



**FIRST
TECH
CHALLENGE**

*Training and
Support*



2016-2017 *FIRST®* Tech Challenge PushBot Build Guide

www.firstinspires.org

200 BEDFORD STREET ■ MANCHESTER, NH 03101

FOR INSPIRATION & RECOGNITION OF SCIENCE & TECHNOLOGY

FIRST
FOR INSPIRATION & RECOGNITION OF SCIENCE & TECHNOLOGY

Volunteer Thank You

Thank you for taking the time to volunteer for a FIRST® Tech Challenge event. FIRST® and FIRST® Tech Challenge rely heavily on Volunteers to ensure events run smoothly and are a fun experience for Teams and their families, which could not happen without people like you. With over 4,600 Teams competing annually, your dedication and commitment are paramount to the success of each event and the FIRST Tech Challenge program. Thank you for your time and effort in supporting the mission of FIRST!



Sponsor Thank You

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

FIRST® Tech Challenge Official Program Sponsor

**Rockwell
Collins**

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

PTC®

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

QUALCOMM®

Revision History		
Revision	Date	Description
1.1	08/15/2016	Initial Release

Contents

Introduction.....	5
What is FIRST® Tech Challenge?	5
FIRST Tech Challenge Core Values	5
<i>Gracious Professionalism®</i>	6
<i>Gracious Professionalism®</i> for Volunteers.....	6
Youth Protection Program	7
Youth Protection Expectations and Guidelines.....	7
NOTICE OF NON-DISCRIMINATION	7
What is the <i>FIRST®</i> Tech Challenge PushBot Build Guide?	8
FIRST Tech Challenge Team #2843 United States – PushBot Robot.....	8
Preparing the Robot Controller and Driver Station cell phones	9
Removal of SIM Cards	9
Naming the Cell Phone	19
Deploying the Robot Controller and Driver Station applications	23
Option 1: Downloading the applications from GitHub	24
Option 2: Android Studio.....	27
Recording the Controller Serial Numbers	95
Building the Robot	102
Robot CAD Drawing.....	102
Creating the Chassis.....	103
Adding Drive Wheels.....	110
Adding Omni Wheels	112
Adding the Arm Motor	116
Adding the Arm Rails	122
Adding the Controller Stack Assembly	125
Adding the Drive Motors.....	135
Adding the Power Distribution Module Assembly	140
Adding the Arm Motor Gears.....	148
Adding the Arm	150
Adding the Hands	154
Adding the Phone and Flag Holder Assembly	161
Adding the Cable Holder	166

Adding the Battery	168
Wiring the Robot.....	169
Wiring between the Phone and the Power Distribution Module (PDM)	169
Wiring the Hands to the Servo Controller	171
Wiring the DC Motor Cables.....	175
Wiring between the Controller Stack and PDM.....	180
Wiring the Drive Encoders	183
Installing the Battery and cleaning up wiring	187
Configuring the Robot.....	189
Creating a configuration file.....	189
Configuring the Servo Controller	194
Configuring the Drive Controller	199
Configuring the Auxiliary (Arm) Controller	203
Testing the Configuration	206
Testing the Robot	208
Assigning Gamepads	208
Pairing the Driver Station to the Robot Controller.....	210
Running an Op mode	216
Using the gamepads	218
Adjusting the Servos	220
Making the Robot Competition Ready	222
Adding Team Numbers	222
Appendices.....	223
Appendix A: Resources	224
Game Forum Q&A	224
FIRST Tech Challenge Game Manuals.....	224
FIRST® Headquarters Support.....	224
FIRST Website: firstinspires.org	224
FIRST Tech Challenge Social Media	224
Feedback.....	224
Appendix B: Bill of Material List	225
Appendix C: References.....	227
Appendix C: Project Team Profiles	227

Introduction

What is FIRST® Tech Challenge?

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

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

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

To learn more about FIRST Tech Challenge and other FIRST Programs, visit www.firstinspires.org.

FIRST Tech Challenge is MORE THAN ROBOTSSM! While competing, students develop personal and professional skills they will be able to rely on throughout their life.

FIRST Tech Challenge Core Values

Volunteers are integral to the FIRST community. FIRST Tech Challenge relies on Volunteers to run the program at many levels, from managing a region to Mentoring an individual Team. Our Affiliate Partners coordinate the program in each region or state. These Affiliate Partners fundraise, run Tournaments, hold workshops and demonstrations, market FIRST Tech Challenge locally, handle public relations, and recruit Volunteers and Teams. They are a tremendous resource for Mentors and FIRST would not exist without them.

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

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

Gracious Professionalism®

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

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

Some possible meanings of *Gracious Professionalism®* include:

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

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

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

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

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



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

- Dr. Woodie Flowers, National Advisor for FIRST

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

Gracious Professionalism® for Volunteers

It is a good idea to spend time going over this concept with Volunteers. Provide Volunteers with real-life examples of *Gracious Professionalism* in practice before, during, and after the event and recognize great *Gracious Professionalism* when you see it in action!

Youth Protection Program

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

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

Youth Protection Expectations and Guidelines

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

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

Forms are available here: <http://www.firstinspires.org/sites/default/files/uploads/about/FORMS.zip>

Information on the US Screening process is available here:

<http://www.firstinspires.org/sites/default/files/uploads/about/us-screening-2016-2017.pdf>

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

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

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

NOTICE OF NON-DISCRIMINATION

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

What is the FIRST® Tech Challenge PushBot Build Guide?

In early 2014, a design challenge was set by FIRST for experienced FIRST Tech Challenge Teams to design a basic Robot and manual for Rookie FIRST Tech Challenge Teams. The Robot and manual had to provide Rookie Teams with simple, basic instructions and images for building a PushBot that can be successful in competition for any Game Challenge. This manual is the result of the challenge and has been updated to match this year's technology and lessons learned from last year's competition.

PushBot 3 (PB3) needs to meet all FIRST Tech Challenge Competition guidelines using the current [Game Manual and Forum rules](#), but using ONLY the Competition Kit of Parts and materials.

As part of the task, the PushBot needs to have basic programming for autonomous mode; as well as a program for manual mode. The code is to be developed in the Java programming language and deployed using the Android Studio IDE.

Depending on Team size and student interests, some sections of this manual can be done in parallel. One group can build the Robot, while another group is installing the software, and a third group can build the electronics test bench. The efforts of these groups would come together during the wiring and testing portion.

FIRST Tech Challenge Team #2843 United States – PushBot Robot.

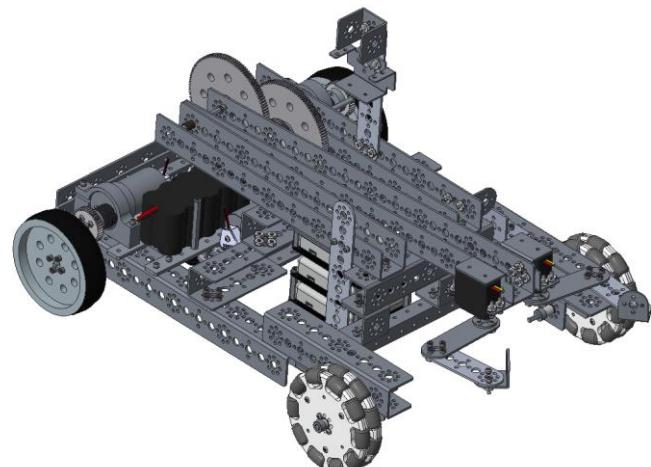
2016 – 2017 Robot Name: PB3 (PushBot 3)

Purpose: Assist new FIRST Tech Challenge competitors to develop a sense of how a FIRST Tech Challenge Robot is constructed with the TETRIX Robotic Construction system and is programmed in Java.

Design: The PushBot has 3 DC motors and 2 servos. Its design is based around a rectangular frame. This allows structural engineering principles to be explored by Teams. The Robot has an 'arm' powered by a single DC motor, a 'hand' that is driven by two servos, and 'feet' powered by two DC motors. The 'arm' is geared down to allow the DC motor to lift heavier objects. PB2 has high clearance for driving over objects, but a lower center of gravity than last year. Last year, it experienced many roll-overs down the mountain. PB2 can push objects using its gripper or a front panel can be installed to 'push' objects. This Guide will assist rookie FIRST Tech Challenge competitors in gaining an understanding of how a simple Robot built with the TETRIX Construction System can be programmed in Java. It will also guide Teams through using the new electronics available in the 2016-2017 season.

The Robot is constructed with the TETRIX Kit of Parts and an electronics kit, which includes cell phones as the communication devices and other USB devices that provide communication with new USB modules.

A link to the TETRIX Kit of Parts description is available on the [FIRST Tech Challenge Team Management](#) page, under "Budget".



Preparing the Robot Controller and Driver Station cell phones

Removal of SIM Cards

To prevent the phones from attempting a connection with a mobile network, the SIM cards should be removed from both the driver station phone and the Robot controller phone.

Step 1: Gather both phones and both cases. If the phones are on, then turn them off. The power button is on the top right edge of the phone.



Step 2: Remove the back covers from the phones, leaving the phones face down. There is a gap between the front and back covers located on the lower right hand corner of the device that can be used to remove the back cover. Inserting a fingernail in the gap and then sliding the fingernail all around the case will cause the case to pop apart.



Step 3: The batteries are on the left and the SIM cards (circled) on the right. Gently push the SIM card INTO the phone.



Step 4: Let the spring gently push the SIM cards from the phones. Pushing the SIM cards activates the spring mechanisms; releasing pressure allows the springs to push the SIM cards out of the phones.



Step 5: Store the SIM card in a safe place. In the image below, the cards have been taped to the inside back cover of the phones.



Step 6: Replace the back covers.



Step 7: Place the protective cases' back covers onto the back of the phones.



Step 8: Turns phones face up.



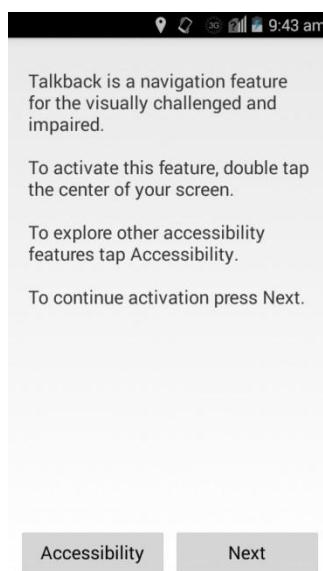
Step 9: Place the protective case front covers into the front of the phones.



Step 10: Turn phones on. The circle is where the power switch is located. The phones will begin booting. The sequence of images shows the booting sequence.

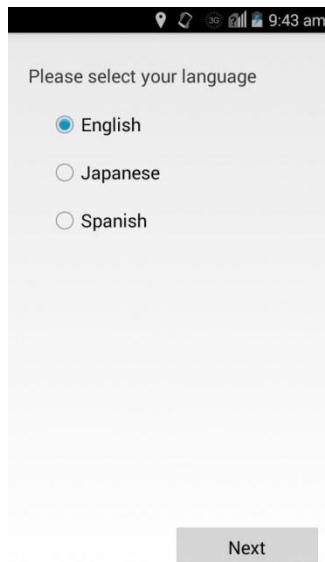


Step 11: Choose 'Next' to accept the default talkback setting.

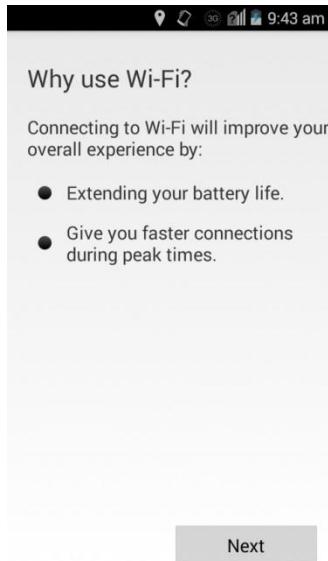


This and all following steps may only show one image, but apply the step to both phones.

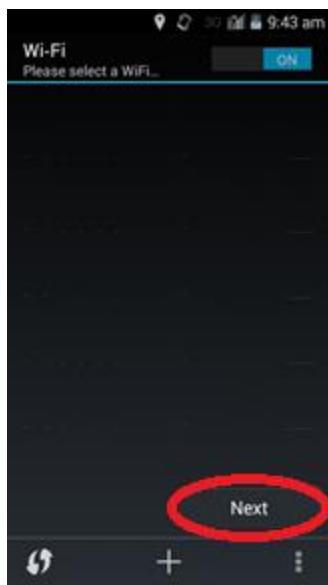
Step 12: Select 'English'.



Step 13: Choose 'Next' to configure Wi-Fi.



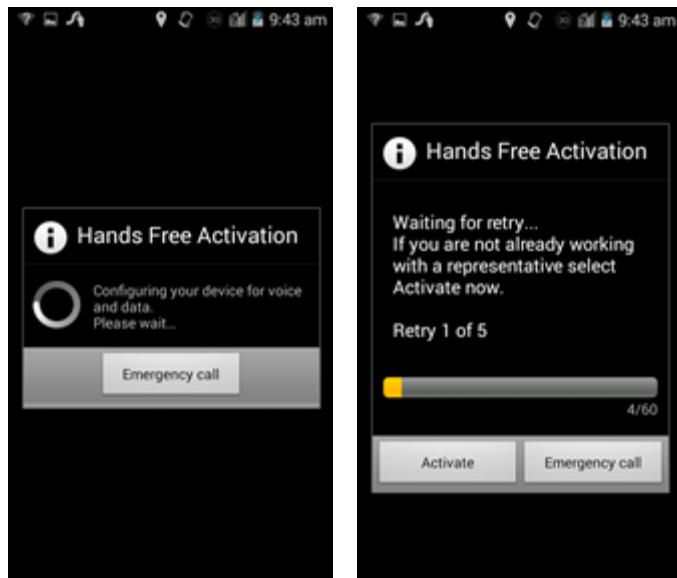
Step 14: Do not connect to Wi-Fi at this time. Select 'Next'.



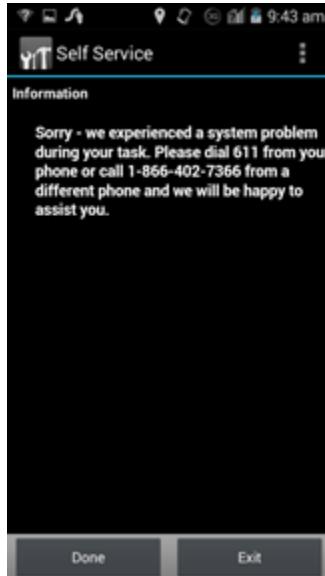
Step 15: When the UICC Error screen is shown, select 'OK'.



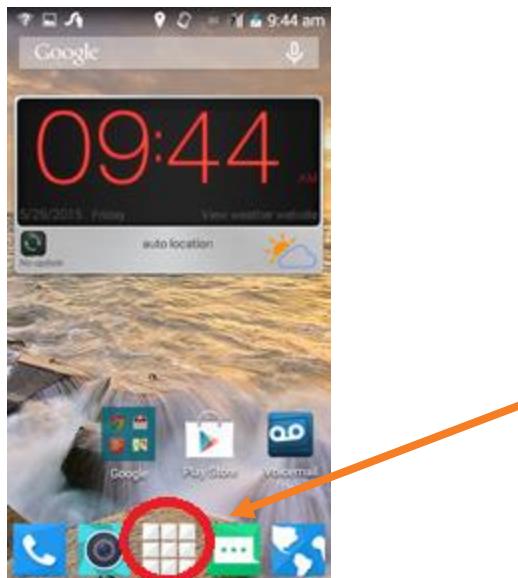
Step 16: Select 'Activate' when the Hands Free Activation activity provides the option. When the activity first starts, it provides the ability to make an emergency call, but then will provide an 'Activate' button.



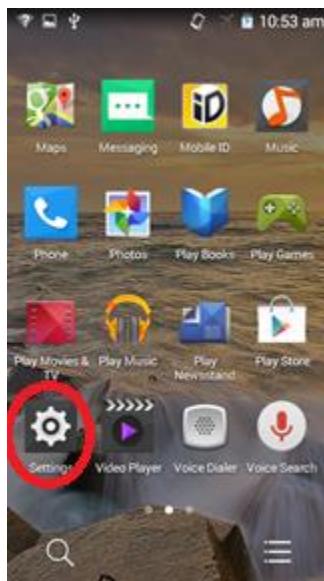
Step 17: The activation should fail. Select ‘Done’.



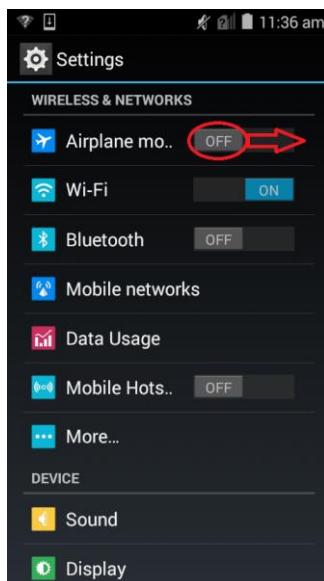
Step 18: Android will return to the home screen. Open the application activity (also known as the App Tray) by selecting its icon (shown in the red circle).



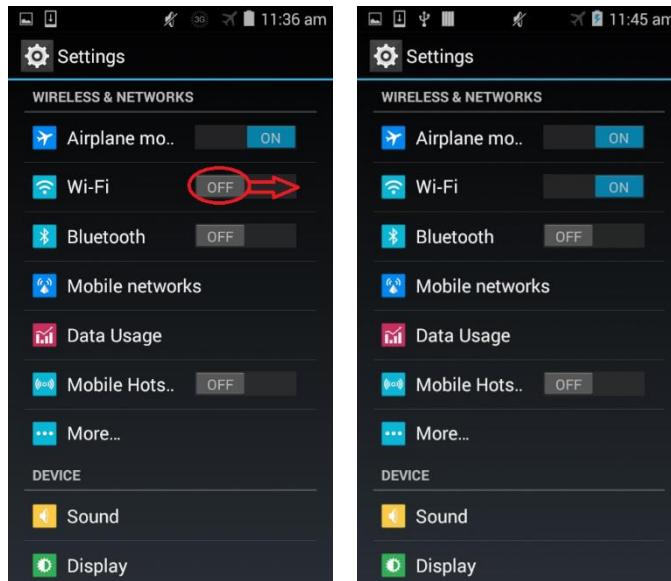
Step 19: Open the settings activity by selecting its icon (shown in the red circle). It might be necessary to swipe the screen left or right.



Step 20: Put the phones into Airplane Mode by sliding the button from left to right until the icon comes into view.



Step 21: Turn Wi-Fi on, if it's off, by sliding the button from left to right. The phones' screens should appear as the image on the right.



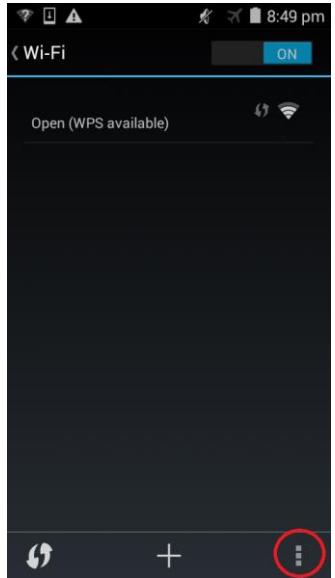
Naming the Cell Phone

According to the Robot software rules in the [Game Manual](#), the phones need to be named properly. The first portion of the name will be the Team's number. The Driver Station phone's name will be postfixed with “-DS”. The Robot Controller's will be postfixed with “-RC”.

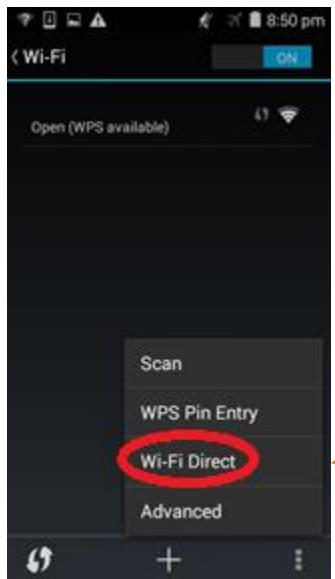
Step 1: (Continuing from the previous section – the settings page is displayed) Select Wi-Fi.



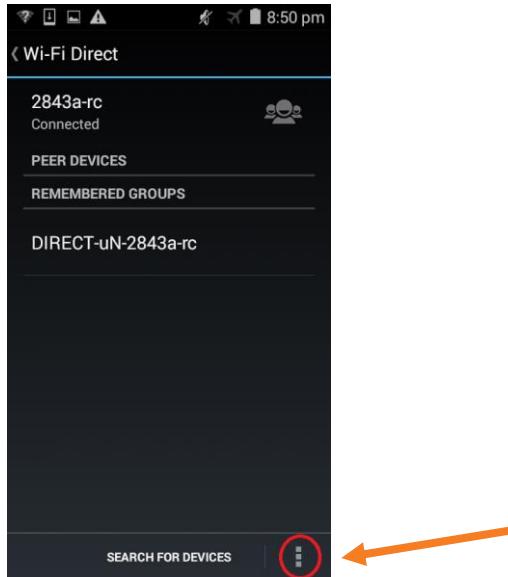
Step 2: Select the overflow settings (three dots on the bottom right).



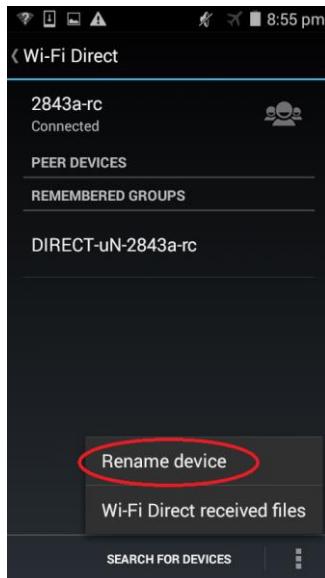
Step 3: Select Wi-Fi Direct.

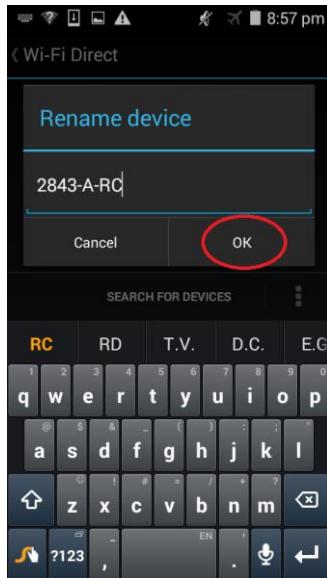


Step 4: If “RENAME DEVICE” is not shown at the bottom of the screen, then select the overflow settings and then select “Rename device”. In the image below, the phone had already been named. New phones will have a generic name.

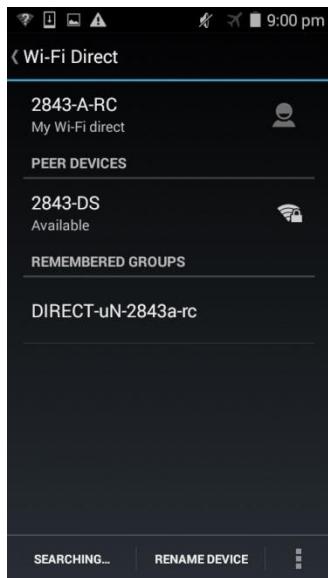


Step 5: Enter the new name according to the rules in [Game Manual Part I](#).



Step 6: Select OK.

Step 7: From the page shown below, return to the settings activity by pressing the back button. It may take several presses to return to the settings activity.



If you have multiple phones,
add “-A-”, “-B-”, etc. in
between the Team Number
and the -RS or -DS postfix.

Deploying the Robot Controller and Driver Station applications

For this section choose one phone to be the Driver Station and the other to be the Robot Controller.

- The Driver Station phone will be the one
 - connected to the gamepads that allow the students to operate the robot
 - hosting the Driver Station app
 - having the postfix “-DS”
- The Robot Controller phone will be the one
 - connected to the Power Distribution Module
 - hosting the Robot Controller application
 - having the postfix “-RS”

There are quite a few options for a Team to control the actions of the robot. This document doesn't cover all of them, but does provide two options. The two options described in this document are [Downloading from GitHub](#) and [Deploying using Android Studio](#). The first option requires only the contents of the kit of parts. Android Studio and the other options require an on-line account, a personal cell phone, a computer and possibly installed or Internet-available software. All options require the Kit of Parts.

- Downloading
 - GitHub: The Robot Controller and Driver Station phones need access to the Internet via a local network.
 - Google Play: The Robot Controller and Driver Station phones need access to the Internet via a local network and a Google Account.
- Sideloadng
 - Personal Cell Phone and Computer: A personal cell phone downloads the apps from GitHub; a computer connects to the phone and sideloads the apps onto the RC and DS phones.
 - Personal Computer with Internet Access: A personal computer downloads the apps from GitHub and sideloads them onto the RC and DS phones.
- Deploying
 - Android Studio: A personal computer with access to the Internet downloads Java, Android Studio and the FTC SDK to deploy the Robot Controller app onto the RC phone. One of the other methods is used to install the Driver Station phone onto the DS phone.
 - MIT App Inventor: A personal computer with access to the Internet downloads a virtual machine player and the MIT App Inventory virtual machine, which is used to deploy the Robot Controller app onto the RC phone. One of the other methods is used to install the Driver Station phone onto the DS phone.

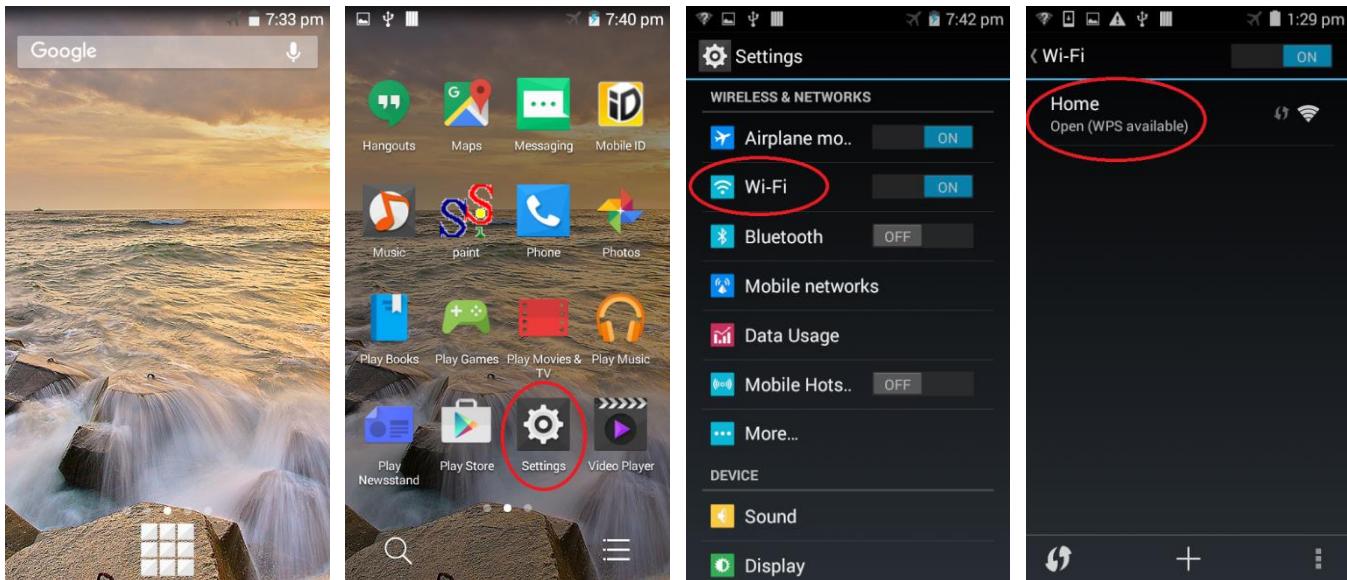
NOTE: Android Studio has required computer properties. Check your computer before downloading.

Option 1: Downloading the applications from GitHub

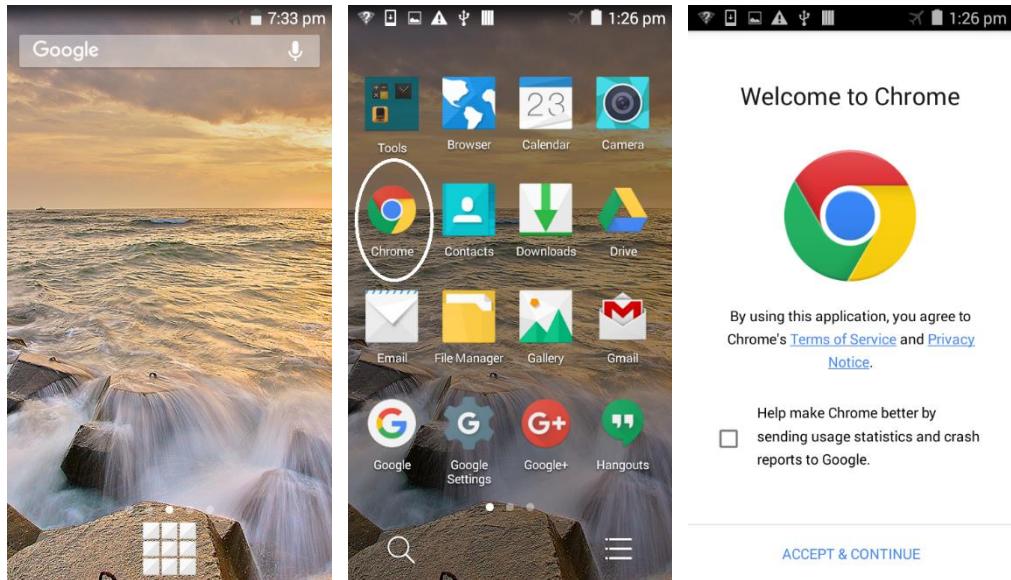
This option is a good choice when only the phone and an Internet connection is available, not a PC, personal cell phone, or Google account. It does, however, require an Internet connection to which the Robot Controller (RC) and Driver Station (DS) can connect.

Step 1: Connect the Robot Controller (RC) to the Internet using Wi-Fi. Open the Settings App. Open the Wi-Fi app by selecting the text, not the On/Off slider. The phone may display a pop-up that indicates the access point is not secure. **It is the Team's responsibility to weigh the risks of connecting to an unknown access point.** It is best to use a known Wi-Fi access point. Also, the Chrome app may display an activity that stores your bookmarks, history, etc. Use the settings to control access to personal information.

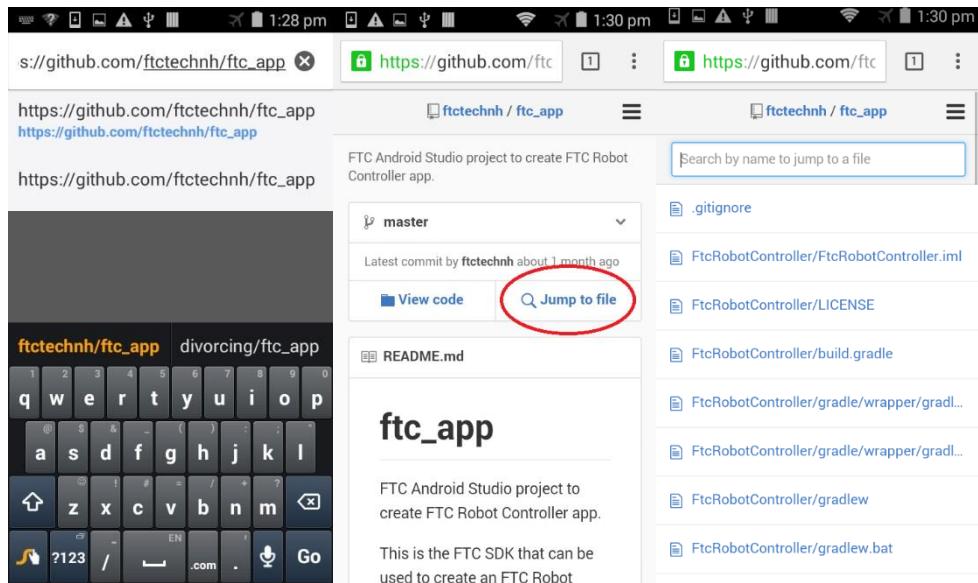
NOTE: that the steps in this section do not use bookmarks or passwords.



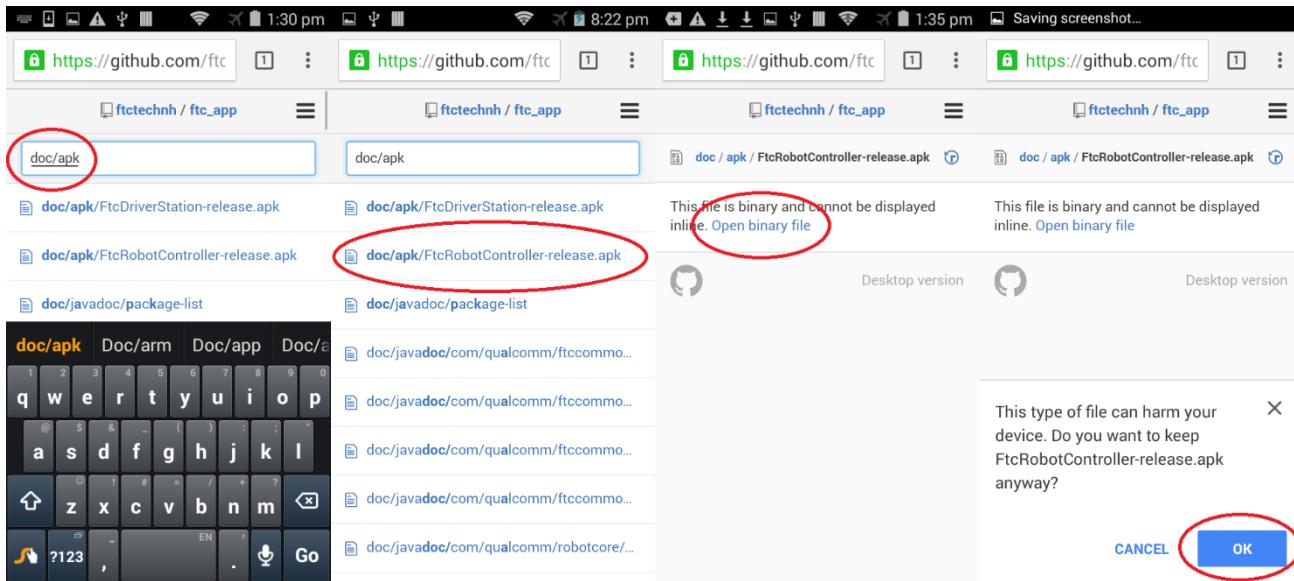
Step 2: Open Chrome. From left to right, the images show one possible method.



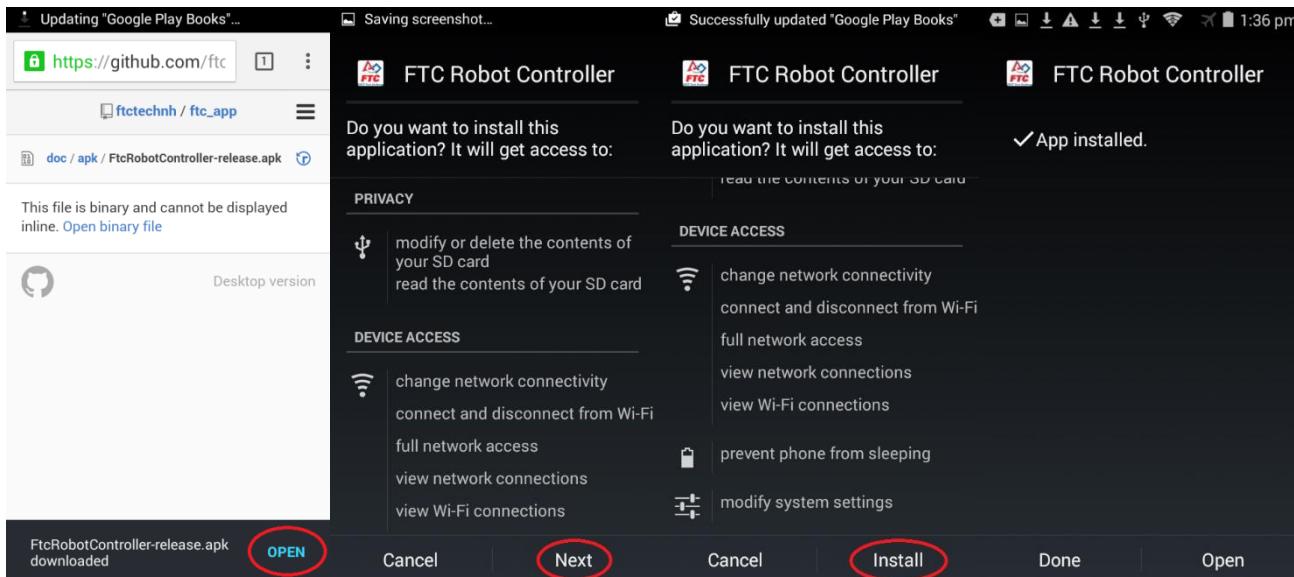
Step 3: Enter the following link into the address bar: https://github.com/ftctechnh/ftc_app. Select 'Jump to file'.



Step 4: Enter the following into the text box: **doc/apk**. The list will update as text is typed. Short-click on the **ftcRobotController-release.apk** entry. Select ‘Open binary file’. Select ‘OK’ and the file will be downloaded to the phone.



Step 5: Select Open and the phone will install the Robot Controller app. Click Next to view all of the necessary permissions. This completes the installation of the app. The phone is not needed again until a later section of this document.



Step 6: Perform the above steps using the Driver Station phone with the exception that the **FtcDriverStation-release.apk** entry should be downloaded and installed.

Option 2: Android Studio

Android Studio is one of the Integrated Development Environments approved for FIRST Tech Challenge Team use by FIRST. It is a text-based IDE that can be used to build, deploy, and debug Java applications. The IDE is available without cost for Windows, Mac OS X, and Linux and can be downloaded from the <https://developer.android.com> website. Select “Develop”, then “Tools” and then the next selection will depend onto which operating system Android Studio will be installed. This document will show all of the steps to download the Windows 64-bit version of Android Studio and install it onto a Windows 10 computer. Windows 10 is the newest operating system released by Microsoft.

Android Studio is built around Java, so the Java Development Kit (JDK) must be installed before Android Studio is installed. This year’s FTC SDK no longer requires version 7, so this document will show all of the steps to download and install version 8 onto a Windows 10 computer.

The FTC SDK contains the Robot controller module which is the application that runs on the cell phone located on the Robot. That cell phone controls the electronics on the Robot and communicates with another cell phone that acts as the driver station. The driver station allows the Team to select Op modes, run the selected Op modes and send gamepad inputs to the Robot controller.

The PushBot software package contains Op modes designed specifically for the PushBot. A simple manual (i.e. TeleOp) Op mode is included in the package that allows a Team to control the Robot’s drive wheels using one gamepad’s joysticks, control the Robot’s arm using a second gamepad’s left joystick, and control the Robot’s hands (i.e. servos) using two of the buttons on the second gamepad.

Installation will begin with the FTC SDK which contains the PushBot software. It will continue with the installation of Android Studio and finish with deploying the Robot controller application to the cell phone.

NOTE: The sections in this portion build on actions made in the previous sections. It is recommended that sections be completed in the order shown.

This document shows some steps that use the Windows Start key. After pressing the Windows key (shown below), typing the name of the application will cause a list of known applications to be shown. The arrow keys can be used to select an item in the list. Once a list item is selected, pressing enter will start the application.

