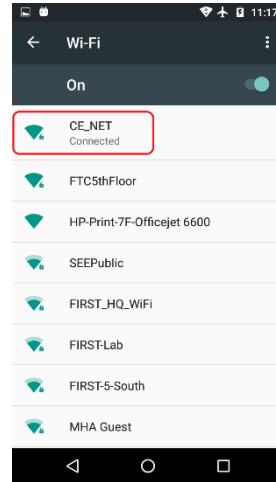
**Important note:** When you install the FTC apps, **only install one FTC app** (FTC Robot Controller or FTC Driver Station) **per phone**. You should avoid installing both apps onto the same phone. Doing so can cause Wi-Fi connection problems. You should only install the FTC Robot Controller app onto the phone that will be the Robot Controller and the FTC Driver Station app onto the phone that will be the Driver Station.



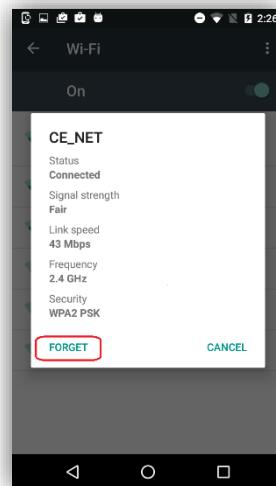
**Step 7:**

After you have successfully installed the app, you should forget the external wireless network on your phone.

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.

**Step 8:**

Select **FORGET** button to forget the wireless network.



### 3.4 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.

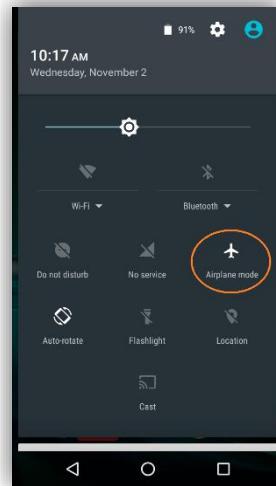
#### Placing Phones into Airplane Mode with Wi-Fi On (Time Needed to Complete Task: 2.5 Minutes per Phone)

##### Step 1:

On the main Android screen of each smartphone, slide from the top of the screen down towards the bottom to display the quick configuration screen.

**NOTE:** 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.

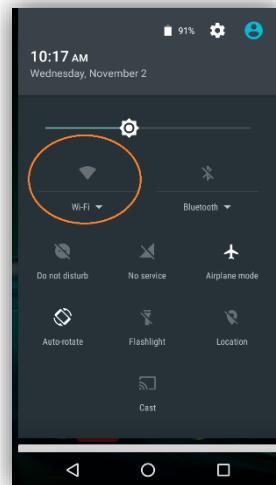
Activate Airplane mode icon (which is shaped like an airplane).



##### Step 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.

**NOTE:** Touch the “Wi-Fi” icon on the quick configuration screen to turn the Wi-Fi radio back on.



### 3.5 Pairing the Driver Station to the Robot Controller

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.

#### 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.”

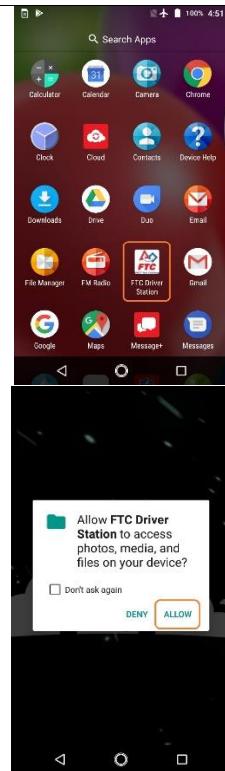
**NOTE:** Control Hub does not have its own internal battery. Before you can connect a Driver Station to the Control Hub, you must connect the Control Hub to a 12V battery.

<b>Pairing the Driver Station to the Control Hub</b> <b>(Time Needed to Complete Task: 10 minutes)</b>	
<p><b>Step 1:</b></p> <p>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><b>Step 2:</b></p> <p>It takes approximately 40 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><b>NOTE:</b> The light blinks blue every ~5 seconds to indicate that the Control Hub is healthy.</p>	

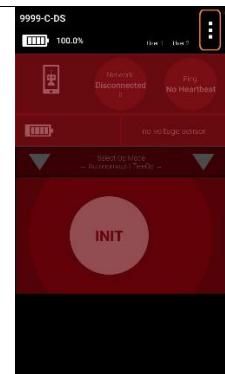
**Step 3:**

On the Driver Station device, select the **FTC Driver Station** icon.

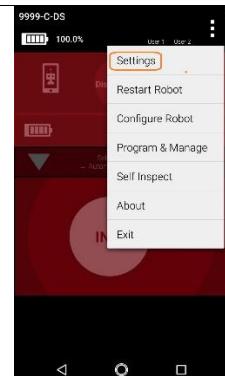
**NOTE:** The first time you launch the app your Android device might prompt you for permissions that the app will need to run properly. Whenever prompted, press **Allow** to grant the requested permission.

**Step 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.

**Step 5:**

Select **Settings** from the pop-up menu.

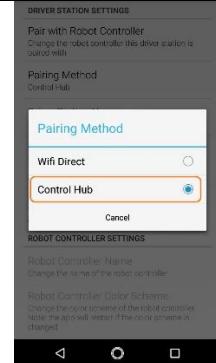


**Step 6:**

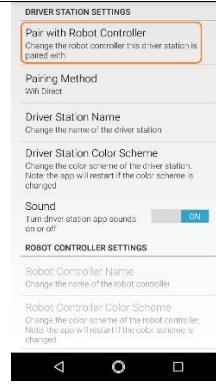
From the **Settings** screen, select **Pairing Method** to launch the **Pairing Method** screen.

**Step 7:**

Touch the words **Control Hub** to indicate that this Driver Station will be pairing with a Control Hub.

**Step 8:**

From the **Settings** screen, select **Pair with Robot Controller** to launch the **Pair with Robot Controller** screen.

**Step 9:**

From **Pair with Robot Controller** screen, press the **Wifi Settings** button to launch the device's Wifi Settings screen.



**Step 10:**

Find and select the name of your Control Hub's wireless network from the list of available WiFi networks.

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).

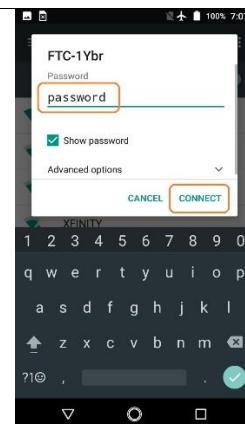
The default network name should be listed on a sticker attached to the bottom side of the Control Hub.

**Step 11:**

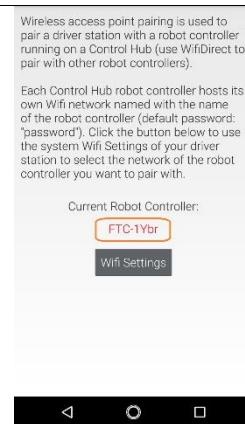
When prompted, specify the password for the Control Hub's WiFi network and press **Connect** to connect to the Hub.

**NOTE:** The default password for the Control Hub network is "password".

**NOTE:** When you connect to the Control Hub's WiFi network successfully, the Driver Station will not have access to the Internet.

**Step 12:**

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 to return to the Settings screen. Then press the back-arrow key again to return to the main Driver Station screen.



**Step 13:**

Verify that the Driver Station screen is now connected to the Control Hub.

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.

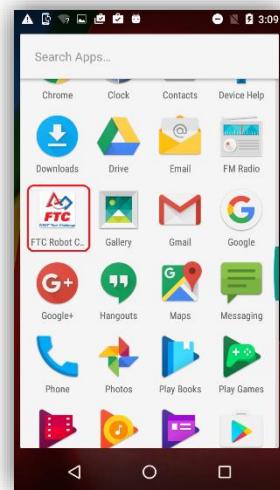
**Users with Two Android Smartphones**

**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."

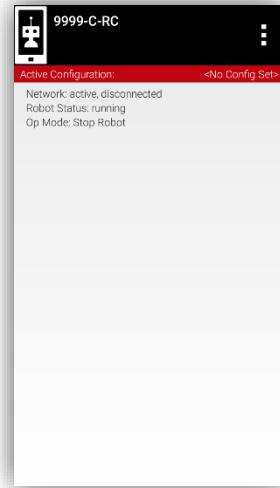
**Pairing the Driver Station to the Robot Controller: Expansion Hub Users  
(Time Needed to Complete Task: 10 minutes)**
**Step 1:**

On the main Robot Controller smartphone, select the FTC Robot Controller icon to open the app.



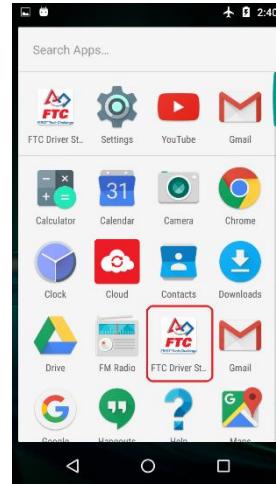
## Step 2:

Verify that the Robot Controller app is running.  
The **Robot Status** field should read “running” if it is working properly.



**Step 3:**

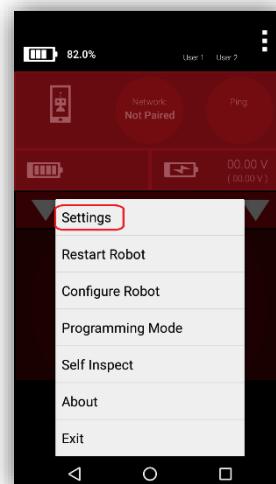
On your Driver Station phone, find the **FTC Driver Station** icon. Tap on the icon to launch the Driver Station app.

**Step 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.

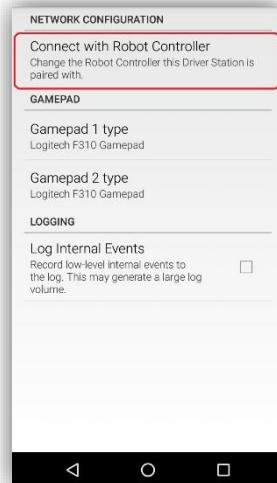
**Step 5:**

Select **Settings** from the pop-up menu.



**Step 6:**

From the Settings screen, select **Connect with Robot Controller** to launch the Connect with Robot Controller screen.

**Step 7:**

Find the name of your Robot Controller from the list and select it.

After you have made your selection, use the back-arrow key to return to the Settings screen.

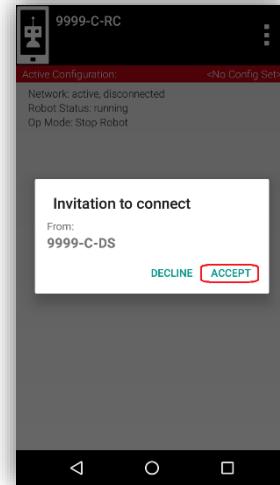
Then press the back-arrow key one more time to return to the main Driver Station screen.



**Step 8:**

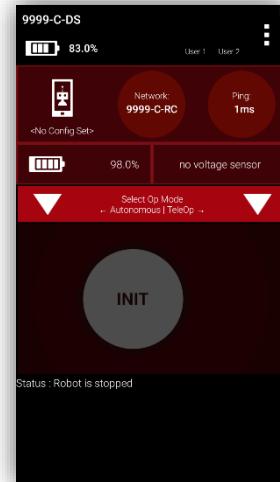
When the Driver Station returns to its main screen, the first time you attempt to connect to the Robot Controller a prompt should appear on the Robot Controller screen.

Select **ACCEPT** the connection request from the Driver Station.

**Step 9:**

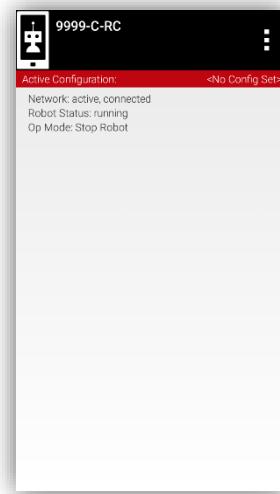
Verify that the Driver Station screen has changed and that it now indicates that it is connected to the Robot Controller.

The name of the Robot Controller's remote network ("9999-C-RC" in this example) should be displayed in the **Network** field on the Driver Station.

**Step 10:**

Verify that the Robot Controller screen has changed and that it now indicates that it is connected to the Driver Station.

The **Network** status should read "active, connected" on the Robot Controller's main screen.



## 4 Connecting Devices to an 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.

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.

### 4.1 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 the power on/off.

<b>Connecting 12V Power to the Hub</b> <b>(Time Needed to Complete Task: 5 minutes)</b>	
<p><b>Step 1:</b>            If your 12V battery has a Tamiya-style connector, connect the Tamiya to XT30 adapter cable to the matching end of the switch cable.</p> <p>Do not connect the 12V battery to the Tamiya adapter yet. We will connect the battery during a later step.</p>	
<p><b>Step 2:</b>            Connect the other end of the switch cable to a matching XT30 port on the Hub.</p>	

**Step 3:**

Verify that the switch is in the OFF position.

**Step 4:**

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

**Step 5:**

Turn on the switch and verify that the Hub is drawing power from the battery.

**NOTE:** Hub's LED should be illuminated (blue LED in image).

**Step 6:**

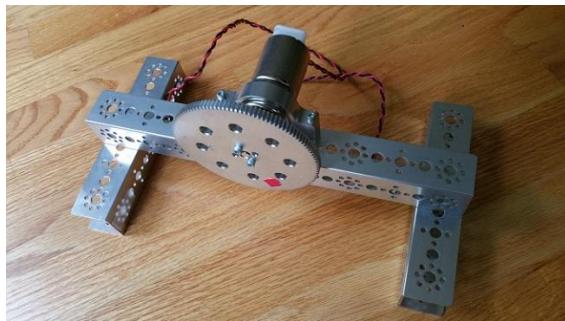
Turn off the switch and verify that the Expansion Hub is off.

**NOTE:** The Expansion Hub's LED should not be illuminated.



## 4.2 Connecting a Motor to the Hub

The Hub can drive up to four (4) 12V DC motors. 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 is used to connect the Anderson Powerpole connectors to the Expansion Hub motor port, see [FIRST Tech Challenge Robot Wiring Guide](#) for more information).



*Figure 2 - Motor test rig made from Tetrix build components.*

For the examples in this document, 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 Tetrix motor installed in a rig built with a Tetrix motor mount and some Tetrix C-channels. A gear was mounted on the motor shaft to make it easier for the user to see the rotation of the shaft.

### Connecting a 12V DC Motor to the Hub (Time Needed to Complete Task: 2.5 minutes)

#### Step 1:

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



#### Step 2:

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



#### 4.3 Connecting a Servo to the Hub

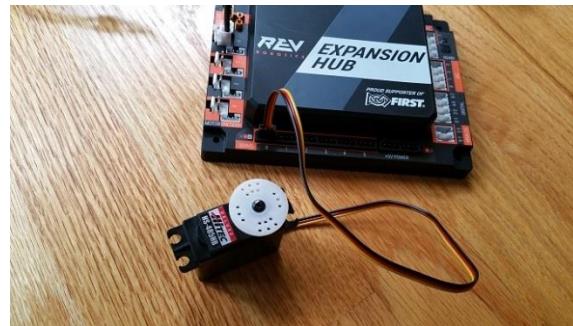
The REV Robotics Hub has 6 built-in servo ports. The servo ports accept the standard 3-wire header-style connectors commonly found on servos. NOTE: The ground pin is on the left side of the servo port.

##### Connecting a Servo to the Hub (Time Needed to Complete Task: 2.5 minutes)

###### Step 1:

Connect the servo cable to the servo port labeled “0” on the Hub.

NOTE: The ground pin is on the left side of the servo port.



###### Step 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).



#### 4.4 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.

##### Connecting a Color-Distance Sensor to the Hub (Time Needed to Complete Task: 2.5 minutes)

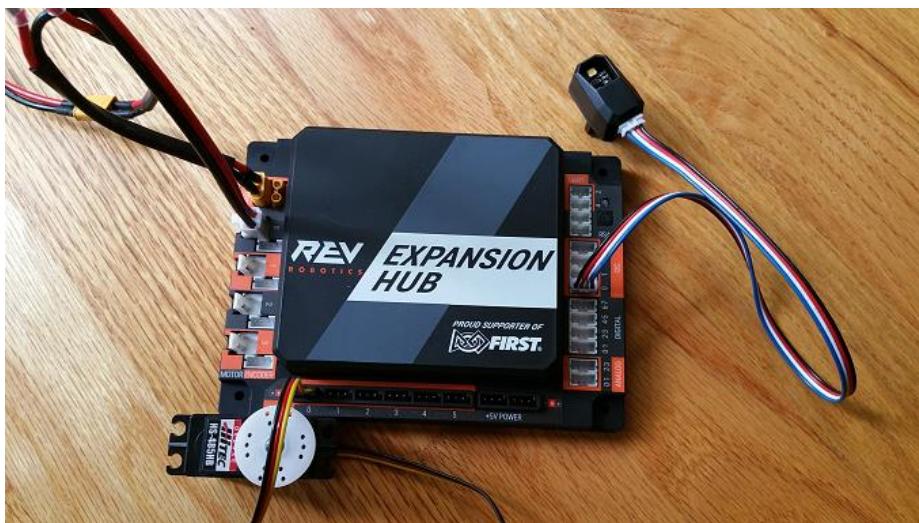
###### Step 1:

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



###### Step 2:

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



## 4.5 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 the Hub. You will connect a REV Robotics Touch sensor to one of the digital I/O ports.

NOTE: 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 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.

### Connecting a Touch Sensor to the Hub (Time Needed to Complete Task: 2.5 minutes)

#### Step 1:

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



#### Step 2:

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



## 5 Configuring Your Hardware on the Robot Controller

Before you can communicate with the motor, servo and sensors that are connected to the Expansion or Control 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 Hub's external ports.

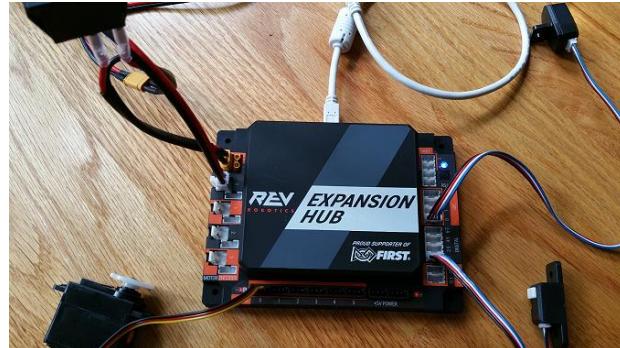
### 5.1 Connecting an Android Smartphone to an Expansion Hub

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.