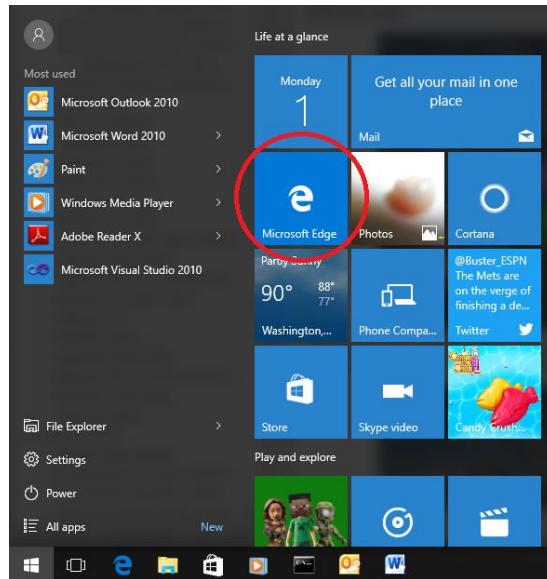
**NOTE: Before beginning,
charge the phones.**



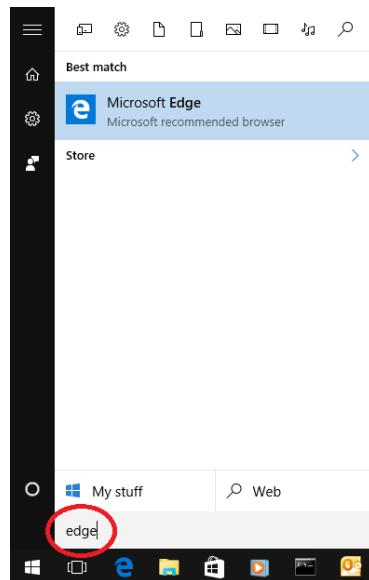
Opening an Application under Windows 10

Some of the steps in this section use the new Microsoft browser known as Edge. This section shows how to start an application using the Windows logo key. A similar procedure can be used to open other applications (as shown in the next section – [Opening File Explorer](#)). The instructions for opening Edge are shown once here and referenced in the sections that follow. Two options are shown below.

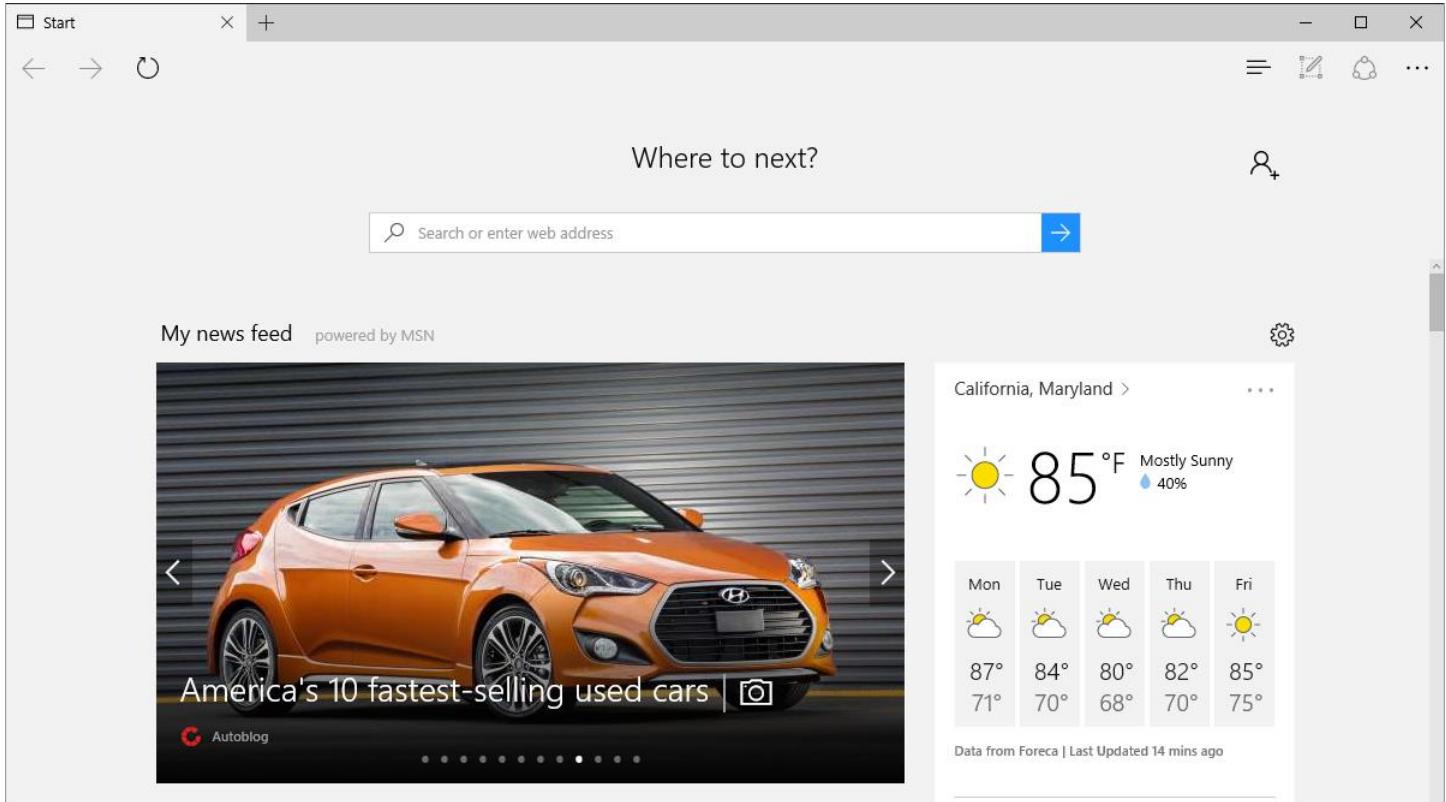
Option A: Press the Windows key and select the Microsoft Edge icon from the Windows start screen (shown in the red circle).



Option B: Press the Windows key and type the name of the application – in this case the name is ‘Edge’. Remember to press enter after typing the name.



The Edge application is shown below.

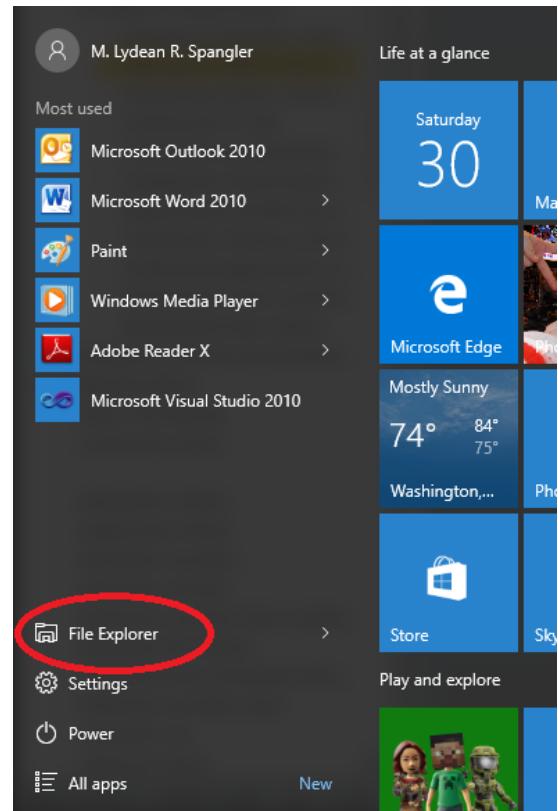


Opening File Explorer (aka Windows Explorer)

Some of the steps in this section use File Explorer (FE). The instructions for opening it are show once here and referenced in the sections that follow. Two options are shown below.

Option A: Press the Windows key and select the File Explorer icon from the Windows start screen (shown in the red circle).

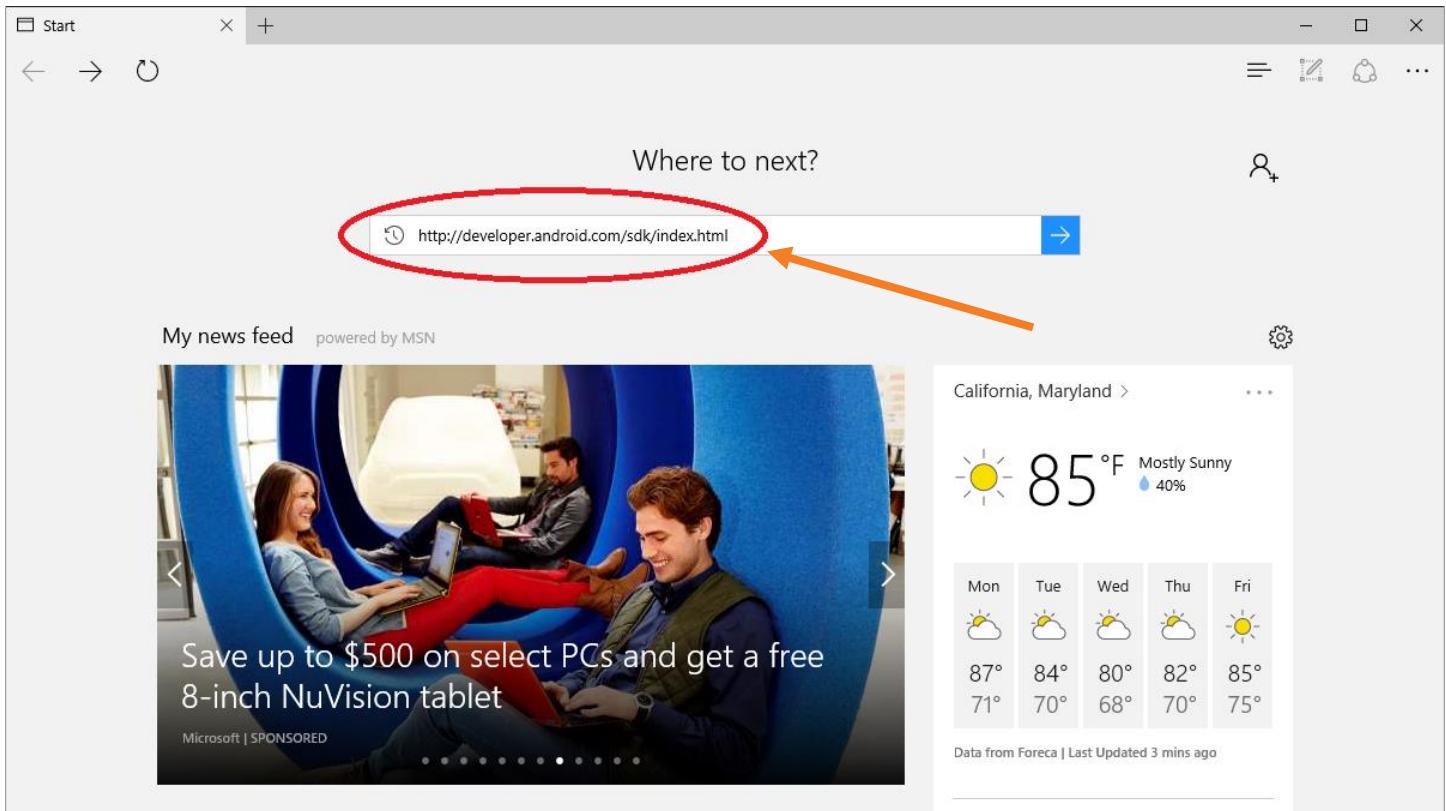
Option B: Press the Windows key and type the name of the application – in this case the name is 'file explorer'. Remember to press enter after typing the name.



Determining whether Android Studio can be installed

Android Studio publishes the qualities a computer must have to be able to run the IDE. Before installing the Android Studio IDE, Java, or the FTC SDK, check whether the machine can run the IDE. If the machine can't support the Android Studio IDE's minimum requirements, then there is no need to install Java or the FTC SDK, so perform this check before installing anything. If the test fails, then a different machine will be needed.

Step 1: Open Edge and enter '<https://developer.android.com/studio>' into the Address field.



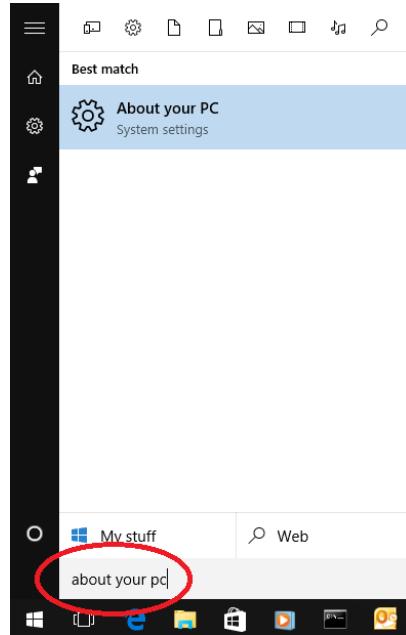
Step 2: Scroll down until System Requirements are shown. The requirements for the Windows operating system are shown in the red box.

The screenshot shows a web browser displaying the Android Studio download page at developer.android.com/studio/index.html. The left sidebar has 'DOWNLOAD' selected. The main content area is titled 'System Requirements' and lists requirements for Windows, Mac, and Linux. The Windows requirements section is highlighted with a red box. At the bottom, there is a cookie consent banner.

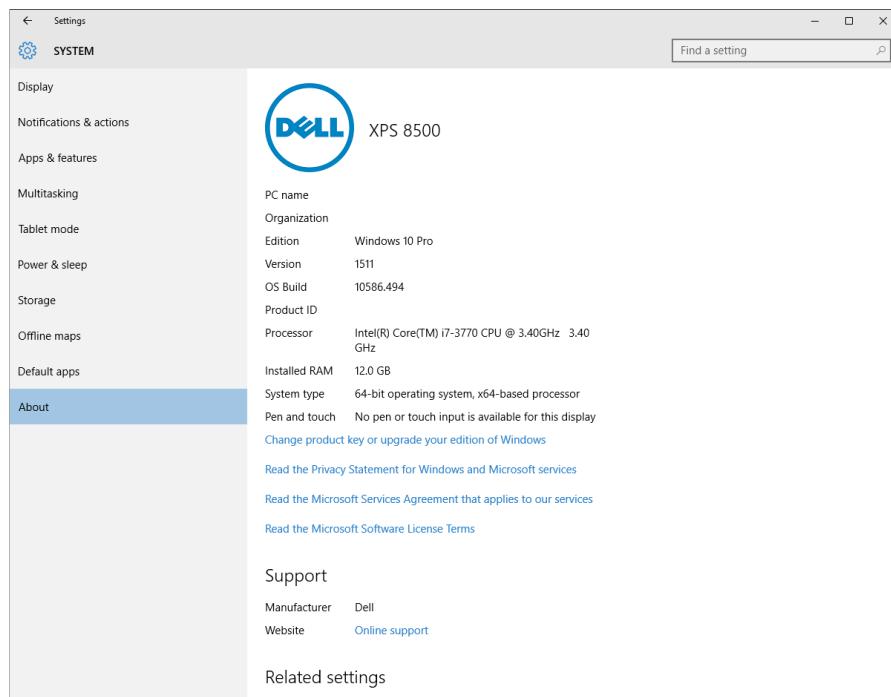
Platform	Requirements
Windows	<ul style="list-style-type: none">Microsoft® Windows® 7/8/10 (32- or 64-bit)2 GB RAM minimum, 8 GB RAM recommended2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)1280 x 800 minimum screen resolutionJava Development Kit (JDK) 8For accelerated emulator: 64-bit operating system and Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality
Mac	<ul style="list-style-type: none">Mac® OS X® 10.8.5 or higher, up to 10.11.4 (El Capitan)2 GB RAM minimum, 8 GB RAM recommended2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)1280 x 800 minimum screen resolutionJava Development Kit (JDK) 6
Linux	<ul style="list-style-type: none">GNOME or KDE desktopTested on Ubuntu® 12.04, Precise Pangolin (64-bit distribution capable of running 32-bit applications)64-bit distribution capable of running 32-bit applicationsGNU C Library (glibc) 2.11 or later2 GB RAM minimum, 8 GB RAM recommended2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)1280 x 800 minimum screen resolutionJava Development Kit (JDK) 8For accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality, or AMD processor with support for AMD Virtualization™ (AMD-V™)

This site uses cookies to store your preferences for site-specific language and display options. OK

Step 3: Press the Windows key and type ‘About Your PC’. Remember to press enter after typing this.



Step 4: Match the information from the following window with the requirements shown in step 2 – the process is shown below the following image. Note the System Type; it will be used later when installing the Java Development Kit. For this document the system type is 64-bit.



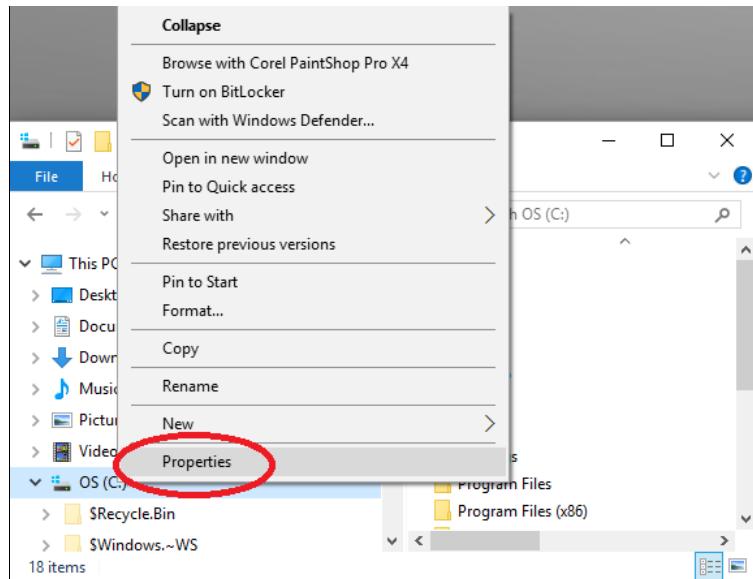
Requirement 1: Microsoft Windows 7/8/10 (32 or 64-bit). Since Windows 10 is shown, the requirement has been met.

PC name
Organization
Edition Windows 10 Pro
Version 1511
OS Build 10586.494
Product ID

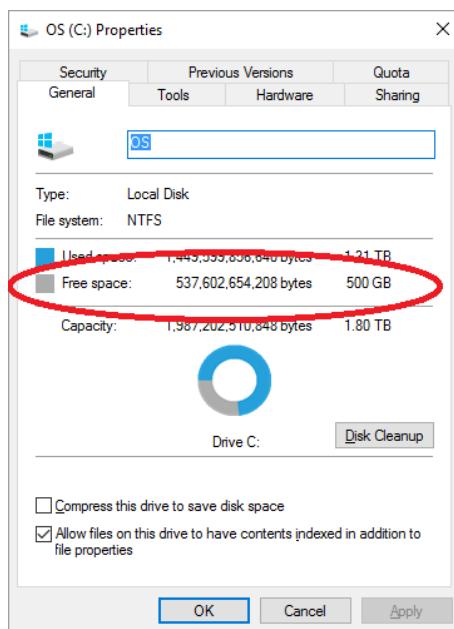
Requirement 2: 2GB RAM minimum, 8GB RAM recommended. Since 8GB is greater than 4GB, the requirement has been met.

Edition	Windows 10 Pro
Version	1511
OS Build	10586.494
Product ID	
Processor	Intel(R) Core(TM) i7-3770 CPU GHz
Installed RAM	12.0 GB
System type	64-bit operating system, x64-based
Pen and touch	No pen or touch input is available

Step 6: To compare the hard disk space, another Windows tool will be needed. Open File Explorer (known as Windows Explorer on older systems). Right click “OS (C:)” and select Properties. If the computer has a touch screen, hold a stylus or finger on “OS (C:)” until the context menu is shown.

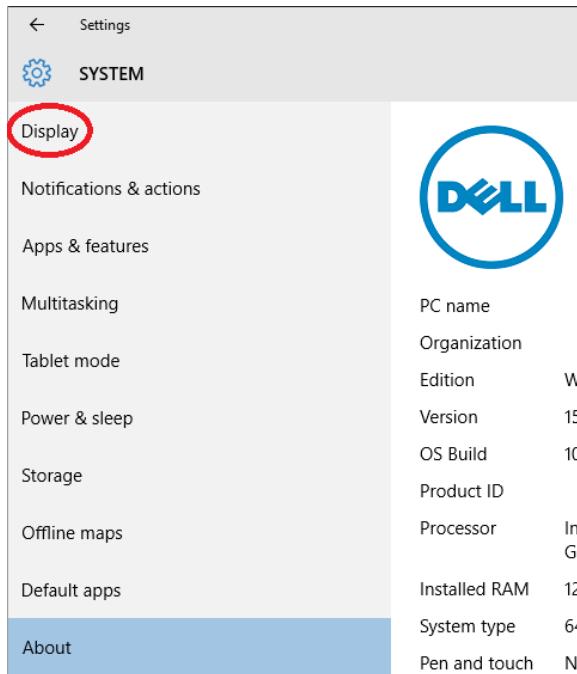


Requirement 3: 2 GB hard disk space. Since 500 GB is greater than 2 GB (1GB=1000MB), the requirement has been met.

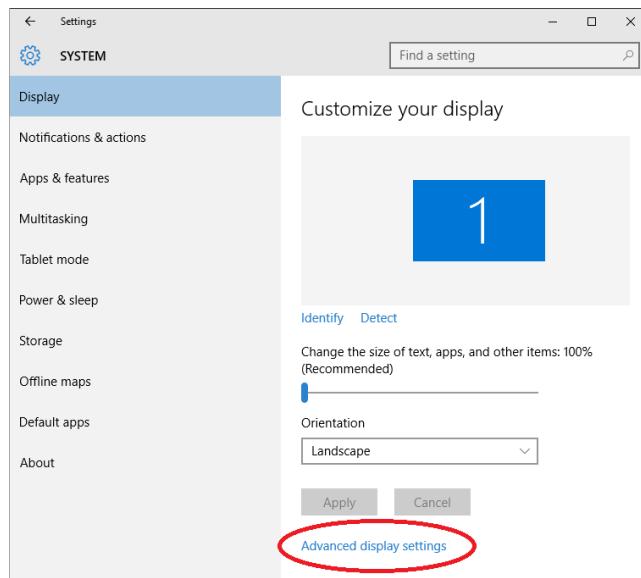


Requirement 4: At least 1.5 GB for Android SDK and emulator system images. Android Studio uses the Android SDK and thus requires hard disk space for the SDK. Since 500 GB (refer to the image above) is greater than 1GB, the requirement has been met.

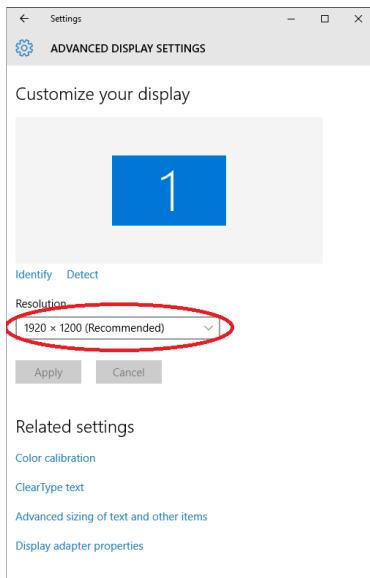
Step 7: To compare the screen resolution, return to the PC Settings page (see steps 3 and 4) and select Display. It might be necessary to enlarge the window by using the maximize button.



Step 8: Select Advanced display settings.



Step 9: Click on any portion of the Resolution drop box and a list of available resolutions will be displayed.



Requirement 5: If the 1280 x 800 minimum screen resolution is shown then the requirement has been met. On the example system, a resolution of 1920 X 1200 is supported. Since the supported width (1920) is greater than (or equal to) the minimum required width (1280) and the supported height (1200) is greater than (or equal to) the minimum required height (800), the requirement has been met.

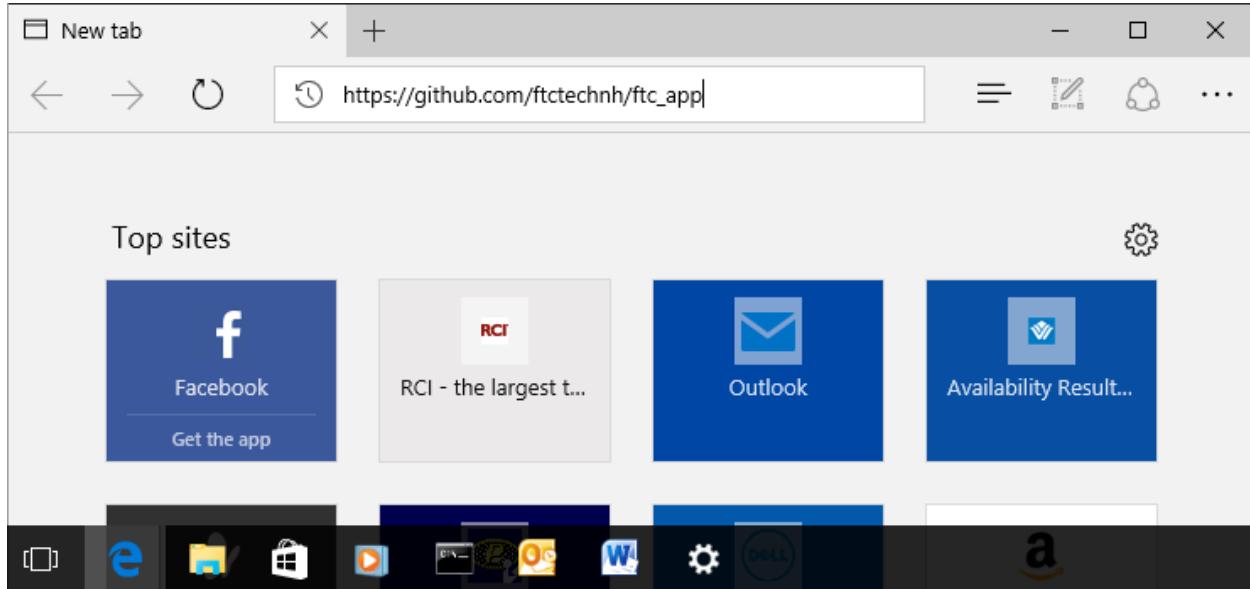
Requirement 6: Java Development Kit (JDK) 8. Since this document describes the process to install JDK 8, this requirement will be met.

Optional Requirement 7: Since the last requirement (accelerated emulator) is optional, this document will not determine whether this requirement has been met.

Installing the FTC SDK

QualComm developed the FTC SDK (Software Development Kit) to allow a cell phone to communicate with other electronic components such as the DC motor controller, the servo controller, and the legacy module. This SDK allows Teams to write Java classes known as op-modes, which allow Teams to customize the functionality of the Robot. These Op modes are compiled using the SDK and deployed to the cell phone using Android Studio. The FTC SDK must be installed onto the same machine that will host the Java SDK and the Android Studio IDE. The FTC SDK is located at https://github.com/ftctechnh/ftc_app.

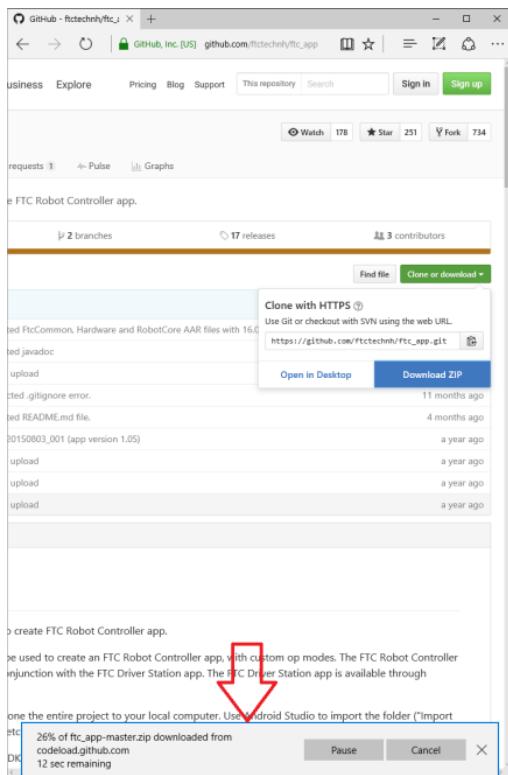
Step 1: Open Edge and enter 'https://github.com/ftctechnh/ftc_app' into the Address field.



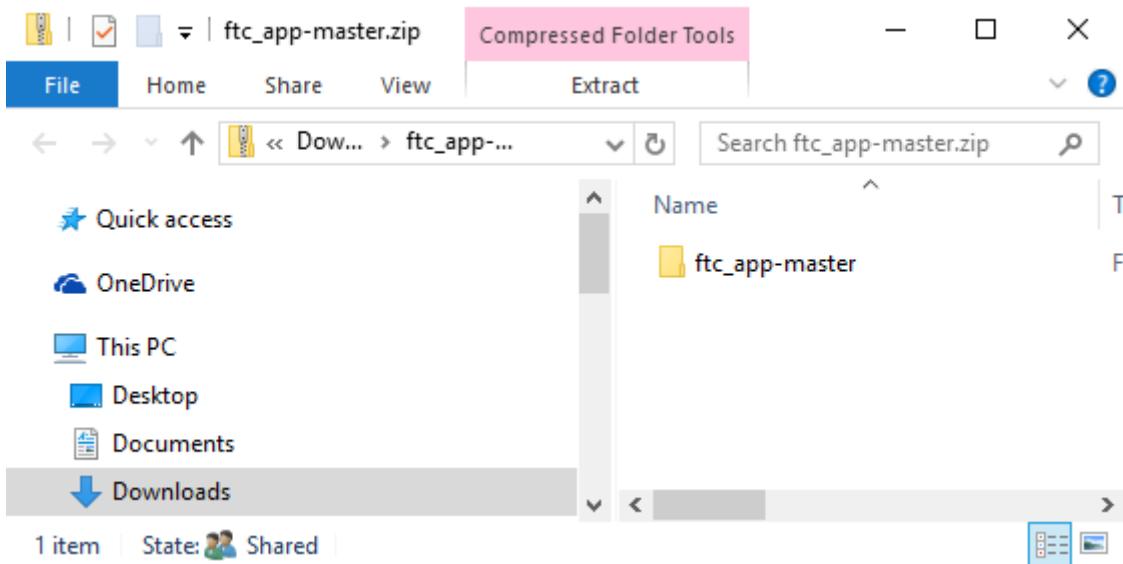
Step 2: After entering the URL from the above step, the following page is shown. Select the “Clone or download” button. Another dialog will display (the right image).

A screenshot of the GitHub repository page for 'ftctechnh/ftc_app'. The page shows basic repository statistics: 39 commits, 2 branches, 17 releases, and 3 contributions. On the right side, there's a 'Clone or download' button highlighted with a red circle and an orange arrow pointing to a larger callout box. This callout box contains a 'Find file' button, a 'Clone with HTTPS' section with a URL input field containing 'https://github.com/ftctechnh/ftc_app.git', and a 'Download ZIP' button also highlighted with a red circle.

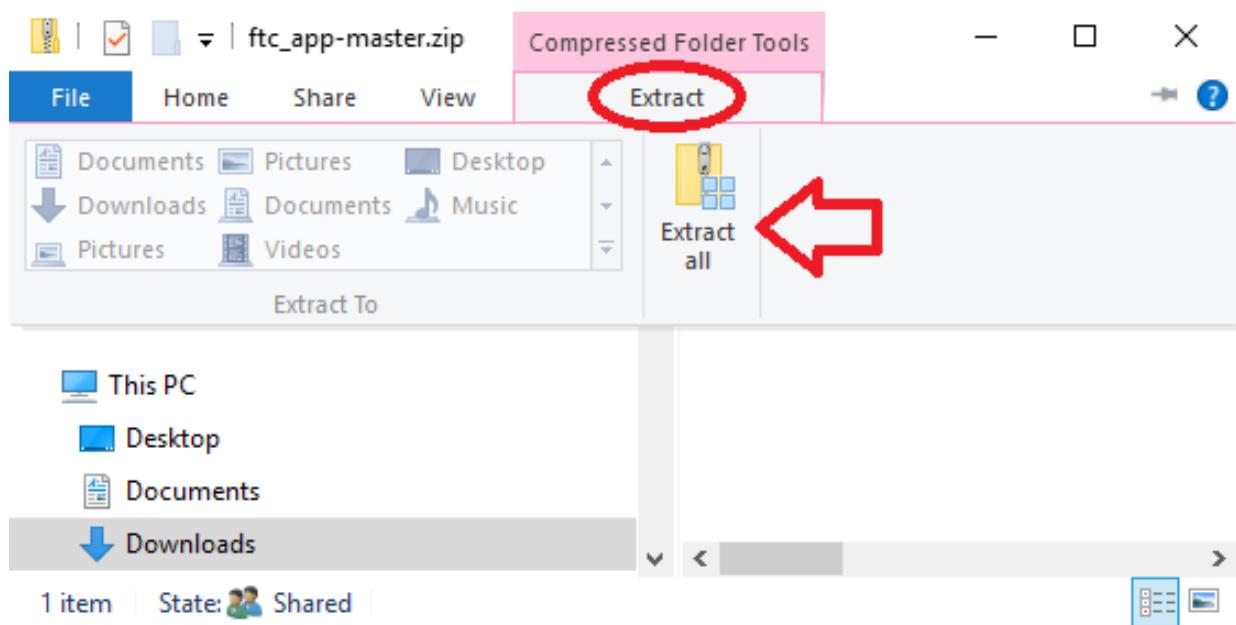
Step 3: After selecting the Download ZIP button, the browser displays a banner that indicates the file is downloading.



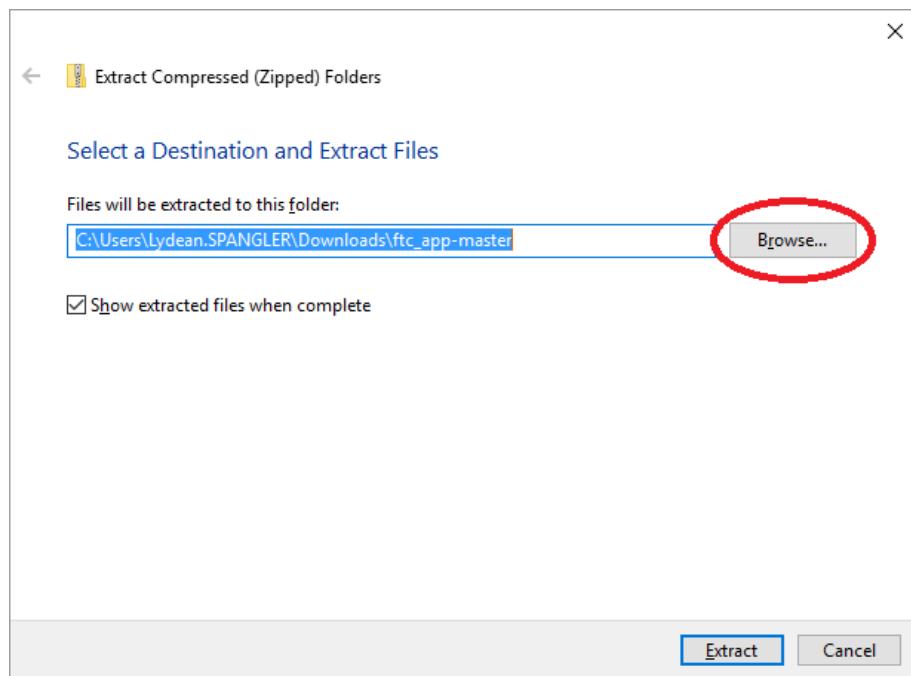
Step 4: After the file has finished downloading, select Open.



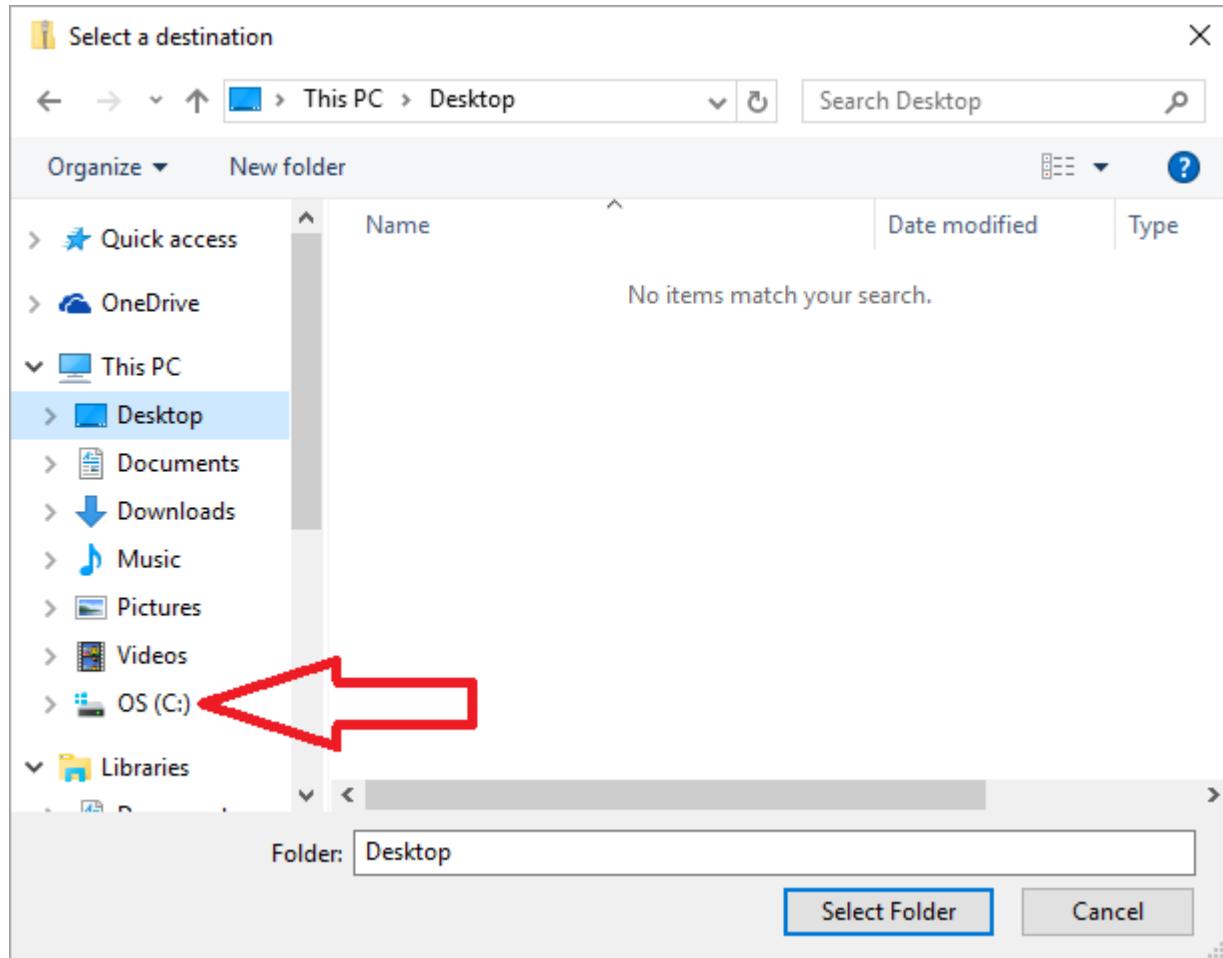
Step 5: After performing the previous step, File Explorer should open. The FTC SDK is a compressed file with a '.zip' extension. The full name of the file is ftc_app-master.zip or ftc_app.zip; the name will vary depending on how the zip file is obtained. If Windows is configured to hide file extensions (as shown below) only the first portion of the file name will be shown, the ".zip" extension will be hidden. Notice that Compressed Folder Tools is displayed in the Title Bar. Select 'Extract' and then 'Extract all'.



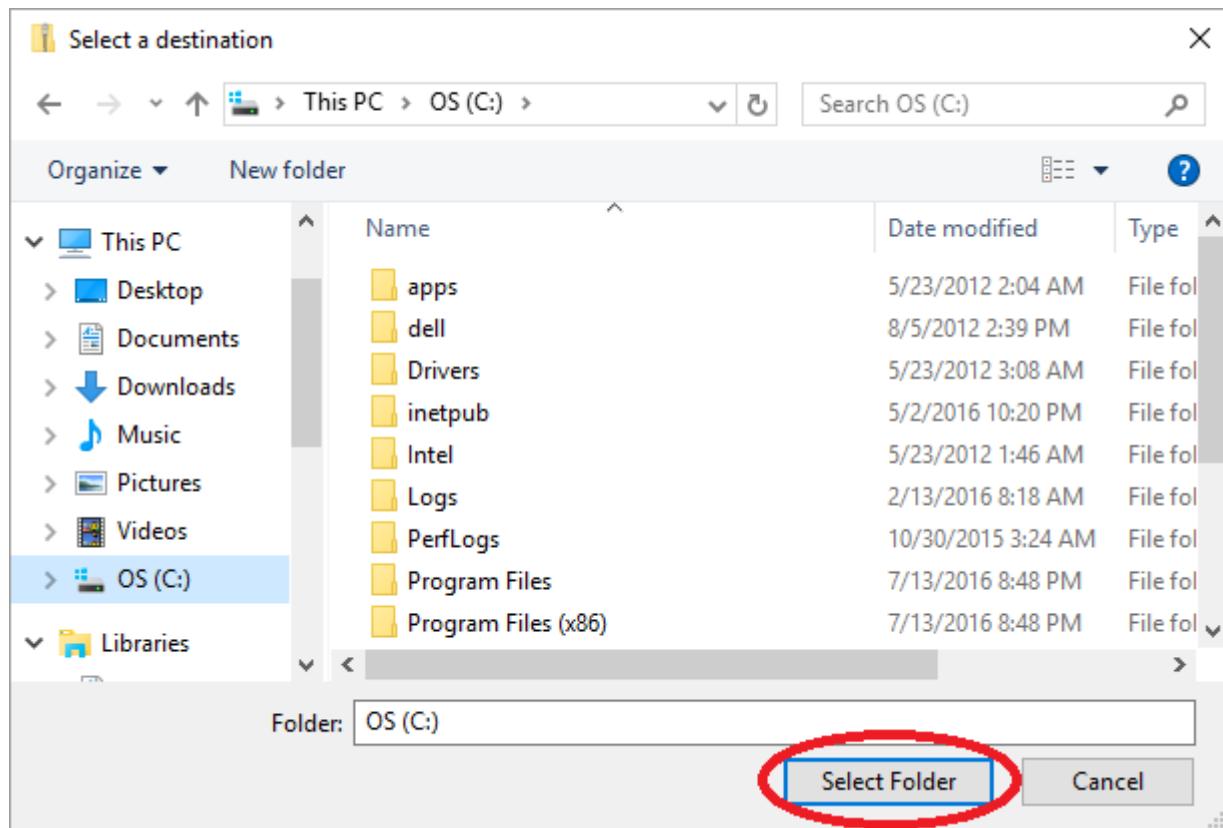
Step 6: Select 'Browse'.



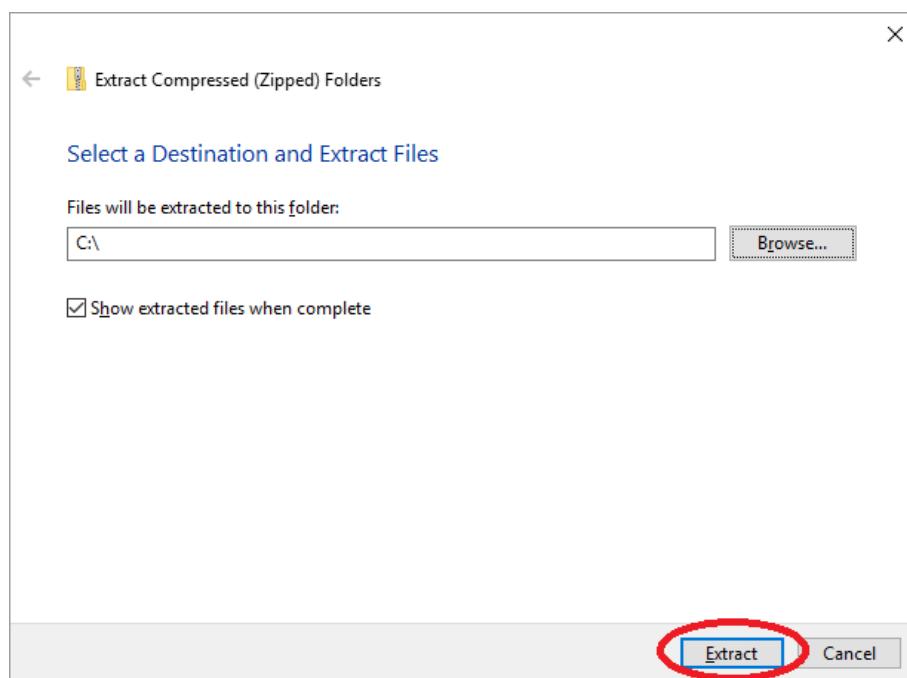
Step 7: Select a directory on your machine's hard drive for the location of the copied file. For this document the "Windows (C:)" folder will eventually be chosen. It will take several steps to find it. If the 'This PC' tree is collapsed then use the small arrow next to it to expand it. Scroll up or down until OS (C:) comes into view. Select it. The OS (C:) folder has been chosen for this document, because all members of our team can access this location from any account. The location your team will choose depends on many factors: the security at your location, whether your team will use a source control tool, etc.



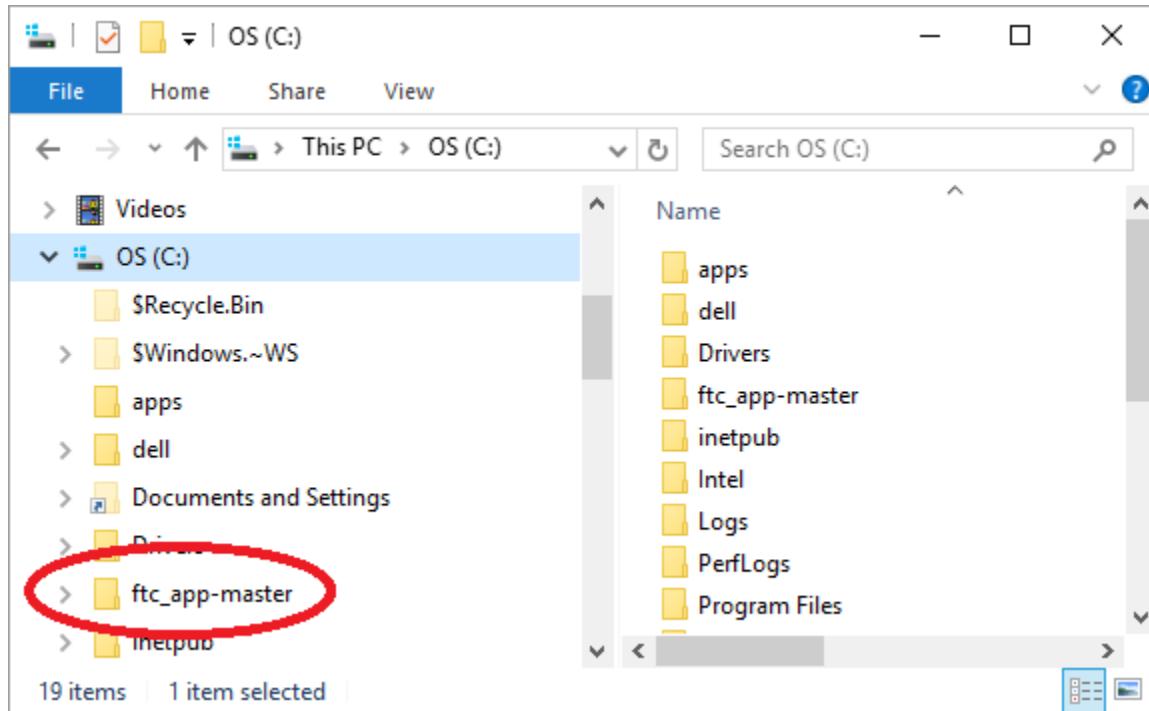
Step 8: Select OSelect Folder and the dialog will close and the Extract Compressed (Zipped) Folders dialog will become active (shown in the next step).



Step 10: Select 'Extract'. While it is extracting, another window may be shown.



Step 11: The 'ftc_app-master' folder should then exist under the 'Window (C:)' folder.



Installing the Java Development Kit (JDK)

The Java Development Kit must be installed before Android Studio. It provides all of the classes that are used by the FTC driver station and Robot controller applications. This section will describe how to install it.

Step 1: Go to this URL:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. Accept the License Agreement.

Note: The Java Development Kit must be installed before Android Studio.

Java SE Development Kit 8u101

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

Step 2: Select the download that matches your computer. Refer to section (Determining whether Android Studio can be installed, step 4 to determine the system type. For this example, it is jdk-8u101-windows-x64.exe.

Java SE Development Kit 8u101

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	jdk-8u101-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.72 MB	jdk-8u101-linux-arm64-vfp-hflt.tar.gz
Linux x86	160.28 MB	jdk-8u101-linux-i586.rpm
Linux x86	174.96 MB	jdk-8u101-linux-i586.tar.gz
Linux x64	158.27 MB	jdk-8u101-linux-x64.rpm
Linux x64	172.95 MB	jdk-8u101-linux-x64.tar.gz
Mac OS X	227.36 MB	jdk-8u101-macosx-x64.dmg
Solaris SPARC 64-bit	139.66 MB	jdk-8u101-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	98.96 MB	jdk-8u101-solaris-sparcv9.tar.gz
Solaris x64	140.33 MB	jdk-8u101-solaris-x64.tar.Z
Solaris x64	96.78 MB	jdk-8u101-solaris-x64.tar.gz
Windows x86	188.32 MB	jdk-8u101-windows-i586.exe
Windows x64	193.68 MB	jdk-8u101-windows-x64.exe

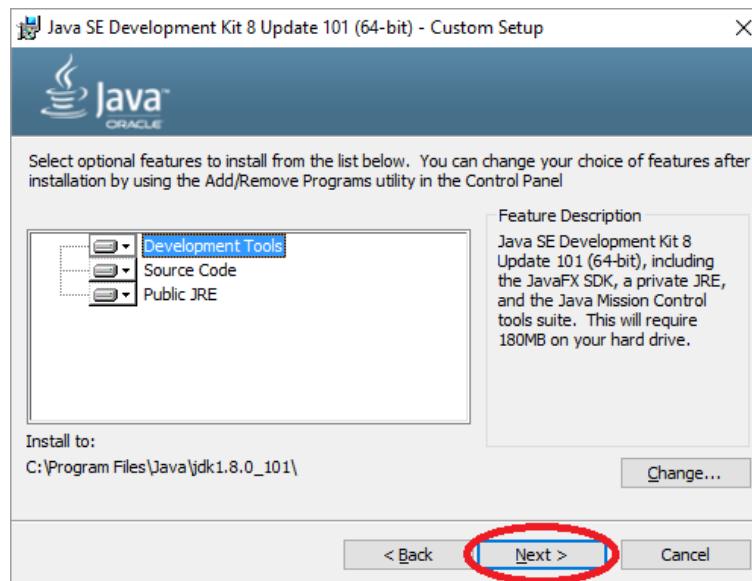
Step 3: Edge will download the file. After the download is complete, select Run. If a window is displayed that asks “Do you want to allow the following program to make changes to this computer?”, select Yes..



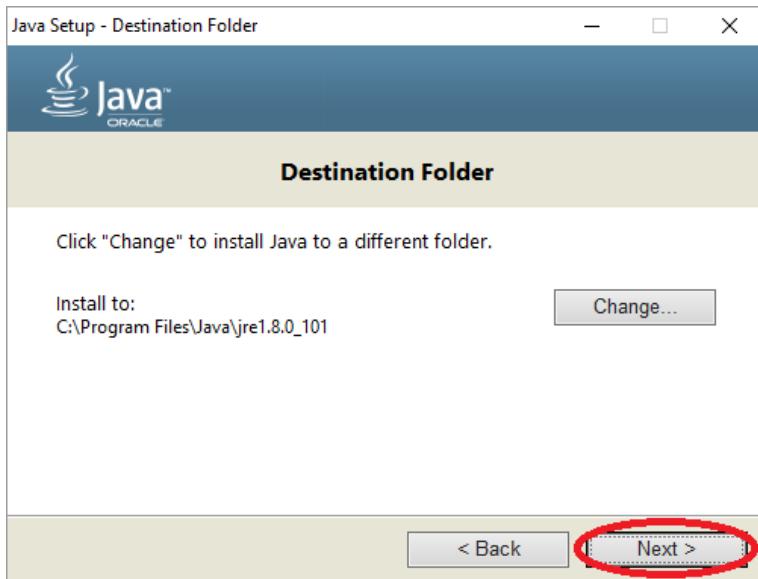
Step 4: The installation of Java will begin by displaying the following window. Select Next.



Step 5: The following dialog can be used to select which parts are installed and where the files will be located. Choose Next to accept the default installation.



Step 6: The installation continues by selecting where the files will be located. Choose Next to accept the default installation.



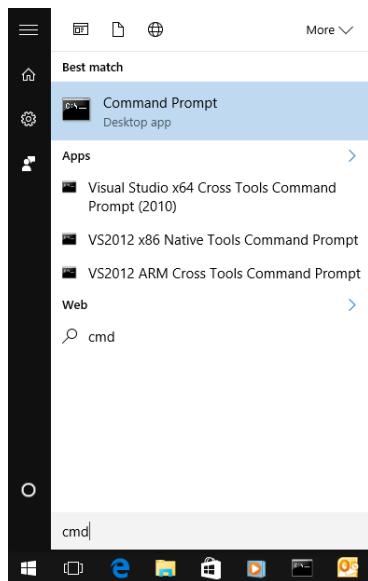
While the files are being installed the following dialog will be displayed.



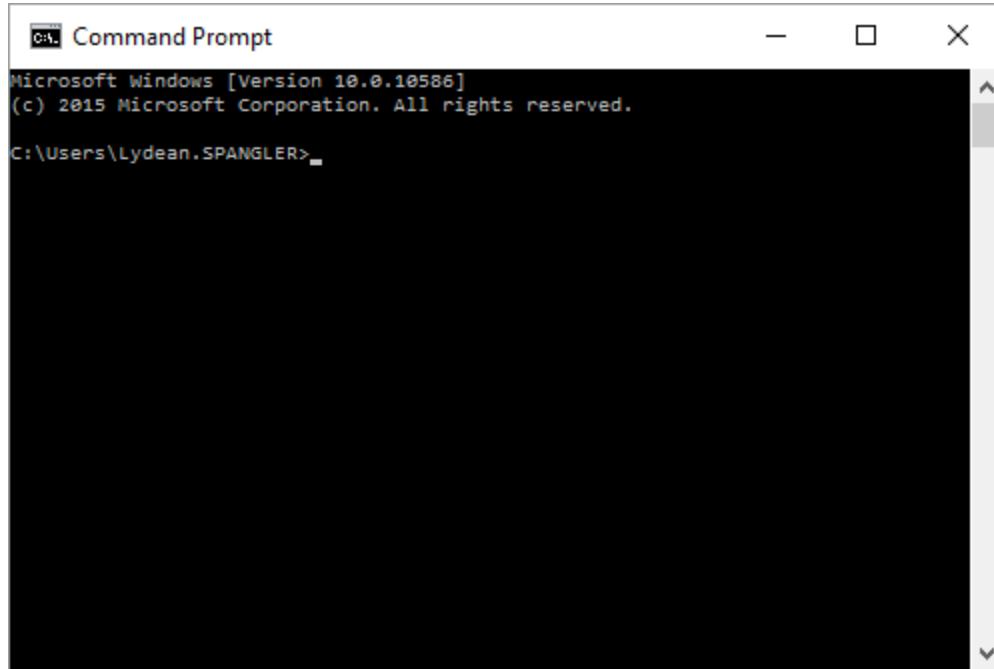
Step 7: After the installation is complete, choose Close to exit the installer.



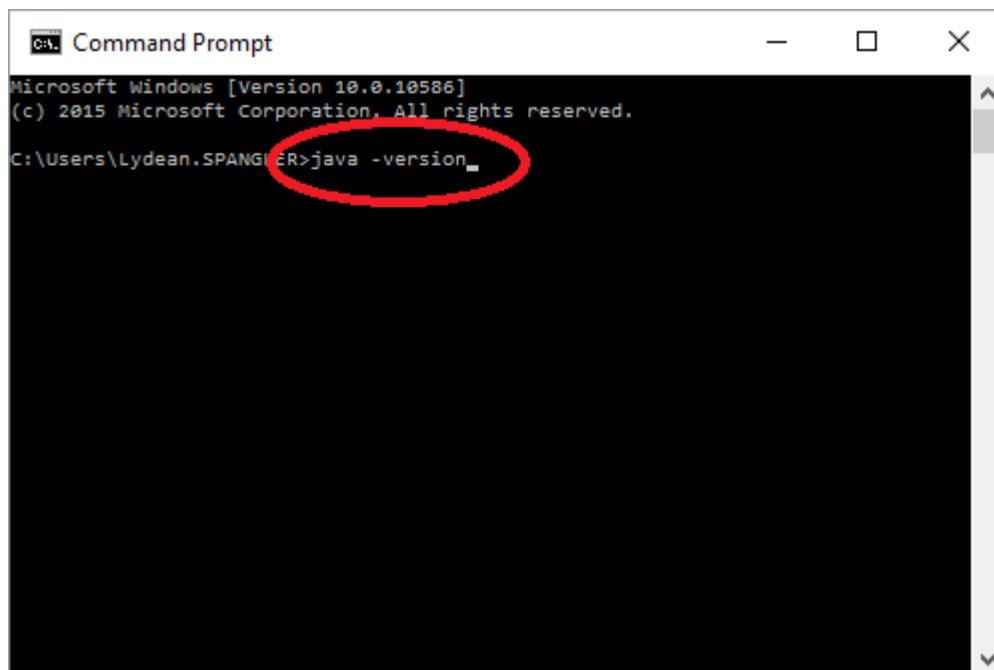
Step 8: To determine whether the installation was a success, use the command prompt. Press the Windows key and type 'cmd'. Remember to press enter after typing the name.



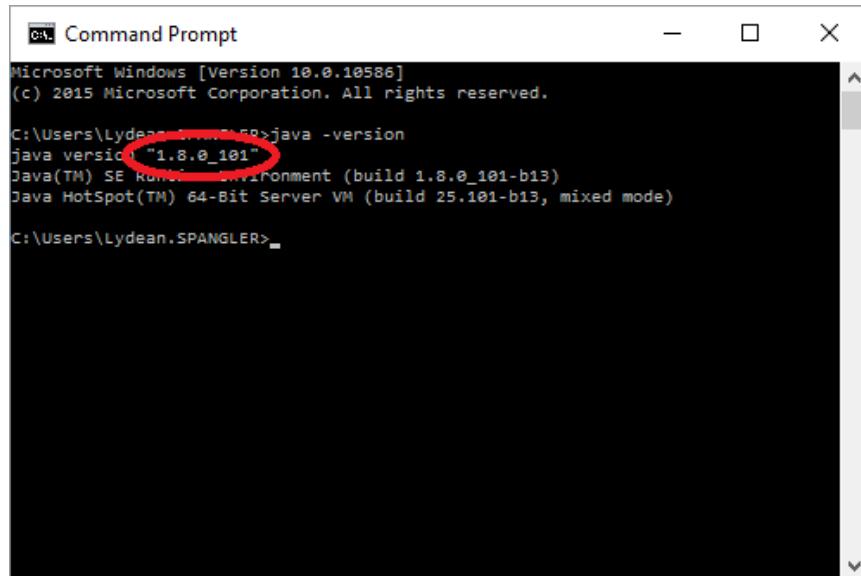
The following window will be displayed.



Step 9: Type “java –version” and press return/enter



Step 10: Verify that the version is “1.7.XXX”. The close button () can be used to close the window.



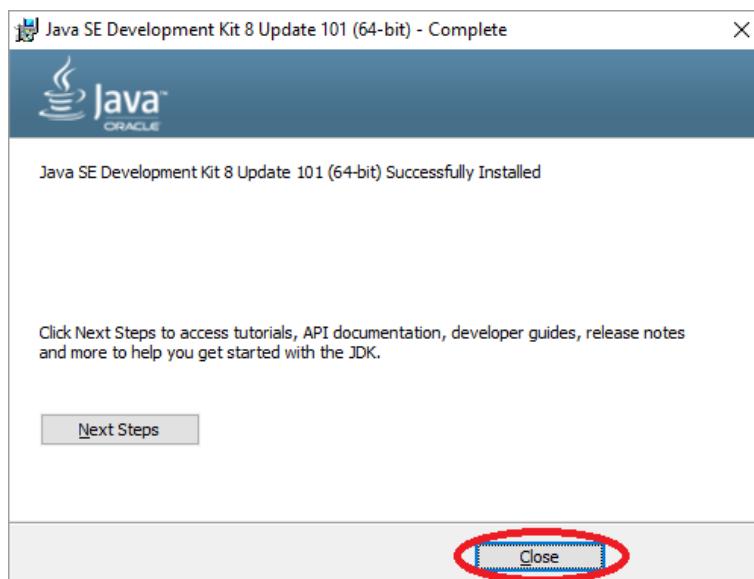
A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text:
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Lydean.SPANGLER>java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)

C:\Users\Lydean.SPANGLER>

The line "java version "1.8.0_101"" is circled in red.

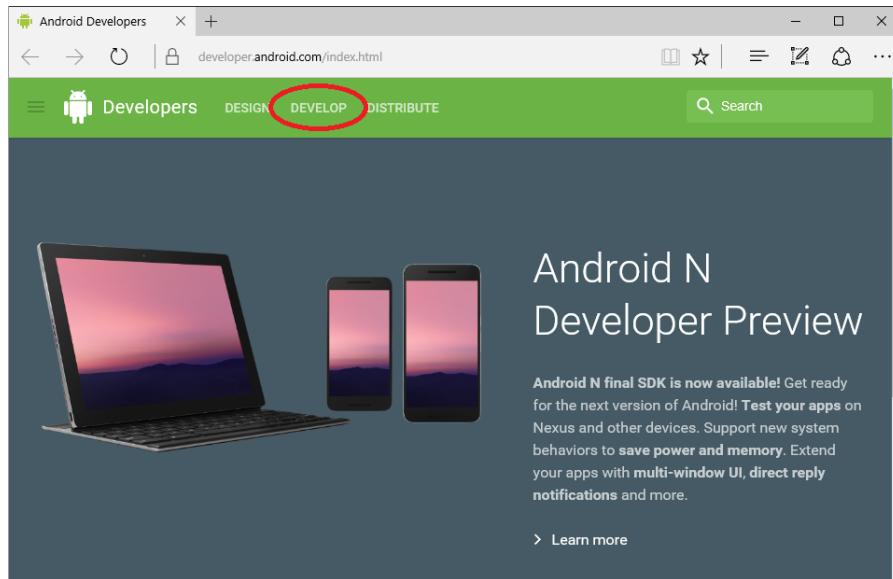
Step 11: Select Close.



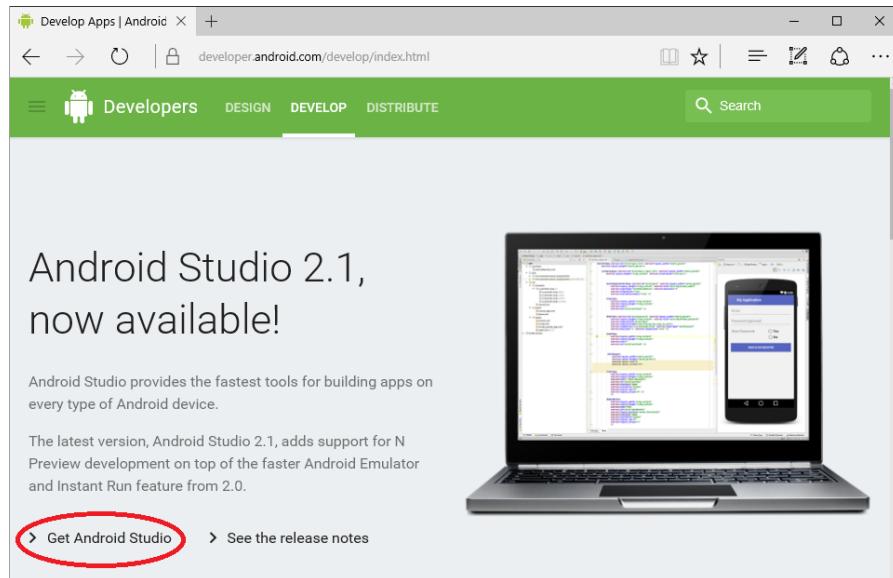
Installing the Android Studio Integrated Development Environment (IDE)

The FTC SDK and Java Development kit have previously been installed. The next task is to install the Android Studio IDE.

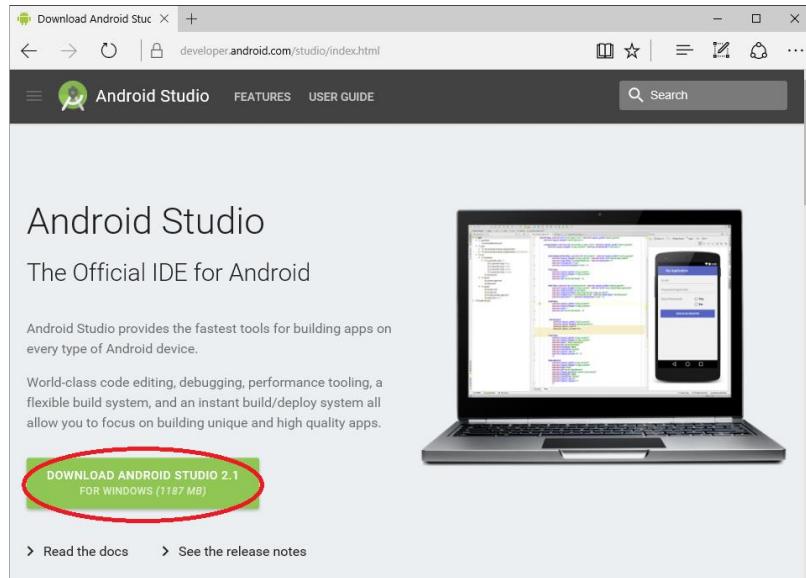
Step 1: The IDE is available for Windows, Mac OS X and Linux. It can be downloaded from <https://developer.android.com>. Select Develop.



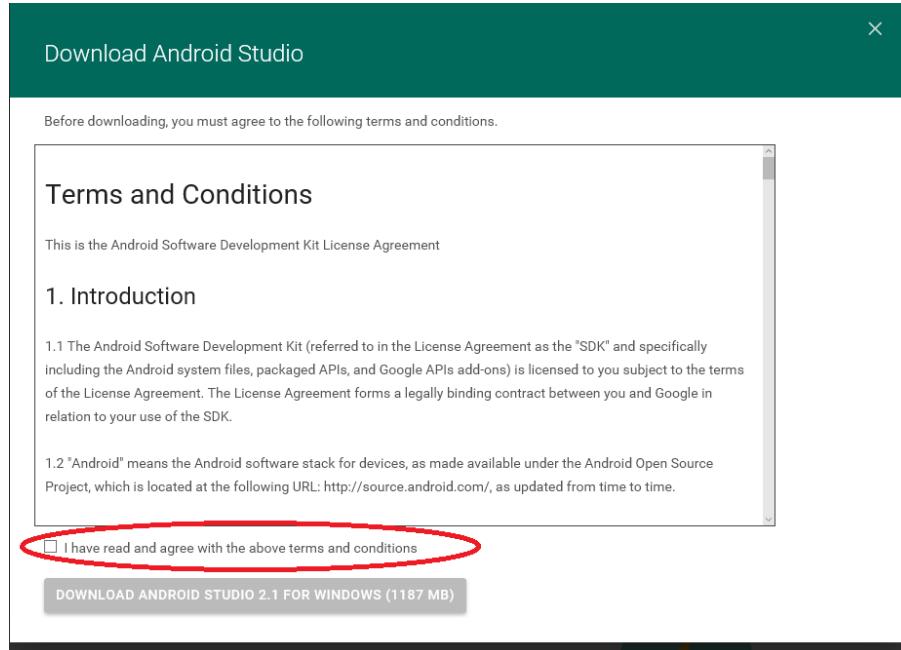
Step 2: Select Tools on the submenu.



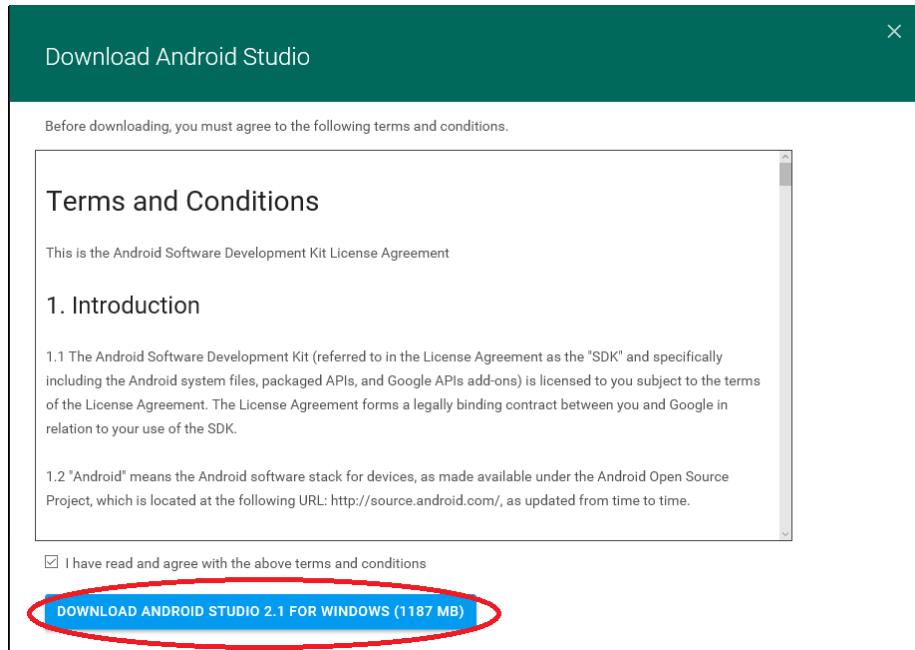
Step 3: Select the green “Download Android Studio for Windows” button.



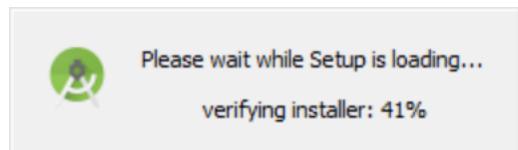
Step 4: Accept the terms and conditions from the next page to begin the download.



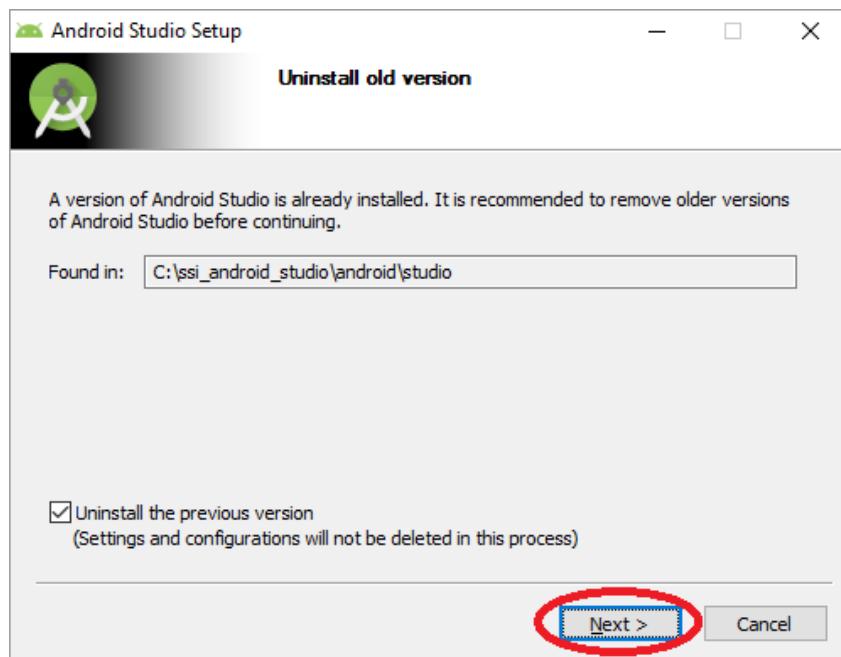
Step 5: After checking the terms and conditions, the download button will be selectable. Select the Download Android Studio for Windows button. Edge should display a downloading banner. Select Save. This might take some time to fully download



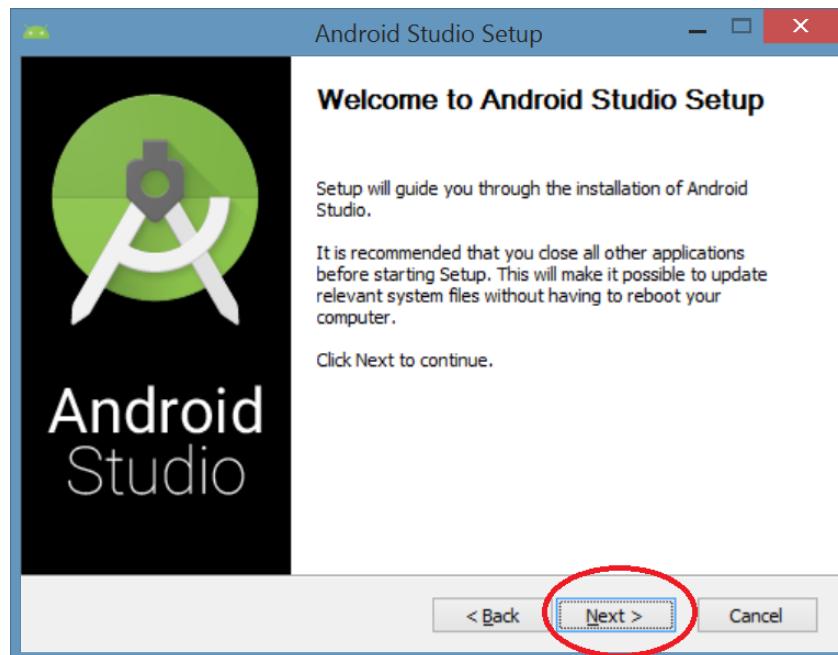
Step 6: After the download is complete, select Run. As the installer is loading, the image below the following image may be shown.



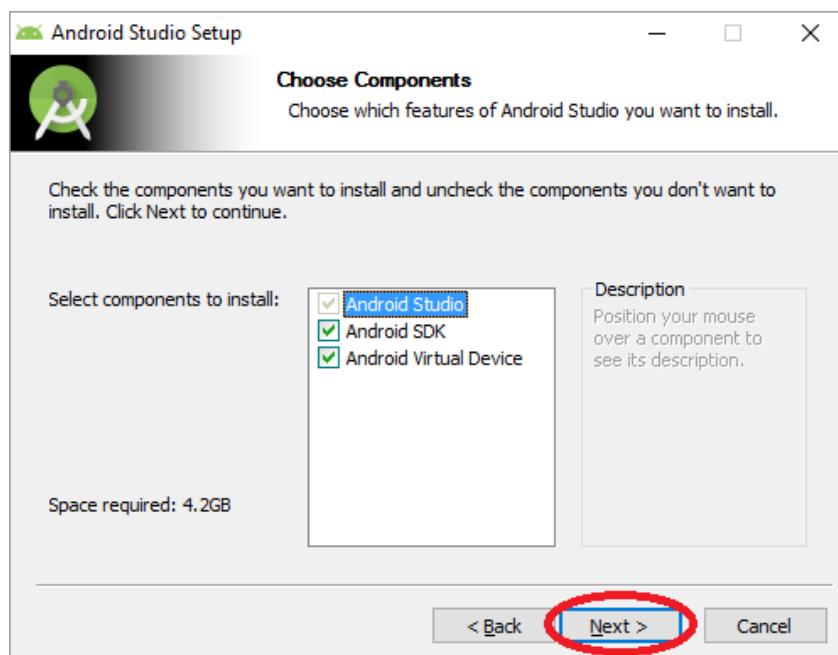
Step 7: If Android Studio has already been installed, the following window may be displayed. If this window isn't displayed continue to the next step.



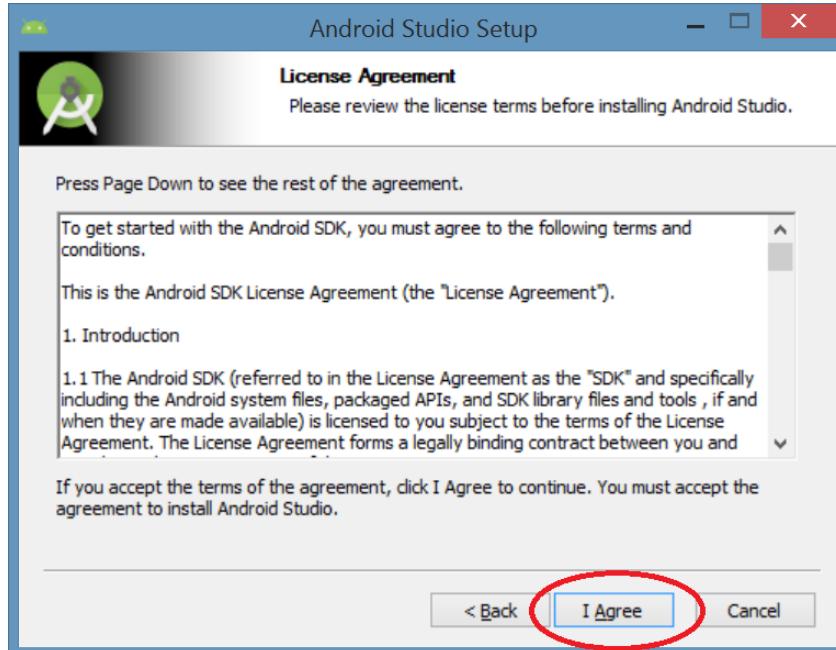
Step 8: To begin the installation process select Next. If a window is displayed that asks “Do you want to allow the following program to make changes to this computer?” select Yes.