#### Creating a Configuration File on the Robot Controller (Time Needed to Complete Task: 20 minutes) Connecting the Phone to the Expansion Hub

##### Step 1:

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



##### Step 2:

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



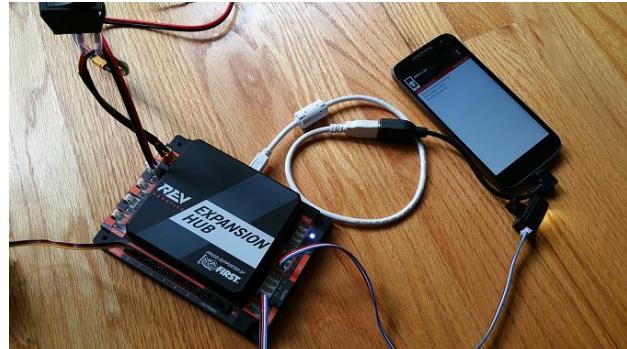
**Step 3:**

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

**Step 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.

**NOTE:** 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.

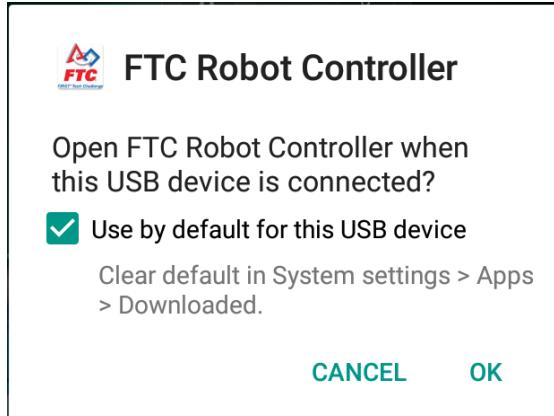
**Step 5:**

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.

**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.



## 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.

### 5.2 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 an Android smartphone Robot Controller.

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

##### Step 1:

Touch the three vertical dots in the upper right-hand corner of the Driver Station.

This will launch a pop-up menu.



##### Step 2:

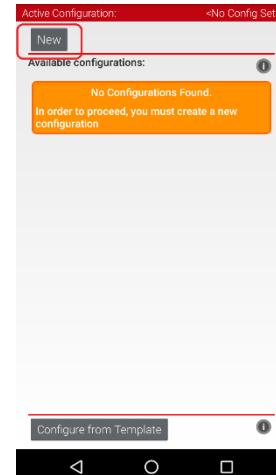
Select **Configure Robot** from the pop-up menu to display the Configuration screen.



**Step 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.

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

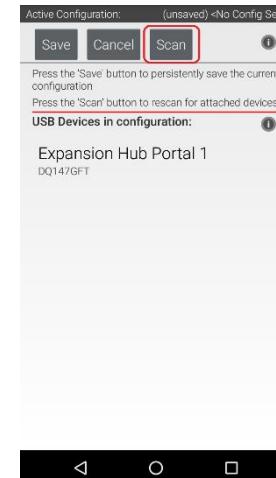
**Step 4:**

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.

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.

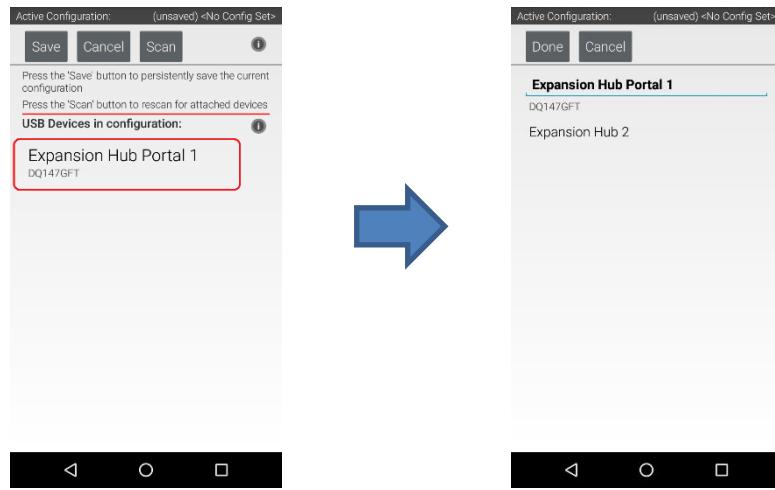
Your Expansion Hub is listed as a *Portal* because it is directly connected to the Robot Controller phone through the USB cable.

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



**Step 5:**

Touch the Portal listing **Expansion Hub Portal 1** 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**

**Step 6:**

Touch the Expansion Hub listing **Expansion Hub 2** to display the Input/Output ports for that device.



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

### 5.3 Configuring the DC Motor

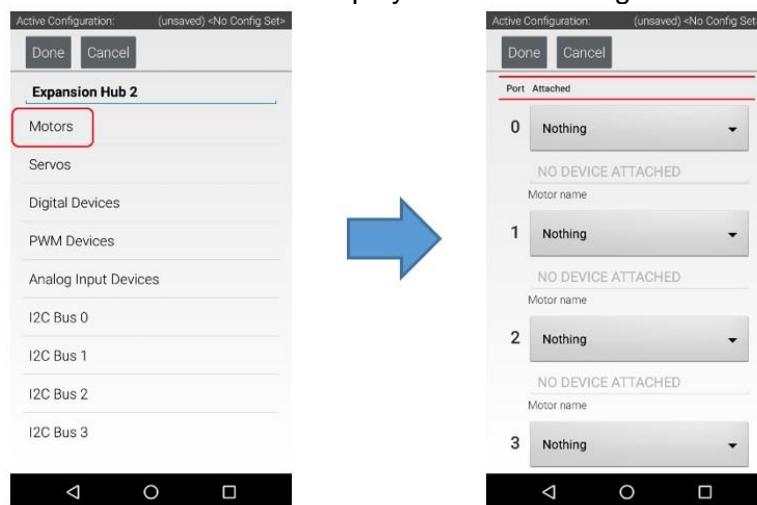
Now that you've created a file, you will need to add a DC Motor to the configuration file.

**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 the DC Motor

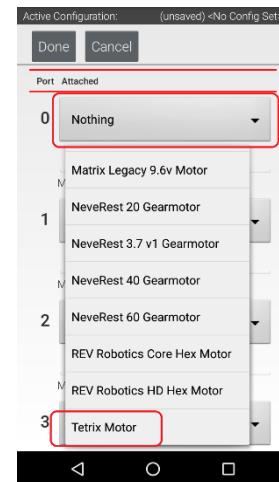
##### Step 1:

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



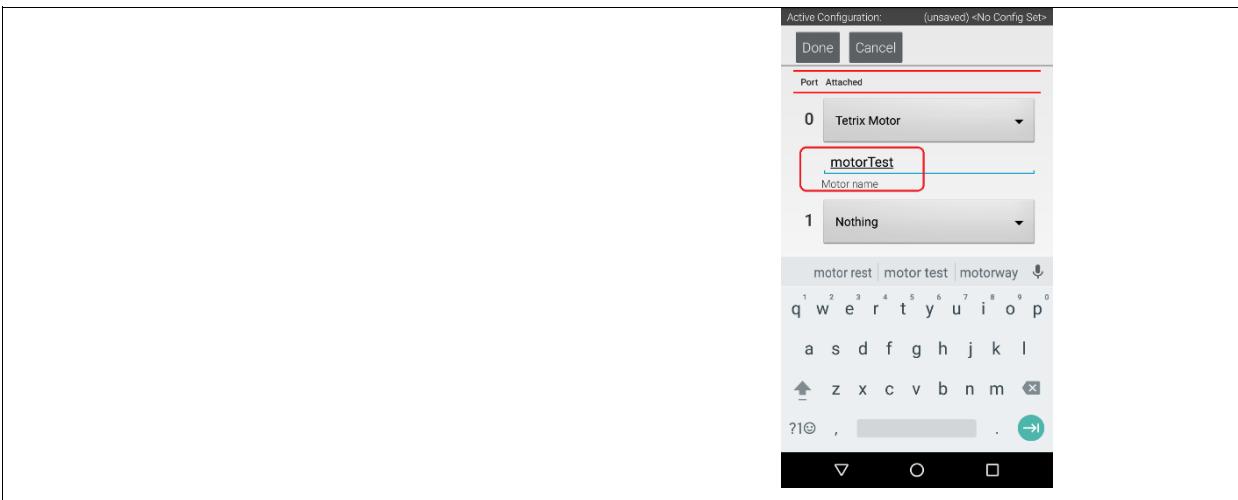
##### Step 2:

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).

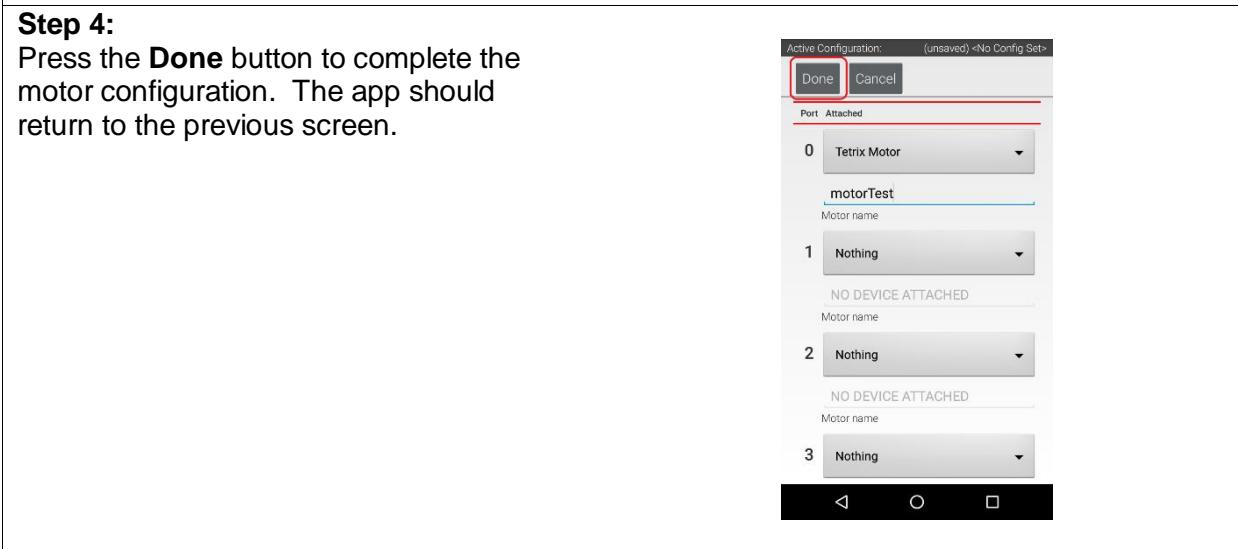


##### Step 3:

Specify a name for your motor (**motorTest** in this example).

**Step 4:**

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



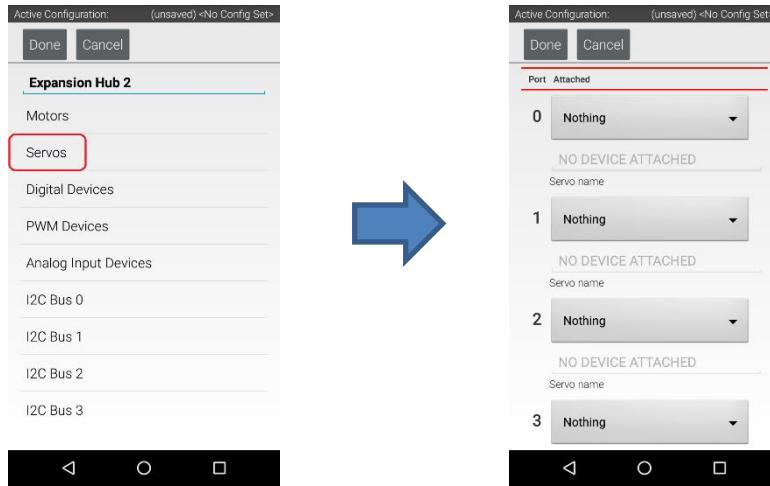
## 5.4 Configuring a Servo

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

### Configuring the Servo

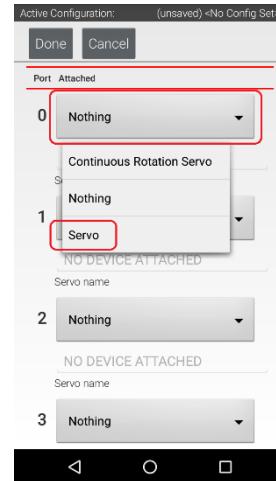
#### Step 1:

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



#### Step 2:

Use the dropdown control to select **Servo** as the servo type for port #0.

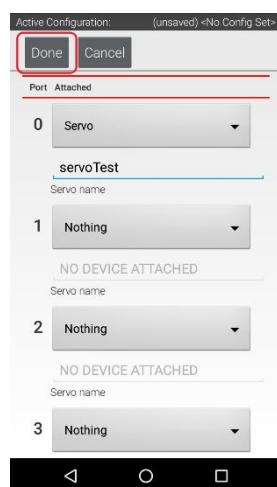


**Step 3:**

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

**Step 4:**

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



## 5.5 Configuring a Color Distance Sensor

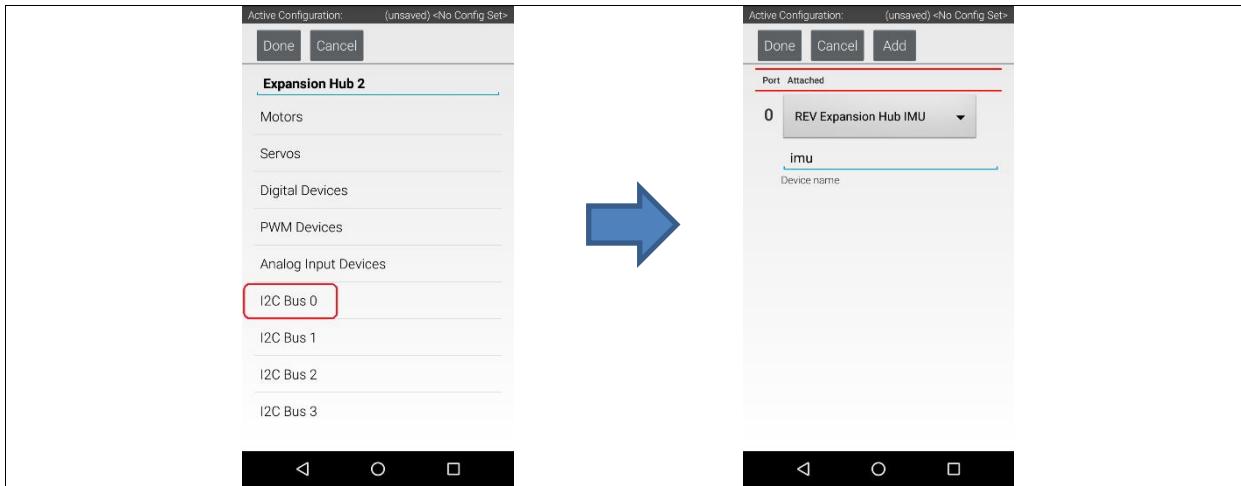
The REV Robotics Color Distance Sensor is an I2C sensor. It combines two sensor functions into a single device. It is a color sensor, that can determine the color of an object. It is also a distance or range sensor, that can be used to measure short range distances.

NOTE: In this tutorial, the word "distance" is used interchangeably with the word "range".

### Configuring the Color Sensor

**Step 1:**

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



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.

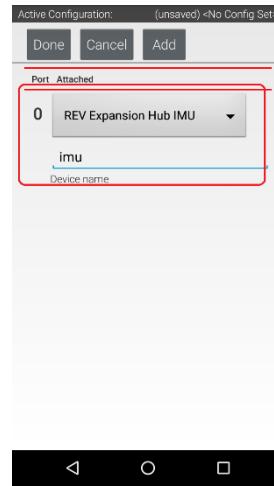
### Step 2:

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.

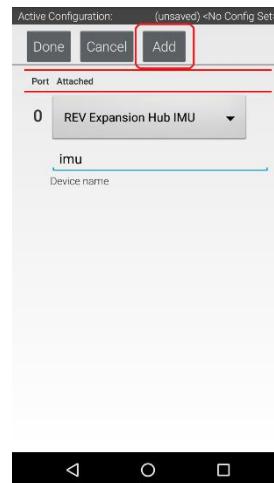
The built-in IMU is internally connected to the I2C Bus 0 on each Expansion Hub. Whenever you configure an Expansion Hub using the Robot Controller, the app automatically configures the IMU for I2C Bus 0.

**NOTE:** You will need to add another I2C device for this bus to be able to configure the color sensor.

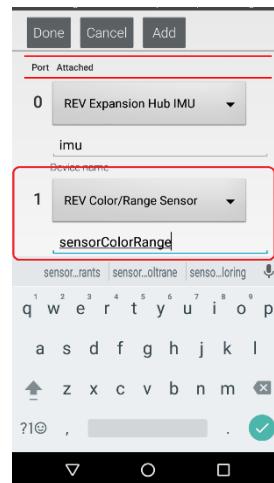


**Step 3:**

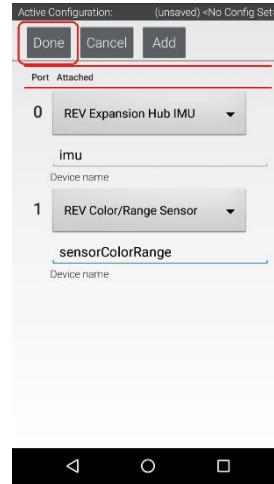
Press the **Add** button to add another I2C device to this bus.

**Step 4:**

Select **REV Color/Range Sensor** from the dropdown menu for this new device. Name this device “sensorColorRange”.

**Step 5:**

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



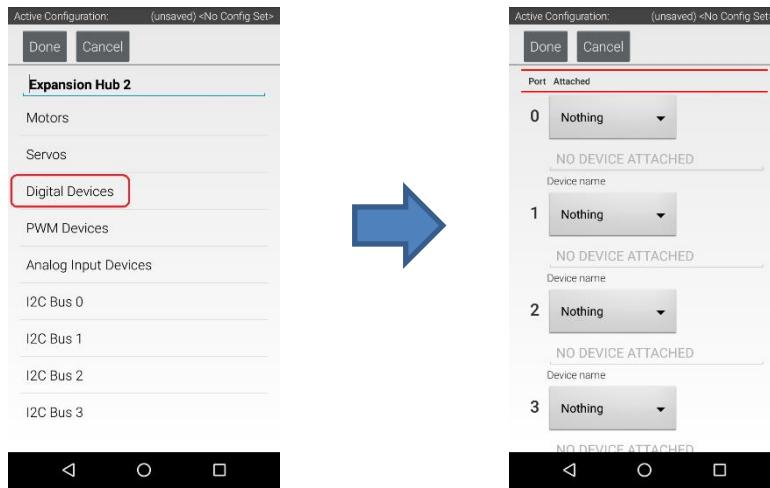
## 5.6 Configuring a Digital Touch Sensor

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 the Digital Touch Sensor

#### Step 1:

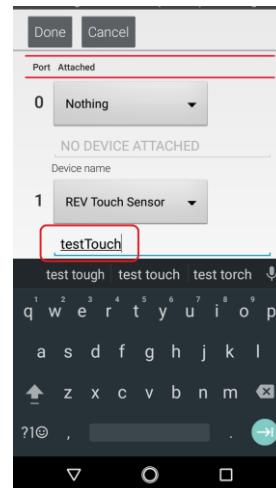
Touch the words **Digital Devices** on the screen to launch the Digital I/O configuration screen.