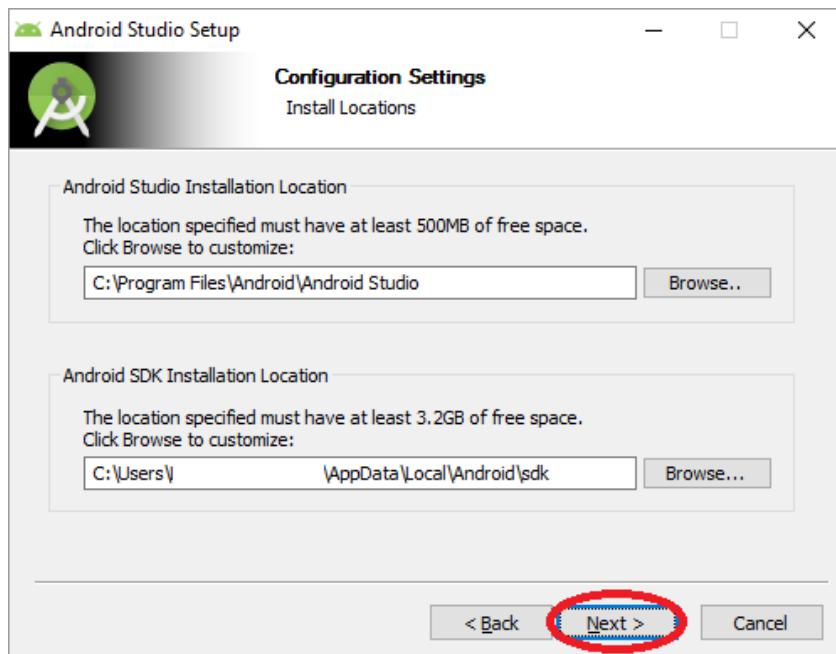
Step 9: The following dialog can be used to select which parts are installed. Choose Next to accept the default installation.



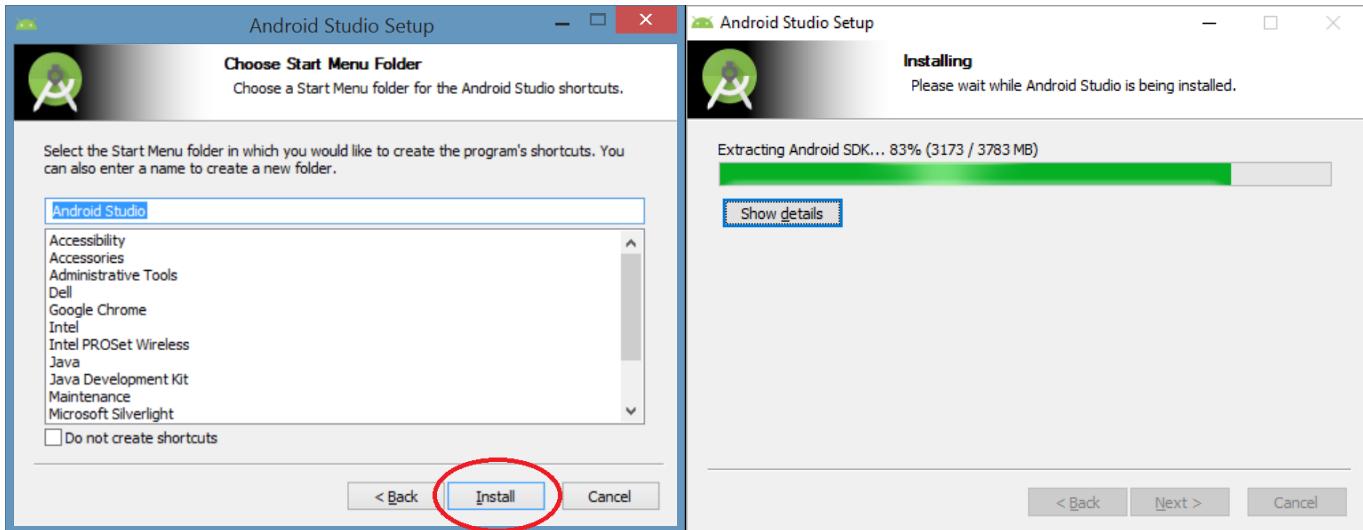
Step 10: Accept the License Agreement for the **Android SDK** by selecting 'I Agree'.



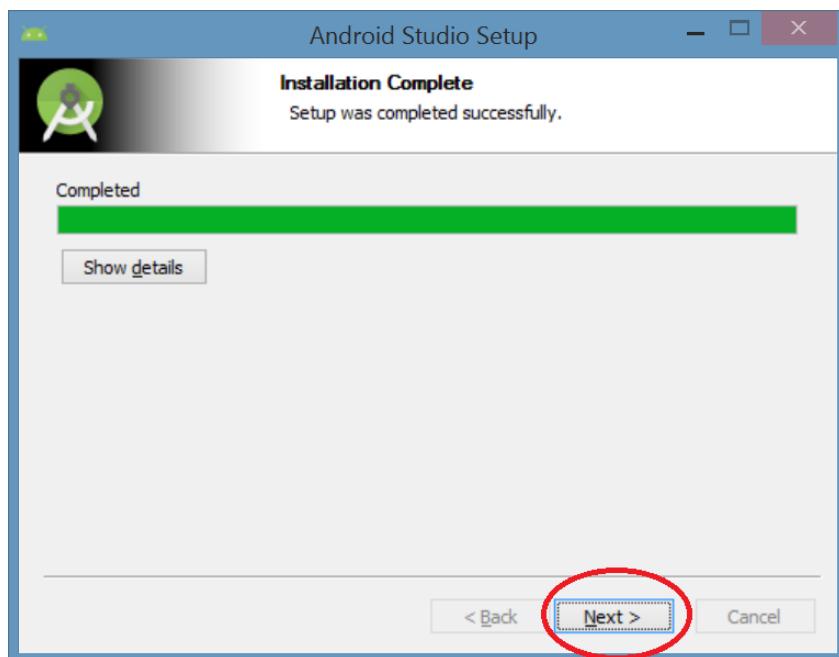
Step 11: The following dialog can be used to select the installation location. Choose Next to accept the default installation.



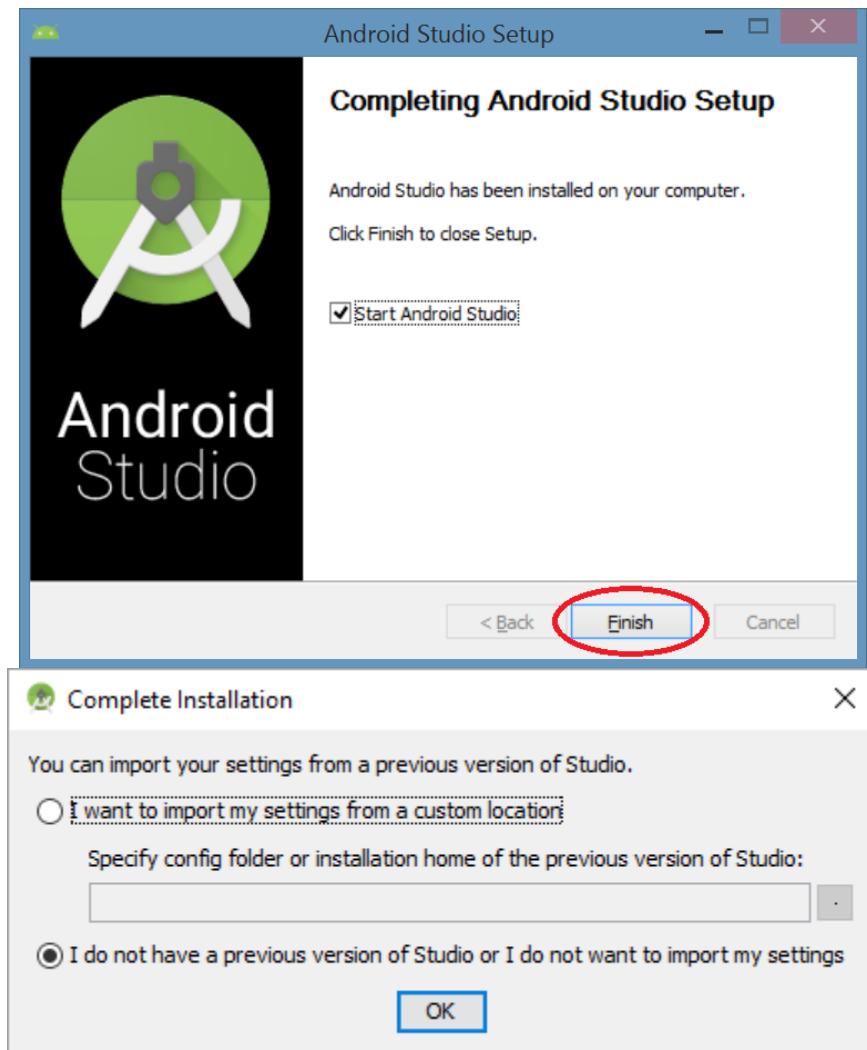
Step 12: Choose 'Install' to accept the default start menu folder.



Step 13: When the installation is complete, select 'Next'.



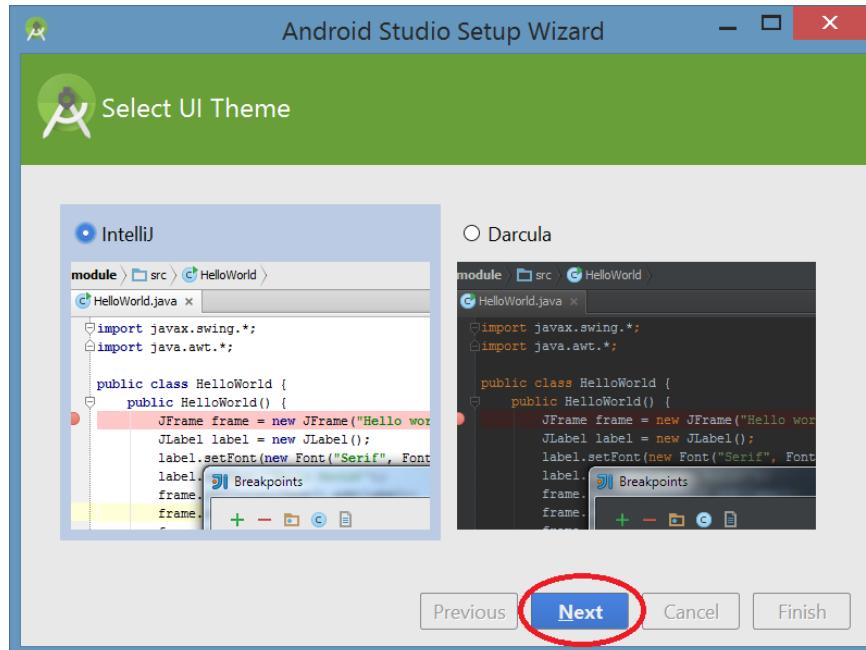
Step 14: Open Android Studio by selecting ‘Finish’. If Android Studio was previously installed, then choose whether the previous settings will be retained (bottom image).



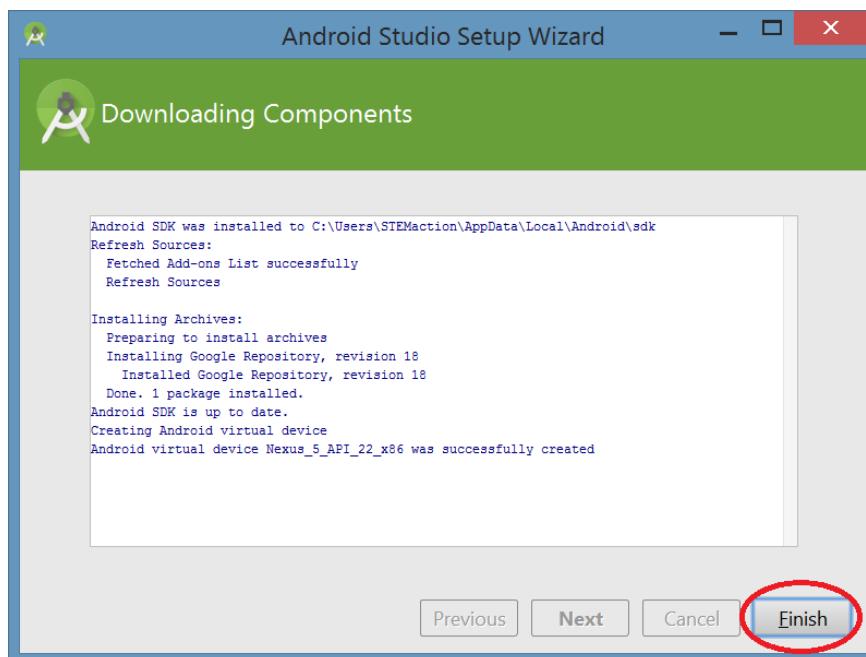
The following screen will be displayed while Android Studio is opening.



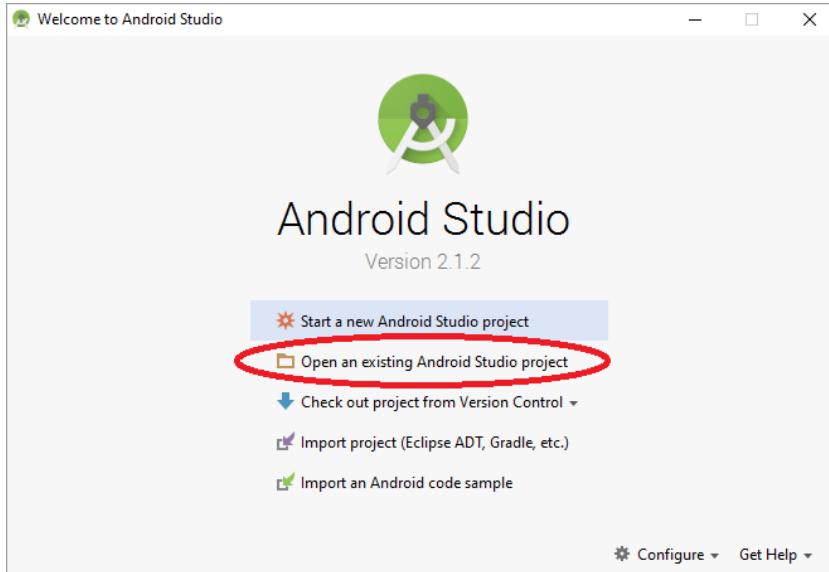
Step 15: Choose 'Next' to select the default UI Theme.



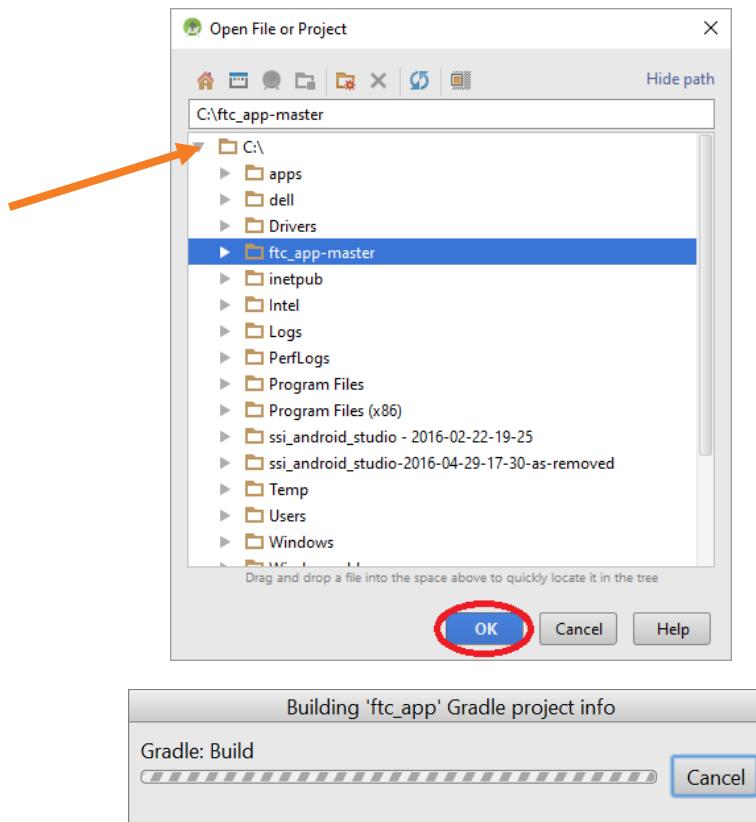
Step 16: Select 'Finish' and the Android Studio Welcome page will be shown.



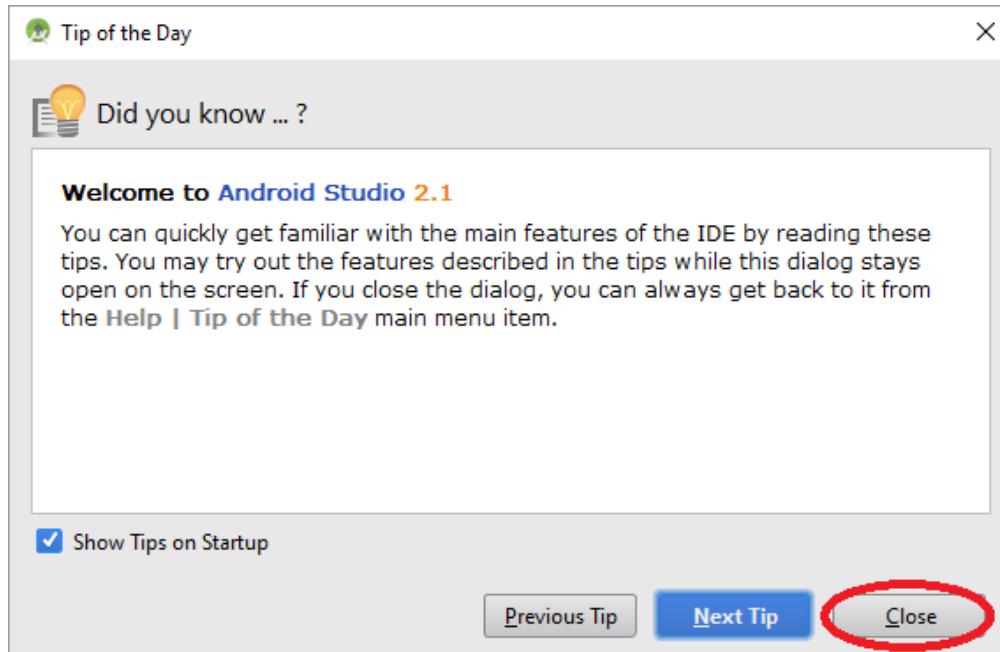
Step 17: Open the FTC SDK by selecting ‘Open an existing Android Studio project’.



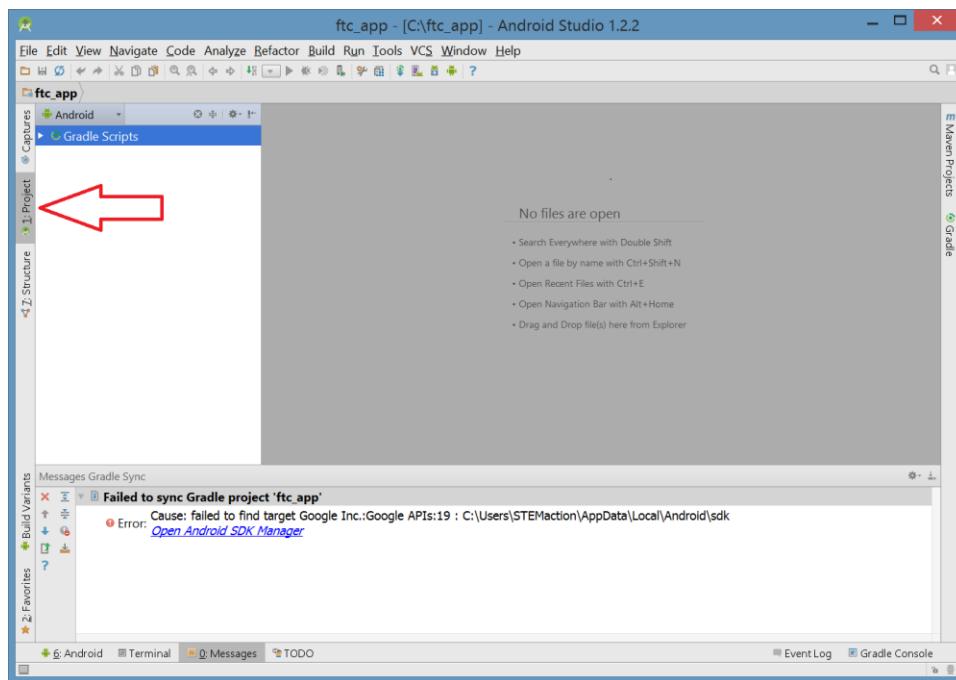
Step 18: It might take a while for the dialog to completely load. If ‘ftc_app-master’ doesn’t show in the list after a long while, close the “c:\” folder by clicking the triangle to the left of ‘C:\’ and reopen the folder by clicking the triangle again. Single click the ‘ftc_app-master’ folder. Select ‘OK’. Android Studio will attempt to build the project and several windows will be displayed before the IDE opens.



Step 19: Close the Tip of the Day window.

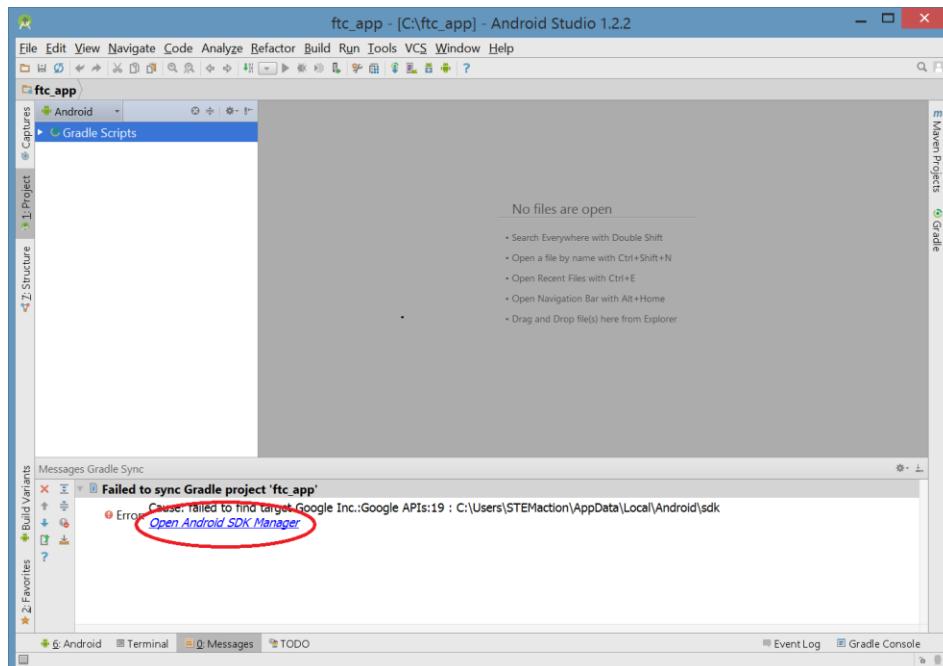


Step 20: If the ftc_app-master project isn't shown, click on the triangle next to Android (near the top left portion and select Project from the drop down list. The image below is sometimes how the project opens (i.e. without the SDK showing).

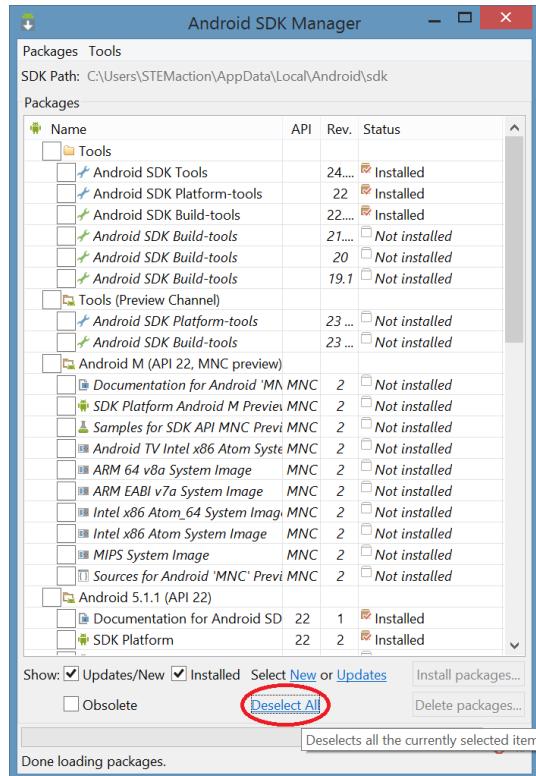


Step 21: If an error message is displayed at the bottom of the window, then the following steps may help.

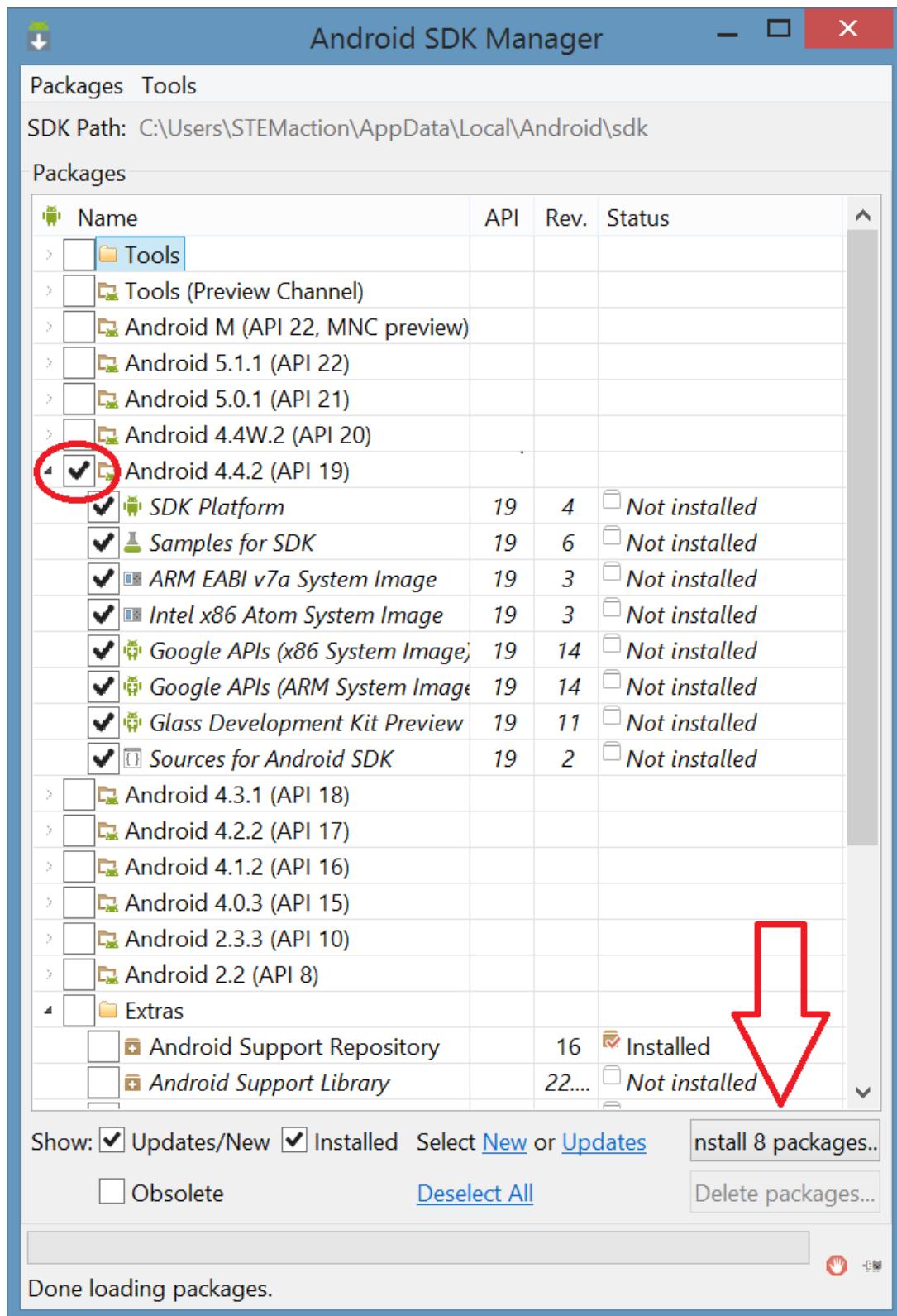
Step 21 A1: If Open Android SDK Manager is being displayed, then install the Android SDK Manager by selecting the link near the bottom of the page. If there is no message that indicates “failed to find Build Tools...”, then the installation is complete and proceed to the next section “Installing the Push Bot Op modes”.



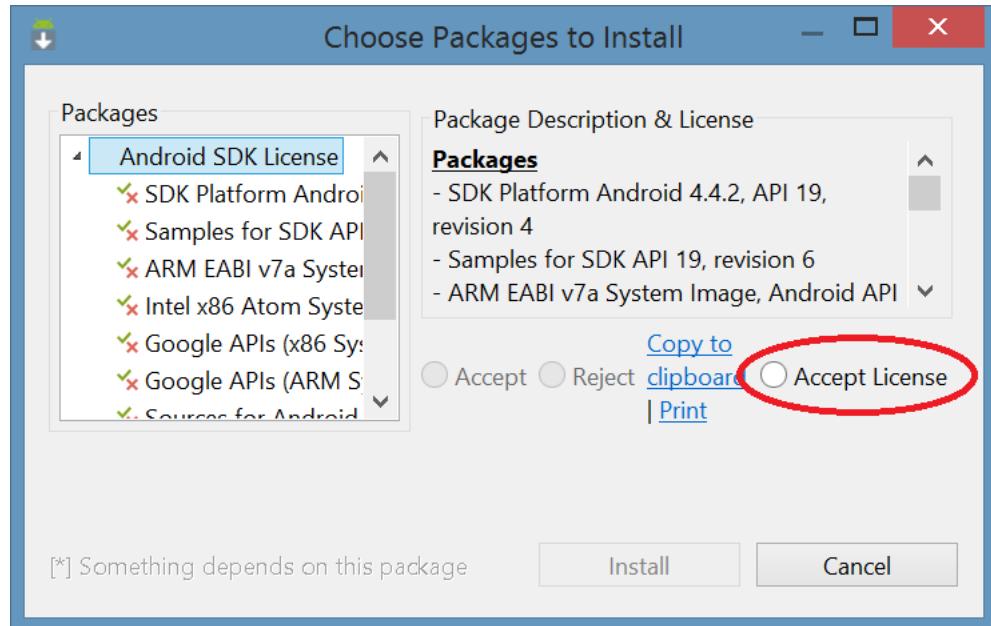
Step 21 A2: Deselect all components using the ‘Deselect All’ button near the bottom of the window.



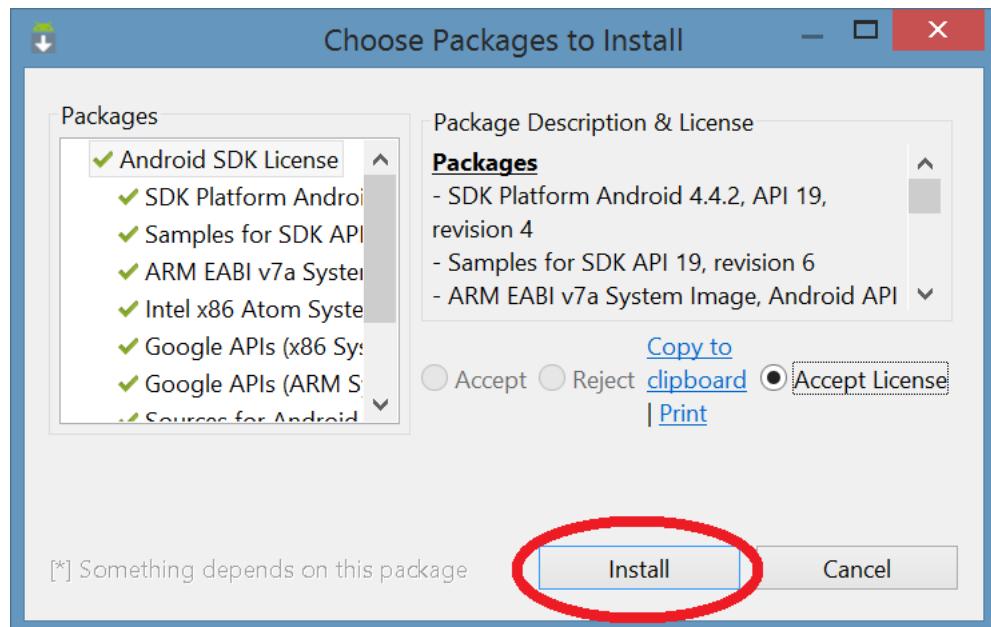
Step 21 A3: Select the box (red circle) next to **Android 4.4.2 (API 19)**. All components beneath that element will become checked. Install the components (red arrow).

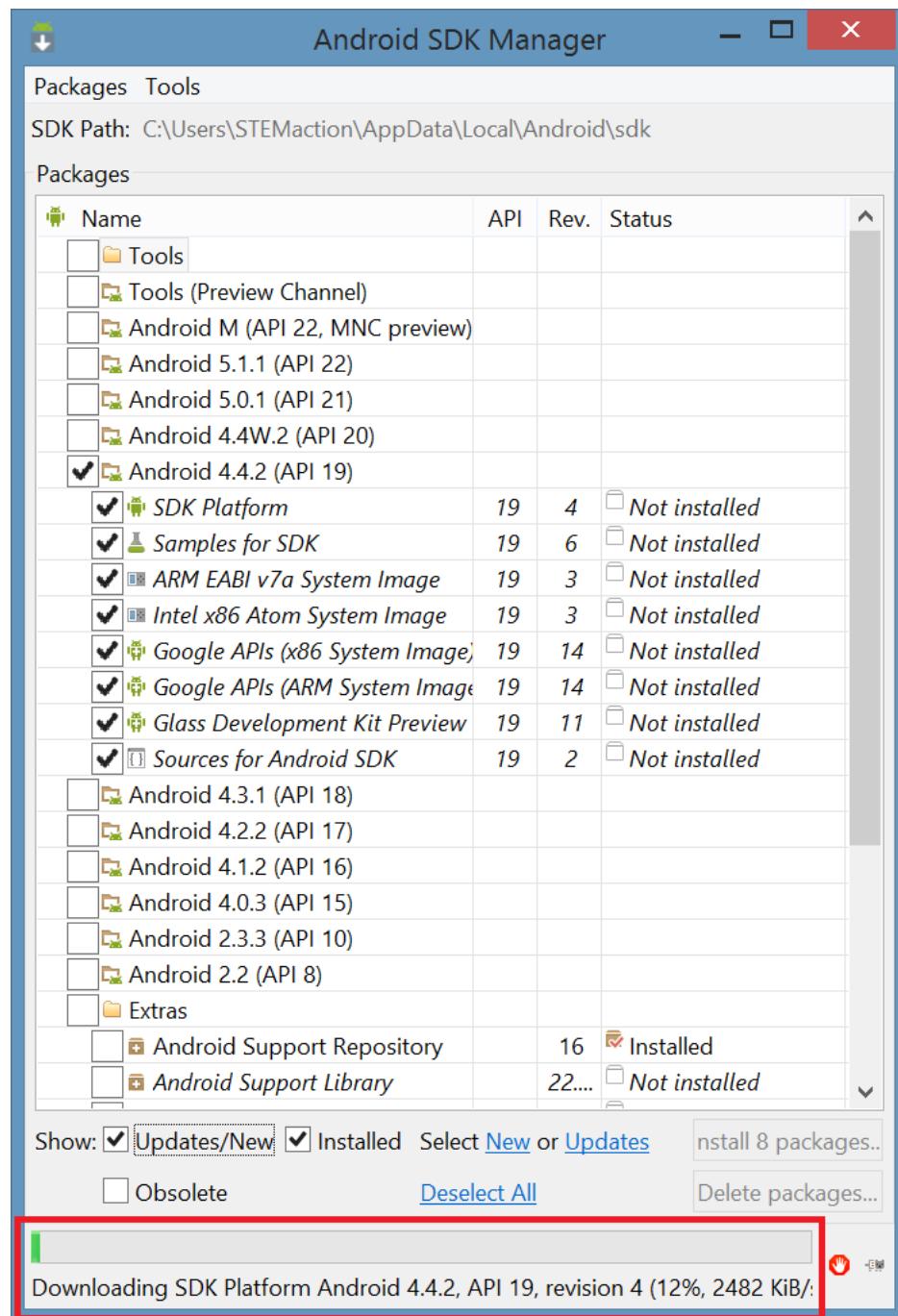


Step 21 A4: Accept the License Agreements.

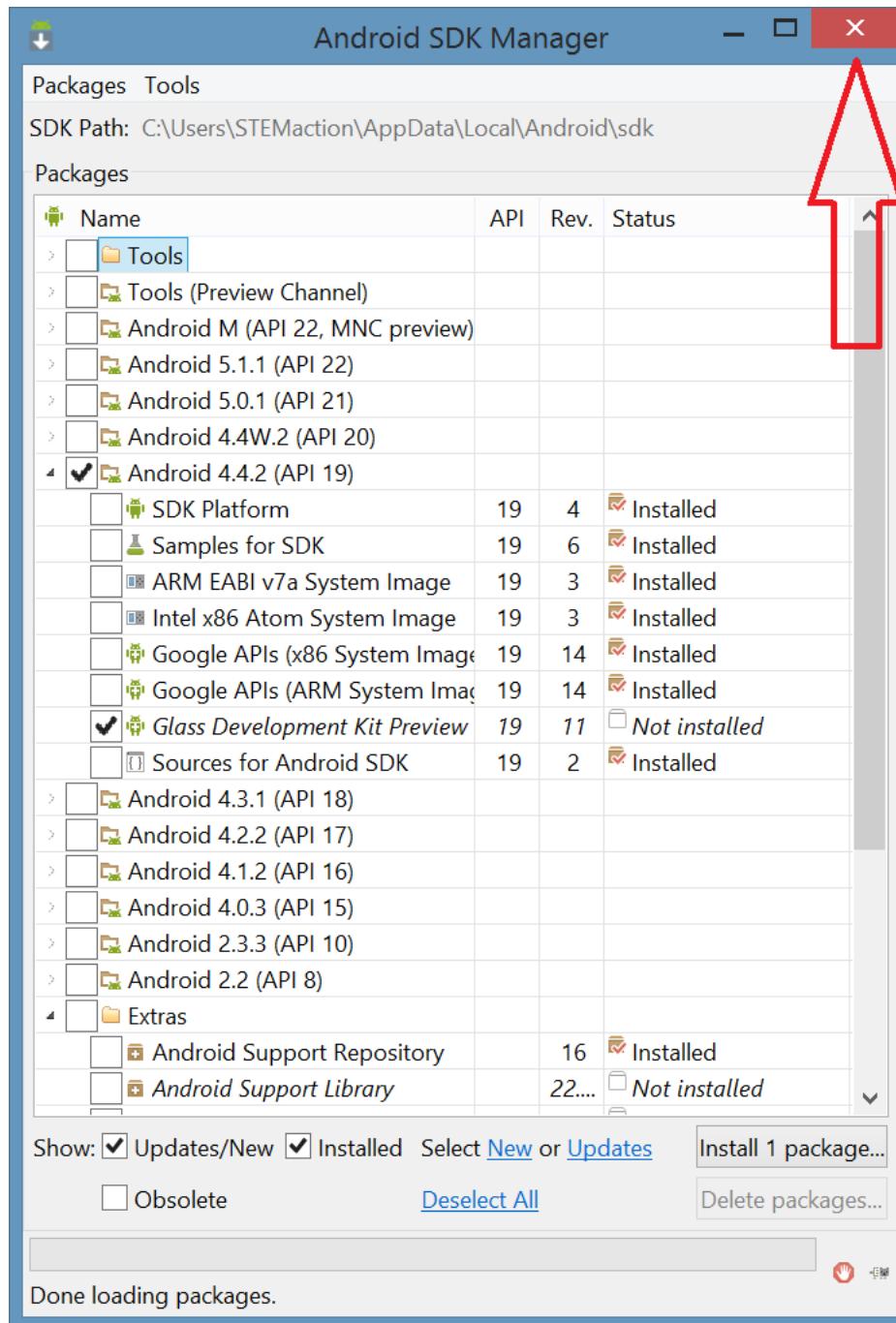


Step 21 A5: Install the packages. While the packages are being downloaded and installed, the status bar at the bottom of the Android SDK Manager will update (image below the following image).

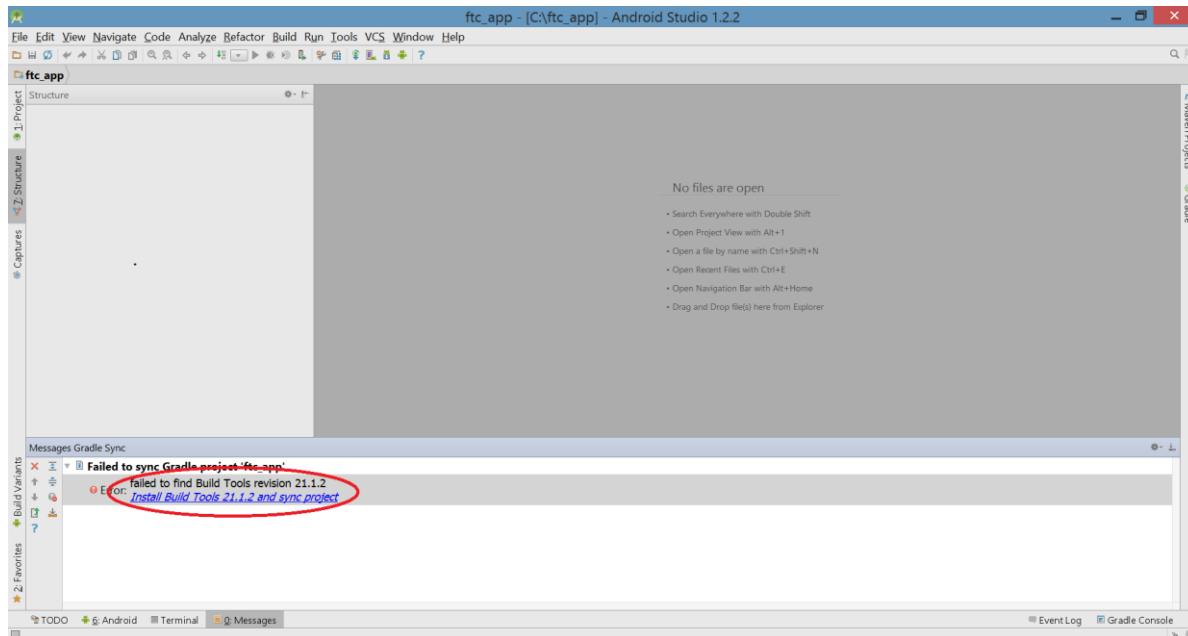




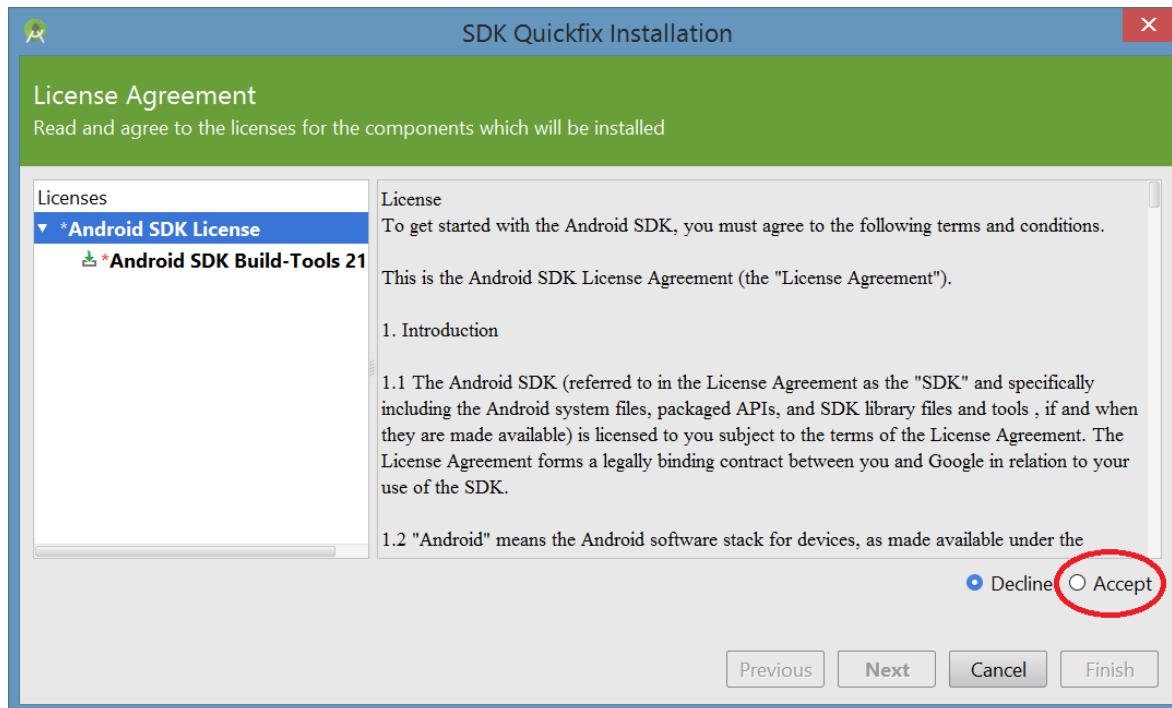
Step 21 A6: After the download has completed, close the dialog.



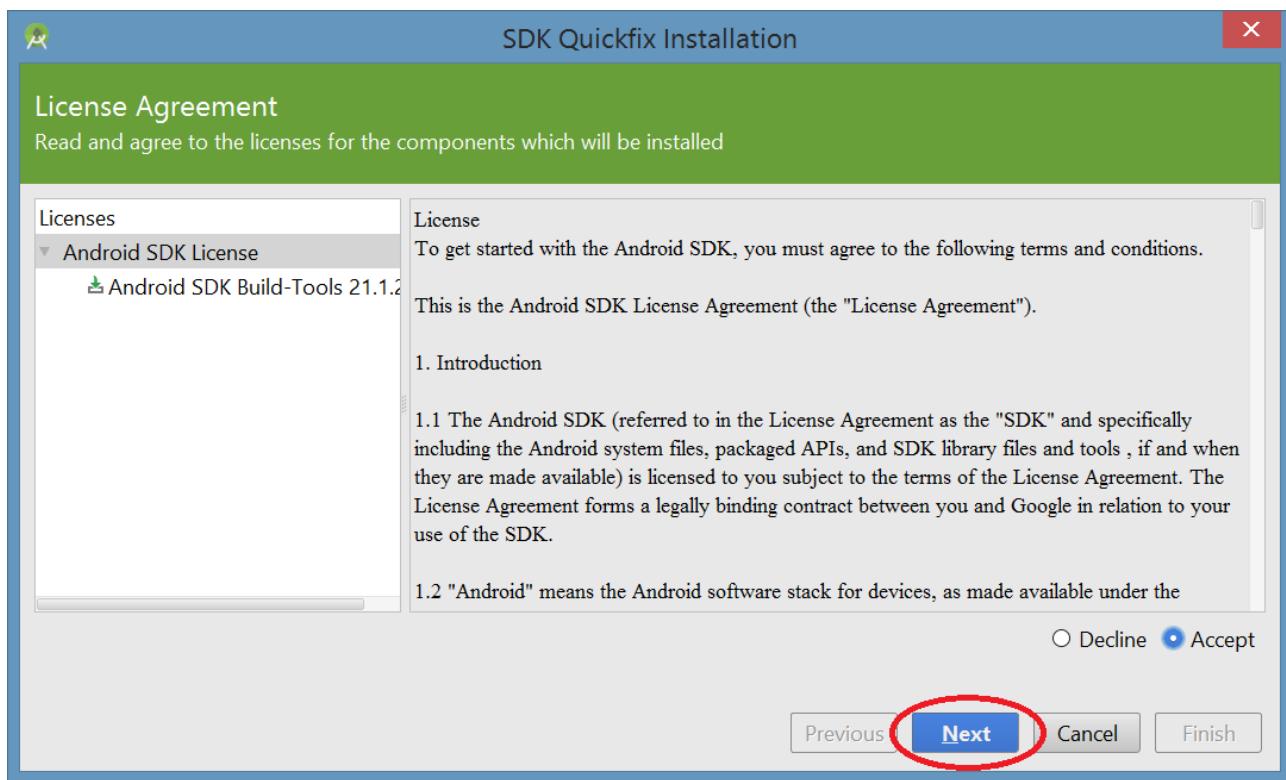
Step 21 A7: If Android Studio doesn't automatically attempt a rebuild, close Android Studio and restart it. If a different error message than the previous error message, select the blue link again to install Build Tools 21.



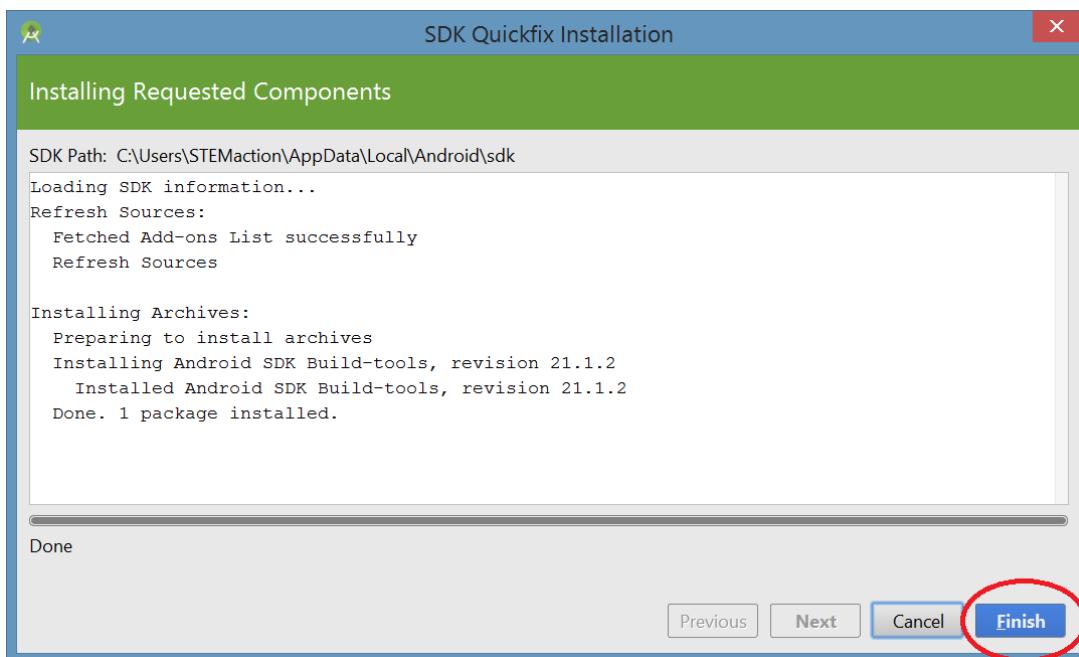
Step 21 A8: Accept the quick fix license. The following window might display Build-Tools 19, which is required by the current FTC SDK.



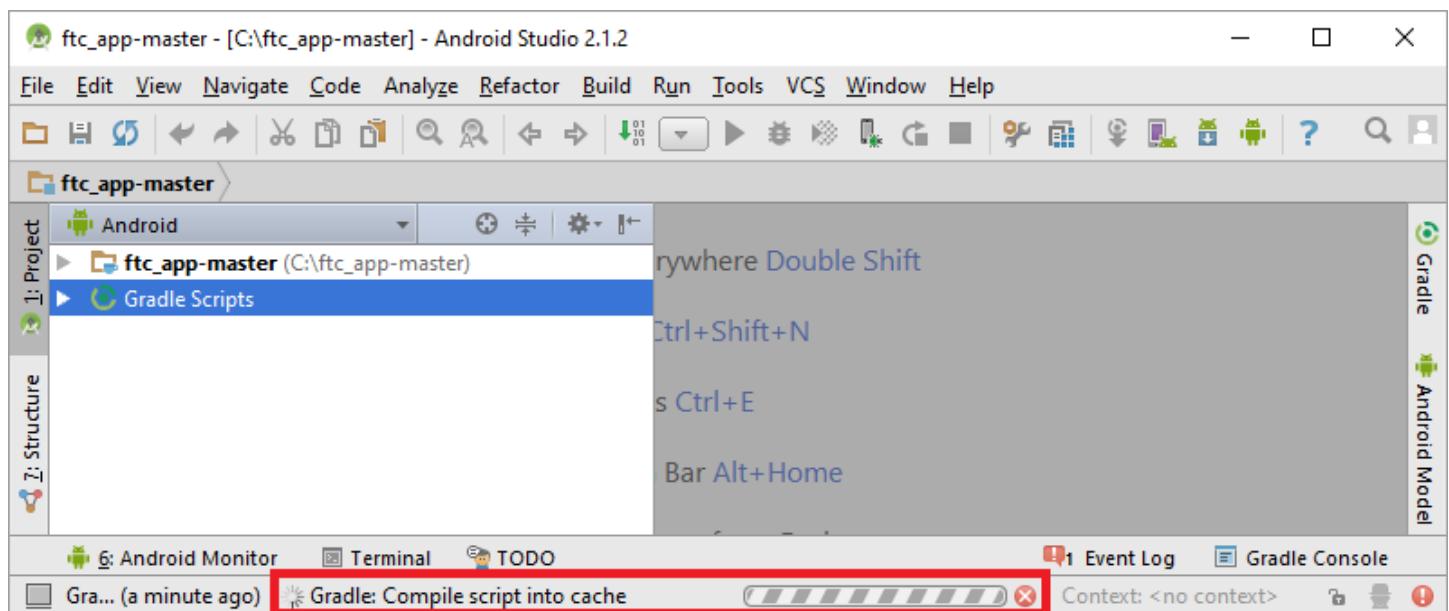
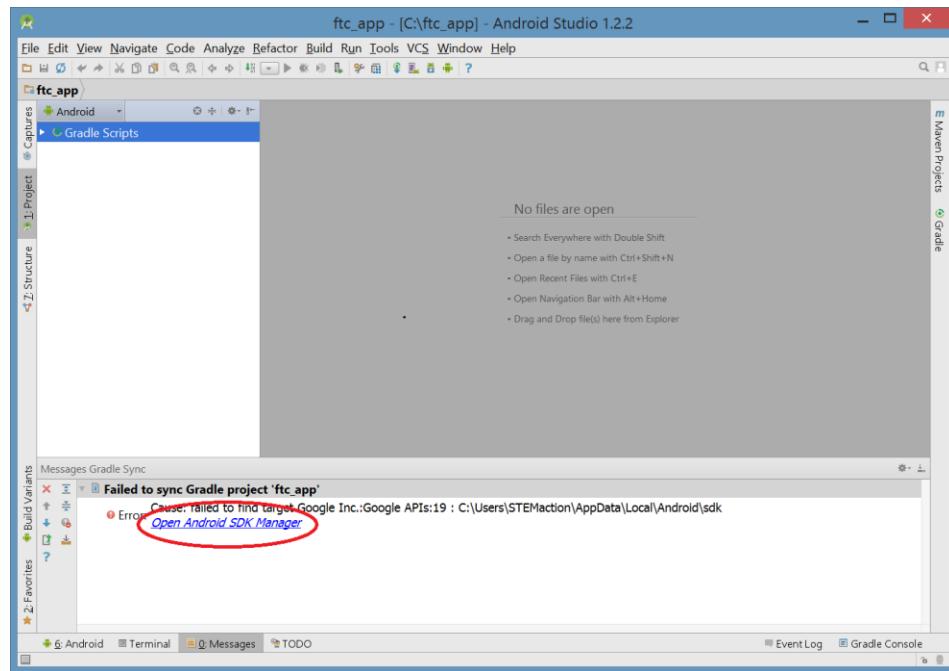
Step 21 A9: Click 'Next' to install the components. The following window might display Build-Tools 19, which is required by the current FTC SDK.



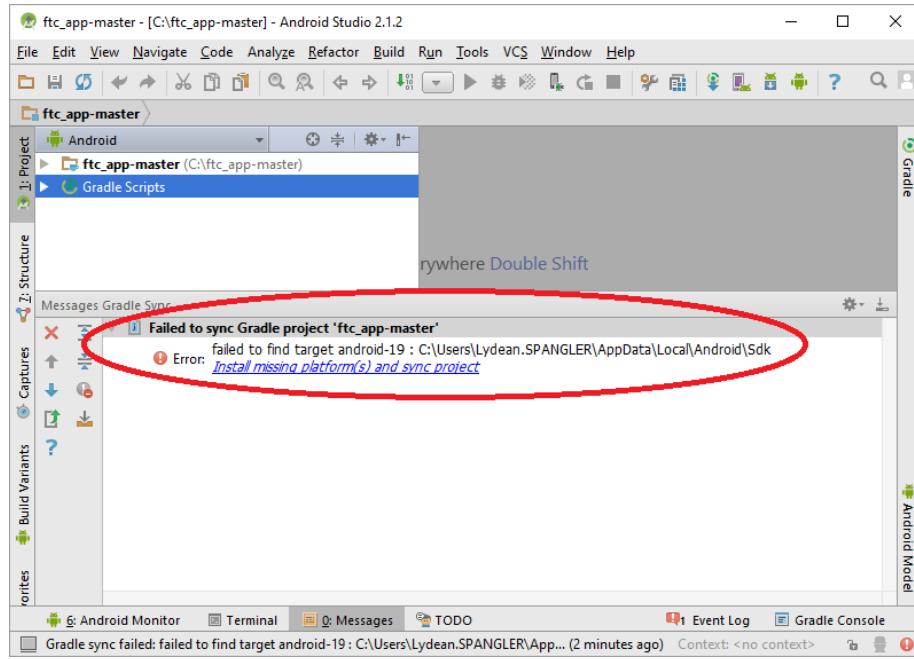
Step 21 A10: Return to Android Studio by selecting 'Finish'.



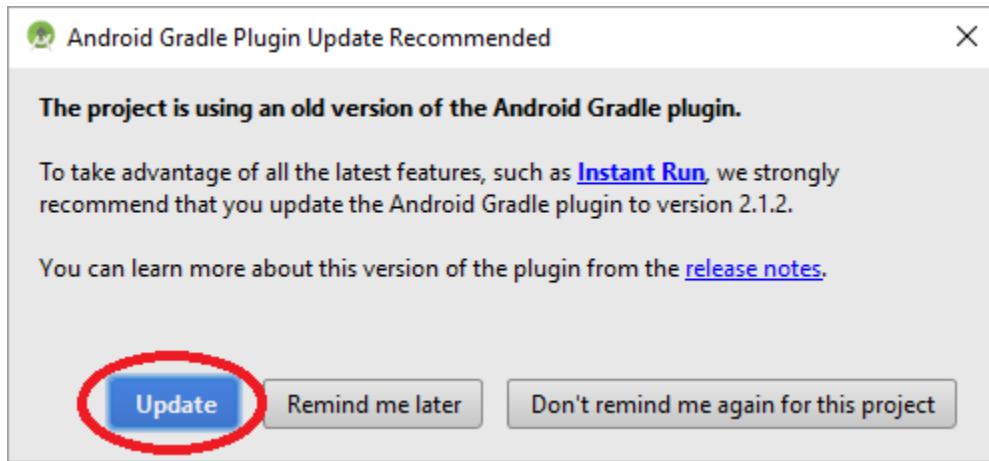
Step 21 B1: If Please fix the project's Gradle settings is being displayed, then select the link near the bottom of the page. The progress of the operation will be shown at the bottom of Android Studio - to the left of a gray scrolling status window, Refreshing ‘ftc_app-master’ Gradle project will be displayed. If there is no message that indicates “failed to find Build Tools...”, then the installation is complete and proceed to the next section “Installing the Push Bot Op modes”.



Step 21 C1: If Install missing platform(s) and sync project is being displayed, then select the link near the bottom of the page. The window in 21 A8 is displayed. Complete the steps for Build Tools 19 AND Build Tools 21.1.2.



Step 21 D1: If Android Gradle Plugin Update Recommended is being displayed, then follow the indicated steps to download and install it. The status of the process will be shown on the lower frame of the Android Studio window.

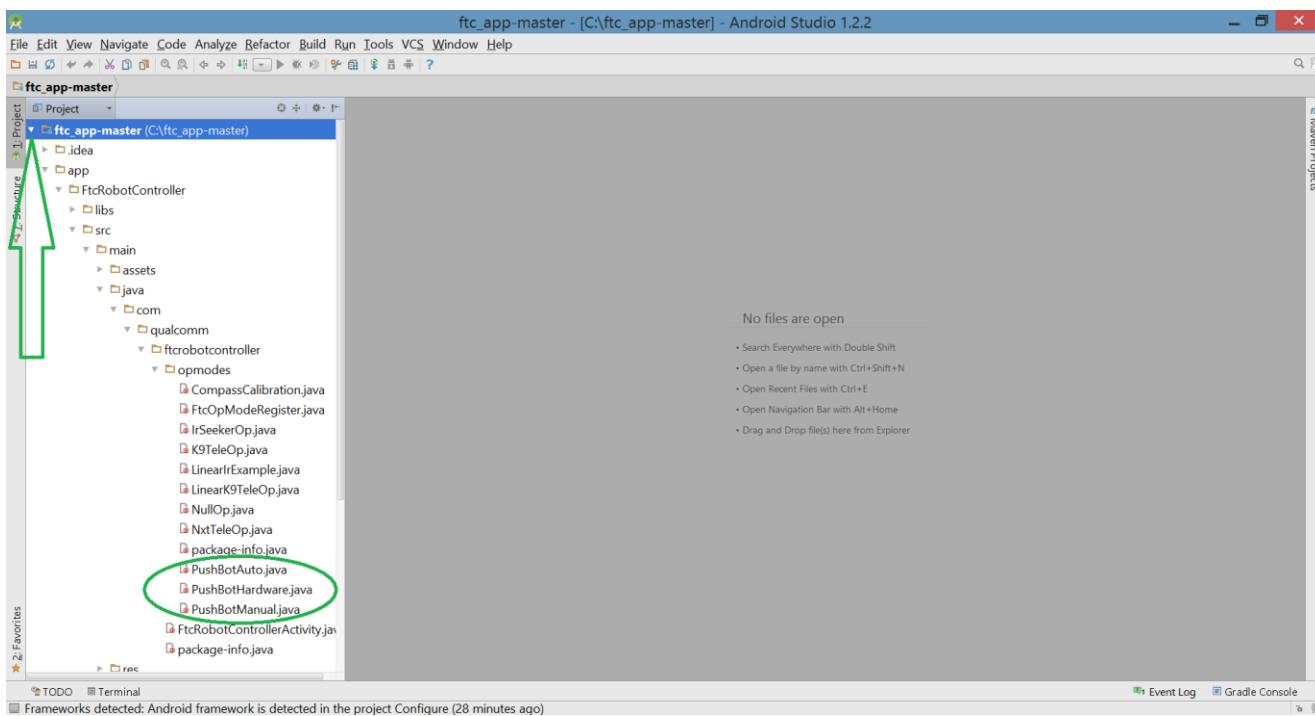


Step 31: Installation should be complete at this point. If an error message continues to be displayed, refer to the Android Studio website or [FIRST Tech Challenge forum](#) for more information.

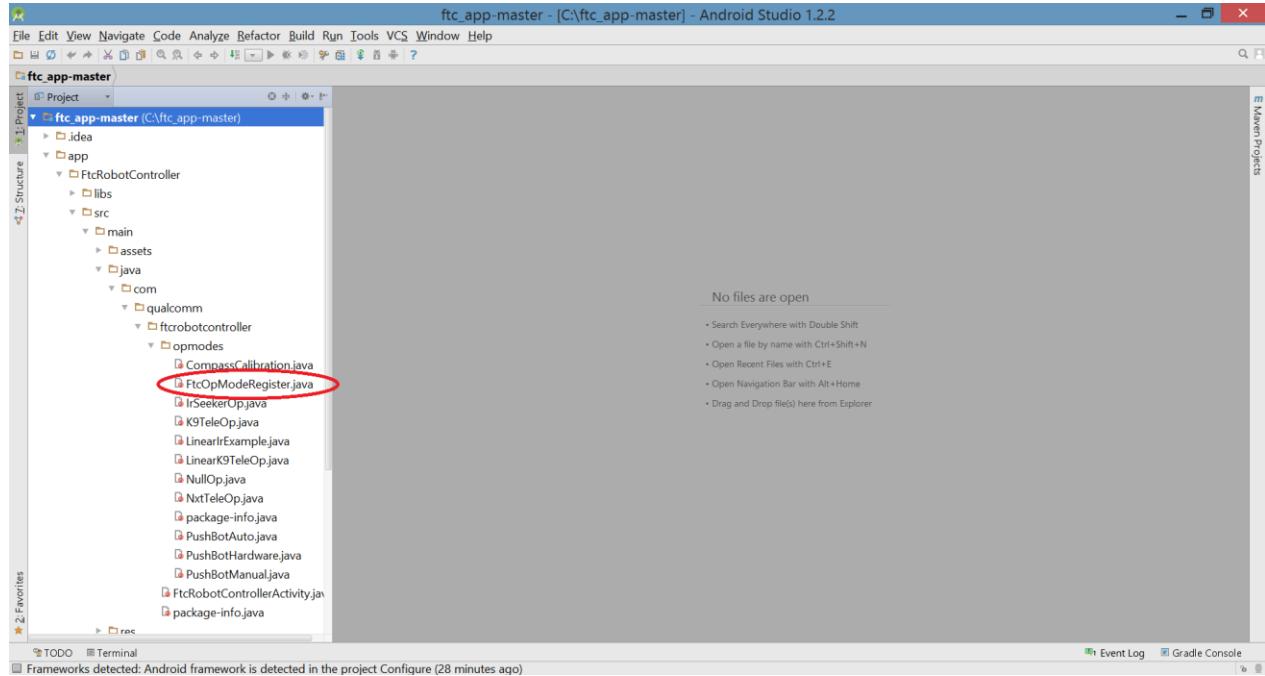
Registering the PushBot Op modes

Before the Driver Station will display an Op mode, it must be registered (i.e. added to the FTC Robot Controller application's list of Op modes). You should verify that the PushBotAuto and PushBotManual Op modes are registered so that they will be listed as an available Op mode on the Driver Station.

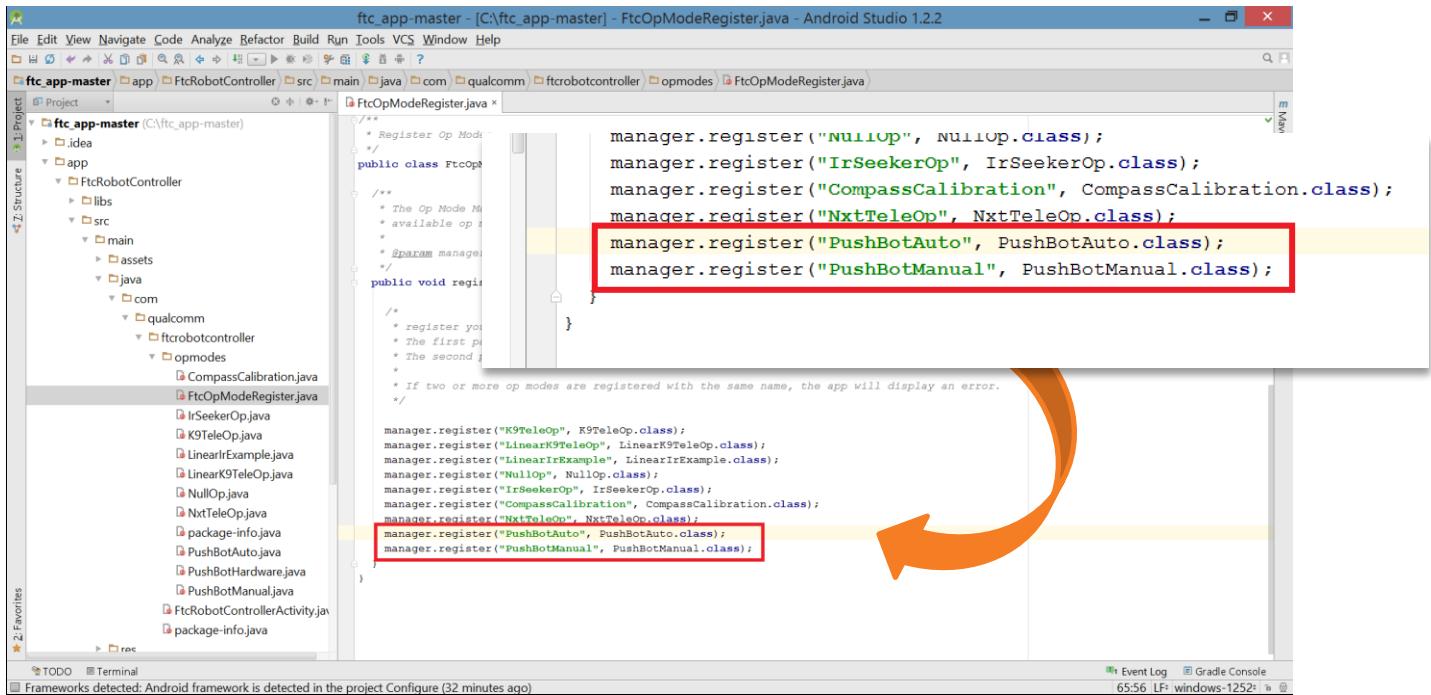
Step 1: If Android Studio isn't still open, open it now. Open the FtcRobotController tree using the small arrow to its left. Continue opening the tree until the Op modes item is shown (i.e. open FtcRobotController, then Java, then com.qualcomm.ftcRobotcontroller, then Op modes). Verify that the PushBot Op modes are in the tree. Note that there are multiple entries in the tree that begin with 'PushBot': PushBotAuto, PushBotHardware and PushBotManual (green circle). Other files have been added since PBv1, which aren't shown in the image below.



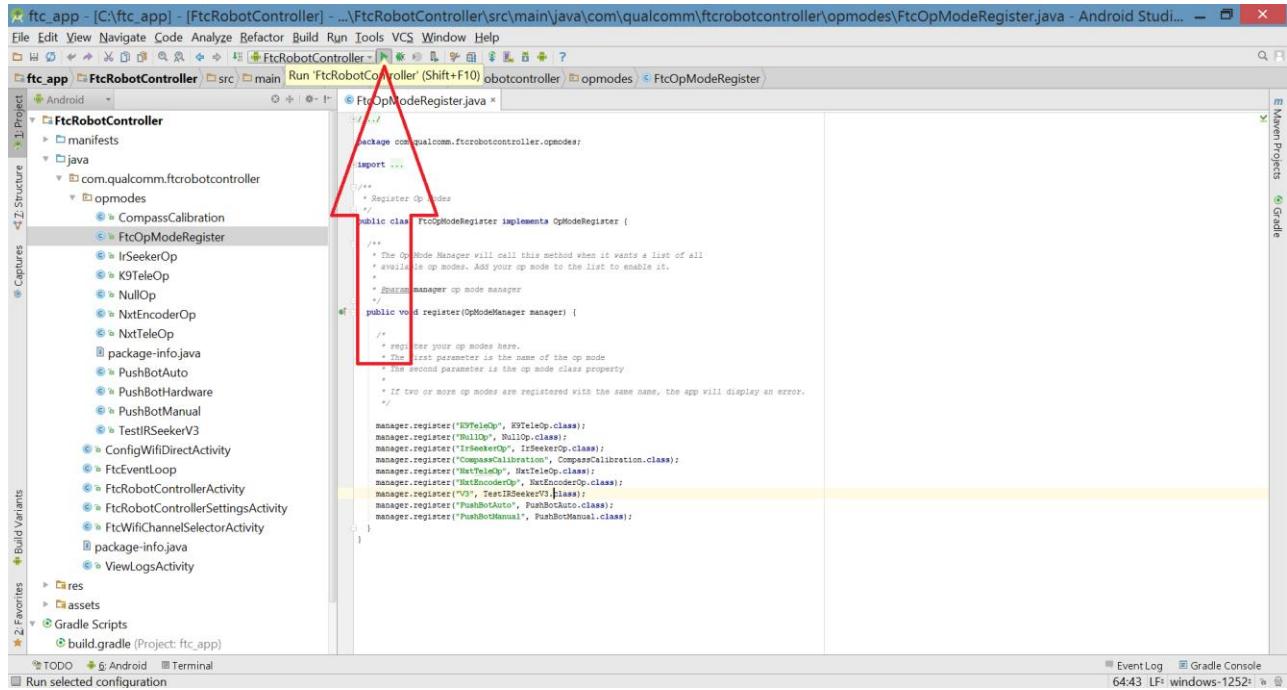
Step 2: Open the FtcOpModeRegister file (red oval) by double clicking it.



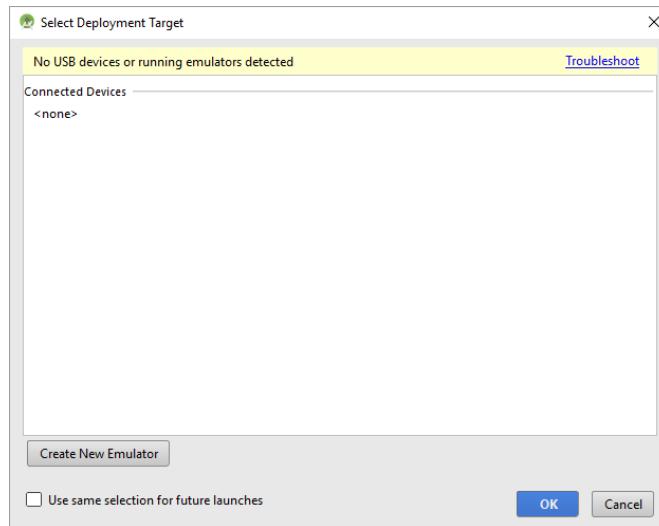
Step 3: Scroll through the file until the 'manager.register' lines are visible; they will have green (strings) and blue (class names) text. Using the existing entries as a guide, type “**manager.register (“PushBotAuto”, PushBotAuto.class);**” and “**manager.register (“PushBotManual”, PushBotManual.class);**”. Do **NOT** create a line for the PushBotHardware class (or the other new Push Bot classes). The new lines are shown in the red rectangle. These lines may already exist in the FtcOpModeRegister file.



Step 4: Build by selecting the green triangular button near the top of the window (red arrow). Note the results of the previous step (green oval). This may take a while. Status is displayed at the bottom of the window.



Step 5: Observe that the build completed successfully and the Choose Device window opens.



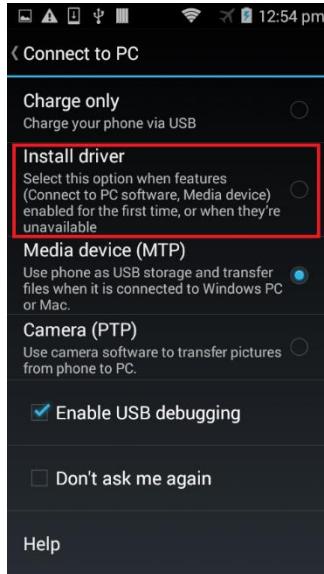
Installing the USB Device Driver

This section applies only to the Robot Controller cell phone. The correct USB driver will need to be installed onto the Windows computer hosting Android Studio. Without the correct driver, Android Studio cannot connect to the cell phone when it is connected via a USB cable. Also the computer will treat the device as a media player or storage device (like a FLASH drive). File Explorer (aka Windows Explorer) might be able to access files on the cell phone in this mode, but Android Studio will not recognize the device as an available Android device.

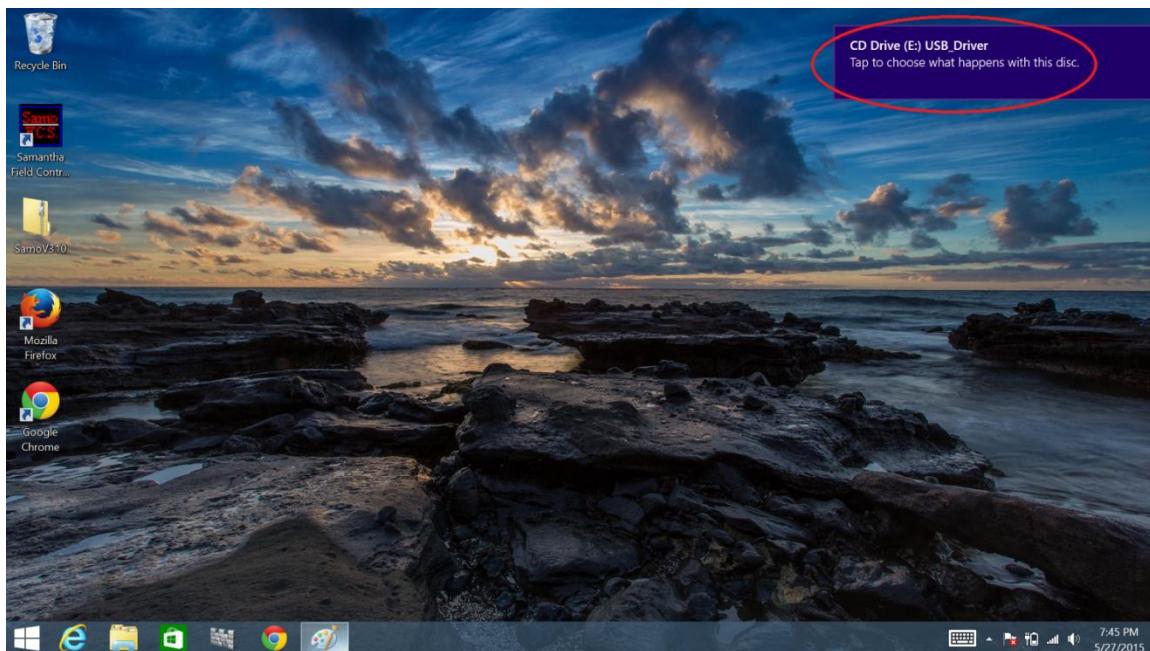
Note: You must perform these steps to utilize Android Studio. Without Installing the Driver, your phone will be treated like a media player/storage device, not a Robot Controller.

Step 1: Using the cell phone's power cable (with the charging block removed so one end is a USB connector), plug the USB connector into a USB port on the phone.

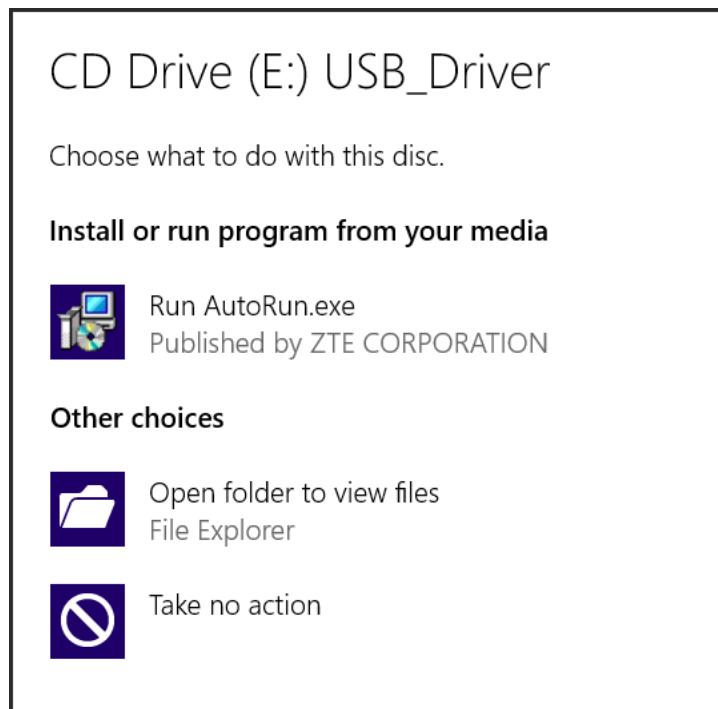
Step 2: Select 'Install driver' on phone.