#### Step 2:

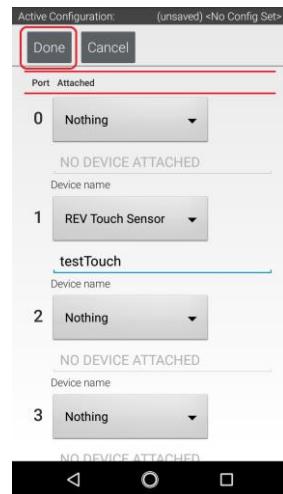
Use the touch screen to add a **Digital Device** for port #1 and name the device “testTouch”.

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.



**Step 3:**

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



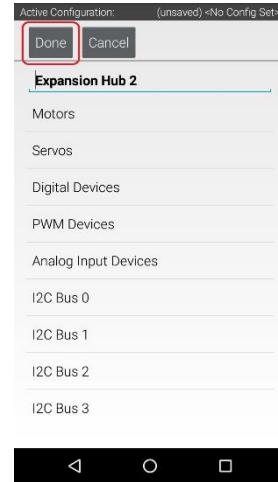
## 5.7 Saving the Configuration Information

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

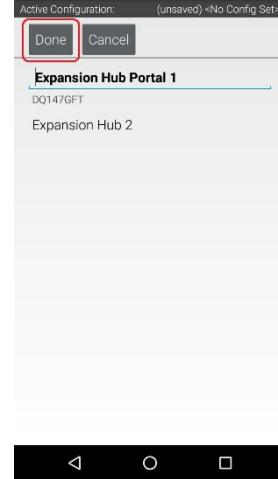
#### Step 1:

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



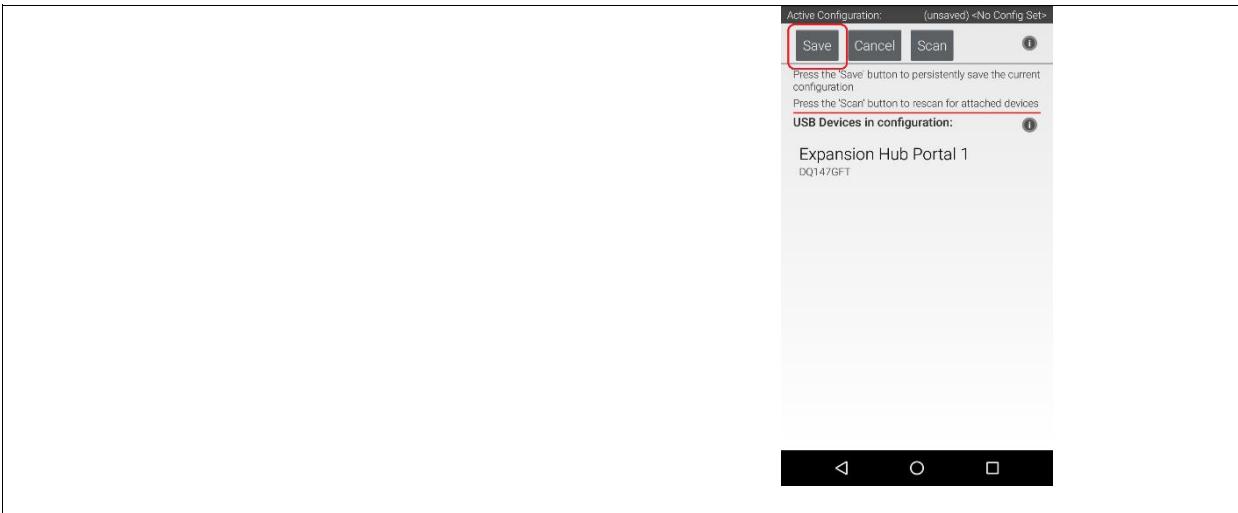
#### Step 2:

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

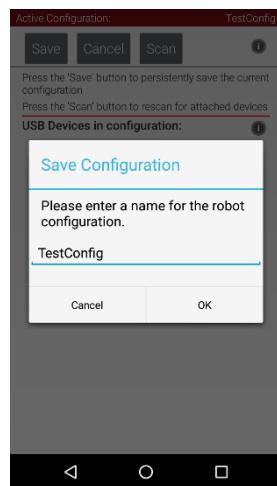


#### Step 3:

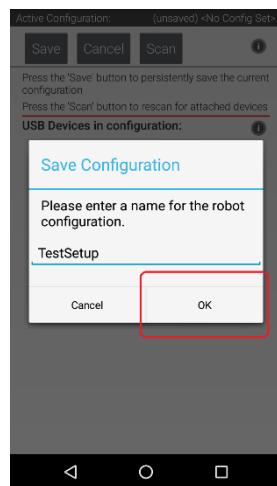
Press the **Save** button.

**Step 4:**

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

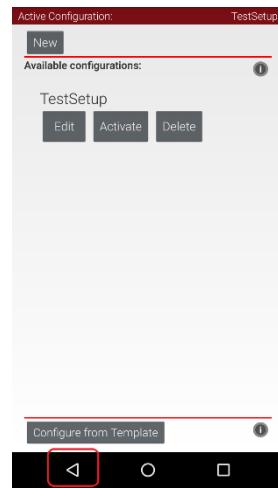
**Step 5:**

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

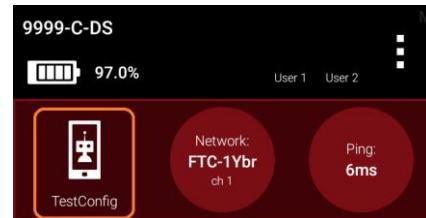


**Step 6:**

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

**Step 7:**

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



## 6 Writing an Op Mode

### 6.1 What's an Op Mode?

During a typical FIRST Tech Challenge match, a team's robot must perform a variety of tasks to score points. For example, a team might want their robot to follow a white line on the competition floor, then score a game element (such as a ball) into a goal autonomously during a match. Teams write programs called “op modes” (which stand for “operational modes”) to specify the behavior for their robot. These op modes run on the Robot Controller after being selected on the Driver Station.

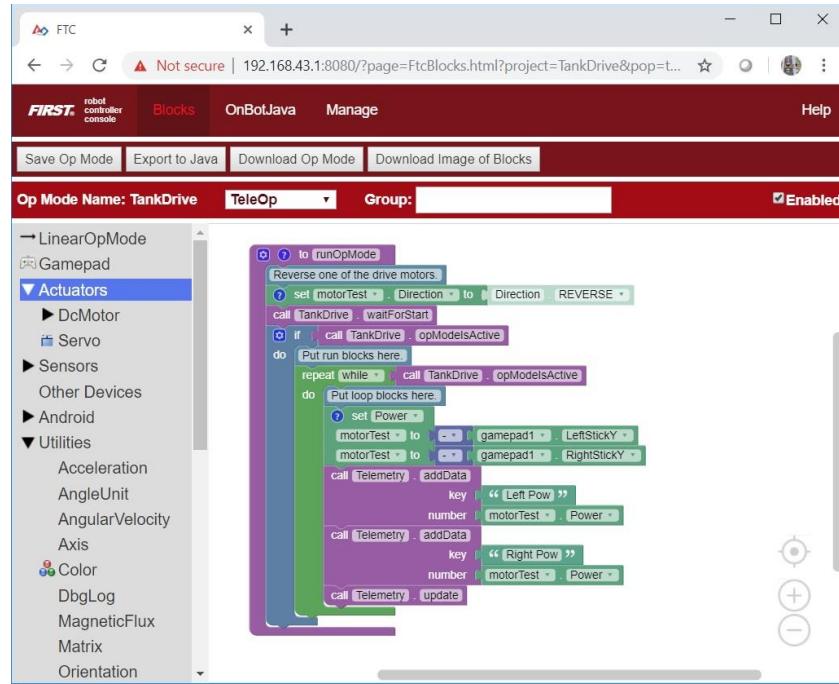
Teams who are participating in FIRST Tech Challenge have a variety of programming tools that they can use to create their own op modes. Teams can use a visual (“drag and drop”) programming tool called the *FTC Blocks Programming Tool* to create their op modes. Teams can also use a text-based Java tool 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.

### 6.2 The FTC Blocks Programming Tool

The *FTC Blocks Programming Tool* is a user-friendly programming tool that is served up by the Robot Controller. A user can create custom op modes for their robot using this tool, then save these op modes directly onto the Robot Controller.

Users drag and drop jigsaw-shaped programming blocks onto a design “canvas” and arrange these blocks to create the program logic for their op mode.

The FTC Blocks Programming Tool is powered by Google's *Blockly* software and was developed with support from Google.



*Figure 3 - Users arrange jigsaw-shaped programming blocks to create the logic for their op modes.*

The examples in this document use a Windows laptop computer to connect to the Robot Controller. This Windows laptop computer has a Javascript-enabled web browser (Google Chrome) installed that is used to access the FTC Blocks Programming Tool.



NOTE: The process used to create and edit an op mode is identical if you are using a Control Hub as your Robot Controller.



Laptop



WiFi Connection



Control Hub

**NOTE:** If you prefer, you can use an alternate device, such as an Apple Mac laptop, or a Chromebook, instead of a Windows computer to access the FTC Blocks Programming Tool.

### 6.3 Installing a Javascript-Enabled Browser

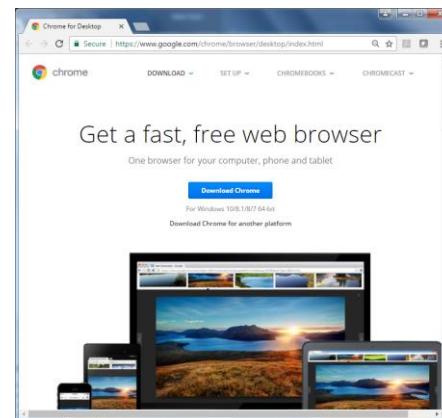
To be able to program using the blocks programming mode server, your laptop will need a Javascript-enabled browser. FIRST recommends the use of Google Chrome as the Javascript-enabled browser for the FTC Blocks Programming Tool. The FTC Blocks Programming Tool has been thoroughly tested using Google Chrome.

#### Installing Javascript-Enabled Browser (Time Needed to Complete Task: 15 minutes)

##### Step 1:

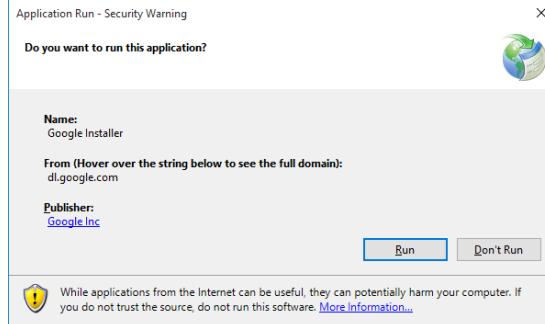
Visit the Google Chrome website (using your computer's existing browser) and follow onscreen instructions to download and install Chrome.

<https://www.google.com/chrome/browser/desktop/index.html>



##### Step 2:

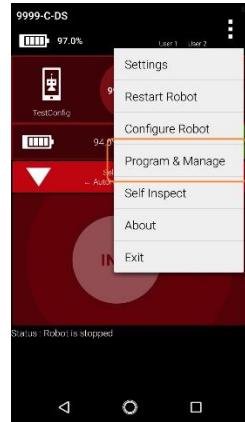
Note that your computer might prompt you with a security warning during the installation process. If you are prompted with this warning, Select the “Run” button to continue with the installation.



## 6.4 Connecting Your Laptop to the Program and Manage Network

In order to write an Op Mode, you will need to connect your programming laptop to the Program & Manage Wi-Fi network. The Program & Manage Wi-Fi network is a wireless network created by your Robot Controller.

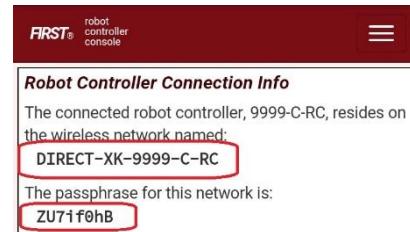
Before you begin this exercise, please make sure that your Windows laptop has the most current service packs and system updates from Microsoft installed.<sup>3</sup>

Connecting Your Laptop to the Program and Manage Network (Time Needed to Complete Task: 20 minutes)	
<p><b>Step 1:</b> On the Driver Station, touch the three dots in the upper right-hand corner of the screen to launch the pop-up menu. Select <b>Program &amp; Manage</b> from the pop-up menu to display the <b>Program &amp; Manage</b> access information.</p>	
<p><b>Step 2:</b> The Program &amp; Manage screen displays important information that you can use to connect your laptop to the FTC Blocks or OnBot Java Programming Mode server.</p>	

<sup>3</sup> This example assumes the user has a Windows 10 laptop. If you are not using a Windows laptop, the procedure to connect to the Programming Mode server will differ. Refer to your device's documentation for details on how to connect to a Wi-Fi network.

**Step 3:**

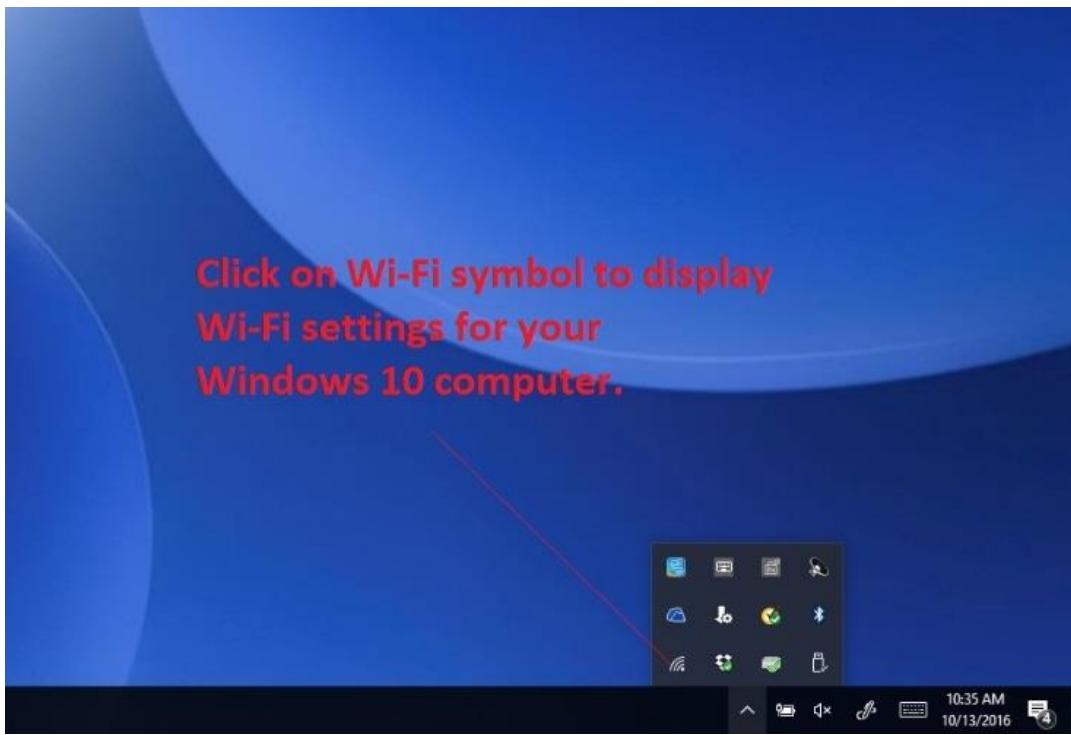
Verify the network name and passphrase for the Program & Manage wireless network. Towards the top of the screen, the name of the Program & Manage wireless network is displayed. If you are using an Android smartphone as your Robot Controller, then the wireless network name will begin with the phrase "DIRECT-".



If you are using a Control Hub, then the wireless network name will be whatever you specified when you configured your Control Hub. If you haven't changed the Control Hub's name yet, then by default the wireless network's name will begin with "FTC-". If you haven't changed its password yet, then by default the wireless network's passphrase will be "password".

**Step 4:**

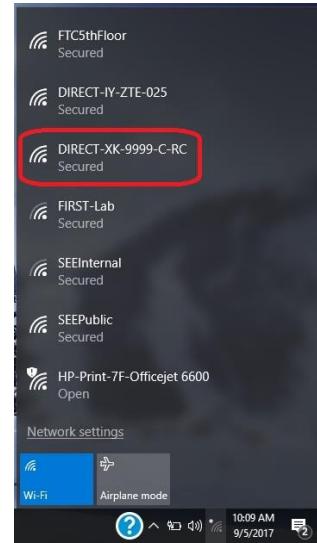
On your Windows 10 computer, look in the lower right-hand corner of your desktop for a Wi-Fi symbol. Select the Wi-Fi symbol to display a list of available Wi-Fi Networks in your vicinity.



**Step 5:**

Look for the wireless network that matches the name displayed on the Program & Manage screen.

In this example, the name of the wireless network is “DIRECT-XK-9999-C-RC” and the network is visible in the list displayed on the Windows 10 computer.

**Step 6:**

Once you have found the target network in the list, select it.

Press the **Connect** button to connect to the network.

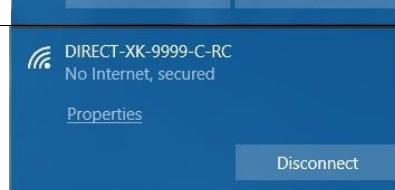
**Step 7:**

When prompted, provide the network passphrase (in this example ZU7if0hB) and press **Next** to continue.

**NOTE:** The passphrase is case sensitive.

**Step 8:**

Once you have successfully established a wireless connection between your Windows 10 laptop and your Robot Controller Android device, the status should be displayed in the wireless settings for your laptop.



If the display is not updated, as shown, after a few seconds, try clicking on Network Connections at the bottom of the blue box showing the Wi-Fi connections. This will bring up a Setting dialog box that includes a link to “Show available networks.”, which can be used to force the list of Wi-Fi connections to be updated.

NOTE: When you are connected to the Blocks programming mode server on your Robot Controller, *your laptop will not have access to the Internet*. It only has direct access to the Robot Controller.

## 6.5 Troubleshooting Your Wireless Connection

If you cannot see your Programming Mode wireless network in the list of available networks or if you are having problems connecting your laptop to Program & Manage wireless network, make sure you answer the following questions:

1. Is the Robot Controller running and connected to the Driver Station?
2. Is your Windows laptop updated with the most current system updates and service packs? Older versions of Windows 8 and 10, for example, had issues that could prevent the laptop from displaying the Program & Manage wireless network in the list of available networks.