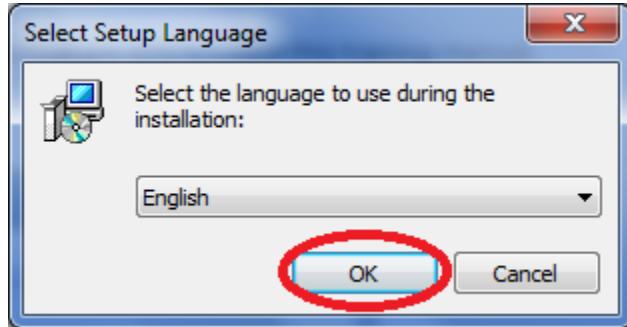
Step 3: When ‘Install driver’ is selected on the phone the computer will detect the connection. Confirm the computer’s prompt.



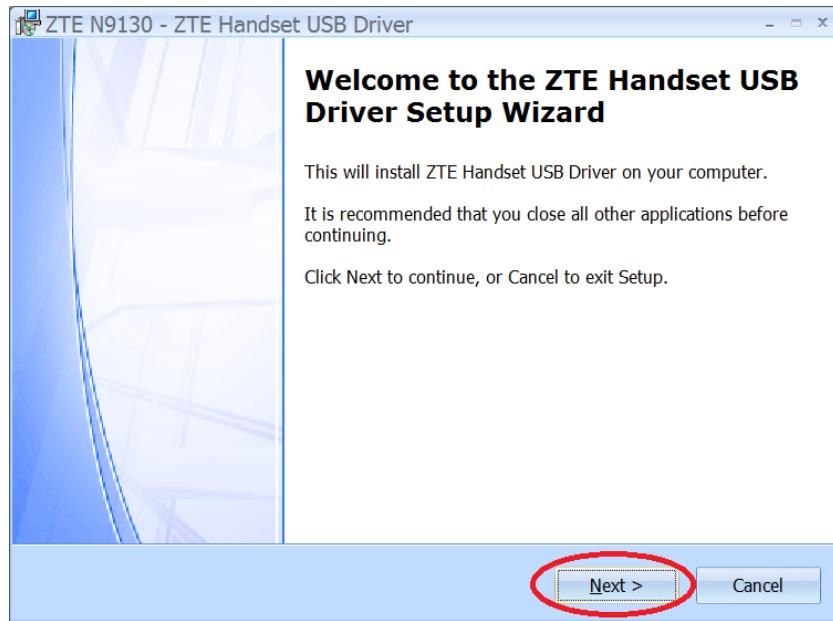
Step 4: After confirming the prompt, another prompt will be displayed. Select ‘Run AutoRun.exe’. If the screen darkens and a dialog asks “Do you want to allow the following program to make changes to this computer?”, select ‘Yes’.



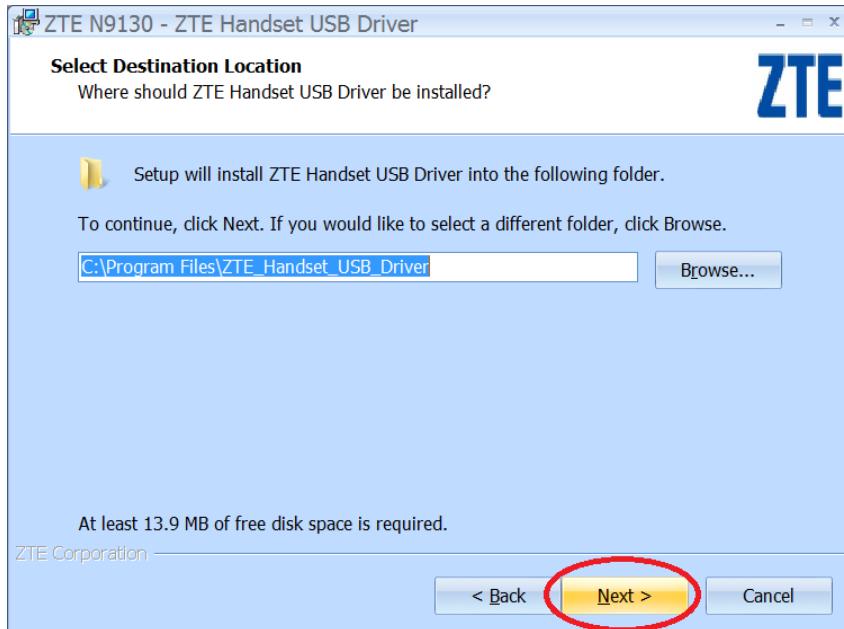
Step 5: The Select Setup Language dialog should be displayed. Select ‘OK’.



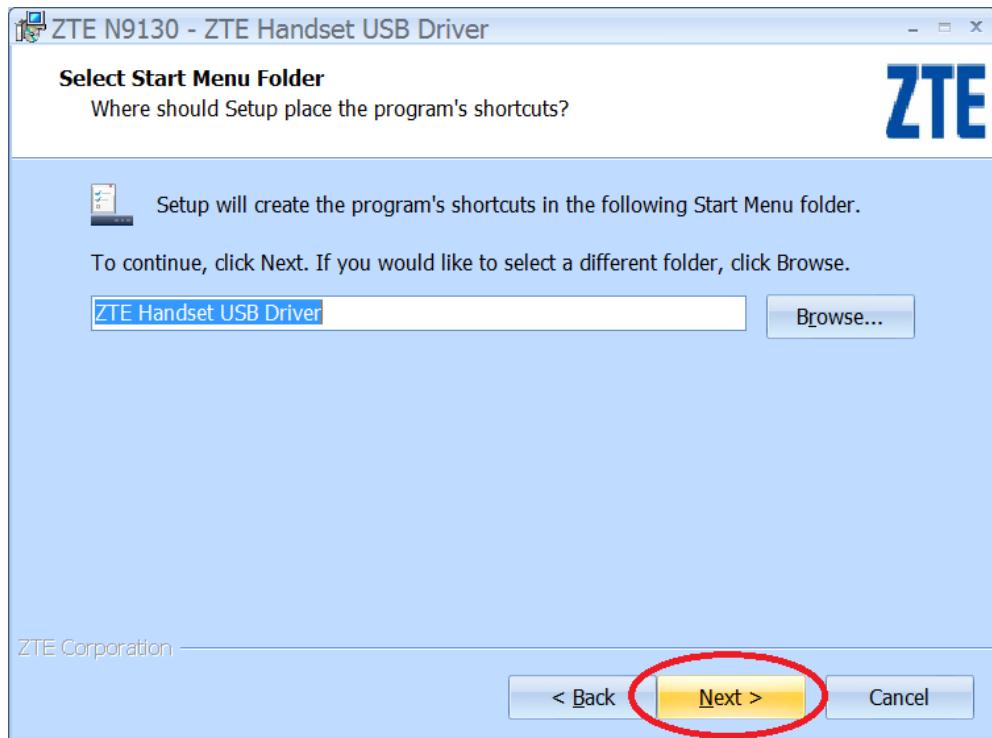
Step 6: Select ‘Next’ to continue with the installation.



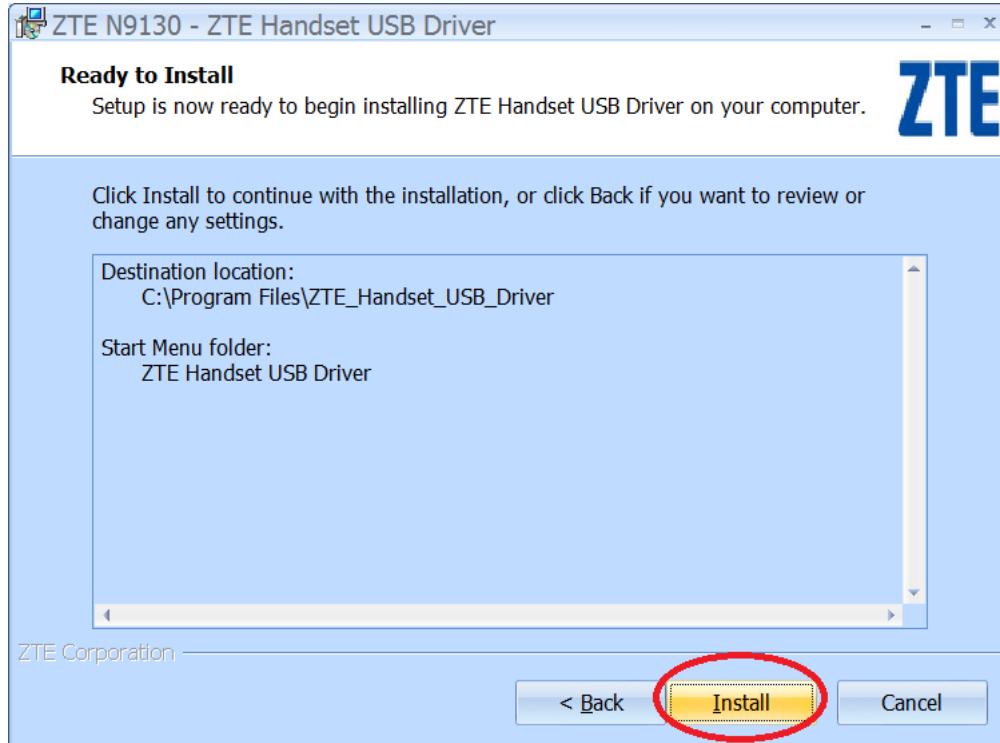
Step 7: Select 'Next' to accept the default location for the driver.

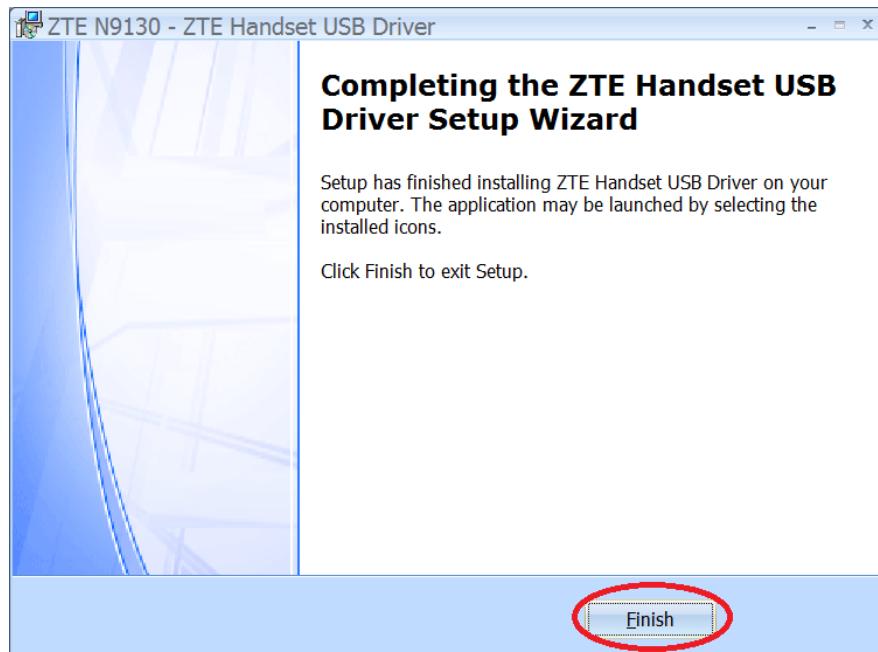
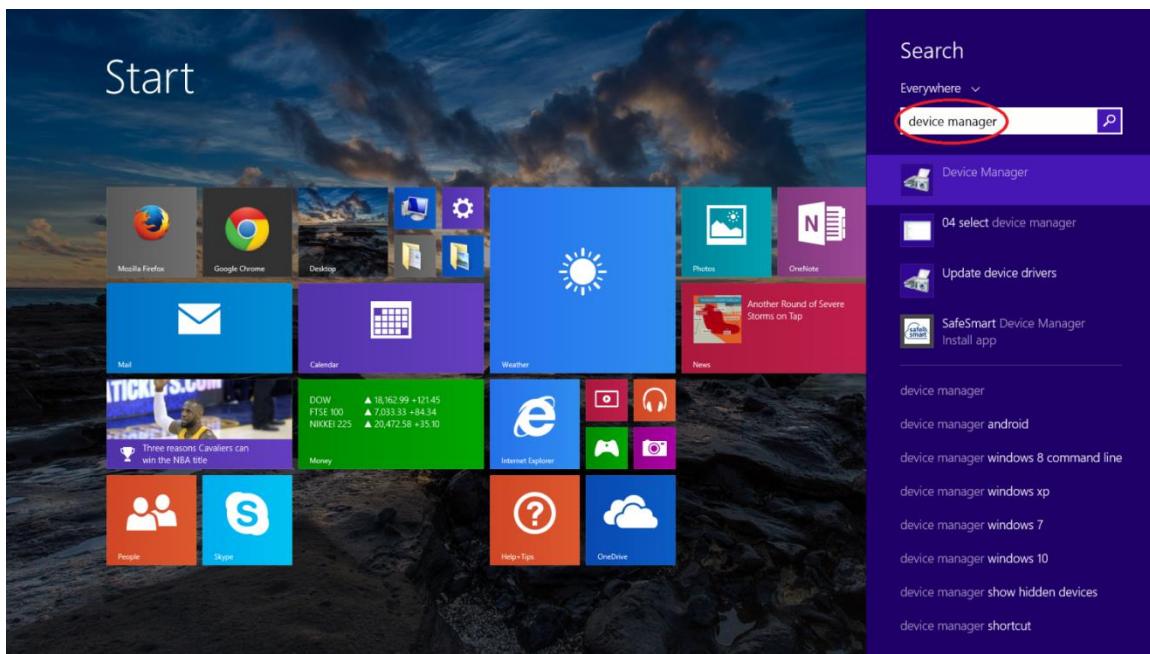


Step 8: Select 'Next' to accept the default location of the driver shortcuts.

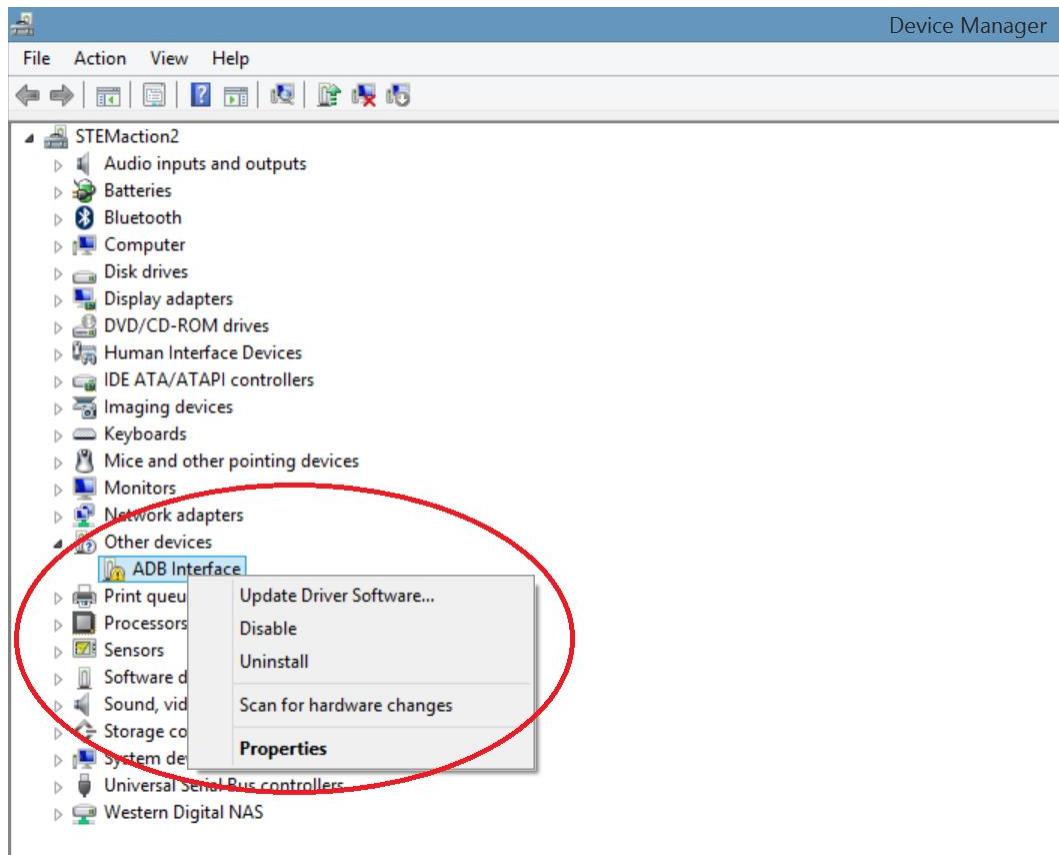


Step 9: Select 'Install'. The image below the following shows status as the driver is being installed.

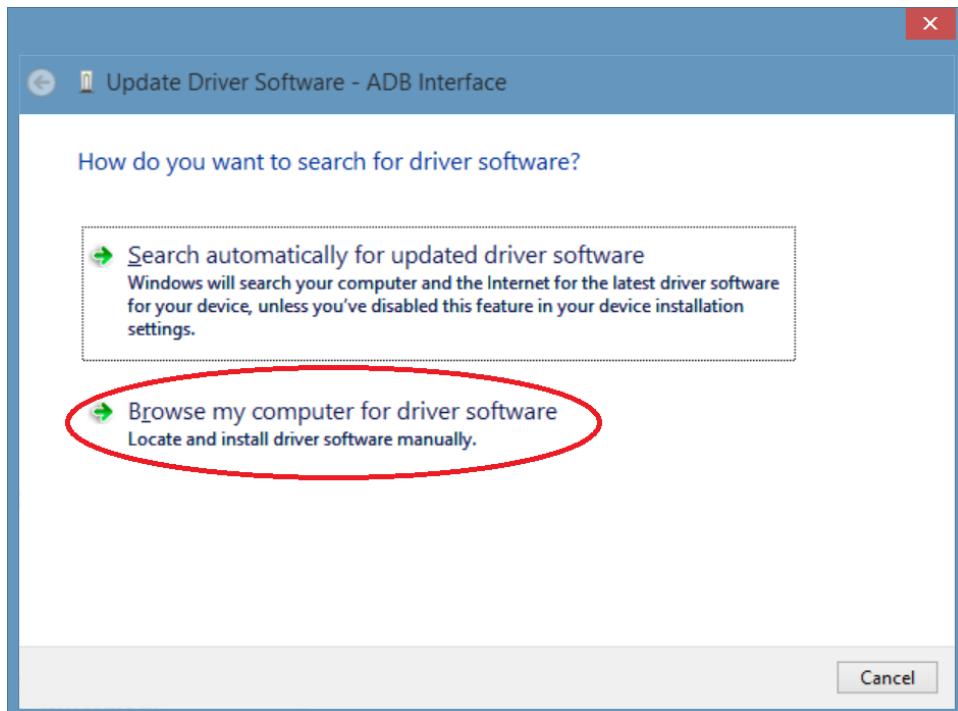


Step 10: Select 'Finish'.**Step 11:** Press the Window key and type 'device manager' and press return (similar to opening Edge).

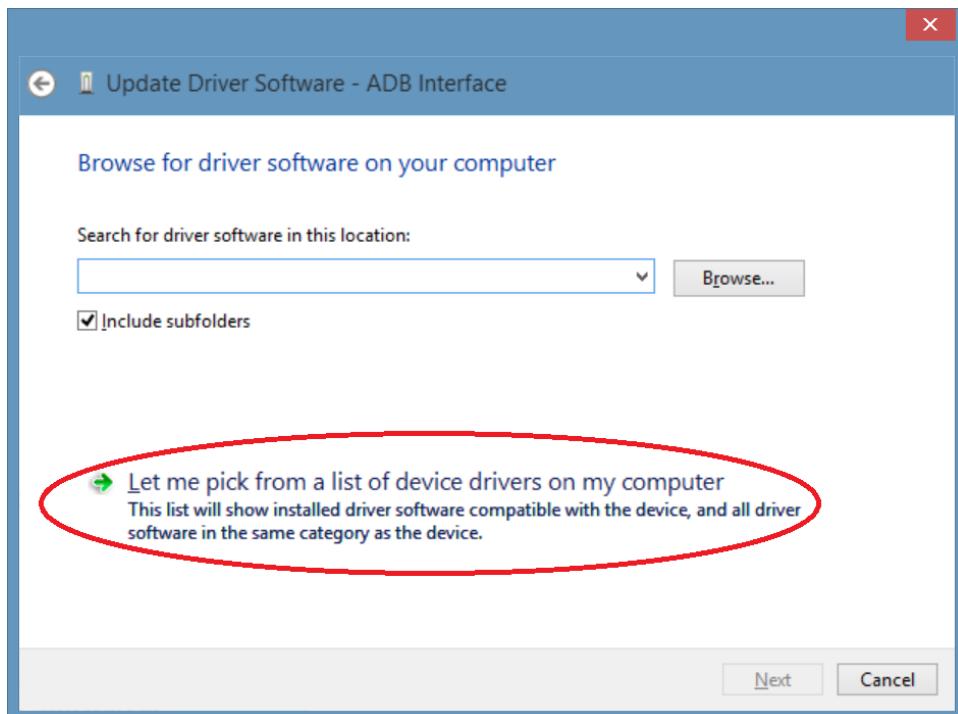
Step 12: Right click on ADB Interface and select Update Driver Software.



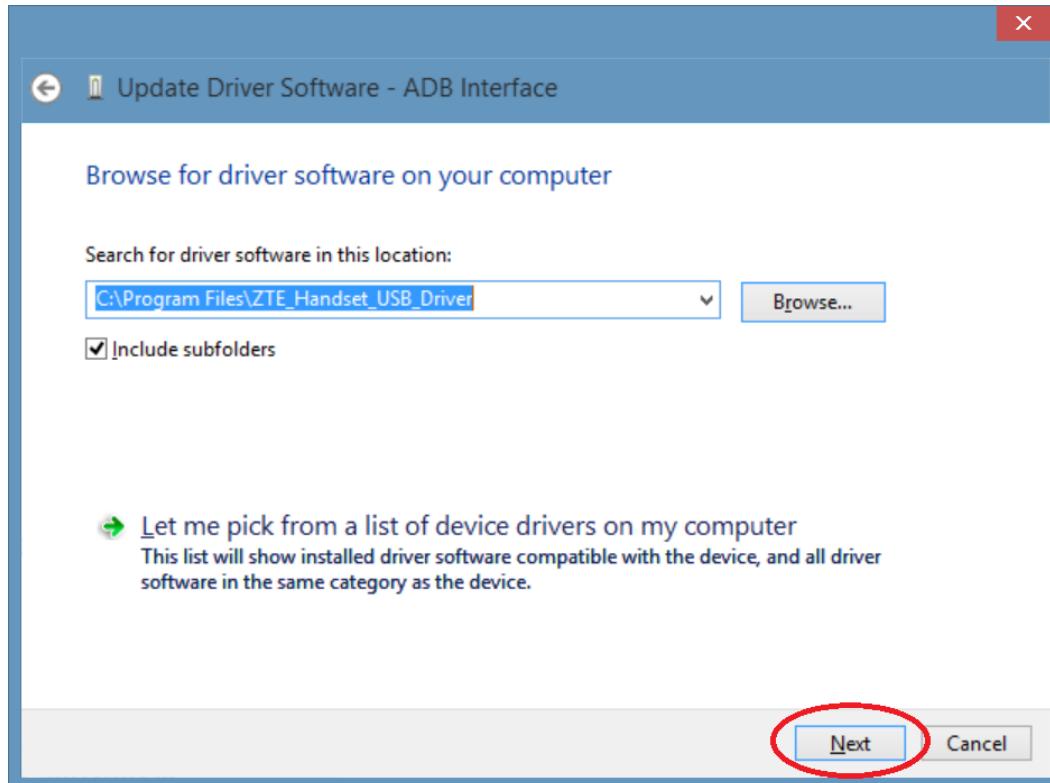
Step 13: Select ‘Browse my computer for driver software’.



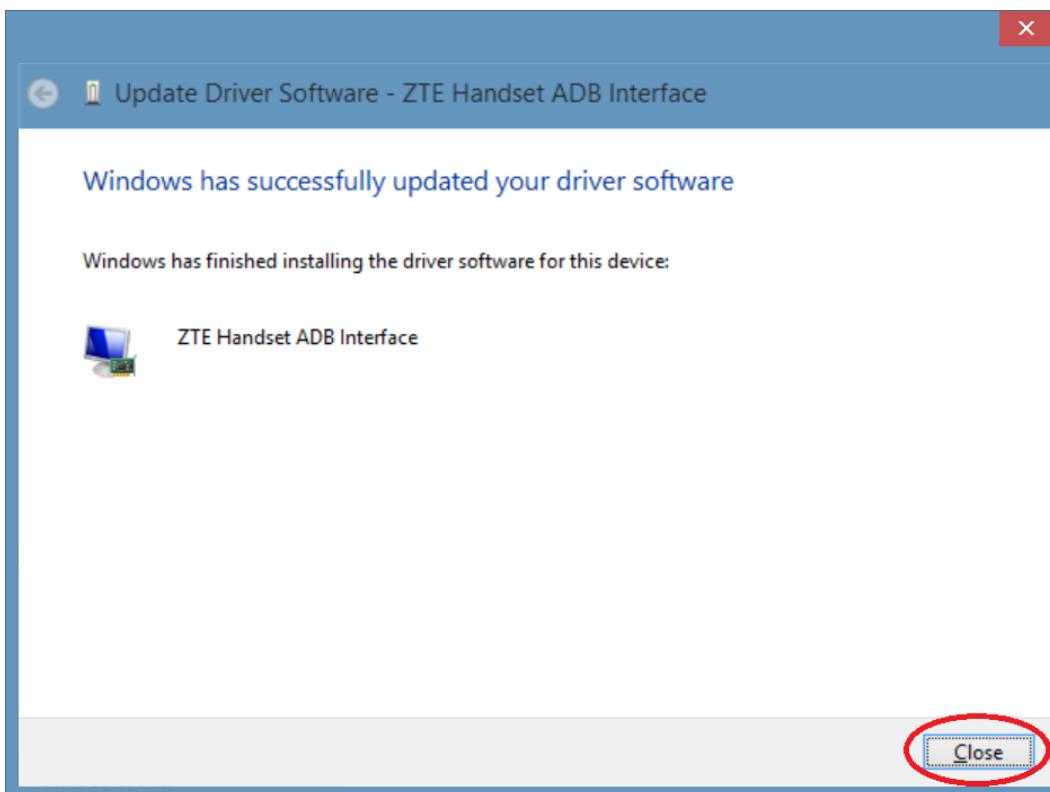
Step 14: Select ‘Let me pick from a list of device drivers on my computer’.



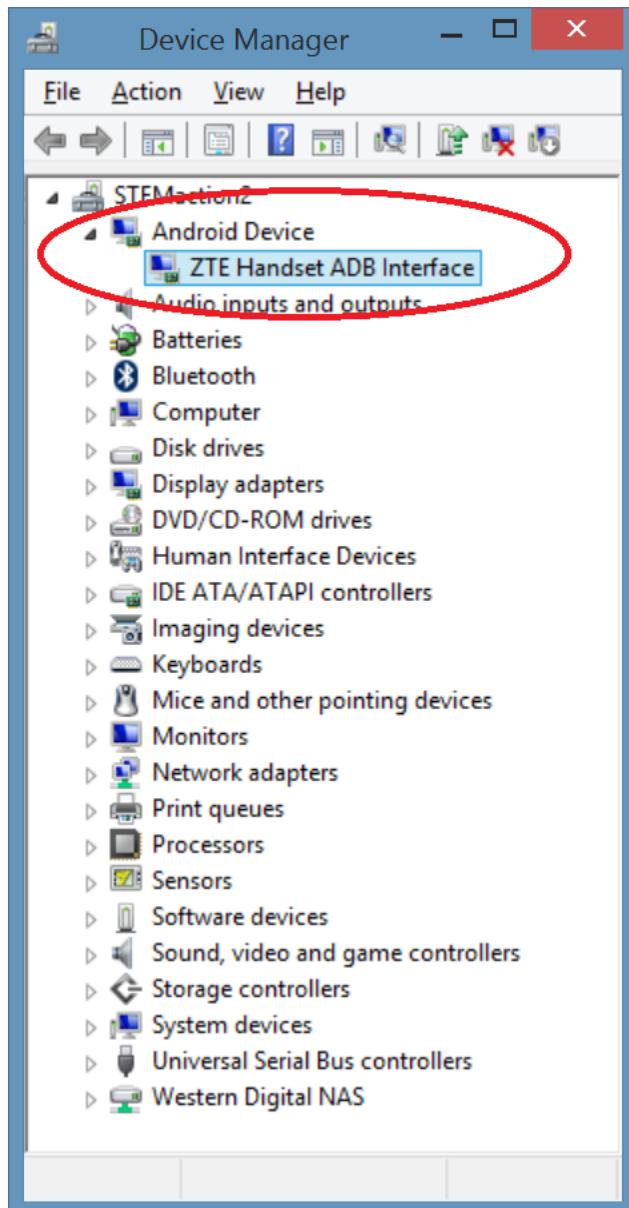
Step 15: Select 'Next'.



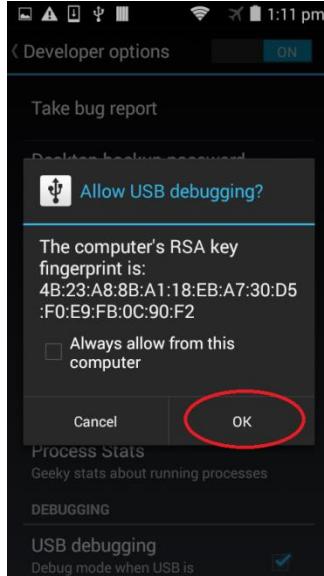
Step 16: Select 'Close'.



Step 17: Observe that the device is now connected. This window can be closed.



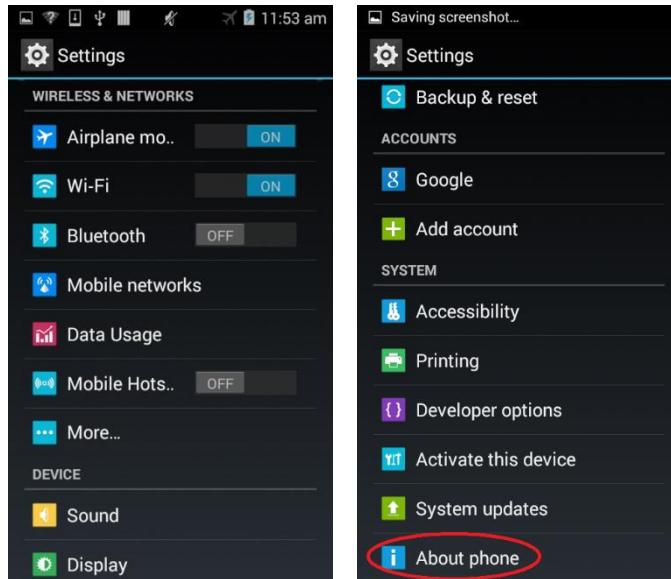
Step 18: Observe that the cell phone is displaying the ‘Allow USB debugging’ window. Select ‘OK’.



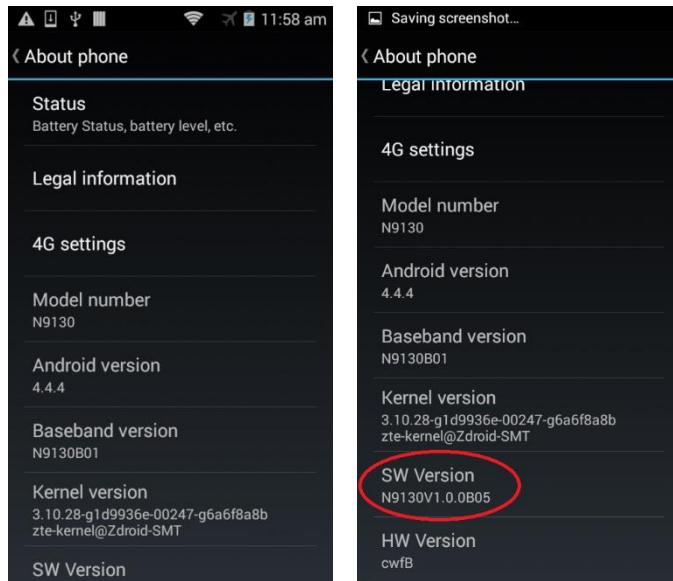
Enabling Debugging Mode on the Robot Controller Cell Phone

For this section choose one phone to be the Driver Station and the other to be the Robot Controller. The Robot Controller will host the Robot Controller app, which implements the PushBot Op modes. Android Studio cannot deploy the Robot Controller app until debugging is enabled on the phone.

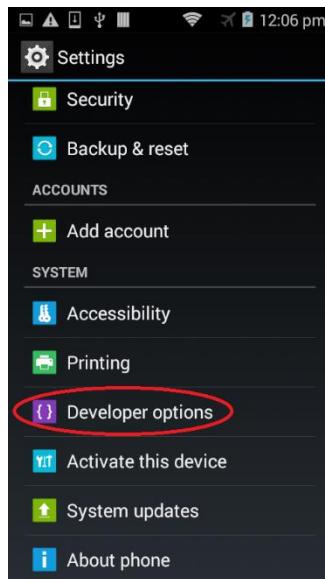
Step 1: From the previous section, the Settings activity should be shown (left image). Scroll downward until ‘About phone.’ Is visible (right image) – it should be at the very bottom. Select ‘About phone’.



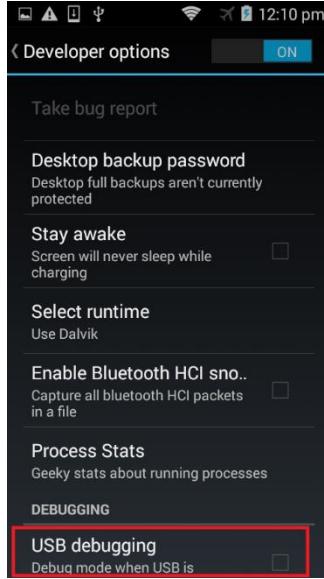
Step 2: The About phone activity (left screen) allows a developer to enable debugging. Scroll downward until ‘SW Version’ is shown (right image) - on some Android devices the text says ‘Build number’. Tap ‘SW Version’ 7 times. A toast (a message that is displayed for a short amount of time) should display as the number of taps increase and finally should indicate that debugging is enabled.



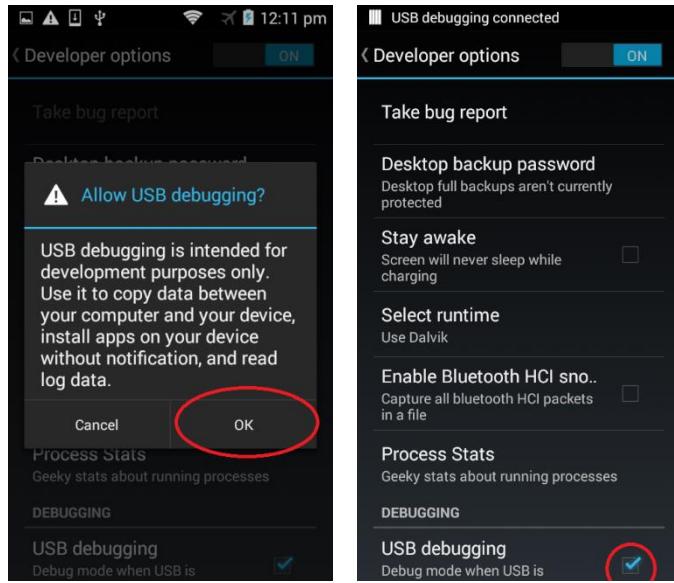
Step 3: If Android hasn't already switched to the Settings activity, then do so now. Scroll to the bottom of the page and select ‘Developer options’. The developer options button is invisible until debugging is enabled (see previous step).



Step 4: Select the USB debugging checkbox.



Step 5: Select 'OK' to allow USB debugging between the phone and a computer when the phone is connected via USB to the computer hosting Android Studio. Note that the box now has a check in it (right image).



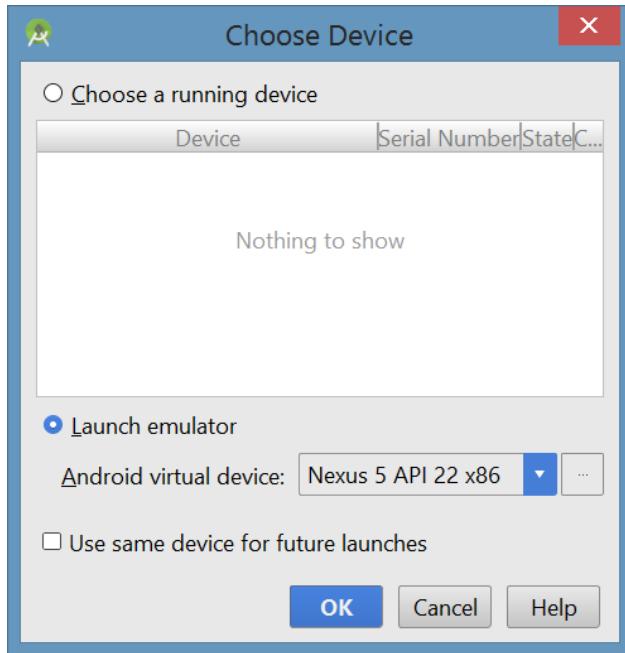
Deploying the Robot Controller

For this section choose one phone to be the Driver Station and the other to be the Robot Controller. The Robot Controller will host the Robot Controller app, which implements the PushBot Op modes. Android Studio can't deploy the Robot Controller app until debugging is enabled on the phone, so that will be the first task.

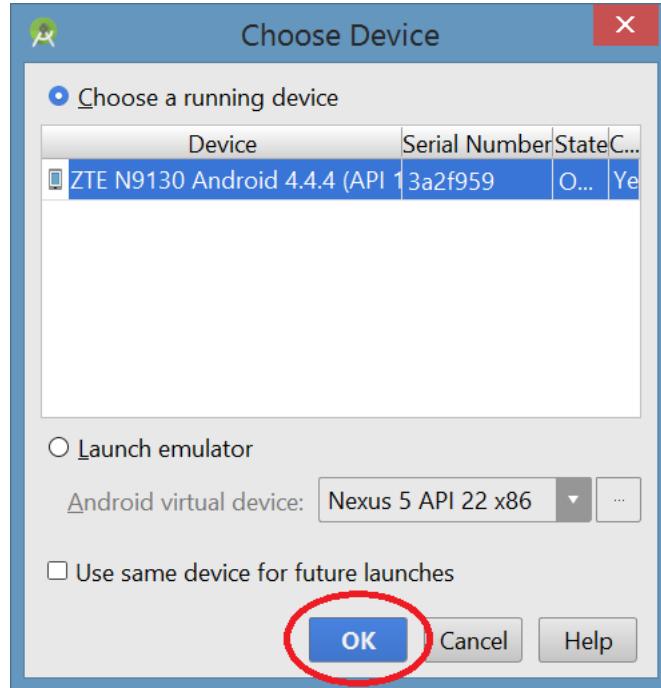
The correct driver for your Android device will need to be installed onto your Windows computer. If your Windows computer does not have the correct driver for your phone installed, then when you connect your device to the computer via a USB cable, the Windows computer will think that the device is a media player or

storage device (like a Flash drive). You might be able to access files on your device in this mode, but your Android Studio software will not recognize the device as an available Android device that it can talk to.