If you are still having issues with connecting the laptop to the Robot Controller, visit [Troubleshooting](#) section of this document for instructions on how to manually connect to the Program & Manage wireless network with a Windows 10 laptop.

## 6.6 Creating Your First Op Mode

If you connected your laptop successfully to the Program & Manage wireless network of the Robot Controller, then you are ready to create your first op mode. In this section, you will use the FTC Blocks Programming Tool to create the program logic for your first op mode.

### Creating Your First Op Mode (Time Needed to Complete Task: 10 minutes)

#### **Step 1:**

Launch the web browser on your laptop (FIRST recommends using Google Chrome).

**Important Note:** If your Robot Controller is an Android smartphone, then the address to access the Program & Manage server is "192.168.49.1:8080".

**Important Note:** If your Robot Controller is a Control Hub, then the address to access the Program & Manage server is "192.168.43.1:8080". Notice the difference in the third octet of the IP addresses (the Control Hub has a "43" instead of a "49").

Once connected, enter the following address into your web browser:

**http://192.168.49.1:8080**

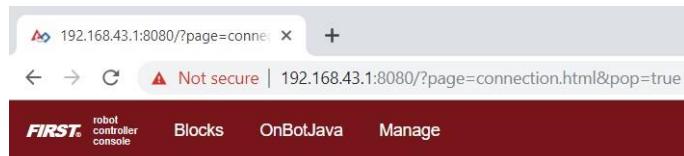
Server status:  
**Server OK (Running since 10/13, 10:17 AM)**

To remotely connect to the controller, connect your laptop's wireless adapter to this network, using the passphrase to gain access. Once connected, enter the following address into your web browser:

**http://192.168.43.1:8080**

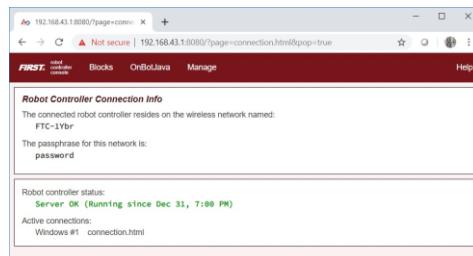
Robot controller status:  
**Server OK (Running since Dec 31, 7:00 PM)**

Type the appropriate web address into the address field of your browser to navigate to the Program & Manage web server.



#### **Step 2:**

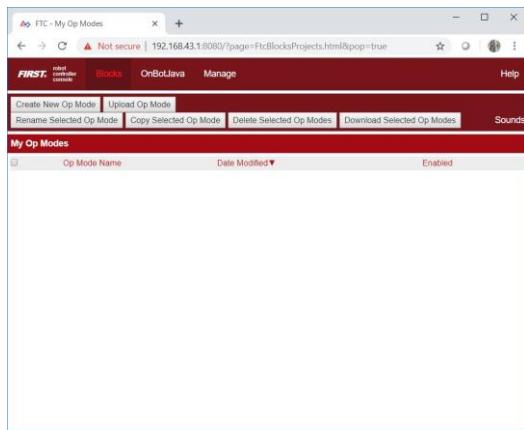
Verify that your web browser is connected to the programming mode server. If it is connected to the programming mode server successfully, the Robot Controller Console should be displayed.



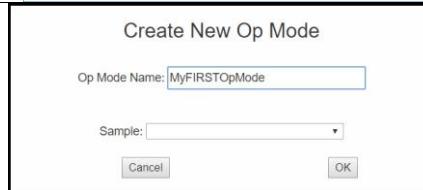
**Step 3:**

Select the **Blocks** to navigate to the main Blocks Programming screen.

The main Blocks Programming screen is where you create new op modes. It is also the screen where you can see a list of existing Blocks Op Modes on a Robot Controller. Initially this list will be empty until you create and save your first op mode.

**Step 4:**

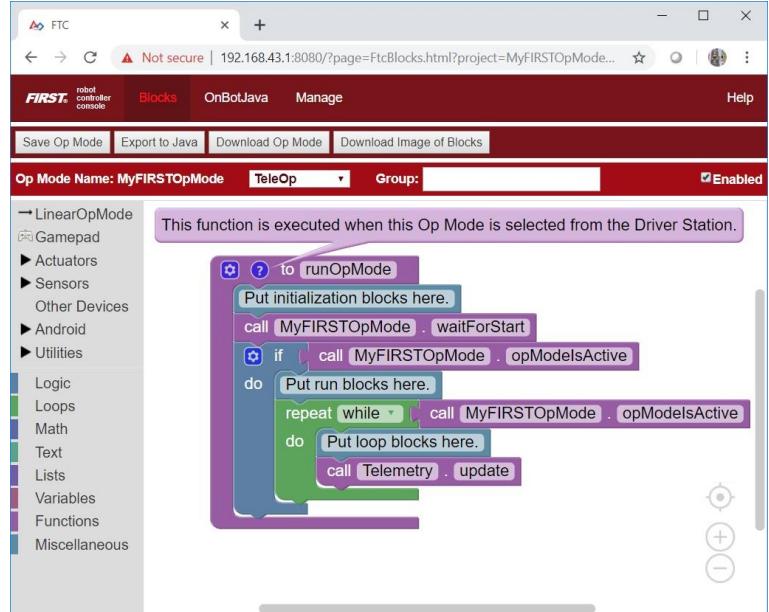
Select **Create New Op Mode** from the upper left-hand corner of the browser window. Specify a name and **OK**.

**Step 5:**

Verify that you created the new op mode. You should see your newly created op mode opened for editing in your web browser's main screen.

Notice that the left-hand side of the browser's screen contains a list of categorized programming blocks. If you select a category, a menu of available related programming blocks opens.

The right-hand side is the design canvas where you arrange your programming blocks to create the logic for your op mode.



## 6.7 Examining the Structure of Your Op Mode

When you create a new op mode, there should already be a set of programming blocks that are placed on the design canvas for your op mode. These blocks are the default with each new op mode that you create. They create the basic structure for your op mode.

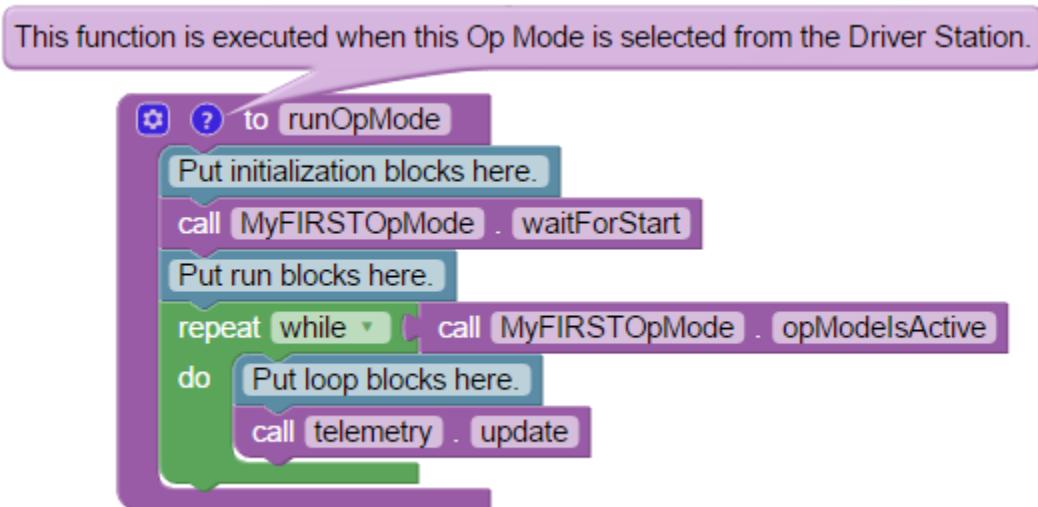


Figure 4 – Each new op mode includes a set of programming blocks that form the basic structure of the op mode.

In the figure above, the main body of the op mode is defined by the outer purple bracket that has the words “to runOpMode” at the top. As the help tip indicates, this function is executed when this op mode (“MyFIRSTOpMode” in this example) is selected from the Driver Station.

It can be helpful to think of an op mode as a list of tasks for the Robot Controller to perform. 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 an op mode.

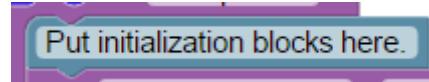


Figure 5 - It can be helpful to think of an op mode as a task list for the Robot Controller.<sup>4</sup>

<sup>4</sup> Clip art image was downloaded from ClipArtBest.com on 10/13/16.

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

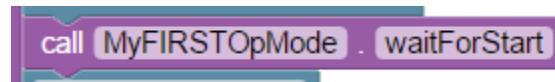
You can hide the help text by selecting the blue question mark (“?”) icon. Let’s look at the flow of this basic op mode. The blue colored block with the words “Put initialization blocks here” is a comment. Comments are placed in an op mode for the benefit of the human user. The robot will ignore any comments in an op mode.



*Figure 6 – Comments are blue-colored and are ignored by the robot.*

Any programming blocks that are placed after the “Put initialization blocks here” comment (and before the “call MyFIRSTOpMode.waitForStart” block) will be executed when the op mode is first selected by a user at the Driver Station.

When the Robot Controller reaches the block labeled “call MyFIRSTOpMode.waitForStart” it will stop and wait until it receives a Start command from the Driver Station. A Start command will not be sent until the user pushes the Start button on the Driver Station. Any code after the “call MyFIRSTOpMode.waitForStart” block will get executed after the Start button has been pressed.



*Figure 7 – The Robot Controller will stop and wait for a Start command when it reaches this programming block.*

Any blocks that are placed after the “Put run blocks here” comment and before the green block labeled “repeat while call MyFIRSTOpMode.opModelsActive” will be executed sequentially by the Robot Controller after the Start button has been pressed.

The green block labeled “repeat while call MyFIRSTOpMode.opModelsActive” is an iterative or looping control structure.



*Figure 8 - The green programming block is a while loop control block.*

This green control block will perform the steps listed under the “do” portion of the block as long as the condition “call MyFIRSTOpMode.opModelsActive” is true. What this means is that the statements included in the “do” portion of the block will repeatedly be executed as long as the op mode “MyFIRSTOpMode” is running. Once the user presses the Stop button, the “call MyFIRSTOpMode.opModelsActive” clause is no longer true and the “repeat while” loop will stop repeating itself.

## 6.8 Modifying the Op Mode to Control a DC Motor

Let’s modify our op mode to add some blocks that will allow us to control a DC motor with a gamepad.

Important Note: The programming blocks for user configured devices (motors, servos and sensors) will only be visible in the Blocks tool if there is an active configuration file with the configured devices included in the file. If

a type of device is not included in the active configuration file, then its programming blocks will be missing from the palette of blocks.

### Modifying the Op Mode to Control a DC Motor (Time Needed to Complete Task: 15 minutes)

#### **Step 1:**

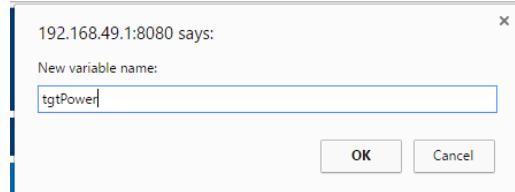
On the left-hand side of the screen select the category called **Variables** to display the list of block commands that are used to create and modify variables within your op mode.

Select “Create variable...” that will represent the target motor power for our op mode.



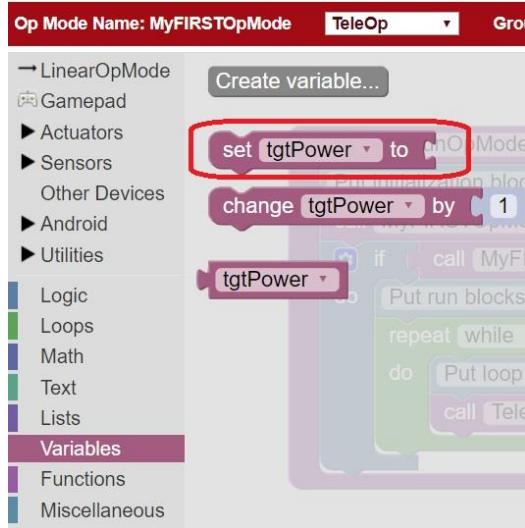
#### **Step 2:**

When prompted, type in a new variable name: tgtPower.



#### **Step 3:**

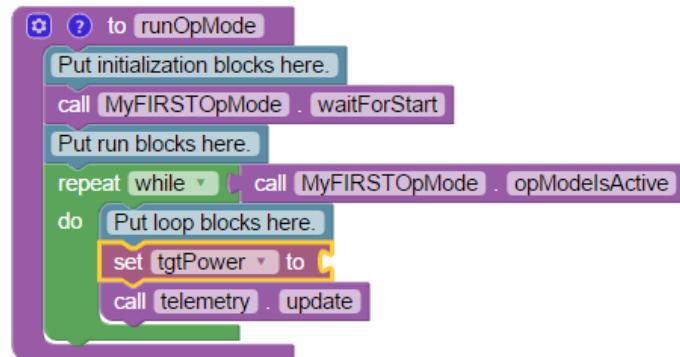
Once you have created your new variable, some additional programming blocks should appear under the Variables block category.



#### **Step 4:**

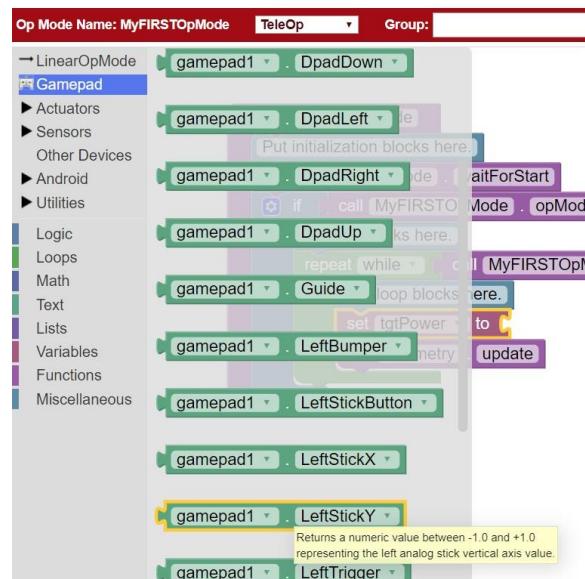
Select “set tgtPower to” block and then drag the block under the “Put loop blocks here” comment block.

The “set tgtPower to” block should highlight and snap into position.



**Step 5:**  
Select **Gamepad** of programming blocks, then the “gamepad1.LeftStickY” block from the list.

NOTE: The control system lets you have up to two gamepads controlling a robot. By selecting “gamepad1” you are telling the op mode to use the control input from the gamepad that is designated as driver #1.

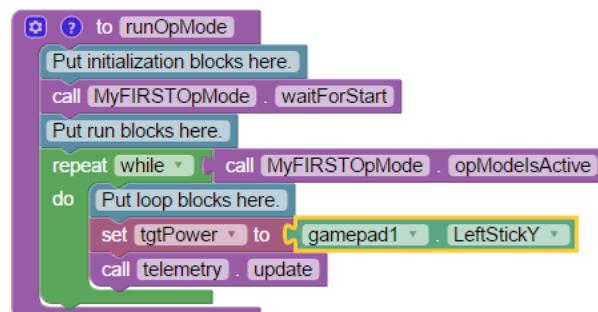


**Step 6:**  
Drag the “gamepad1.LeftStickY” block to the right side of the “set tgtPower to” block.

This set of blocks will continually loop and read the value of gamepad #1's left joystick (the y position) and set the variable tgtPower to the Y value of the left joystick.

NOTE: The F310 gamepads, 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.

This means that for the blocks shown in our example, if the left

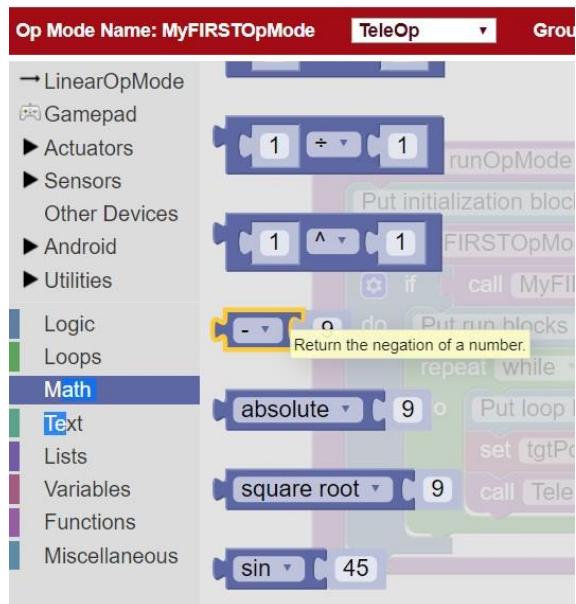


joystick is pushed to the top, the variable tgtPower will have a value of -1.



### Step 7:

Select the **Math** category for the programming blocks then the negative symbol (" - " ).



### Step 8:

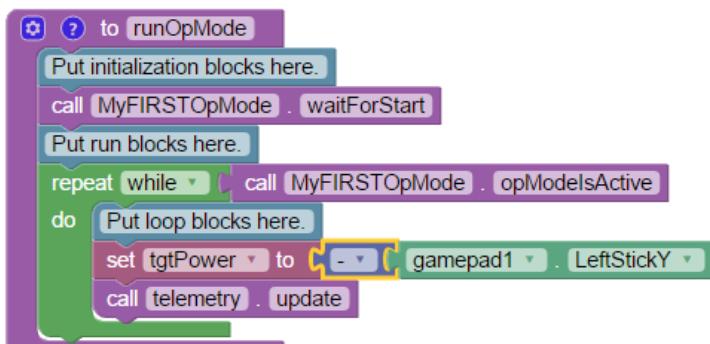
Drag the negative symbol (also known as a “negation operator”) to the left of the “gamepad1.LeftStickY” block.

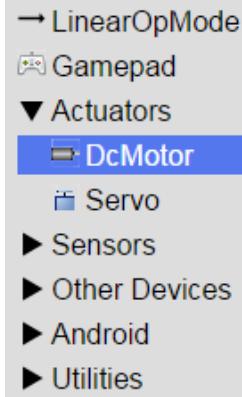
It should click into place after the “set tgtPower to” block and before the “gamepad1.LeftStickY” block.

With this change, the variable tgtPower will be set to +1 if the left joystick is in its topmost position and will be set to -1 if the joystick is in its bottommost position.

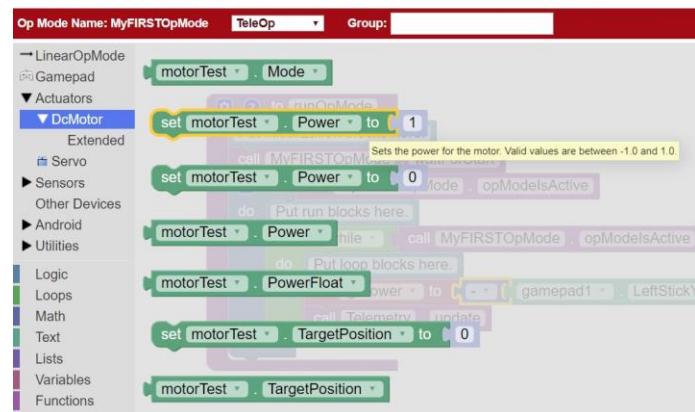
### Step 9:

Select **Actuators** category of blocks, then **DcMotor**.

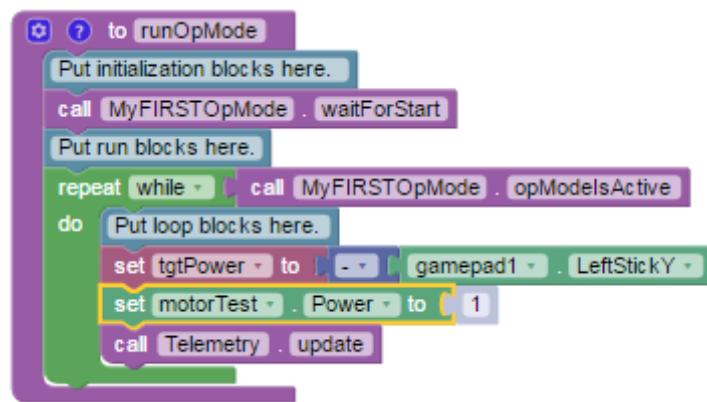


**Step 10:**

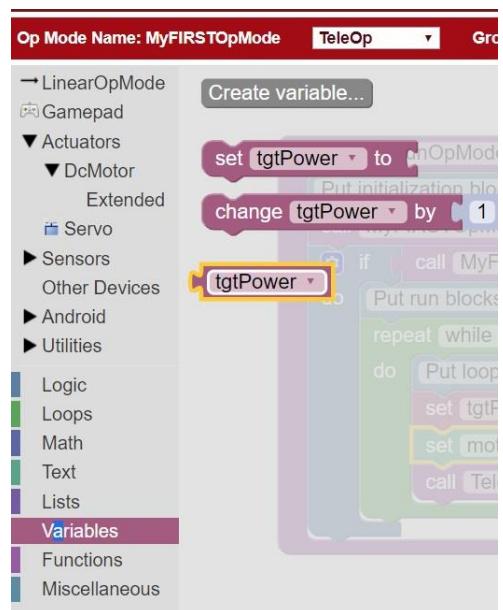
Select the “set motorTest.Power to 1” programming block.

**Step 11:**

Drag “set motorTest.Power to 1” block so that it snaps in place right below the “set tgtPower to” block.

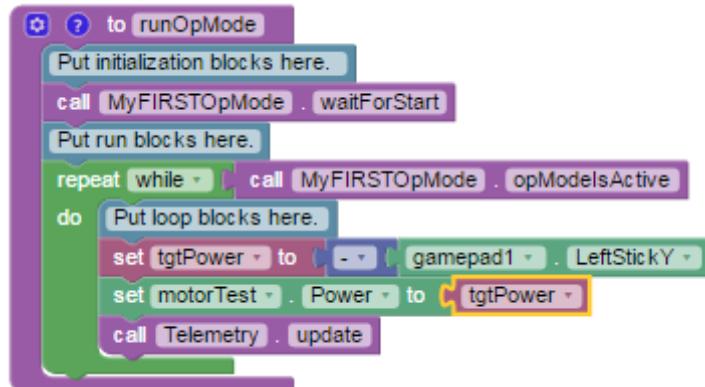


**Step 12:**  
Select the **Variables**, then  
“tgtPower” block.



**Step 13:**  
Drag the “tgtPower” block so it  
snaps in place just to the right of  
the “set motor1.Power to” block.

The “tgtPower” block should  
automatically replace the default  
value of “1” block.



## 6.9 Inserting Telemetry Statements

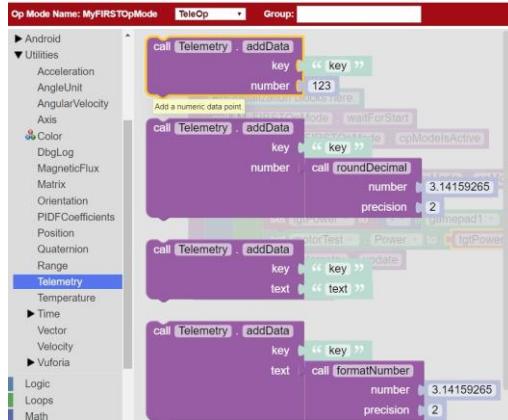
Your op mode is just about ready to run. However, before we continue, let's add a couple of *telemetry statements* that will send information from the Robot Controller to the Driver Station for display on the Driver Station user interface. This *telemetry* mechanism is a useful way to display status information from the robot on the Driver Station. You can use this mechanism to display sensor data, motor status, gamepad state, etc. from the Robot Controller to the Driver Station.

## Inserting Telemetry Statements

**(Time Needed to Complete Task: 15 minutes)**

**Step 1:**

Select **Utilities**, then **Telemetry** and the “call telemetry.addData (key, number)” block.

**Step 2:**

Drag the “call telemetry.addData(key, number)” block and place it below the “set motor1.Power to” block.

Select the green text block “key” and highlight the text and change it to read “Target Power”.

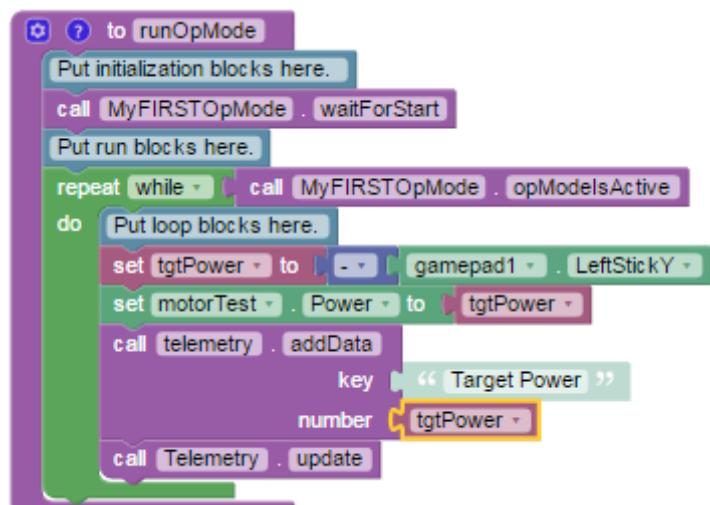
**NOTE:** The “call telemetry.update” block is an important block. Data that is added to the telemetry buffer will not be sent to the Driver Station until the telemetry.update method is called.

**Step 3:**

Select the **Variables** then the “tgtPower” block.

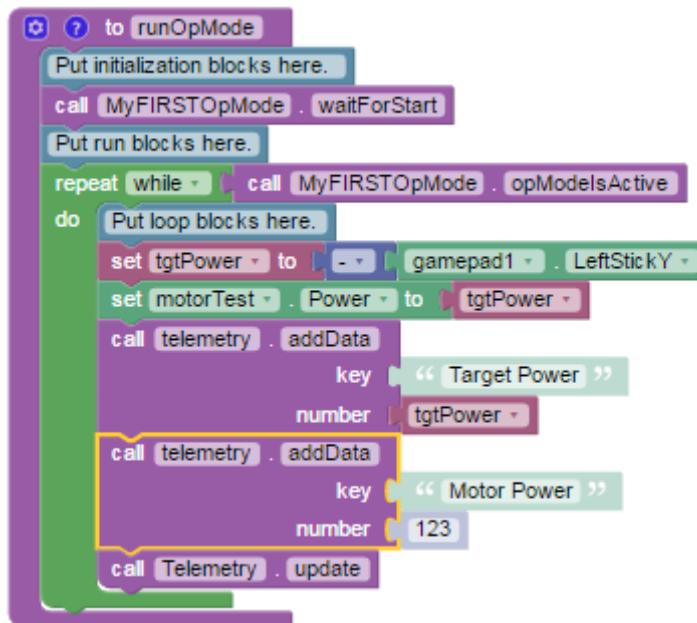
Drag the block so it clicks into place next to the “number” parameter on the telemetry programming block.

The Robot Controller will send the value of the variable “tgtPower” to the Driver Station with a key or label of “Target Power”. The key will be displayed to the left of the value on the Driver Station.



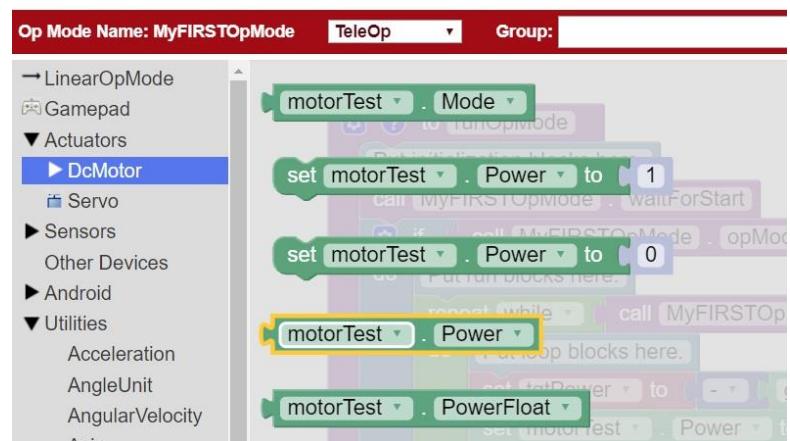
**Step 4:**

Repeat this process and name the new key “Motor Power”.

**Step 5:**

Select **Actuators**, then **DcMotor**.

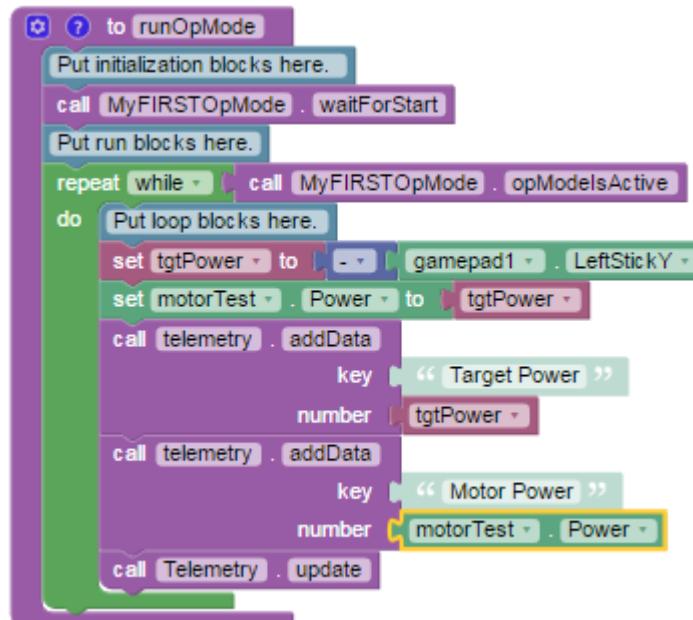
Look for the green programming block labeled “motorTest.Power”.



**Step 6:**

Drag the “motorTest.Power” block to the “number” parameter of the second telemetry block.

Your op mode will now also send the motor power information from the Robot Controller to be displayed on the Driver Station.



## 6.10 Saving Your Op Mode

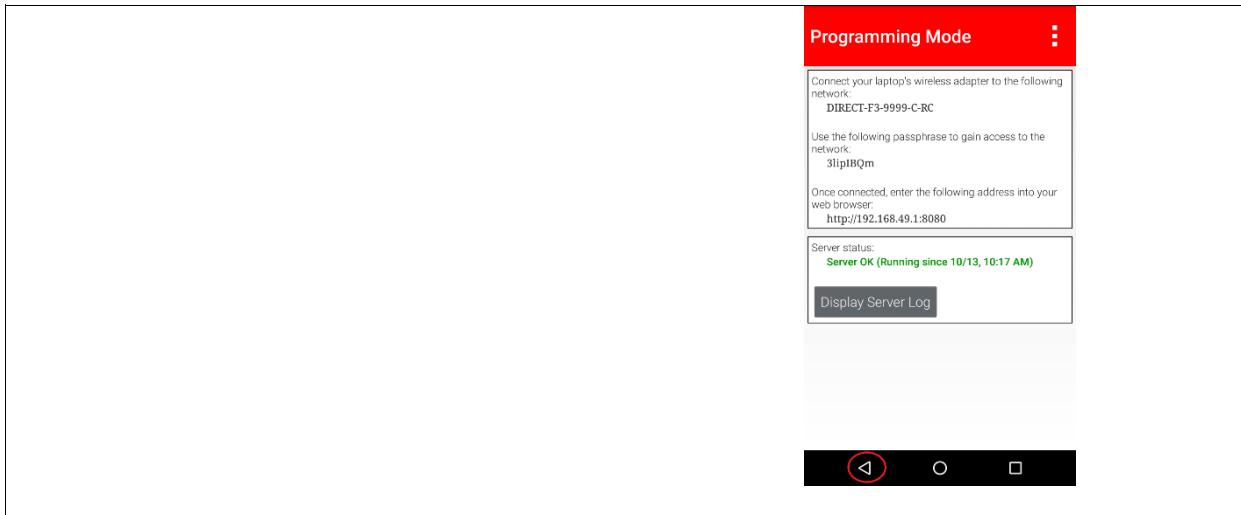
After you have modified your op mode, it is important to save the op mode to the Robot Controller.

<b>Saving Your Op Mode</b> (Time Needed to Complete Task: 1 minutes)
<p><b>Step 1:</b>  <b>Save Op Mode</b> to the Robot Controller’s programming mode server. If your save was successful, you should see <b>Save completed successfully</b> in green letters.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> </div>

## 6.11 Exiting Programming Mode

After you have modified and saved your op mode, you need to exit Programming Mode before you will be able to run your op mode.

<b>Saving Your Op Mode</b> (Time Needed to Complete Task: 1 minutes)
<p><b>Step 1:</b>          Press the Android back arrow to exit Programming Mode. You need to exit Programming Mode before you can run your op mode.</p>



## 7 Running Your Op Mode

Your op mode requires 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.

### Running Your Op Mode (Time Needed to Complete Task: 10 minutes)

#### **Step 1:**

Before you connect your gamepad to the phone, verify that the switch on the bottom of the gamepad is set to the “X” position.



**Step 2:**

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

**Step 3:**

Your example op mode is looking for input from the gamepad designated as the user or driver #1.

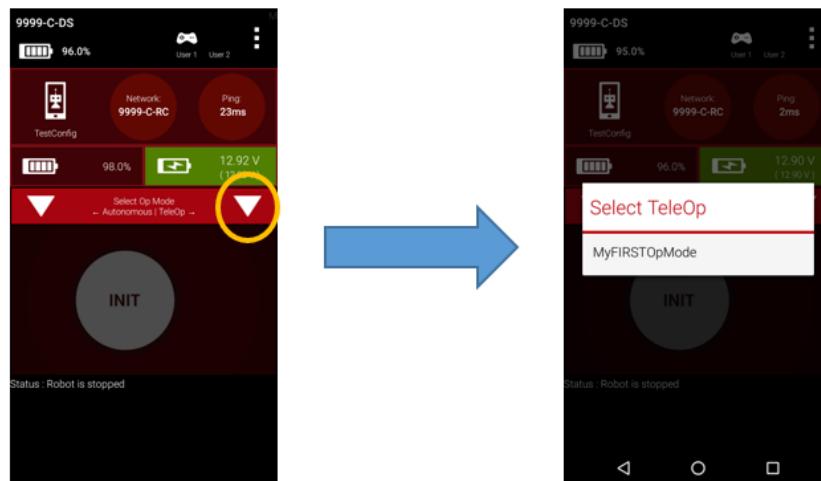
Press the **Start** button and the **A** button simultaneously on the Logitech F310 controller to designate your gamepad as user #1.

**NOTE:** Pushing the **Start** button and the **B** button simultaneously would designate the gamepad as user #2.



**Step 4:**  
On the Driver Station screen, press the triangular-shaped, **TeleOp** to display a list of available op modes.

You should see your recently saved op mode among the list of available op modes that reside on your Robot Controller.

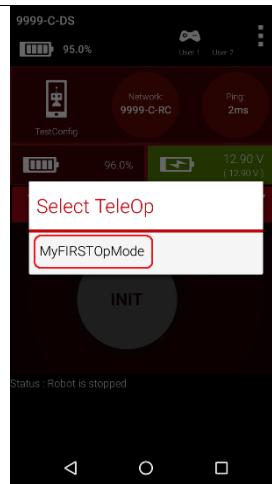


**NOTE:** *TeleOp* is short for *Tele-Operated* and it implies a driver-controlled op mode (i.e., an op mode that gets input from a human driver).

**Step 5:**

Select **MyFIRSTOpMode** to load your op mode on the Robot Controller.

**NOTE:** Even though you are using the Driver Station to select the op mode, the actual op mode instructions will be executed on the Robot Controller.

**Step 6:**

Press **INIT** to initialize your op mode.

**Step 7:**

Press **Start** (designated by the triangular-shaped symbol) to start the op mode run.



**Step 8:**

Use the left joystick of the gamepad to control the operation of the DC motor.

As you manipulate the left joystick up and down, the target power and the motor power should be displayed in the lower left-hand corner of the screen.

If you want to stop your op mode, press the square-shaped **Stop** button on the Driver Station.



## 8 Controlling a Servo Motor with an Op Mode

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

### 8.1 What is a Servo Motor?

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

In the figure below, a “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.



*Figure 9 – An example of a servo motor that is commonly used on FIRST Tech Challenge robots.<sup>5</sup>*

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).

### 8.2 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.

<sup>5</sup> Image taken from the Pitsco website on 10/17/2016 (see [https://c10645061.ssl.cf2.rackcdn.com/product/icongo/icg\\_39197\\_180deg servo.jpg](https://c10645061.ssl.cf2.rackcdn.com/product/icongo/icg_39197_180deg servo.jpg)).  
**Gracious Professionalism®** - “Doing your best work while treating others with respect and kindness - It’s what makes FIRST, first.”

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 Program & Manage server, 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.



Figure 10 - A typical servo motor can rotate to and hold a position from 0 to 180 degrees.

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.



Figure 11 – The colored buttons on the right side of the gamepad will be used to control servo position.