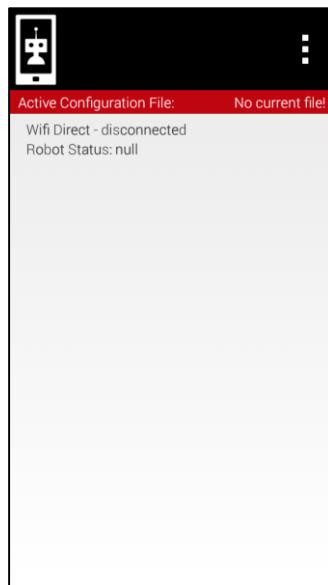
Step 1: If you are following the PushBot Build Guide in sequence, Android Studio may still be open and displaying the Choose Device dialog. Close the Choose Device dialog if it is still open and the device list is empty (shown below). If Android Studio isn't open, then start it. Build the project (small green triangular button –shown in the aforementioned section).



Step 2: The device should be shown in the list. Select 'OK'.



Step 3: Observe that the application starts and the Active Configuration File activity is displayed.



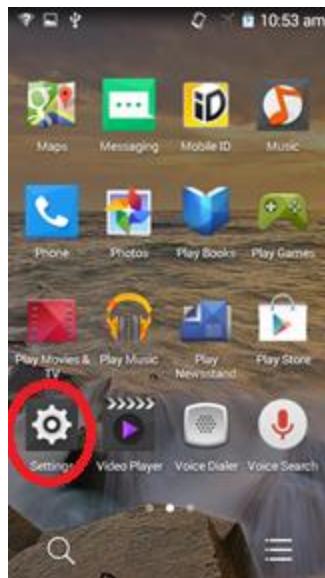
Deploying the Driver Station

The driver station is the Android device that sits near the Team drivers and is connected to one or two USB gamepads. Teams use the touch screen and the gamepads to operate their Robot remotely. The driver station Android device runs a special app called the FTC Driver Station. The instructions in this section are directed at the Driver Station cell phone unless noted otherwise.

Step 1: Open the application activity by selecting its icon (shown in the red circle).



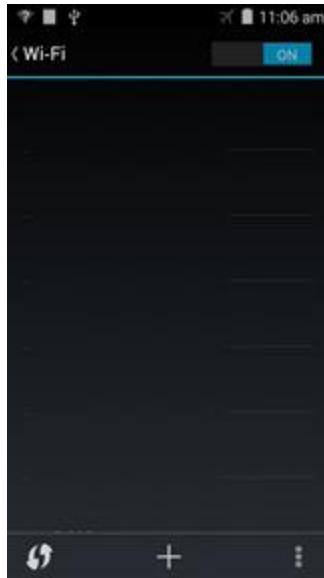
Step 2: Open the settings activity by selecting its icon (shown in the red circle). It might be necessary to swipe the screen left or right until the icon comes into view.



Step 3: Click on the word ‘Wi-Fi’, not the on/off button.



Step 4: Connect to a Wi-Fi network that is able to reach the Internet and to which the login information is known. The image below shows no networks. If networks are available, then they will be shown in the middle of the window. Select a Wi-Fi network and enter any required login information.

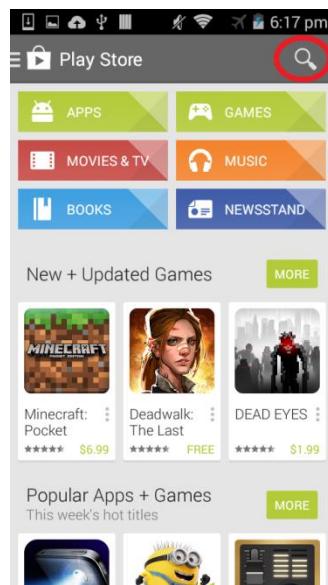


Step 5: Return to the home screen by pressing the home button (the center icon on the bottom of the phone’s screen).

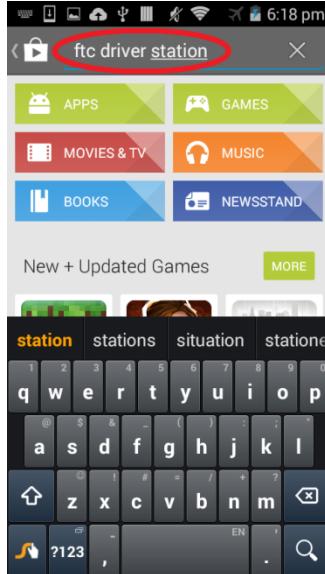
Step 6: Launch the Google Play Store app (orange circle). The Play Store app might prompt you to either login to an existing Google account or create a new one. Follow the onscreen instructions to either create a new (free) account or login to your existing account. The Driver Station app is free and no payment method is required. If the application requests a credit card number or some other method of payment, select the “Skip” button.



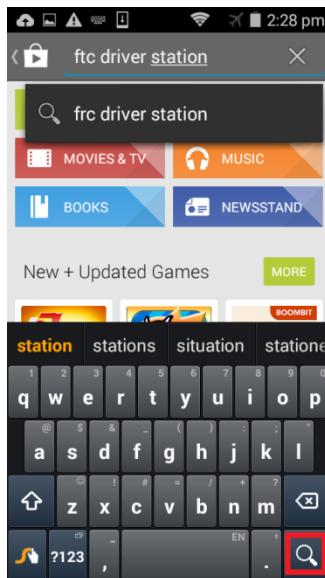
Step 7: Click on the search icon.



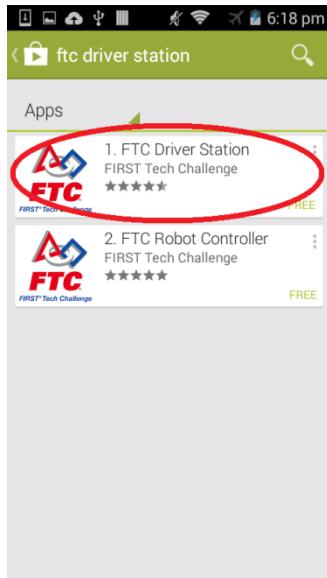
Step 8: Enter the phrase “FTC Driver Station” and press the magnifying glass in the bottom right corner (on the keyboard)



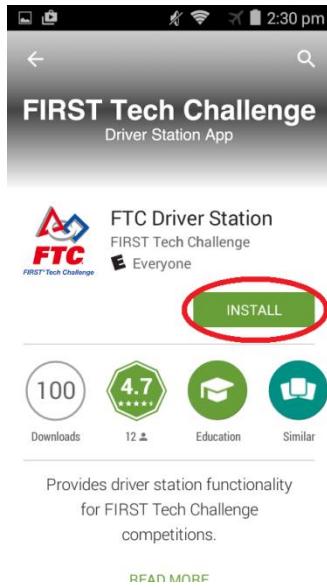
Step 9: And press the magnifying glass in the bottom right corner (on the keyboard).



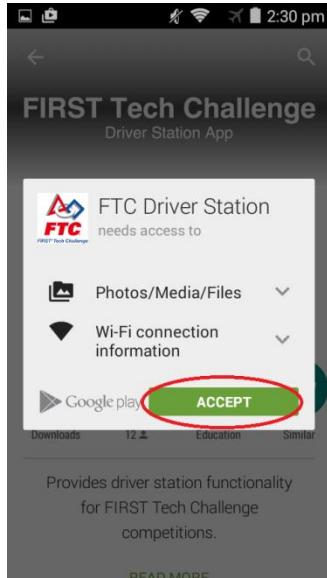
Step 10: Select the FTC Driver Station block.



Step 11: Select 'INSTALL'.



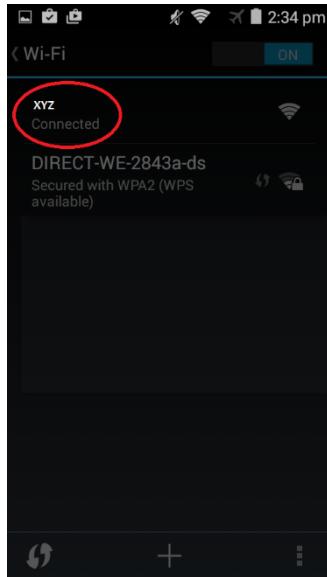
Step 12: Select 'Accept'.



Step 13: Return to the home screen. Observe the FTC Driver Station icon is displayed there. If it isn't shown on the homescreen then search for it in the app drawer.

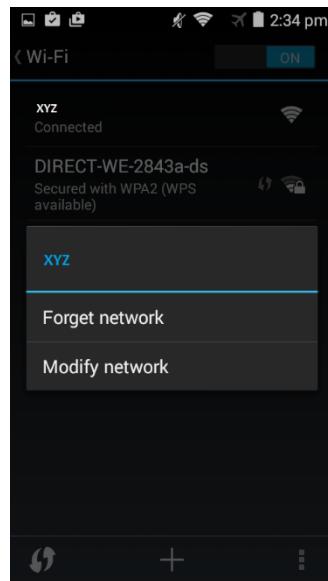


Step 14: (IMPORTANT) The network needs to be forgotten to prevent connection issues between the Driver Station cell phone and the Robot Controller cell phone. Open the Wi-Fi setting activity as shown earlier in this section. Long hold the name of the connected network.



Note: Remember to “forget” your networks!

Step 15: The following screen should be displayed. Select ‘Forget network’. In general, the Driver Station and Robot Controller cell phones should not be connected to any wireless networks with the exception of the Wi-Fi Direct connection between the two cell phones.



Recording the Controller Serial Numbers

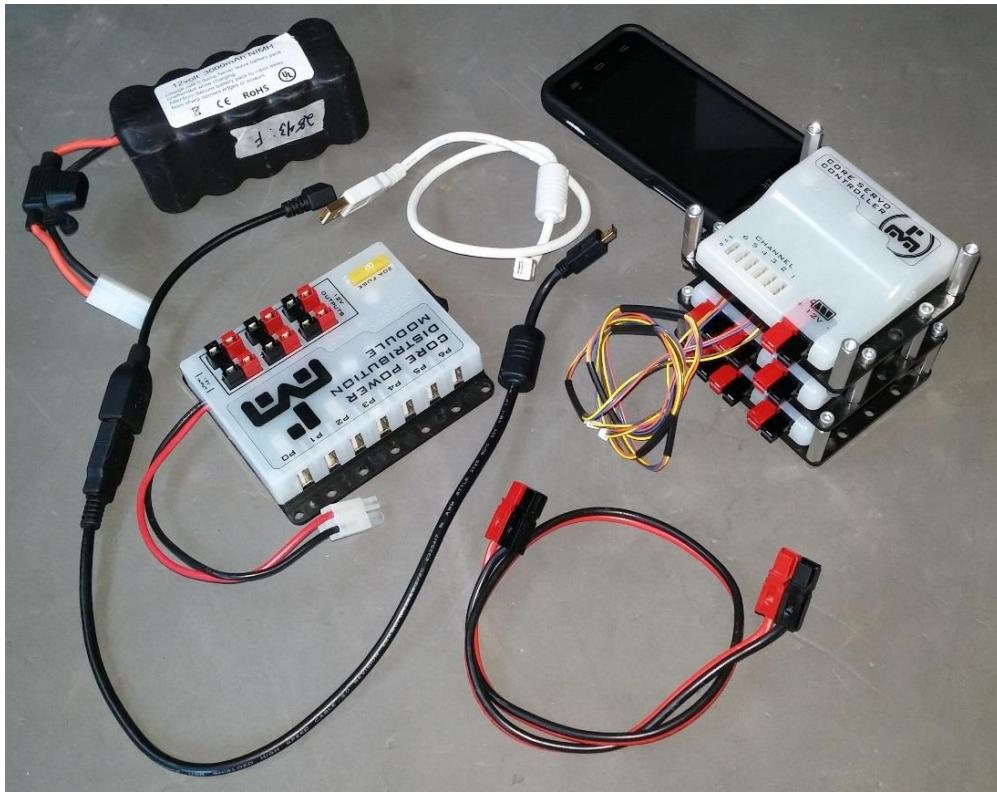
This section discusses the connection between the software and the hardware. It is easiest to perform this section before starting the “Building the Robot” section. The image below shows the controller stack from “Building the Robot” to clarify the steps in this section - sometimes the controller will be referred to as the bottom, middle, or top and/or DC Motor Controller 1, DC Motor Controller 2, Servo Motor Controller 1, etc. However, this section can be completed (and is even expected to be completed) with the controllers disassembled from each other. The encoder cables aren’t necessary - they are shown only to clarify the idea behind the controller stack both in the section and the “Building the Robot” section.

Even though the kit provides multiple DC power, USB to Mini B and USB to Micro B cables, only one of each will be used in this section (as shown below).

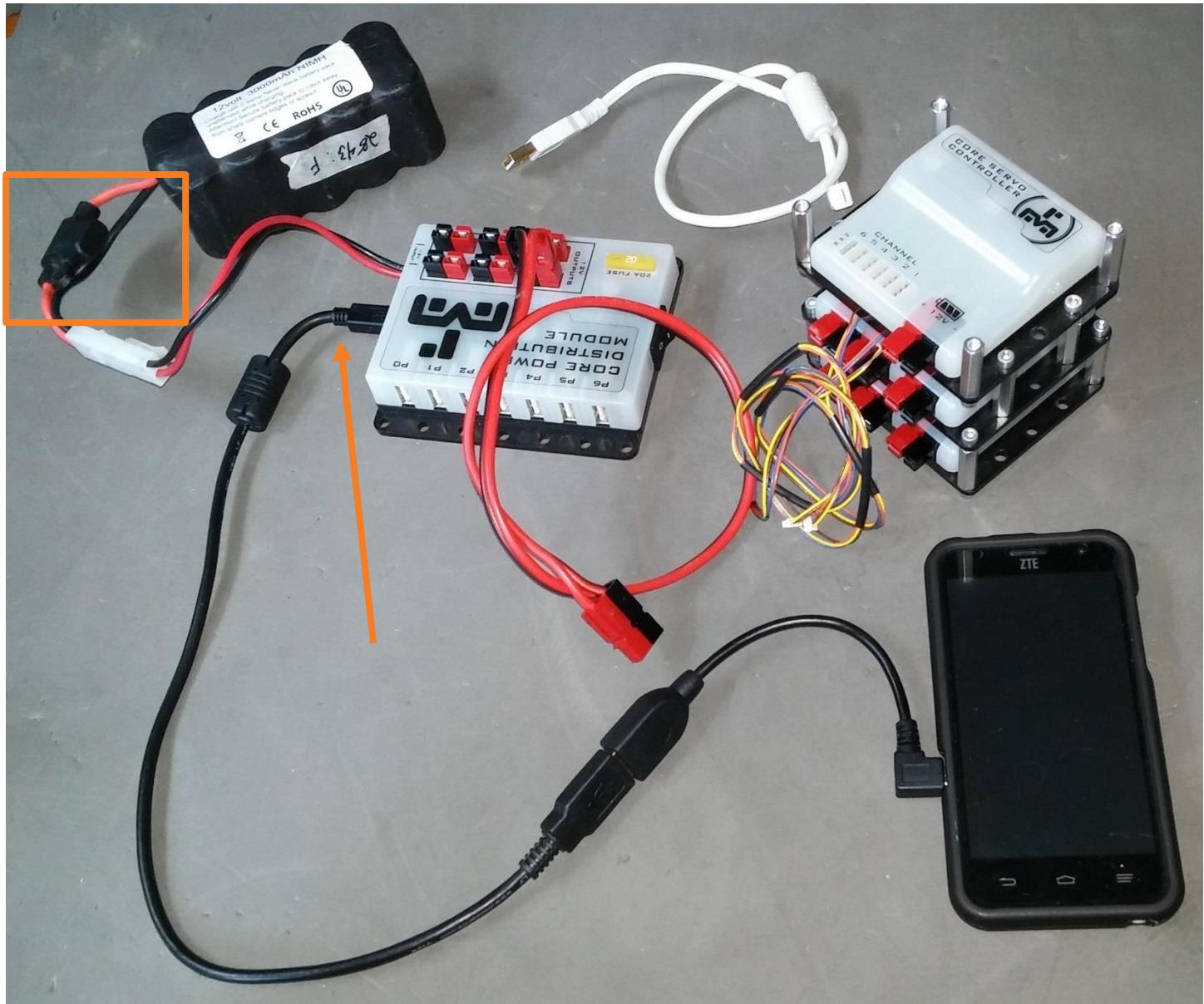
Note that the battery must be fully charged to complete this section.

Note also that the Robot Controller (RC) phone is used and the RC app must be installed (see “Deploying the Robot Controller...” section).

The configuration information collected in this section isn’t stored as a configuration file on the RC phone, because this section assumes the controller stack has not yet been built. If controllers are stacked in a different order in the “Building the Robot” section, the configuration file contents may become invalid. Creating the configuration file is completed in the Configuring the USB devices section.



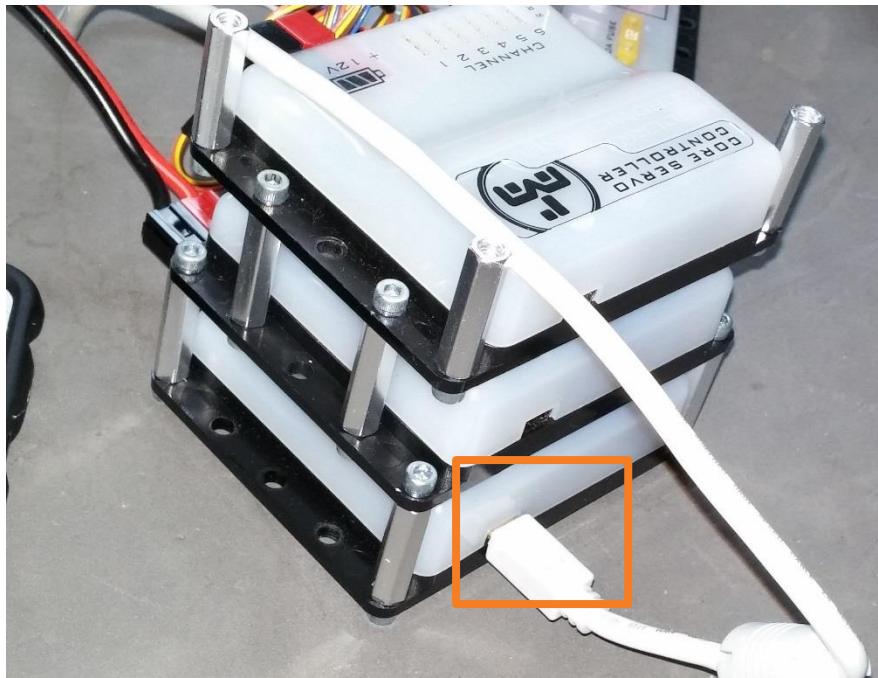
Step 1: Plug the Power Distribution Module (PDM), provided in the kit, into a charged battery. Plug the RC phone into the PDM. The image below shows a two-piece cable between the phone and PDM. The cable(s) provided in the Kit of Parts may vary - it may actually be a single piece cable.



Step 2: Plug the USB end of a USB to Mini B cable into the PDM. Plug the other end into one of the controllers. The top image shows the cable plugged into the PDM. The bottom image shows the other end of the cable is plugged into the controller. Plug a power cable between the PDM and controller.

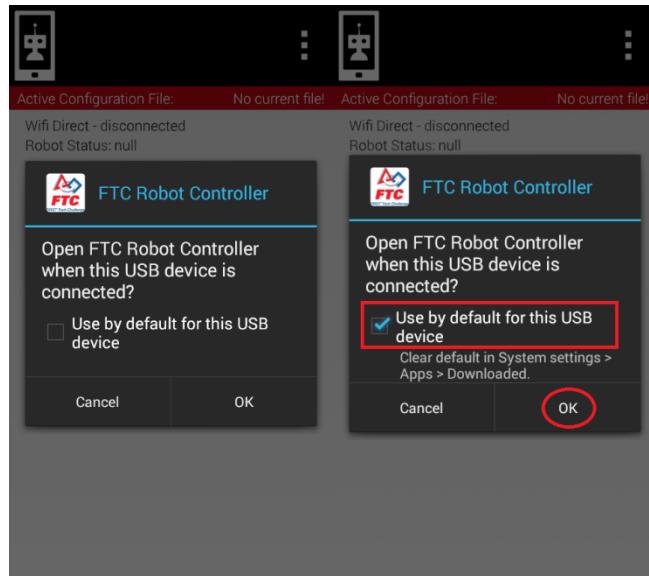


1:Cable plugged into Power Distribution Model

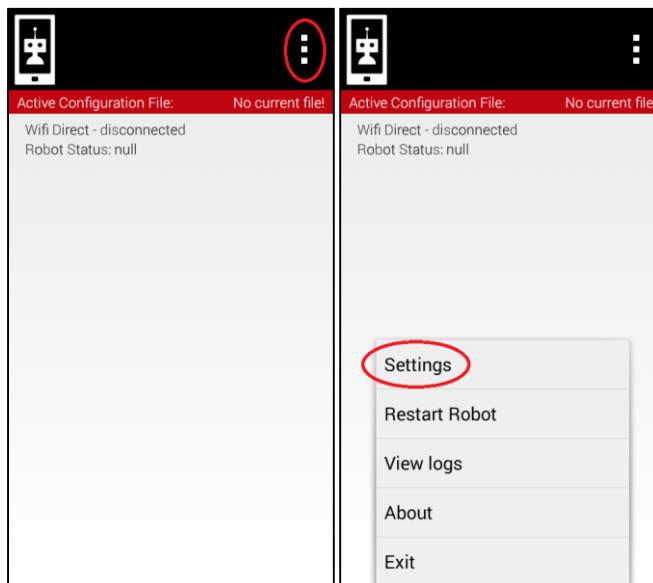


2: Micro End plugged into Core Servo Controller

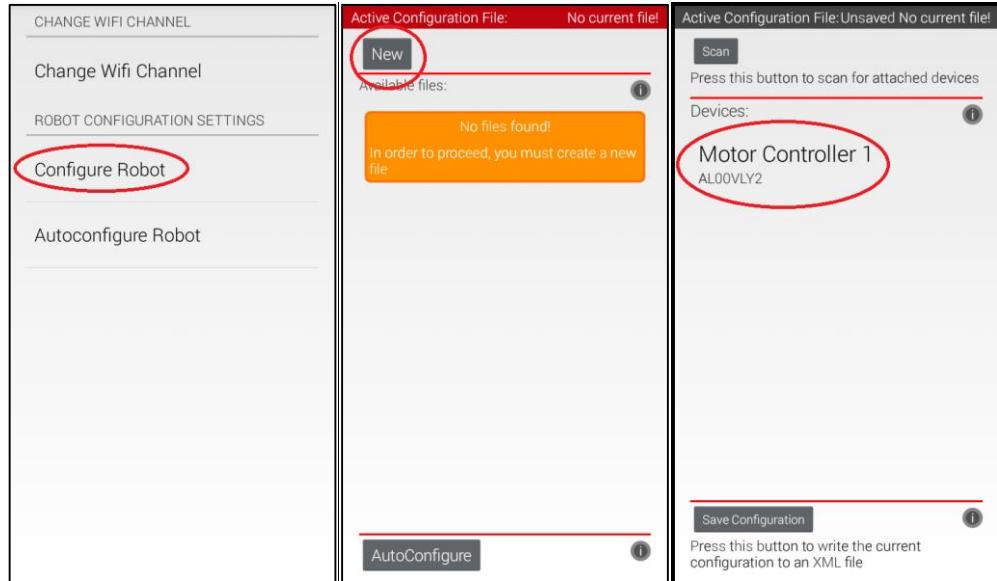
Step 3: Boot the RC phone if it is not already running. Apply power to the Power Distribution Module, which should start the RC app automatically, if it is not already running (far left image). Since this is the first connection of the Power Distribution Module to the phone, the phone will display a message requesting permission for the app to start automatically when that PDM is plugged into it. Check the Use by default box (red box). Click 'OK' (red circle).



Step 4: Open the RC app's settings using the three dots in the top right corner and selecting settings.

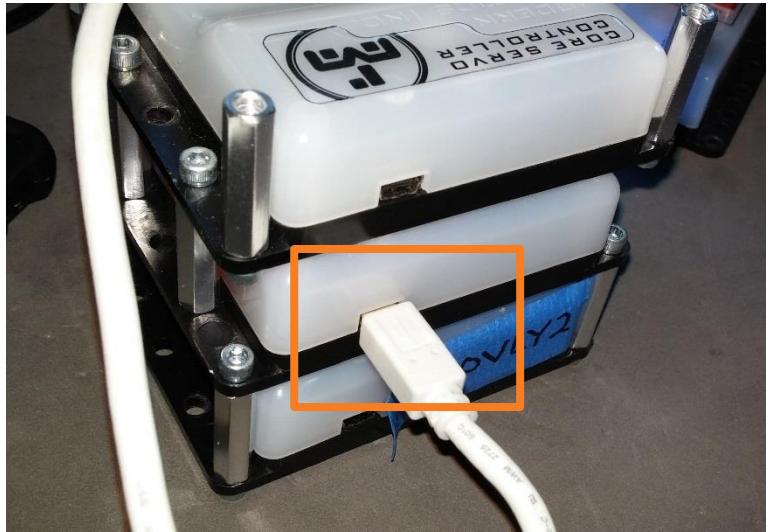
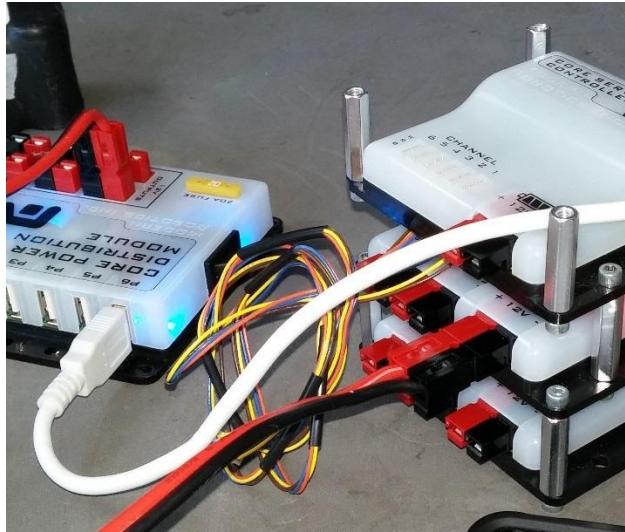


Step 5: Choose ‘Configure Robot’. Choose ‘New’. Select ‘Scan’ and the screen should change to show the connected Controller (third image from the left). Record the serial number onto the controller similarly to the bottom image. This document shows the use of tape and marker. Marker directly onto the case is also an option.

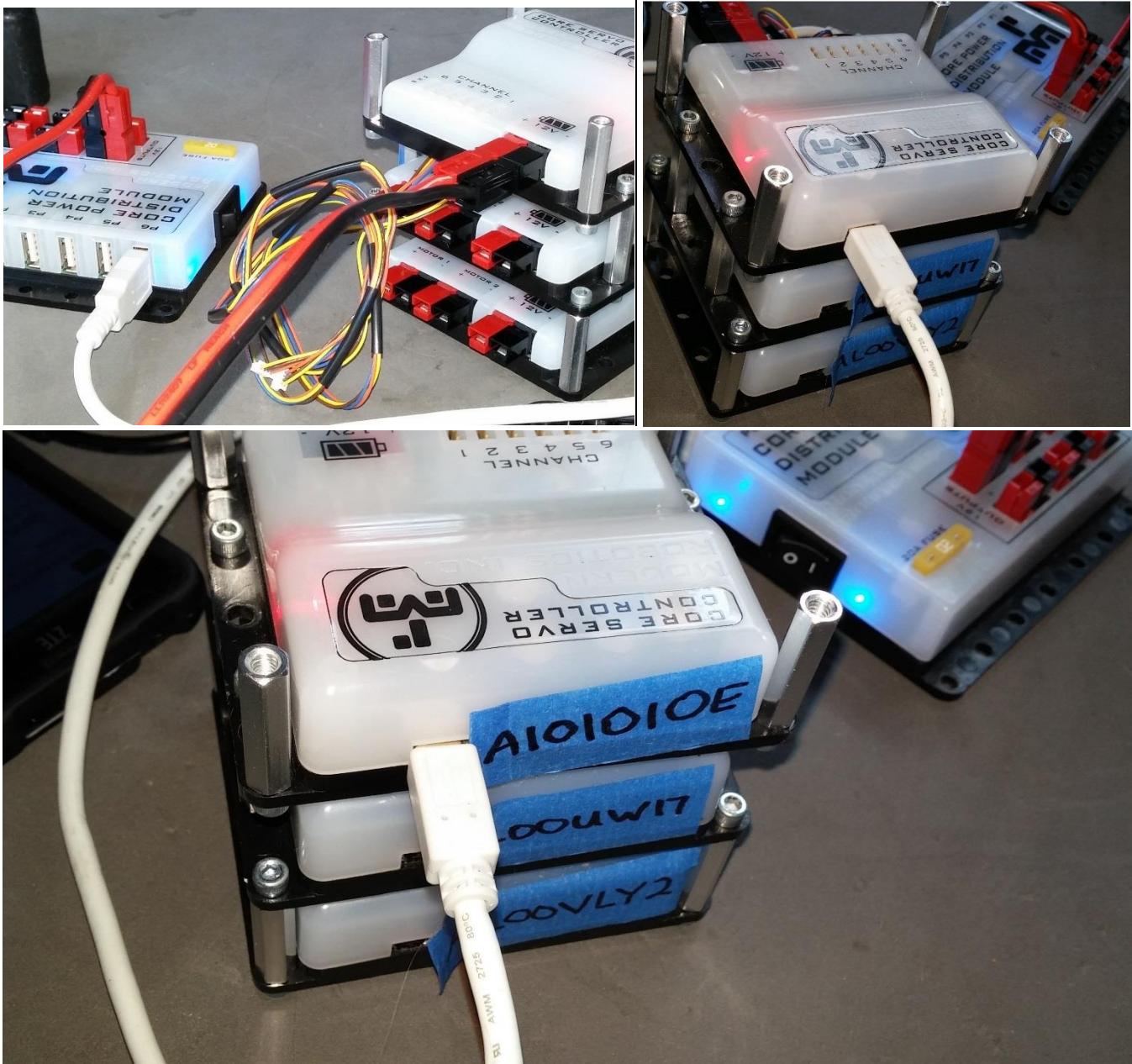


3: Label each component's Serial Number. Here, the Servo Controller's Serial Number is labeled.

Step 6: Move the power cable from the current controller to another controller (left image). Move the USB cable from the current controller to the current powered controller (right image). Repeat steps 4-5 to record the serial number onto the second controller (bottom image).



Step 7: Move the power cable from the current controller to the last controller (left image). Move the USB cable from the current controller to the last controller (right image). Repeat steps 4-5 to record the serial number onto the second controller (bottom image).



Building the Robot

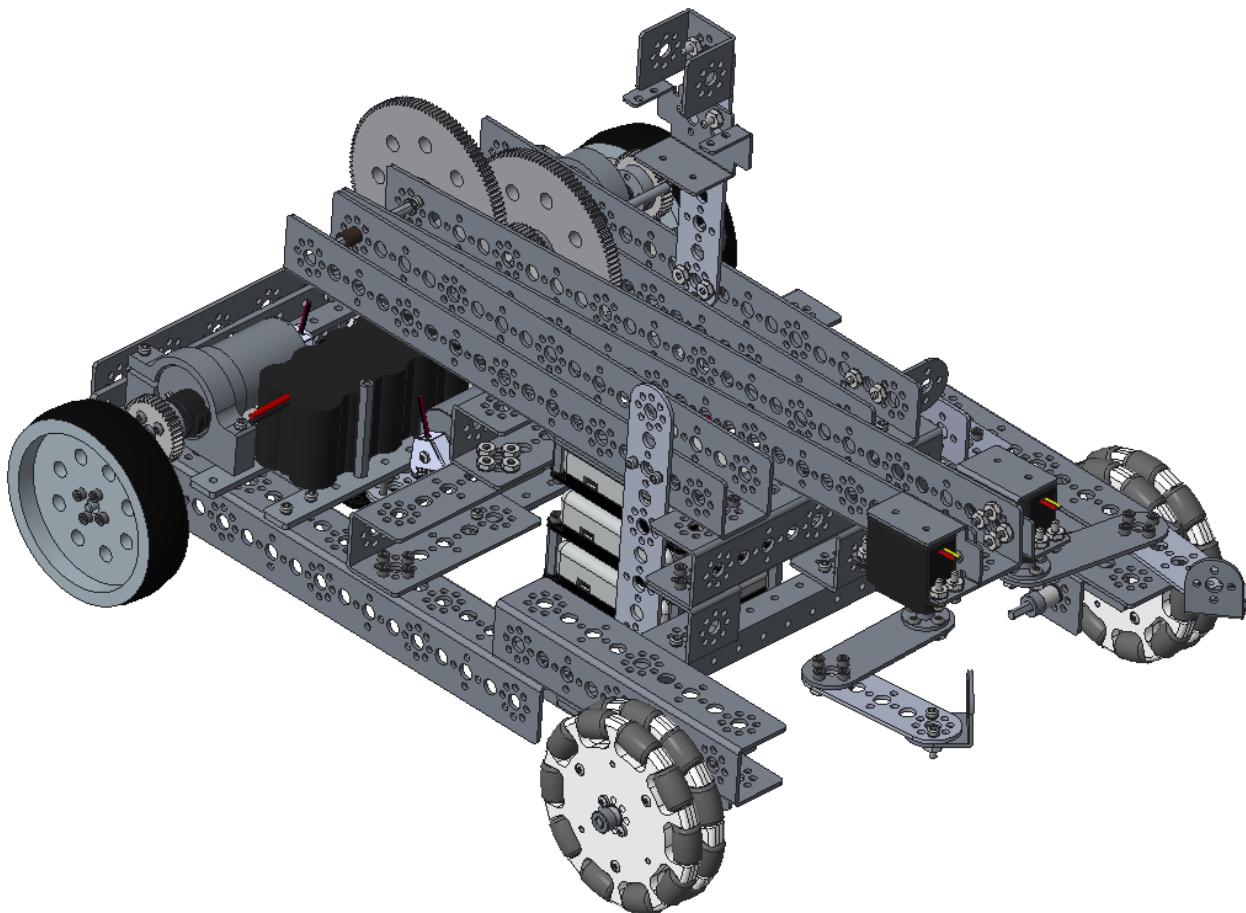
This section will outline the construction of the robot, version 3 of the PushBot. The robot is constructed using only the parts in the TETRIX Kit of Parts. The standard tools included in the Kit of Parts set will be needed to build the robot. In addition, it will be helpful to purchase additional zip ties for securing the wiring to the robot frame. The standard four-inch size works well for this. Also, electrical tape can be used to secure the motor wires to the motor and longer zip ties or Velcro straps can be used to keep the battery from falling out of the robot in case it tips over. Make sure that when using an axle hub, motor hub, or axle collar that the set screw is installed so that they do not just spin on the axle. Also, make sure that all assemblies are square. It is hard to drive a crooked robot straight!

The following sections show the assembly of the robot shown below. For most of the steps in those sections, the top image shows the necessary parts; the lower image shows the completed assembly.

Robot CAD Drawing

The robot was designed using Creo Parametric Computer Aided Design (CAD) software. CAD software is used to design an object on a computer. By designing on the computer first, the design can be tested to insure everything will work together before actual construction. The Creo software is available for free to FIRST teams for use in designing their robots.

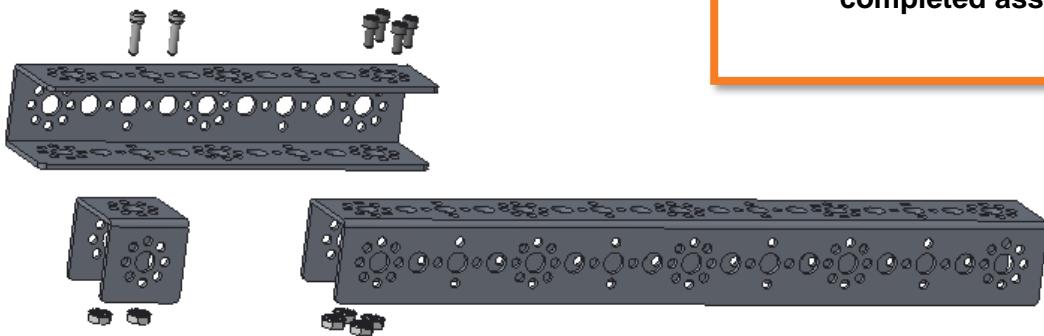
This is a representation of what the robot will look like at the end of the build



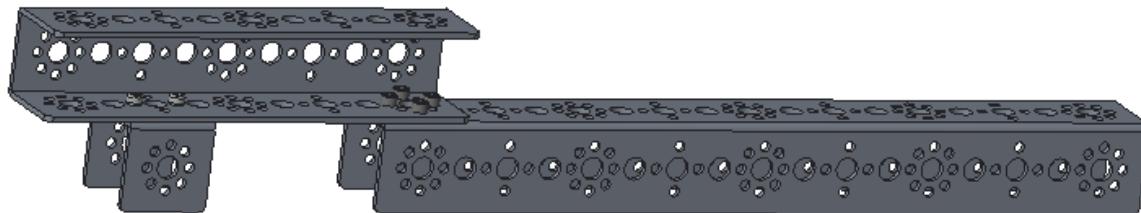
Creating the Chassis

Step 1: Left Chassis Rail. Parts: 32 mm channel (1), 160 mm channel (1), 288 mm channel (1), 5/16" socket head cap screws (4), 1/2" socket head cap screws (2), keps nuts (6).

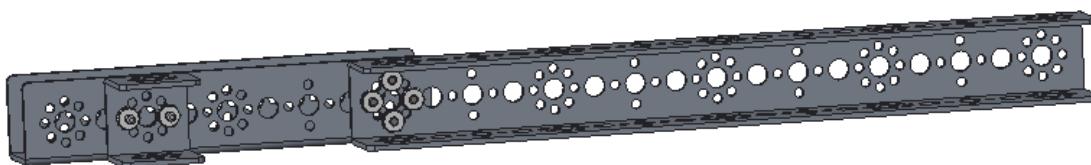
In this and subsequent steps, the parts required for the step are listed first. The top image shows the parts in an exploded view; the lower image shows the completed assembly.