## Modifying the Op Mode to Control a Servo Motor (Time Needed to Complete Task: 20 minutes)

### Step 1:

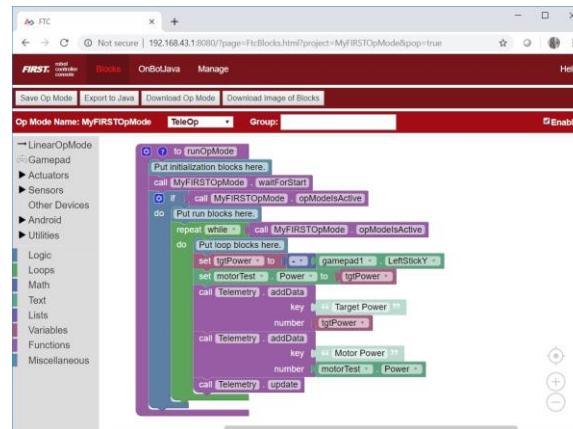
Verify that your laptop is still connected to the Robot Controller's Program & Manage Wi-Fi network.

### Step 2:

Verify that **MyFIRSTOpMode** is opened for editing.

If it is not, select the FIRST logo in the upper left-hand corner of the browser window. This should take you to the main FTC Blocks Development Tool screen.

Select the **MyFIRSTOpMode** project to open it.



### Step 3:

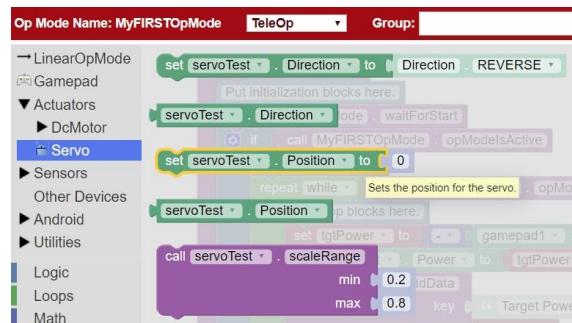
On the left-hand side of the screen select **Actuators** and then **Servo**.

#### Op Mode Name: MyF

- LinearOpMode
- Gamepad
- ▼ Actuators
  - DcMotor
  - Servo**
- Sensors
- Other Devices
- Android
- Utilities

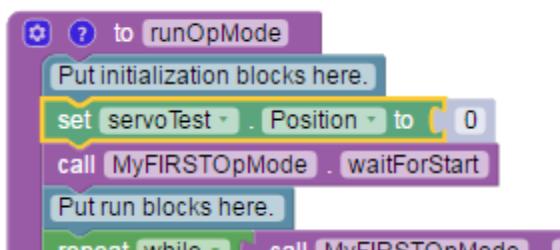
**Step 4:**

Select the “set servoTest.Position to” block from the list of available Servo blocks.

**Step 5:**

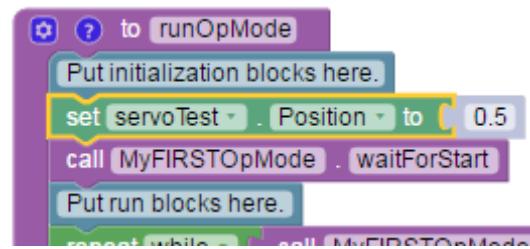
Drag the “set servoTest.Position to” block under the comment block “Put initialization blocks here.”

The block should click into place.

**Step 6:**

Select the number block “0” and change the block’s value to “0.5”.

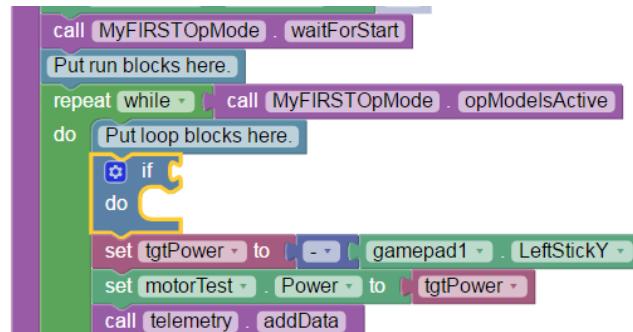
When a user selects this op mode, the servo position will initially be set to the midway point (90-degree position).

**Step 7:**

Select **Logic**, then select the “if do” block from the list of available blocks.

Drag the block under the comment block “Put loop blocks here.”

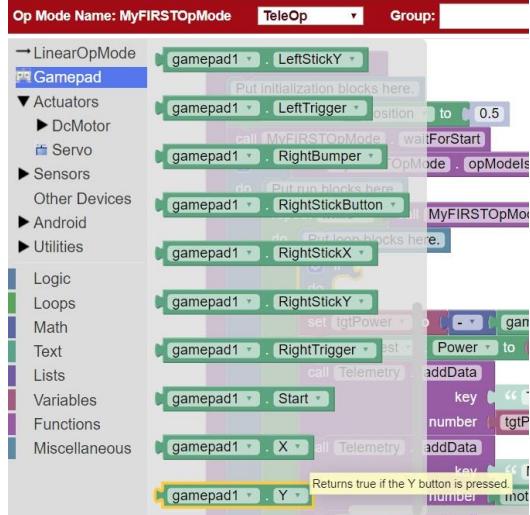
The block should click into place.



**Step 8:**

Select the **Gamepad** and the “gamepad1.Y” block from the list of available blocks.

**NOTE:** This block is towards the bottom of the list of blocks.

**Step 9:**

Drag the “gamepad1.Y” block to the right side of the “if do” block.

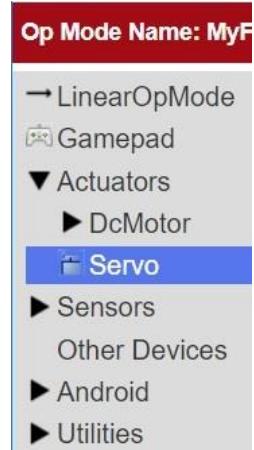
The block should click into place.

The “if do” block will use the state of the “gamepad1.Y” value its test condition. If the “Y” button is pressed, the statements within the “do” portion of the block will be executed.

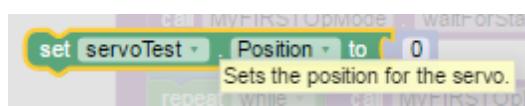


**Step 10:**

On the left-hand side of the screen select the category called **Actuators**, then **Servo**.

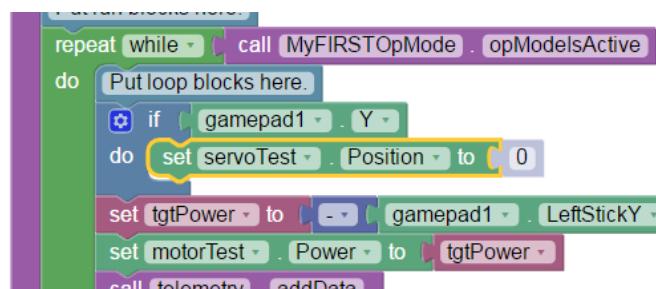
**Step 11:**

Select the “set servoTest.Position to” block from the list of available Servo blocks.

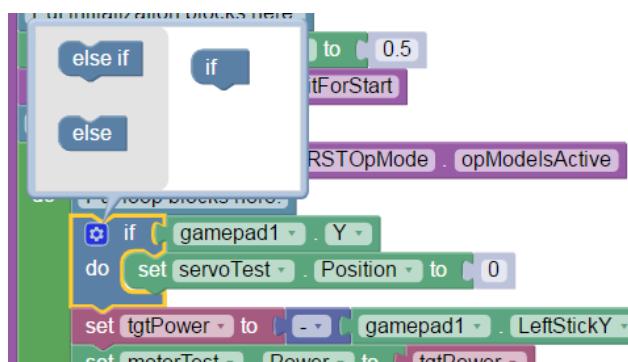
**Step 12:**

Drag the “set servoTest.Position to” block so that it snaps in place in the do portion of the “if do” block.

If the “Y” button is pressed on gamepad #1, the op mode will move the servo’s position to the 0-degree position.

**Step 13:**

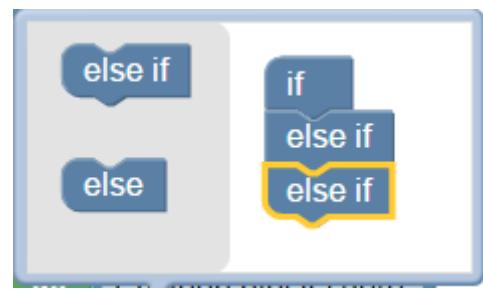
Select the blue and white **Settings** icon for the “if do” block. This will display a pop-up menu that lets you modify the “if do” block.



**Step 14:**

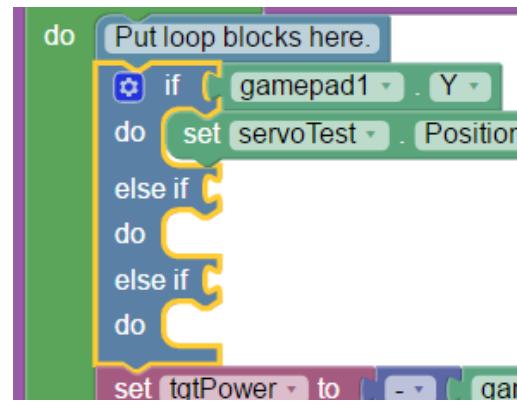
Drag an “else if” block from the left side of the pop-up menu and snap it into place under the “if” block.

Drag a second “else if” block from the left side and snap it into place on the right side under the first “else if” block.

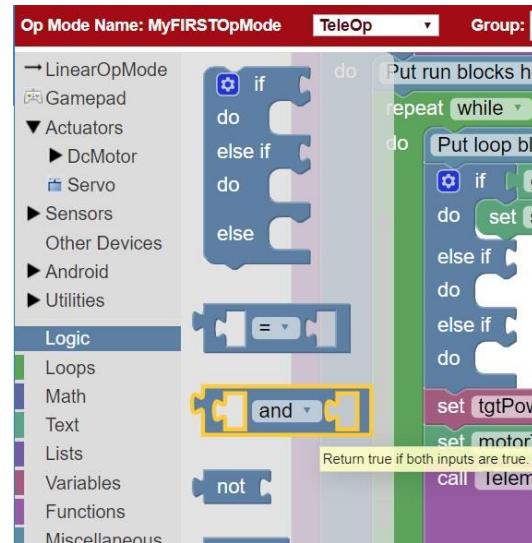
**Step 15:**

Select the **Settings** icon to hide the pop-up menu for the “if do” block.

The “if do” block should now have two “else if” test conditions added.

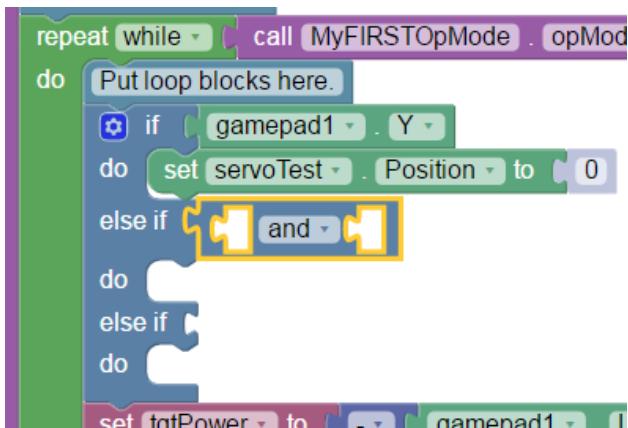
**Step 16:**

Select **Logic** then the “and” block

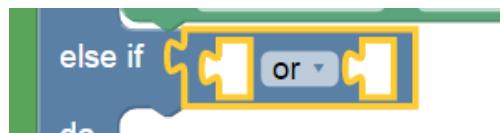


**Step 17:**

Drag the “and” block so it clicks in place as the test condition for the first “else if” block.

**Step 18:**

Select the word “and” and select “or” from the pop-up menu to change the block to a logical “or” block.

**Step 19:**

Select the **Gamepad** and select the “gamepad1.X” block.

Drag the block so that it clicks in place as the first test condition of the logical “or” block.

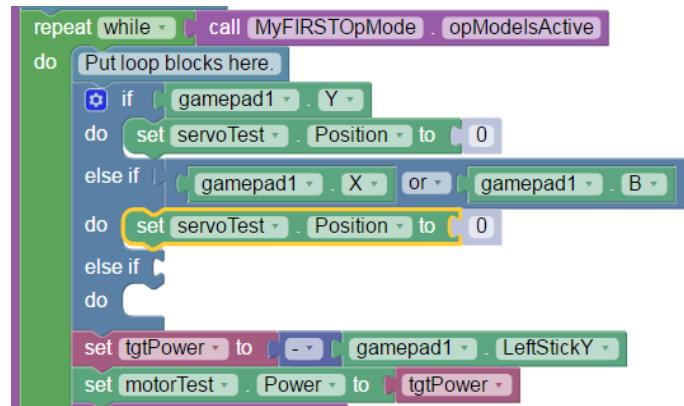
**Step 20:**

Select **Gamepad** and the “gamepad1.B” block.

Drag the block so that it clicks in place as the second test condition of the logical “or” block.

**Step 21:**

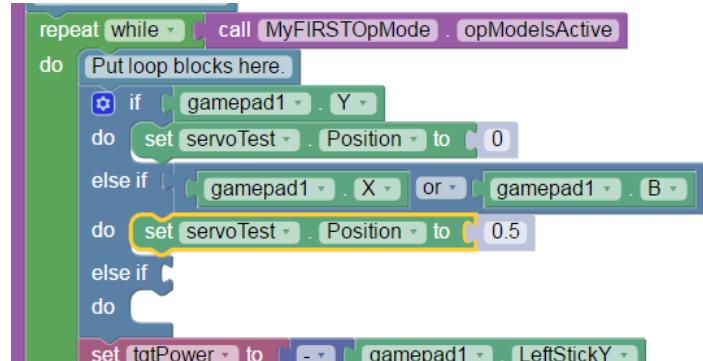
Select a “set servoTest.Position to” block and place it into “do” clause of the first else-if block.



**Step 22:**

Highlight the number “0” and change it to “0.5”.

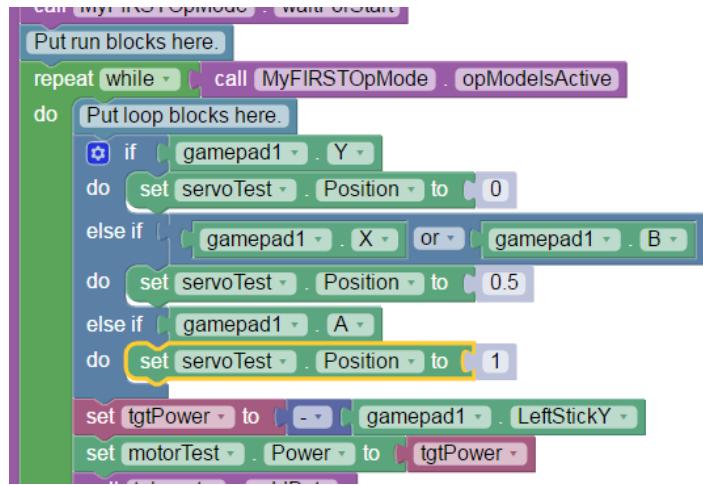
With this change, if the user presses the “X” button or “B” button on gamepad #1, the op mode will move the servo to the midway (90-degree) position.

**Step 23:**

Use a “gamepad1.A” block as the test condition for the second “else if” block.

Drag a “set servoTest.position to” block to the do clause of the second “else if” block and modify the numeric value so that the servo’s position will be set to a value of 1.

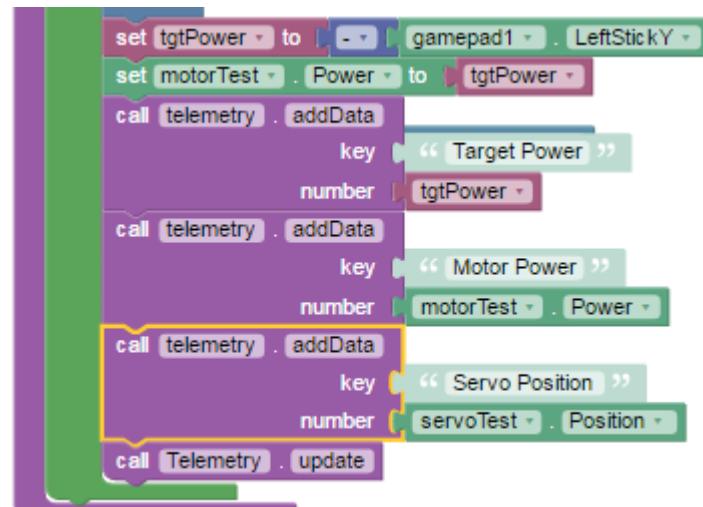
For this clause, if the “A” button is pressed on the #1 gamepad, the op mode will move the servo to the 180-degree position.

**Step 24:**

Insert a “call telemetry.addData” block (numeric) before the “call Telemetry.update” block.

Rename the key field to “Servo Position” and insert a “servoTest.Position” block for the number field.

This set of blocks will send the current servo position value to the Driver Station while the op mode is running.



**Step 25:**

**Save Op Mode** and verify that it was saved successfully to the Robot Controller.

**Step 26:**

Following the procedure outlined in [Running Your Op Mode](#) to run your updated op mode.

Don't forget to exit programming mode before selecting and running the op mode. Also, make sure that your gamepad is designated as User #1 before running your op mode.

You should now be able to control the servo position with the colored buttons. The servo position should be displayed on the Driver Station.



## 9 Using Sensors

### 9.1 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: 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.

## Modifying the Op Mode to Display Distance (Time Needed to Complete Task: 15 minutes)

### Step 1:

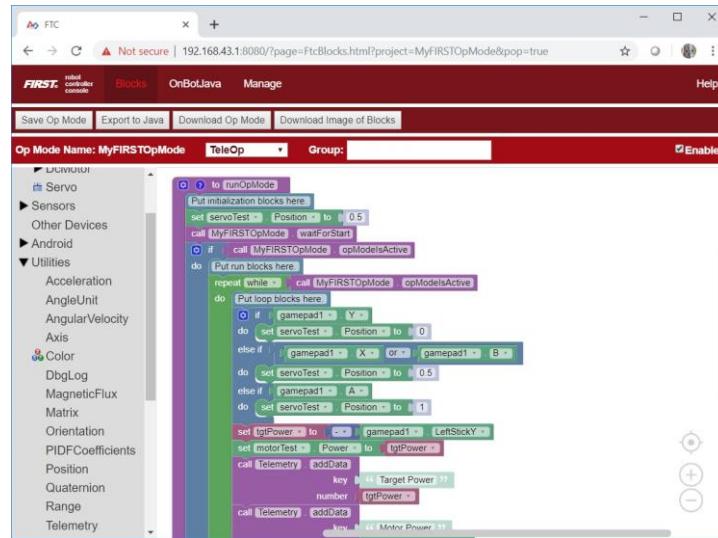
Verify that your laptop is still connected to the Robot Controller's Program & Manage Wi-Fi network.

### Step 2:

Verify that "MyFIRSTOpMode" is opened for editing.

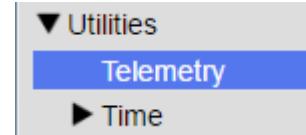
If it is not, you can select the *FIRST* logo in the upper left-hand corner. This should take you to the main FTC Blocks Development Tool project screen.

Select the "MyFIRSTOpMode" project to open it.



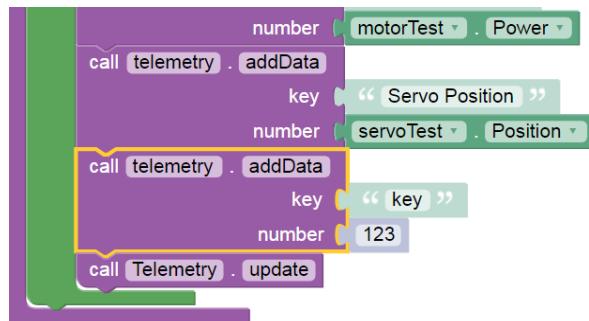
### Step 3:

Select **Utilities**, then **Telemetry**.

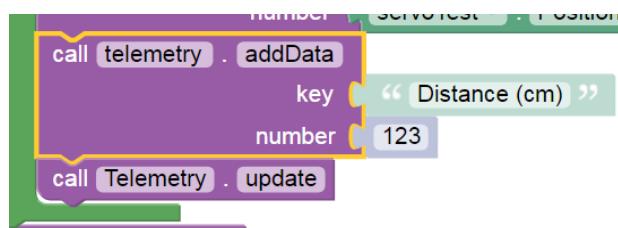


**Step 4:**

Select the “call telemetry.addData” block (the numeric version) and drag it to the spot in your “while” loop block immediately before the “telemetry.update” block.

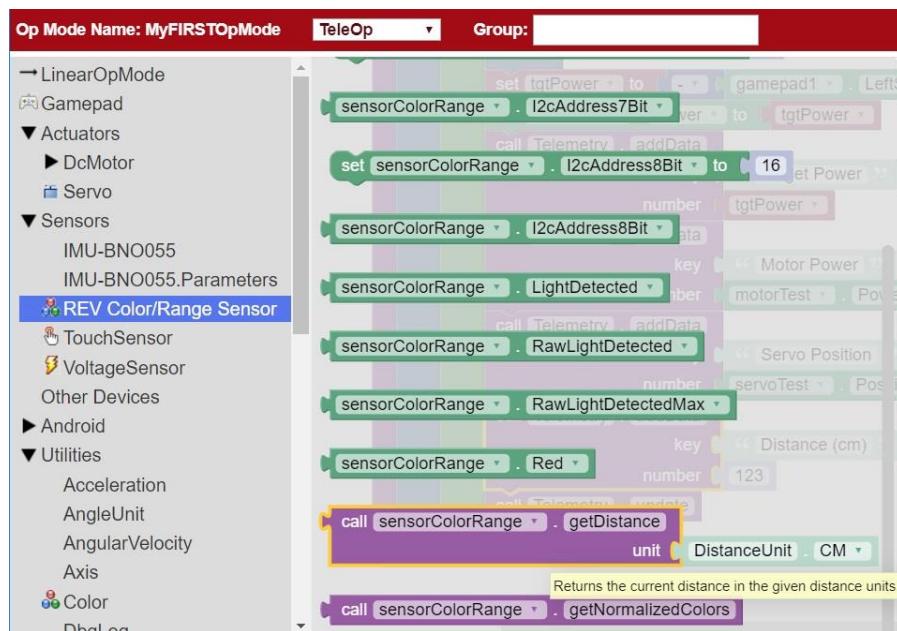
**Step 5:**

Click and highlight the “key” text and change the text so it reads “Distance (cm)”.

**Step 6:**

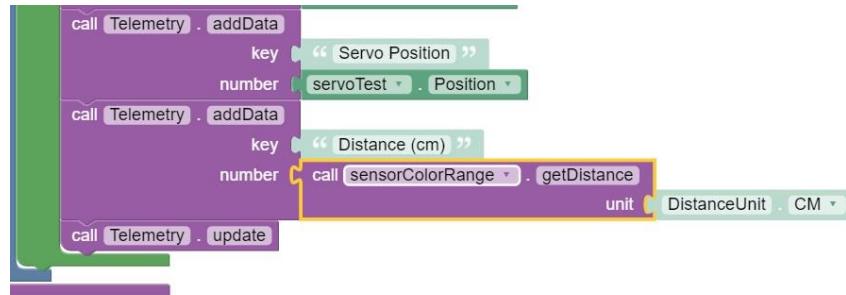
Select and expand the **Sensors** category. Select the “REVColor/RangeSensor” subcategory.

Select the “call sensorColorRange.getDistance” programming block.



**Step 7:**

Drag the “call sensorColorRange.getDistance” programming block to the “number” field of the “call telemetry.addData” programming block.



This will send the measured distance to the target in centimeters back to the Driver Station.

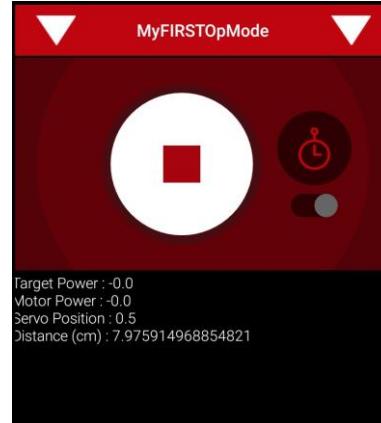
**Step 8:**

Save your op mode and verify that it was saved successfully to the Robot Controller.

**Step 9:**

Follow the procedure outlined in the section [Running Your Op Mode](#) to run your updated op mode.

As you run the op mode, if you move your hand above the color light sensor, you should see the measured distance change on the Driver Station screen. If the expression “NaN” (not a number) is displayed on the Driver Station, the target is most likely out of range (and the sensor does not detect any reflected light).



## 9.2 Touch Sensor

For this example, we assume that the REV Robotics Touch Sensor has been configured as a digital touch sensor in the Robot Controller's active configuration file. We will use the "isPressed" programming block to determine if the button on the sensor is currently pressed or not.

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



*Figure 12 - REV Robotics Touch Sensor.*

The 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 the 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 Hub, the Touch Sensor will be connected to the second pin (labeled “1”) of the port. The first pin (labeled “0”) will stay disconnected.

<b>Modifying the Op Mode to Display Button State (Time Needed to Complete Task: 15 minutes)</b>
---

**Step 1:**

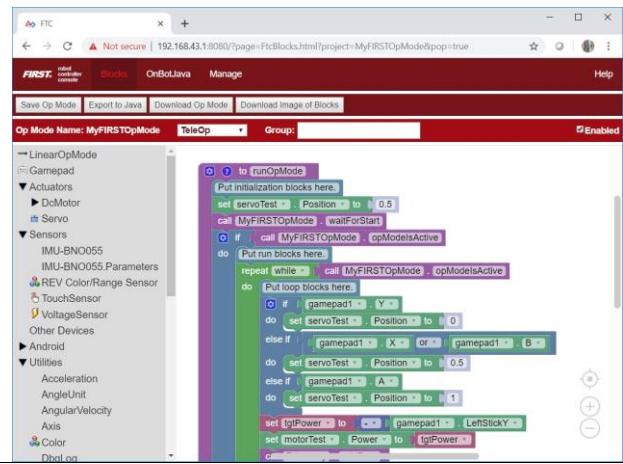
Verify that your laptop is still connected to the Robot Controller's Program & Manage Wi-Fi network.

**Step 2:**

Verify that **MyFIRSTOpMode** is opened for editing.

If it is not, you can select the *FIRST* logo. This should take you to the main FTC Blocks Development Tool project screen.

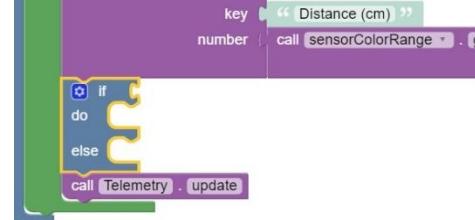
Select the **MyFIRSTOpMode** project to open it.

**Step 3:**

Select **Logic**, then the “if do else” block.

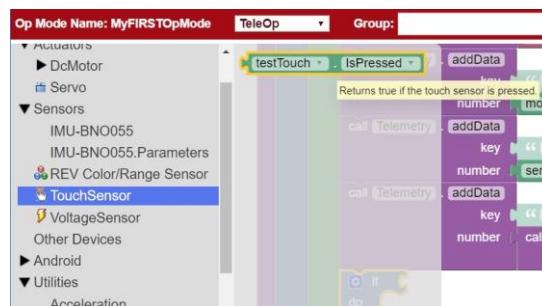
**Step 4:**

Drag the “if do else” block to the position before the “Telemetry.update” block.

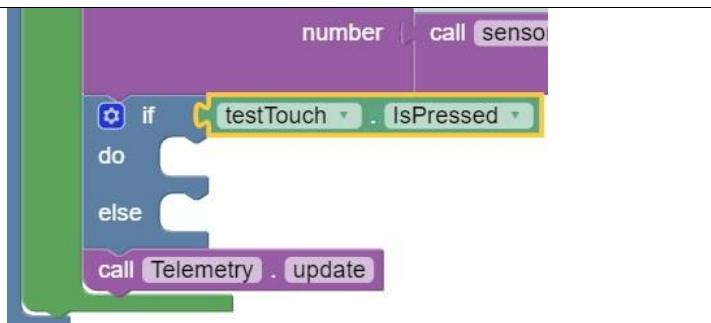
**Step 5:**

Select the **Sensors** category to expand it.

Select the **Touch Sensor** subcategory, then find and select the “.isPressed” block.

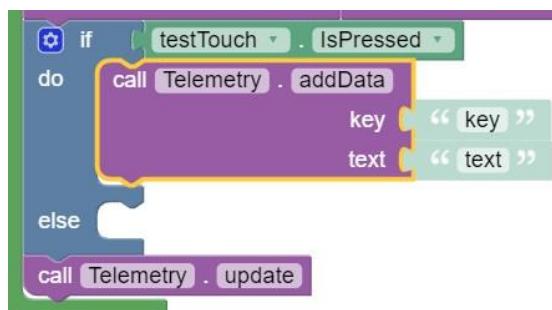
**Step 6:**

Drag the “isPressed” block to the test condition of the “if do else” programming block.

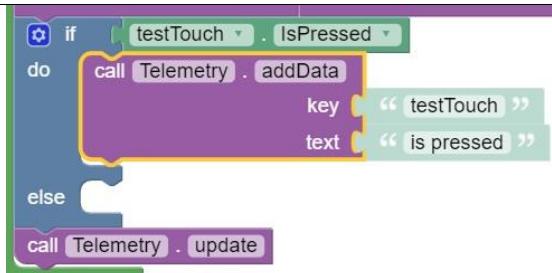
**Step 7:**

Select the **Utilities** category on the left-hand side of your browser. Find and Select the **Telemetry** subcategory.

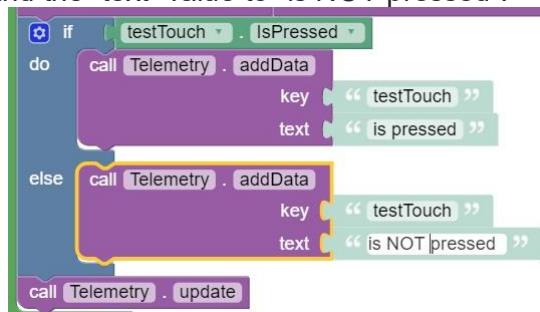
Select the “call telemetry.addData” block (the text version) and drag it to the “do” clause of the “if do else” block.

**Step 8:**

Change the “key” value to “testTouch” and the “text” value to “is pressed”.

**Step 9:**

Insert another “telemetry.addData” block (the text version) to the “else” clause of the “if do else” block. Change the “key” value to “testTouch” and the “text” value to “is NOT pressed”.



**Step 10:**

Save your op mode and verify that it was saved successfully to the Robot Controller.

**Step 11:**

Following the procedure outlined in the section titled [Running Your Op Mode](#).

As you run the op mode and push or release the button, the telemetry message on the Driver Station should update to reflect the current state of the digital Touch Sensor.



## 10 Troubleshooting

### 10.1 Manually Connecting to the Blocks Programming Mode Wi-Fi Network

Section 6.4 describes how to search for the blocks programming mode Wi-Fi network from a list of available networks and then connect to it with a Windows laptop. For some Windows devices, the laptop might not display your blocks programming mode Wi-Fi network in its list of available networks. This problem can occur with some Windows 10 machines (and possibly with some Windows 8 machines), especially if the computer does not have current system updates and service packs.

If you are having problems seeing your FTC Blocks Programming Wi-Fi network in your list of available networks, make sure that your Driver Station is paired and connected to your Robot Controller (see section 3.5 of this document). Also, make sure that your Robot Controller is in Programming Mode (see section 6.4 of this document). Also, make sure that your Windows 10 device has its most current updates installed from Microsoft.

If you have verified that the Driver Station is paired and connected to the Robot Controller and that the Robot Controller is in Programming Mode, and if you have verified that your Windows 10 updates are current, then you might have to manually connect your Windows 10 computer to the blocks programming mode Wi-Fi network.

You can manually connect to this network as if the network were a *hidden* network (i.e., a network that does not broadcast its presence to other Wi-Fi devices).

#### Manually Connecting to the Programming Mode Wi-Fi Network (Time Needed to Complete Task: 15 minutes)

**Step 1:**

In the lower right-hand corner of the Windows 10 desktop, Select the network icon in the system tray to display a list of available Wi-Fi networks.

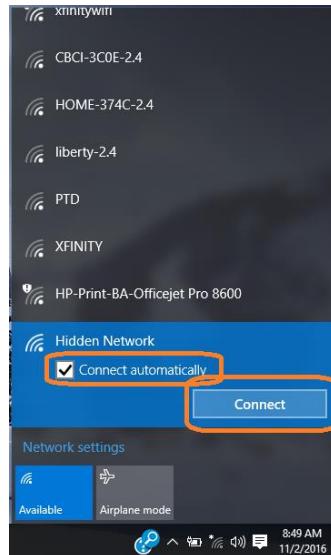
If you still do not see your blocks programming mode network listed, then scroll to the bottom of the list and look for the item “Hidden Network”.



### Step 2:

Select the “Hidden Network” listing to start the connection process. The listing should display a “Connect” button.

Make sure the option “Connect automatically” is checked and then Select the “Connect” button to continue with the process.



### Step 3:

The computer should prompt you for the name or SSID of your blocks programming mode Wi-Fi network. You should type in the network name that is displayed in the Programming Mode window of the Android device.

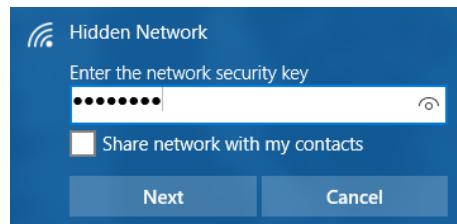
Note, the SSID or network name is case sensitive. Make sure the capitalization of the name that you enter matches the capitalization of the



name displayed in the Programming Mode Window.

#### Step 4:

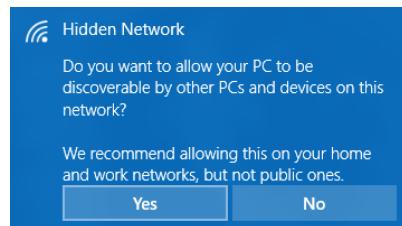
The computer should then prompt you for the passphrase to access this Wi-Fi network. You should type in the network passphrase that is displayed in the Programming Mode window of the Android device.



Note that the passphrase is *case sensitive*. Make sure that your spelling and capitalization matches the original spelling and capitalization shown on the Programming Mode screen.

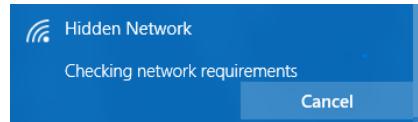
#### Step 5:

Your computer will prompt you on whether you want to make your PC discoverable by other devices on this network. Click “Yes” to continue.



#### Step 6:

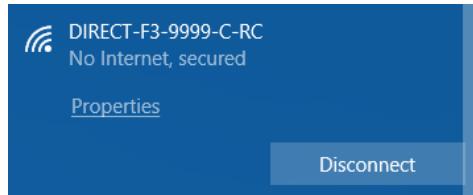
The computer will attempt to connect to your network. Note that it could take several minutes before it connects.



#### Step 7:

If you could successfully connect to the network, it will eventually appear in the list of networks on your computer.

Note that when your computer is connected to the blocks programming server on your Robot Controller phone, *it will not have access to the Internet*.



## 10.2 Troubleshooting Tips

In this section, we provide some basic tips on troubleshooting problems that you might encounter when using the blocks programming mode server to write op modes for your Robot Controller.

### 10.2.1 Cannot See the Blocks Programming Mode Wireless Network

If you are trying to connect to the blocks programming mode wireless network so that you can create/edit an op mode for your Robot Controller, but you cannot see this wireless network listed as an available network for your laptop to connect to, verify the following items:

1. Make sure the Driver Station is successfully paired to the Robot Controller (see section for details). Often the Robot Controller’s Wi-Fi Direct network will time out if it is not connected to the Driver Station.
2. Make sure the Robot Controller has been switched successfully to programming mode (see section 0).

- Power cycle (turn off and then turn back on) your Robot Controller phone and then relaunch the FTC Robot Controller app. Reconnect the Driver Station to the Robot Controller, and then turn off the wireless adapter on your laptop for a few seconds, and then turn it back on (to force a rescan of the available Wi-Fi networks).

### 10.2.2 “Save project failed. Error code 0.”

If you attempt to save the op mode that you are currently editing, but you receive an error message indicating that the “Save project failed. Error code 0.” you might not be connected to the blocks programming mode sever:

- Make sure the Robot Controller is in programming mode.
- Make sure that your laptop is connected to the blocks programming mode Wi-Fi network.
- If you have verified the first two items, press the “Save Op Mode” button again to re-attempt the save operation.

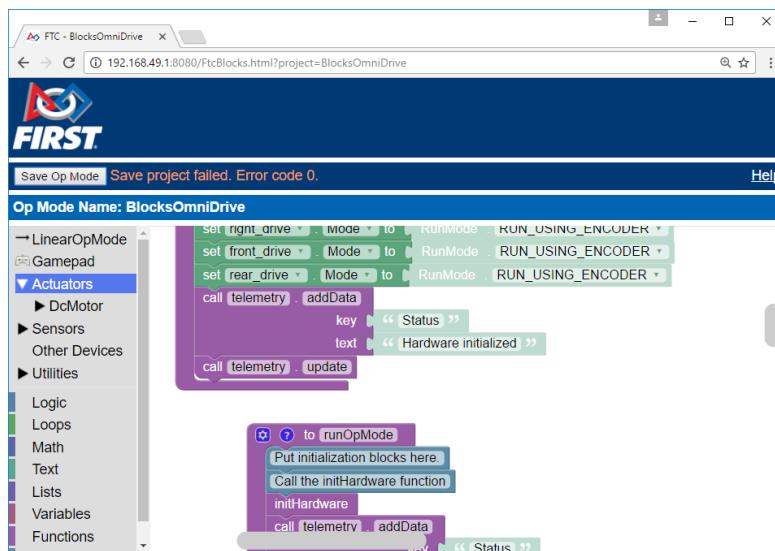


Figure 13 – If the save attempt fails, you might not be connected to the blocks programming mode network and/or server.

### 10.2.3 Op Mode Blocks Are Missing...

If you have opened an existing op mode to edit it in your Javascript-enabled browser, but the programming blocks are missing, check the following:

- Did you remember to save the op mode the last time you edited and then exited the op mode? If you did not save the op mode after the last editing session, you might have lost some of your changes.
- Are the blocks collapsed and/or in an area of the design “canvas” (or design pane) that is outside your current browser window? If so, you can use the *expand* and *cleanup* functions of the blocks programming tool to expand all the blocks on your screen and to organize them in an easy-to-view (and easy-to-find) manner.

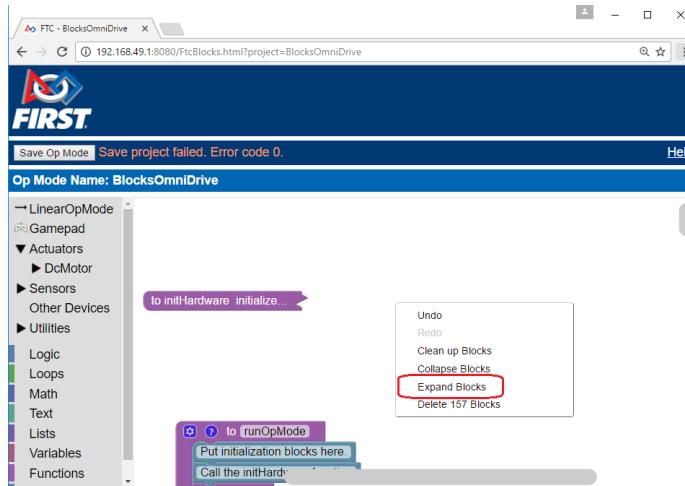


Figure 14 – You can right mouse Select “canvas” and select Expand Blocks to expand all of the blocks in your op mode.

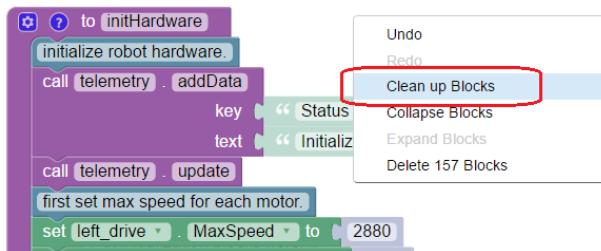
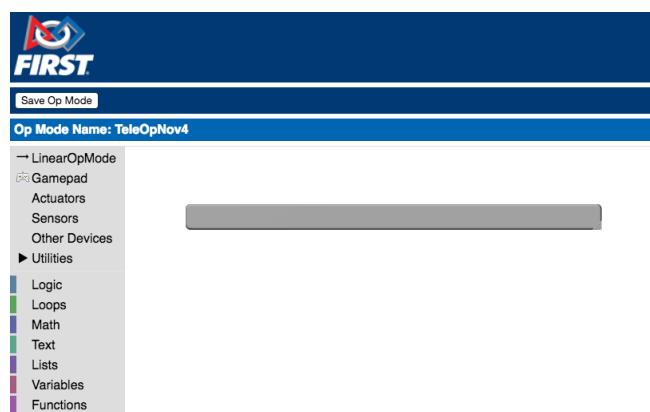


Figure 15 – Right mouse Select the canvas/design pane and select Clean up Blocks to organize all your blocks.

3. Are your programming blocks missing and you only see a solitary gray rectangular block on your screen? If this is the case, then you should check to see if the active configuration file for the Robot Controller is the same configuration file that you originally used to create the op mode. There is a bug in early versions of the blocks programming software that prevents the blocks server from properly rendering the programming blocks if the active configuration of the Robot Controller does not match the original configuration file used to create the op mode. More specifically, if some of the hardware devices (such as the DC motors or servos) from the original file are missing from the current configuration file, the blocks mode server will not properly display the programming blocks in the design pane.



*Figure 16 – Are the programming blocks missing and only a solitary gray rectangular block is visible?*

#### 10.2.4 Driver Station Appears Unresponsive

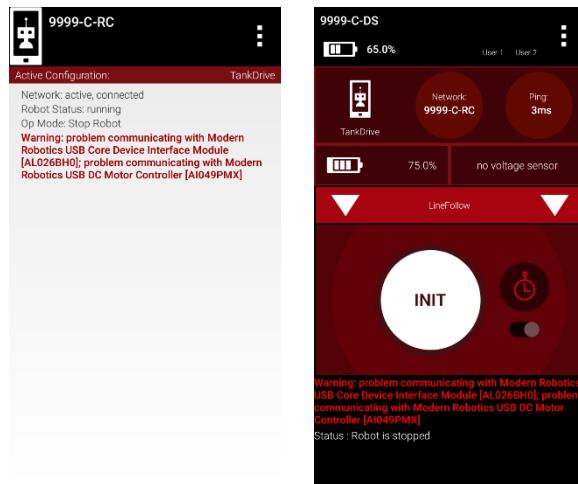
If you are ready to run an op mode, but the Driver Station is unresponsive and you cannot initialize or start your selected op mode, check the following items:

1. Verify that the Driver Station is properly paired to the Robot Controller.
2. Make sure that the Robot Controller is not in Programming Mode.
3. Check the *ping* times on the Driver Station main screen. The ping time is the average time it takes for the Driver Station to send a message to the Robot Controller and for the Robot Controller to acknowledged that it received the message. If the ping time is low (< 20 msec) the wireless connection between the Driver Station and Robot Controller is good. If the ping time is consistently high (> 50 msec) there could be some wireless interference in your venue that is causing the problems between the Driver Station and the Robot Controller.



#### 10.2.5 Warning: problem communicating...

If you are trying to run an op mode and you notice error messages like the ones displayed below, it could be that your wired connection between the phone and the electronic modules is bad.



*Figure 17 - A "problem communicating with..." message often indicates a bad connection between the phone and modules.*

If you notice this error message, here are some things you can try:

1. Verify that the USB cable connecting the phone to the Expansion Hub is secure and well connected.
2. Verify that the 12V power cables connecting the battery to the switch and the Expansion Hub are properly secured and connected. Also, verify that the power switch is in the on position.
3. Try to do a “Restart Robot” from the pop-up menu (touch the three vertical dots in the upper right hand screen of the Robot Controller or Driver Station apps).
4. If that does not work, disconnect the USB cable from the phone, then shut down the main power switch on the Expansion Hub. Wait for 5 seconds, then power the device back on and reconnect the USB cable to the phone.

#### **10.2.6 Additional Disconnect Incidents with Motorola E4, G5 and G5 Plus Phones**

We have observed a few incidents where teams who were using newer Motorola phones have had disconnects similar to the ones described in an earlier Troubleshooting article (see [Motorola E4, G5 and G5 Plus Phones Disconnecting Momentarily](#)). These newer Motorola phones have support for 5GHz Wi-Fi Channels.

What we observed was that these teams saw occasional disconnects, primarily during the autonomous portion of the match with these 5GHz enabled phones. For these newer Motorola phones, if the phone thinks that the WiFi radio is not being used, the phone will take control of the radio and scan the available WiFi channels. When the phone is doing this scanning, the WiFi radio appears to not be available to the FTC app. We do not know the exact purpose of these scans (the manufacturer will not disclose a reason), but if the system thinks the radio is available, it will scan the available channels sequentially. We have documented that when these scans occur with the newer phones (which have the additional 5GHz channels as part of the scan process) the scans can take a relatively long amount of time to complete (greater than 2 seconds) causing the watchdog safety mechanism in our apps to kick in. This watchdog mechanism will cause the Robot Controller to go into an Emergency Stop (E Stop) mode.

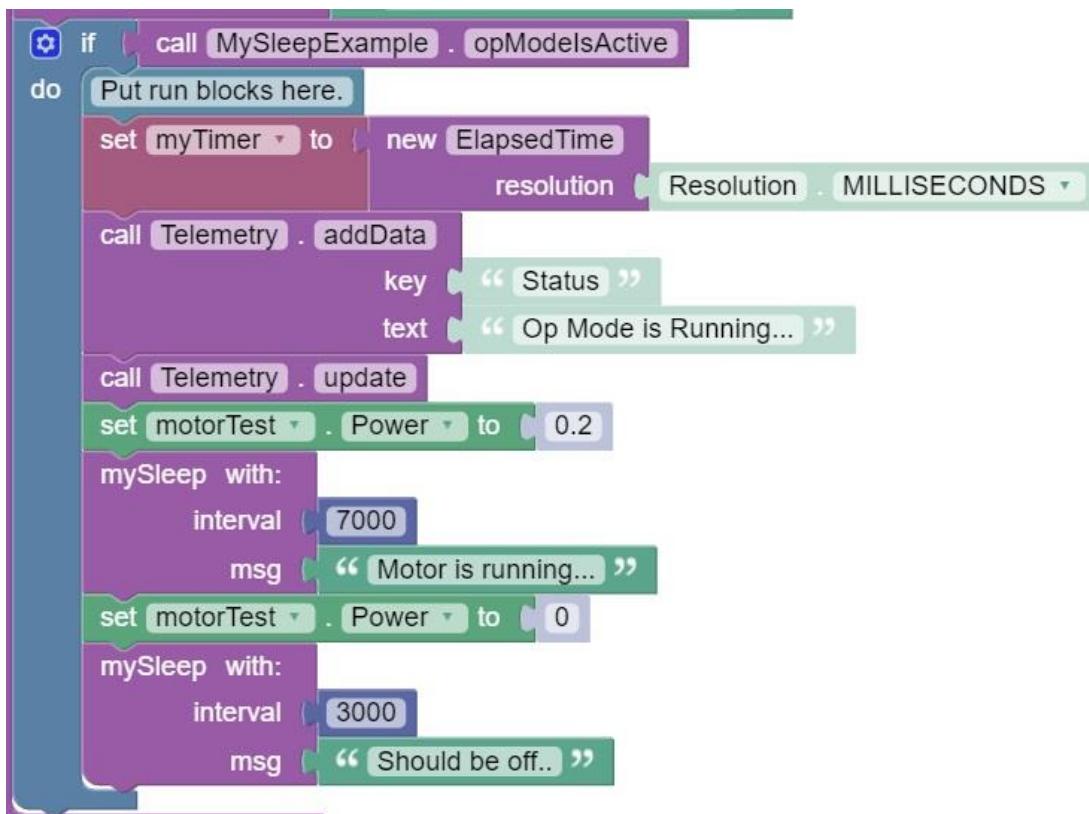
Note that the heartbeat messages sent by the Driver Station (and acknowledged by the Robot Controller) do not seem to suppress this scanning behavior on the Motorola phones. However, telemetry messages sent from the Robot Controller to the Driver Station do seem to suppress the scanning process.

A fix for the FTC apps has been developed that will address this Motorola scanning issue. However, this fix will not be available until ***after Championship 2019***.

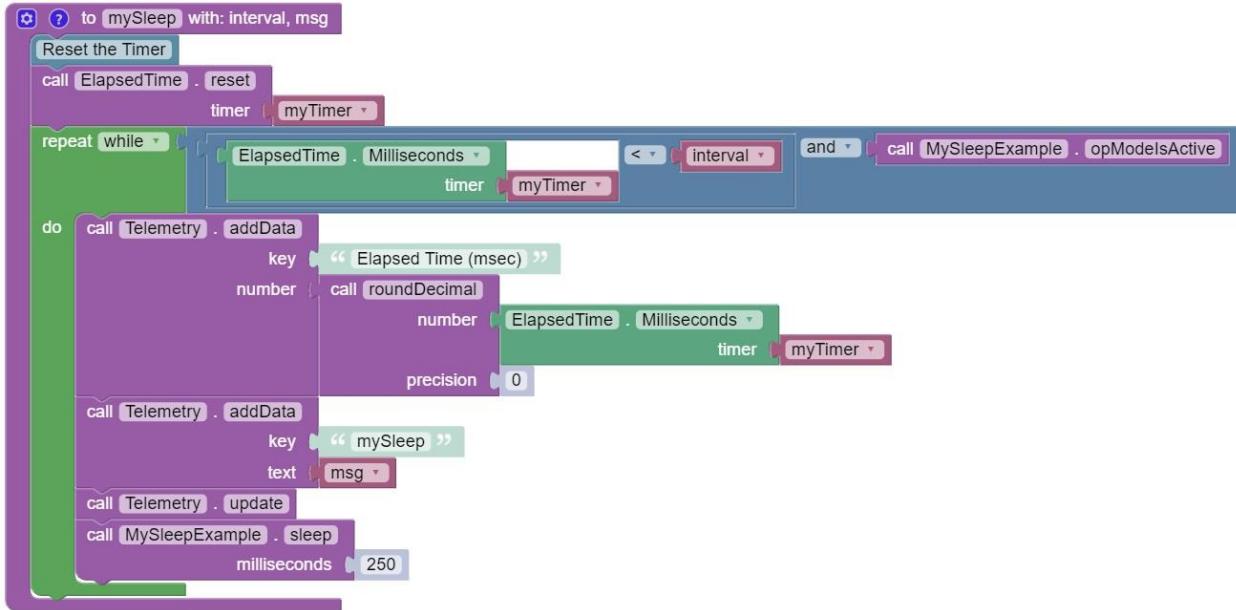
Teams who experienced these disconnects, even after they modified the wait for start process as described in the [previous Troubleshooting report](#), seemed to have the disconnects occur during the autonomous portion of their matches. An inspection of their autonomous op modes revealed that these teams often had linear op modes with long (greater than 2 seconds) sleep intervals. To work around the Motorola scanning issue, the teams rewrote their op modes so that if they needed to sleep for a fair amount of time, they would periodically send telemetry messages from the robot controller while they were sleeping to suppress this scanning behavior.

For a team that is using an iterative op mode, a telemetry statement can be placed in the loop() method to suppress the scanning behavior. For a team that is using a linear op mode, a custom sleep method can be made which will cause the Robot Controller to periodically send telemetry statements and avoid this scanning behavior.

In the following example Blocks op mode, the op mode uses a custom function called "mySleep" rather than the standard sleep block when it needs to sleep during its autonomous run.



In this example, the mySleep user-defined function uses a timer to "sleep" the requested amount. However, in between periodic sleeps, the mySleep function will also send out an occasional telemetry message to suppress the Motorola WiFi scanning behavior.



Similarly, if you are using Java to write your op mode, you can call a custom, user-defined sleep method instead of using the LinearOpMode's sleep method:

```
motorTest.setPower(0.2);
mySleep(7000, "Motor is running...");
motorTest.setPower(0);
mySleep(3000, "Should be off...");
```

The Java version of the mySleep method also uses an elapsed timer to send telemetry messages periodically while sleeping. These telemetry statements should suppress WiFi scanning on the Motorola devices.

```
public void mySleep(long interval, String msg) {
    ElapsedTime myTimer = new ElapsedTime(ElapsedTime.Resolution.MILLISECONDS);
    while(myTimer.milliseconds() < interval && opModelsActive()) {
        telemetry.addData("Elapsed Time (msec)", "%0.00f", myTimer.milliseconds());
        telemetry.addData("mySleep", msg);
        telemetry.update();
        this.sleep(250);
    }
}
```

### 10.2.7 Wi-Fi Blocker at Venue

We have had several incidents where FTC (and even FRC) events have been disrupted by the presence of Wi-Fi Blocking technology. A Wi-Fi Blocker or Suppressor is a device (often built-in as a feature of a wireless access point) that prevents people from using an unauthorized Wi-Fi network in the vicinity of the Blocker.

If there is a Wi-Fi blocker present, then teams will have trouble connecting their Driver Station devices to their Robot Controllers whenever they are in the range of the Blocker. The Driver Station might be able to see, for

example, the Robot Controller listed as an available device, but it would fail to connect or stay connected to the Robot Controller in the presence of the Blocker.

A good way to detect a Wi-Fi blocker is to take a pair of the problematic Android devices outside, away from the school's Wi-Fi system. If the devices can pair outside and stay connected (and can run op modes) outside, and if these devices suddenly disconnect once you move them back inside, then you might have a Wi-Fi Blocker present.

NOTE: We have even had an instance of a region that was at a school where they held an event in a prior year without issue. However, the teams were now having problems connecting their Android devices and it turned out that the school had installed a Wi-Fi Blocker in between the previous and current events.

If you have confirmed that there is a Wi-Fi Blocker present, then the best course of action is to work with the venue's IT administrator to disable the Wi-Fi Blocker for the event.

NOTE: FIRST has seen instances where the IT administrators did not even realize this Wi-Fi Blocking technology was present. It turned out that their wireless access points had this feature built-in, and they had to modify a configuration file (i.e., modify a "white list") to permit devices with a certain IP address range (192.x.x.x) to be allowed in their venue.

At some events, FTAs discovered a WiFi Blocker, but did not have an IT admin available to disable it. The FTAs were able to find the Wi-Fi Blockers in the venue that were located close to the competition field and unplug them for the duration of their event. The FTAs reported that after the devices were powered off, the teams were able to pair and control their robots successfully.

NOTE: that the Wi-Fi Event Checklist suggests that an FTA or similar technical volunteer conduct some preliminary tests in advance of an event to check for things like Wi-Fi Blockers. The best way to deal with a Blocker is to detect it well in advance of your event.

### [Wi-Fi Event Checklist](#)

Also, in the United States of America, the FCC has ruled Wi-Fi Blocking illegal:

### [FCC Warning on Wi-Fi Blocking](#)

## Appendix A – Resources

---

### **Game Forum Q&A**

<http://ftcforum.usfirst.org/forum.php>

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 User Name and Password for your team.

Volunteers that apply for a specific volunteer role will receive an email from [FTCTrainingSupport@firstinspires.org](mailto:FTCTrainingSupport@firstinspires.org) with their username and password to the forum. You will receive access to the forum thread specific to your role.

### **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 Tech Challenge Event On-Call Support**

These numbers are available for event personnel only. Please **do not** call these numbers if you are a team looking for a ruling, a decision, or assistance. We trust that you will not misuse this resource.

Day of Event Robot Control System Support: 603-206-2450

All other Day of Event support: 603-206-2412

### **FIRST Websites**

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

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

[FIRST Tech Challenge Volunteer Resources](#) – To access public Volunteer Manuals.

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

### **FIRST Tech Challenge Social Media**

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

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

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

[FIRST Tech Challenge Blog](#) – Weekly articles for the *FIRST* Tech Challenge community, including Outstanding Volunteer Recognition!

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

### **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!