4: Parts needed in Step 1 of Creating the Chassis

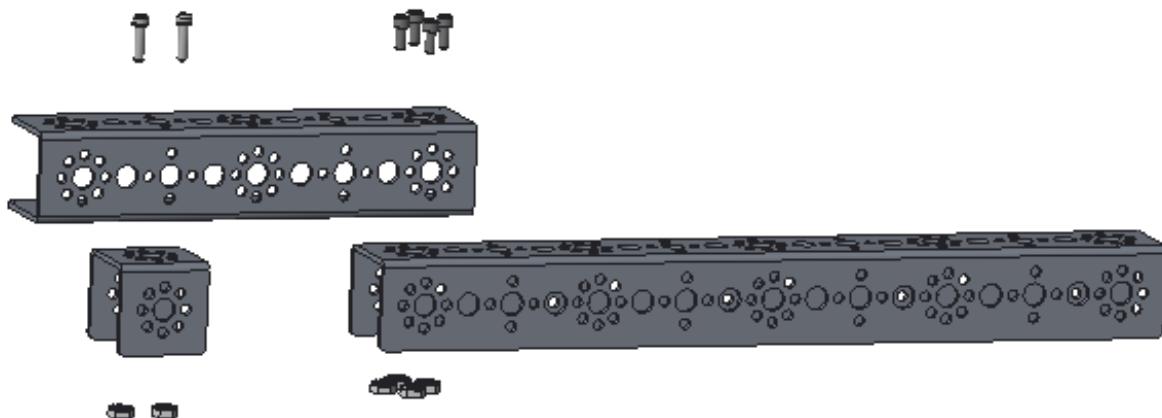


5: Completed Step 1 of Creating the Chassis

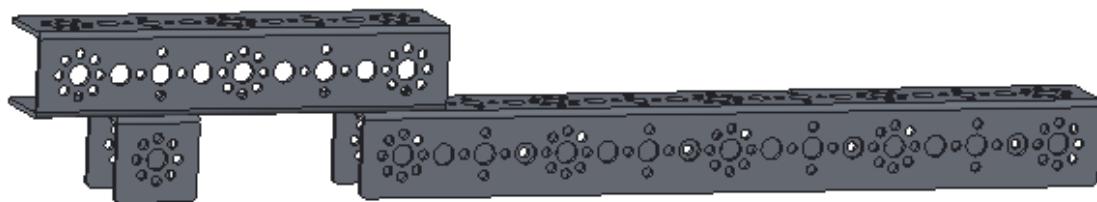


6: Detail View of Step 1 of Creating the Chassis

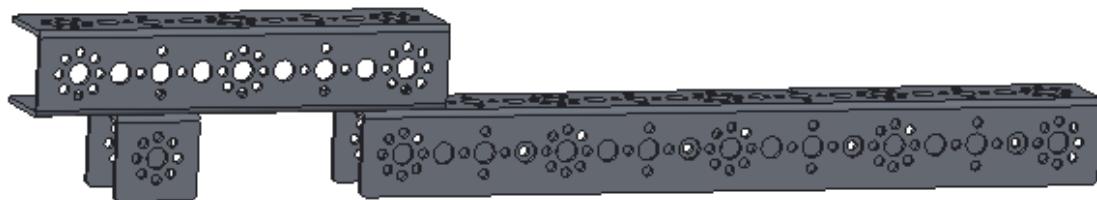
Step 2: [a mirror image of the left chassis rail assembly built in the previous step] Right Chassis Rail. Parts: 32 mm channel (1), 160 mm channel (1), 288 mm channel (1), 5/16" socket head cap screws (4), 1/2" socket head cap screws (2), keps nuts (6).



7: Parts needed in Step 2 of Creating the Chassis

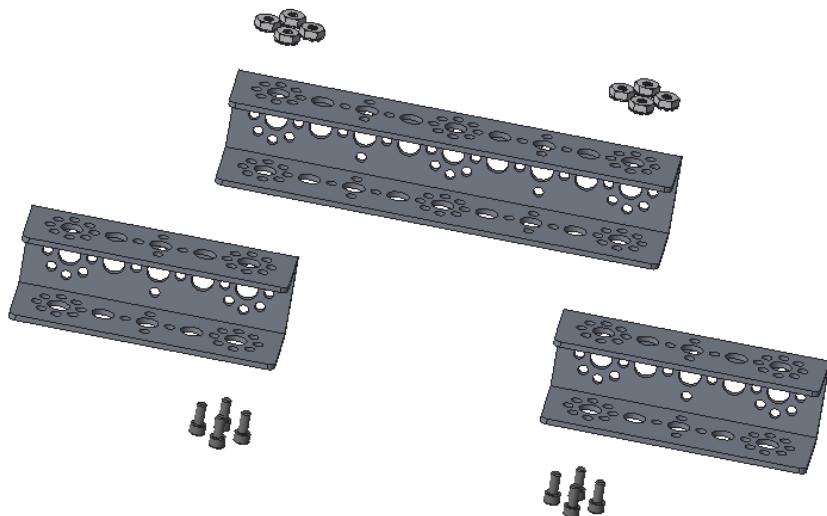


8: Completed Step 2 of Creating the Chassis

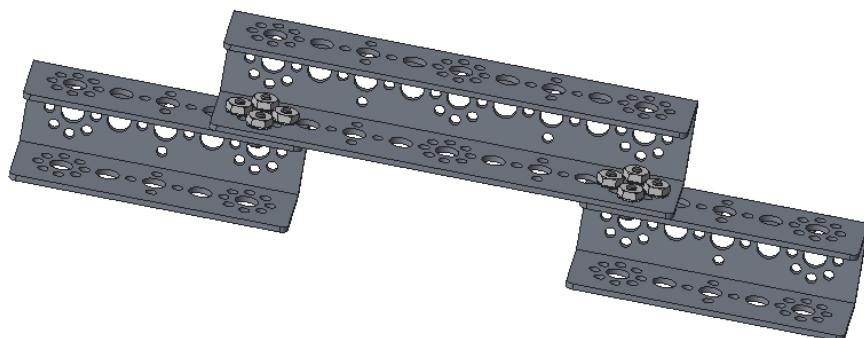


9: Detail View of Step 2 of Creating the Chassis

Step 3: Cross Bar Assembly. Parts: 160 mm channel (1), 96 mm channels (2), 5/16" socket head cap screws (8), keps nuts (8).

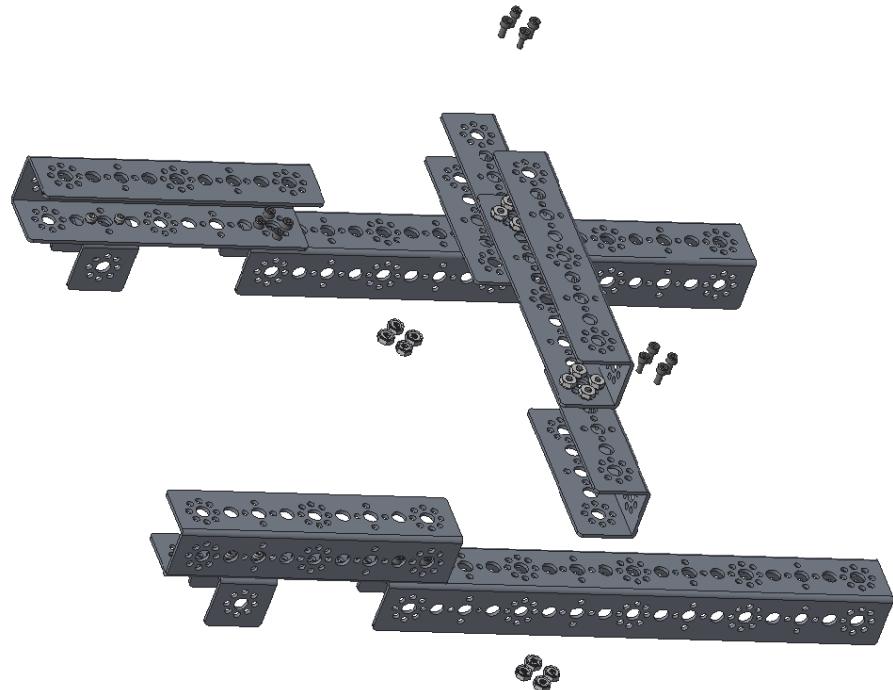


10: Parts needed in Step 3 of Creating the Chassis

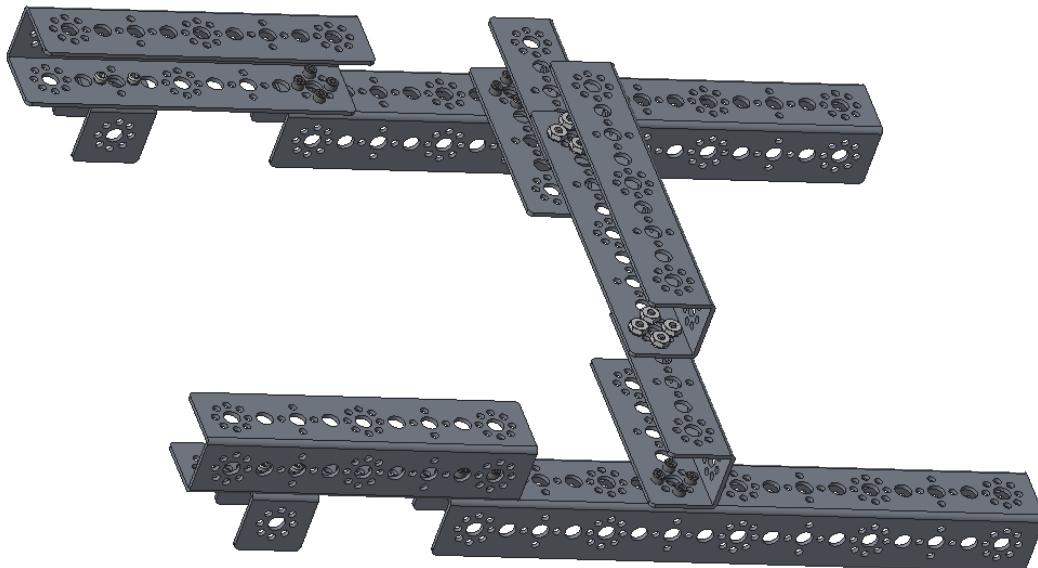


11: Completed Step 3 of Creating the Chassis [Note that there are two copies]

Step 4: right rail assembly, left rail assembly, cross rail assembly, 5/16" socket head cap screws (8), keps nuts (8). In the image, the front of the robot is on the right. The left rail assembly is on the top and the right rail assembly is on the bottom.

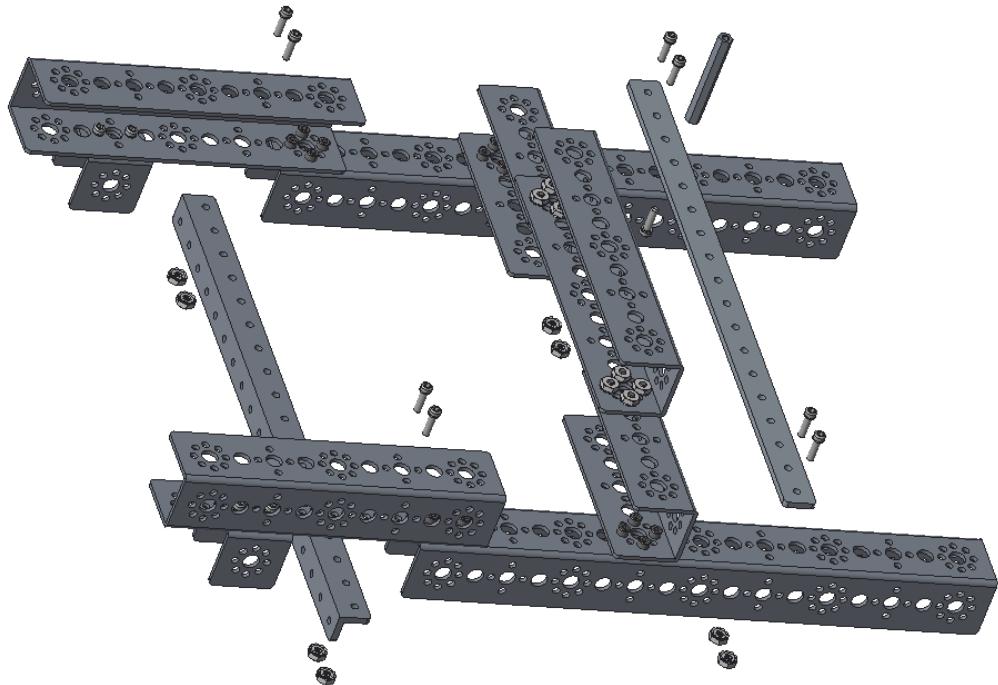


12: Parts needed in Step 4 of Creating the Chassis

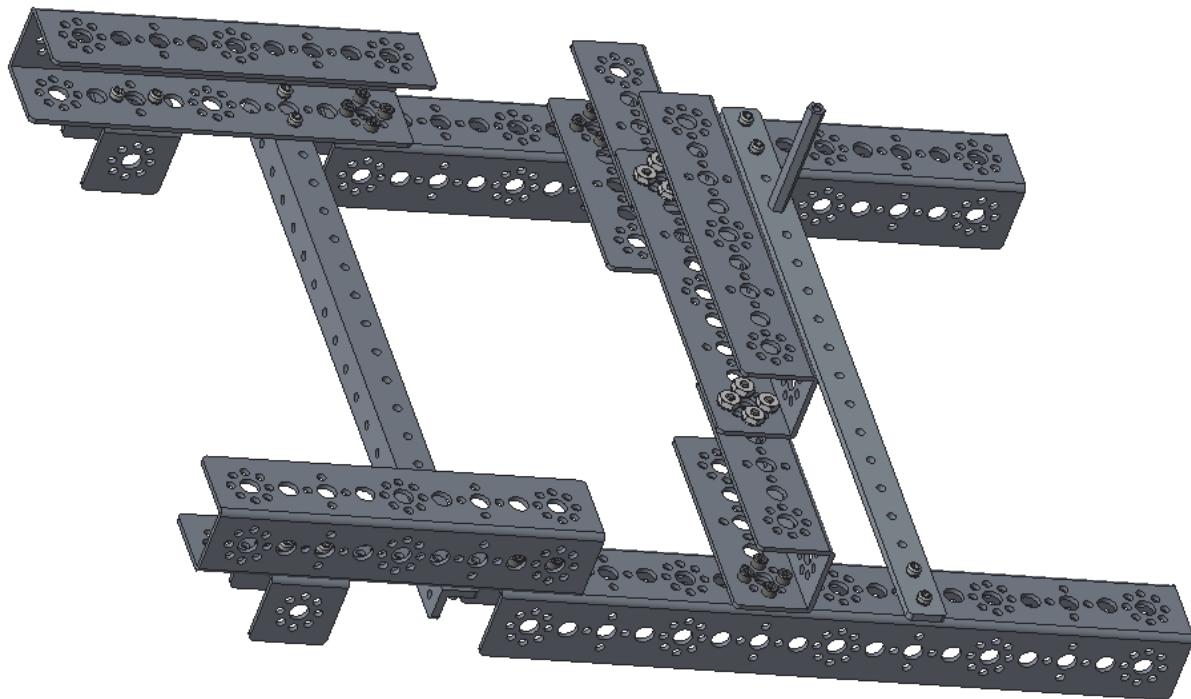


13: Completed Step 4 of Creating the Chassis

Step 5: chassis assembly from Step 4, 288 mm flat bar (1), 288 mm angle (1), 1/2" socket head cap screws (9), keps nuts (8), 2" stand-off (1).

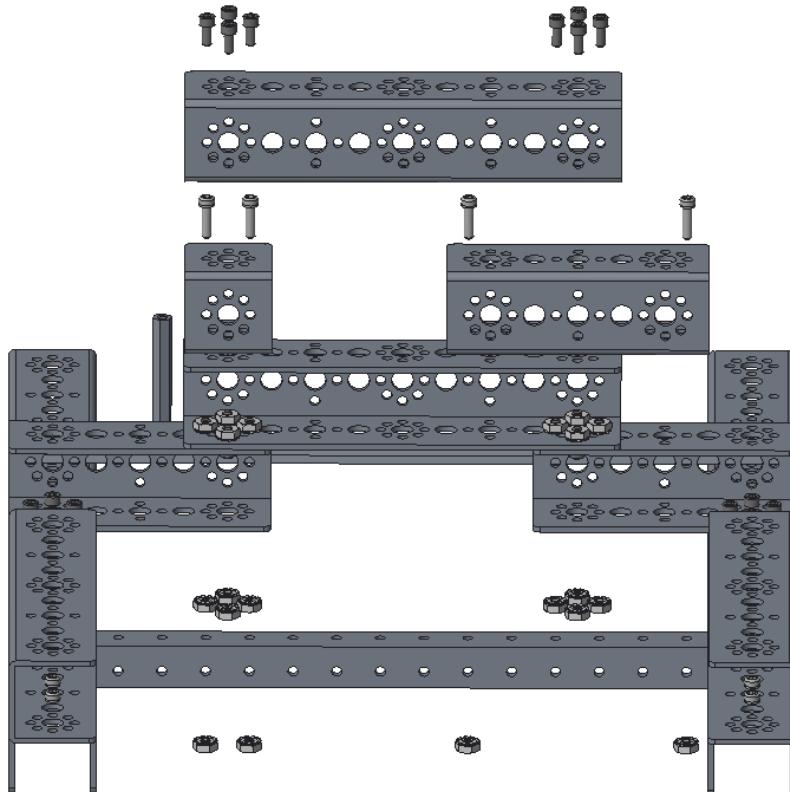


14: Parts needed in Step 5 of Creating the Chassis

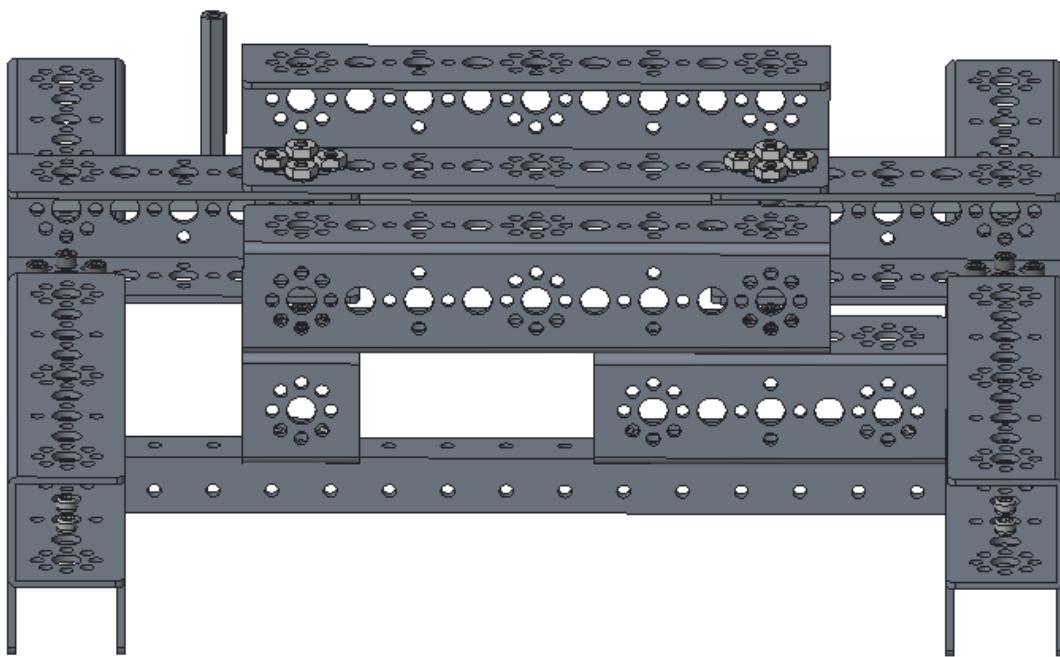


15: Completed Step 5 of Creating the Chassis

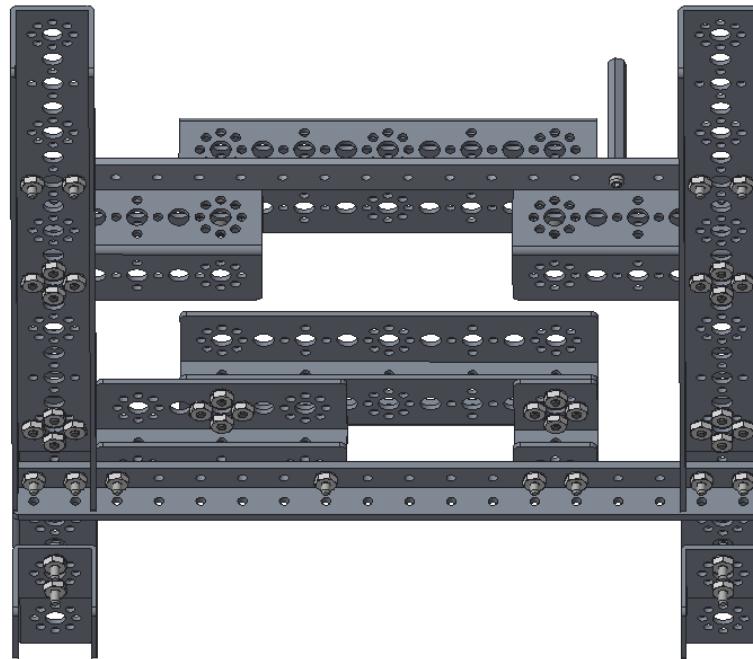
Step 6: chassis assembly, Parts: 32 mm channel (1), 96 mm channel (1), 160 mm channel (1), 1/2" socket head cap screws (4), 5/16" socket head cap screws (8), keps nuts (12).



16: Parts needed in Step 6 of Creating the Chassis Front View



17: Completed Step 6 of Creating the Chassis Front View

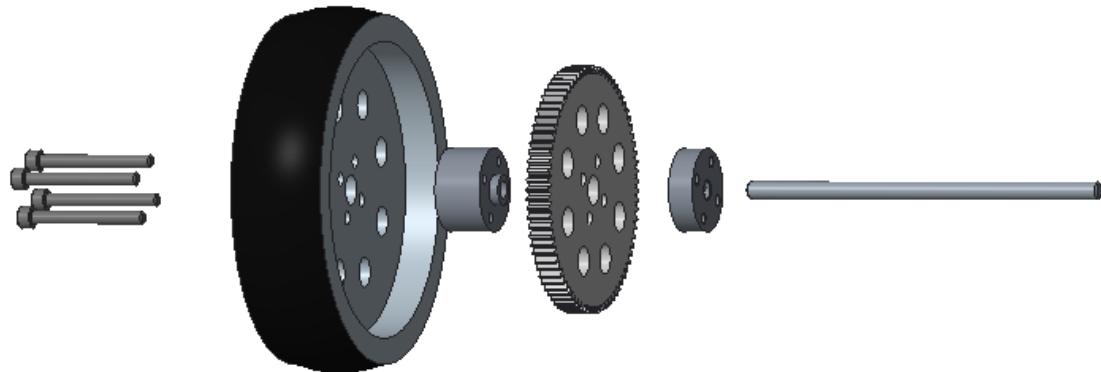


18: Completed Step 6 of Creating the Chassis Underneath Rear View

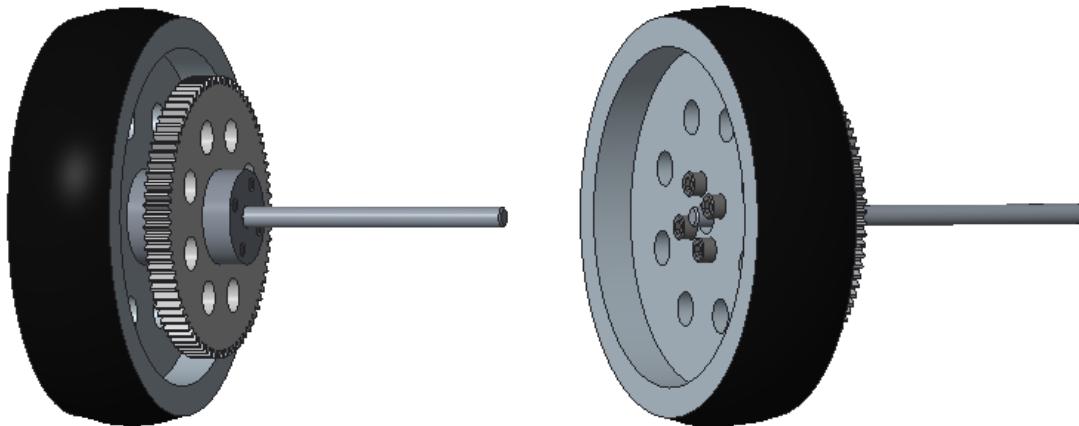
Adding Drive Wheels

[Make 2 copies]

Step 1: Parts: 4" wheel (1), gear hub spacer (1) with included screws (4), axle hub (1), and 100 mm axle (1). The order is wheel, gear hub spacer, gear and then axle hub.

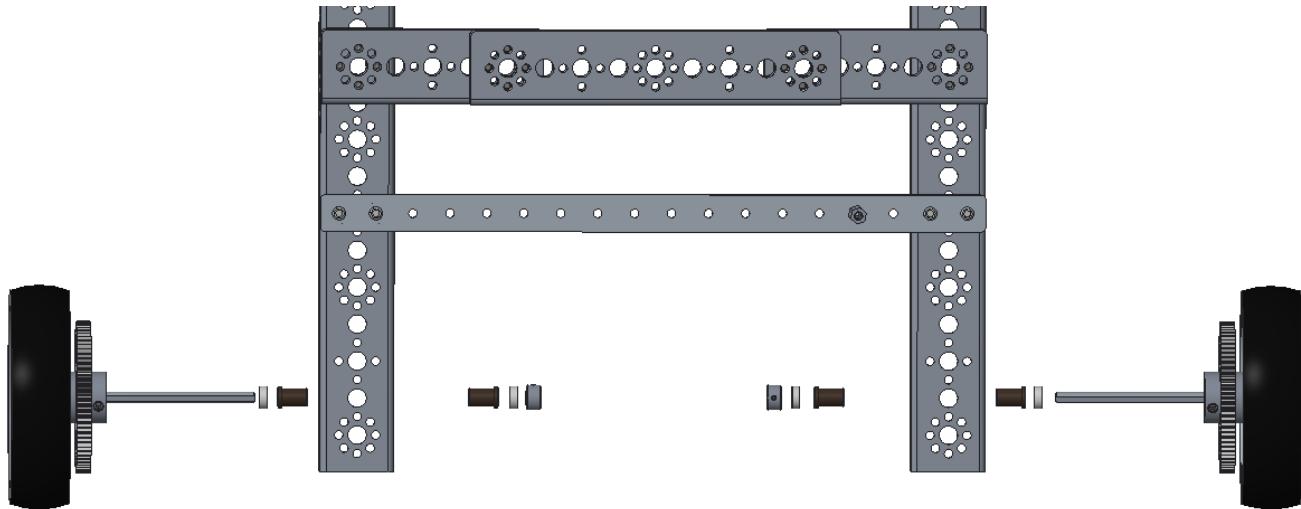


19: Parts needed in Step 1 of Adding Drive Wheels

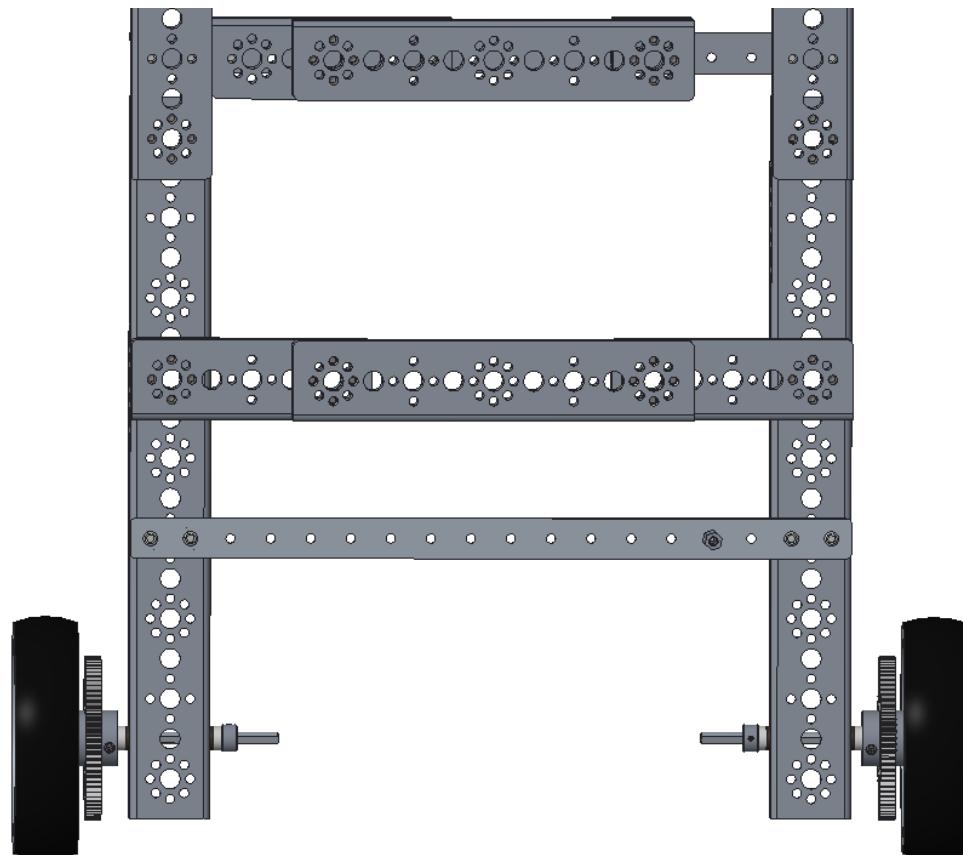


20: Completed Step 1 of Adding Drive Wheels

Step 2: drive wheel assemblies, these are what will power the robot. Parts: 1/8" axle spacers (2), axle collars (1), 11 mm bronze bushings (2). Note: assembly order is drive wheel assembly, 1/8" spacer, bronze bushing, channel, bronze bushing, 1/8" spacer, and axle collar. Note: the drive wheels are on the FRONT of the robot.



21: Parts needed in Step 2 of Adding Drive Wheels

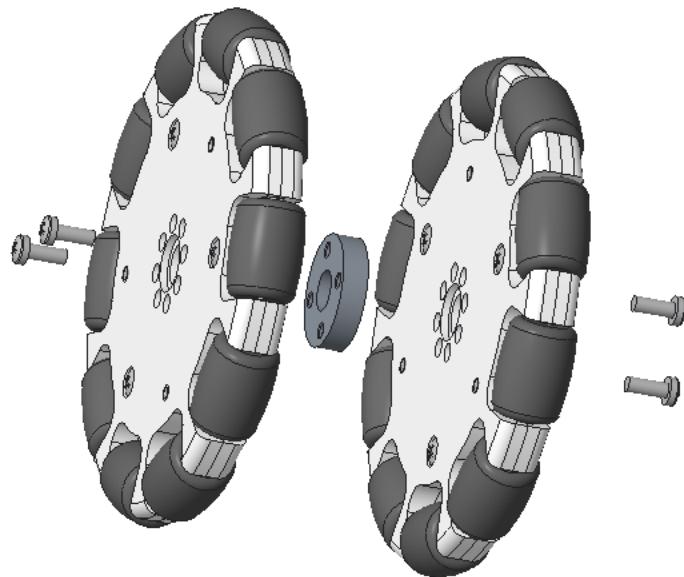


22: Completed Step 2 of Adding Drive Wheels

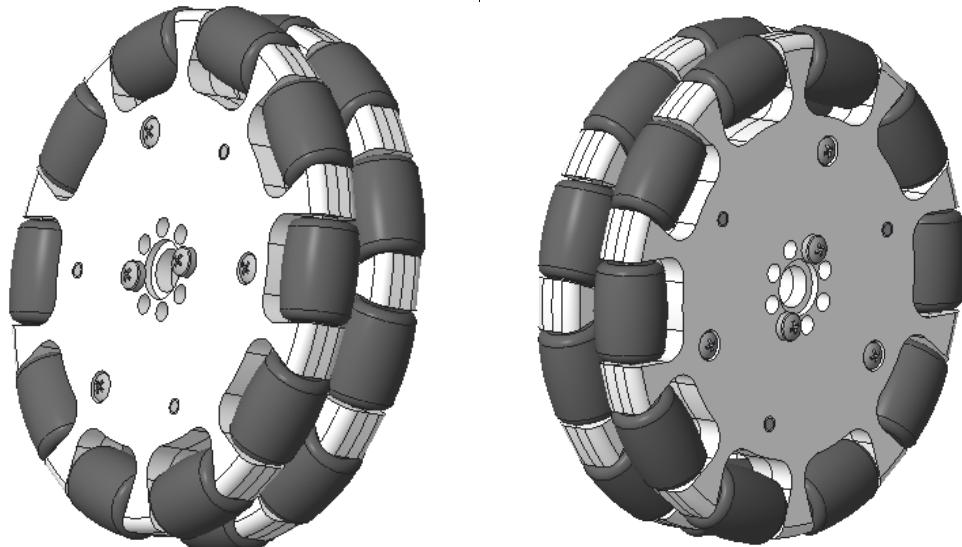
Adding Omni Wheels

[Make 2 copies]

Step 1: The omni wheels on the back of the robot make it easier for the robot to turn. They can slide sideways with very little friction due to the rollers. Parts: 4" omni wheel half (2), joining ring (1) with included screws (4). The order is omni wheel, joining ring, omni wheel. These are assembled according to the instructions that come in the omni wheel pack. Note the screws on one wheel are in opposite holes from the screws on the other wheel.



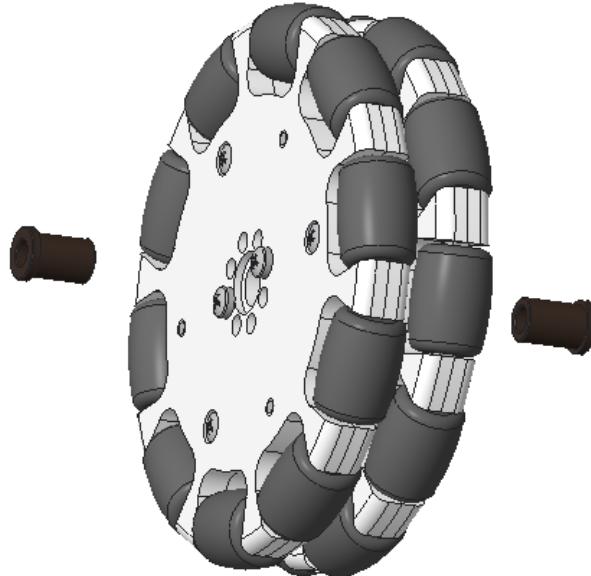
23: Parts needed in Step 1 of Adding Omni Wheels



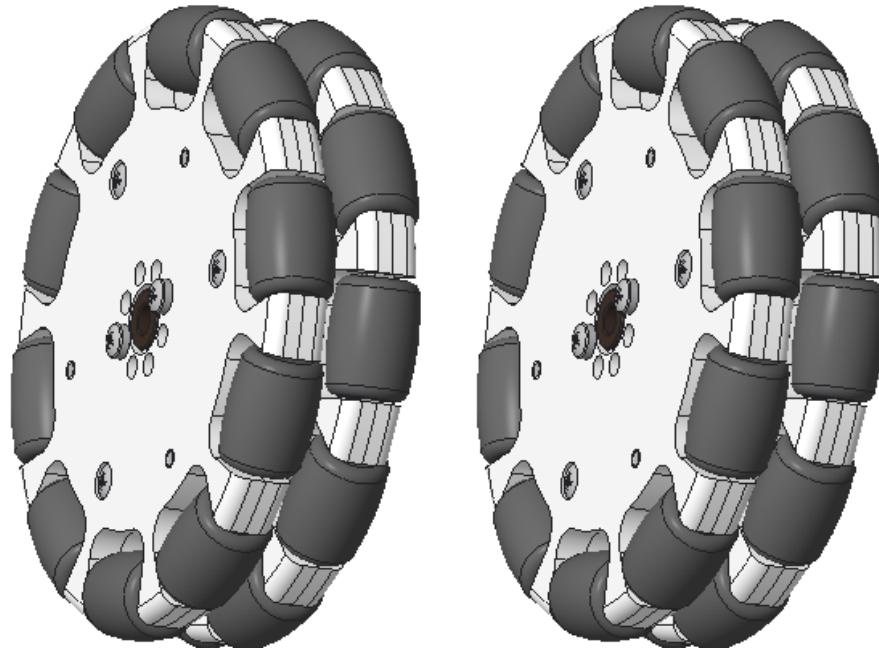
24: Completed Step 1 of Adding Omni Wheels

Step 2: Parts: assemblies from the previous step and 11 mm bronze bushings (4). Sometimes the bronze bushings are hard to insert. To make this easier, put a bronze bushing on a wheel assembly from the Drive Wheel Assembly section. Slide the omni wheel down the axle and press it against the wheel assembly. A video is available that shows this trick

(<https://www.facebook.com/322705934572847/videos/506108019565970/>).

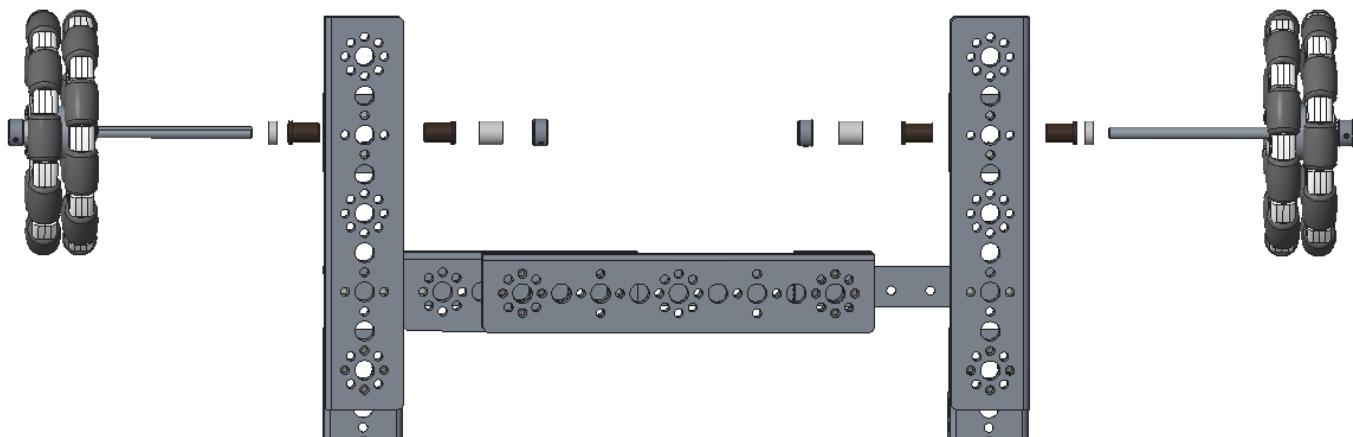


25: Parts needed in Step 2 of Adding Omni Wheels

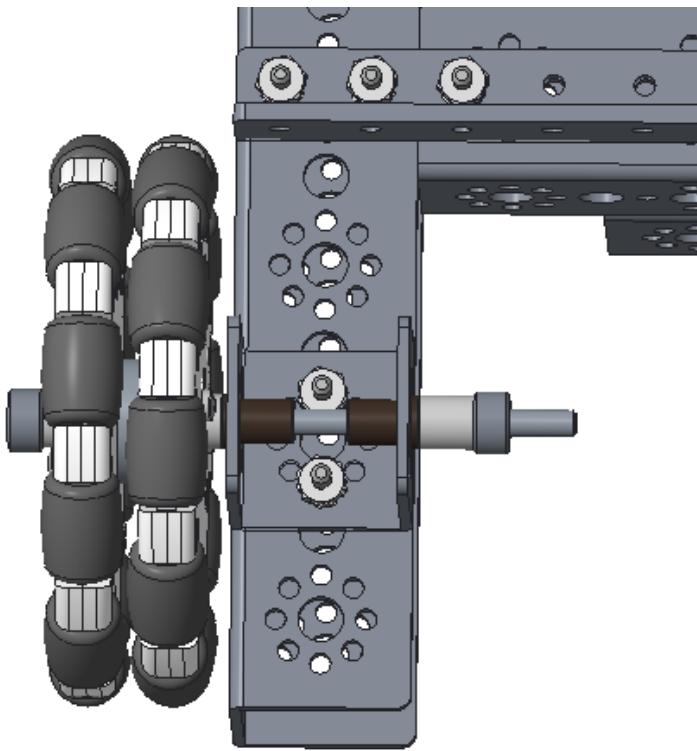


26: Completed Step 2 of Adding Omni Wheels

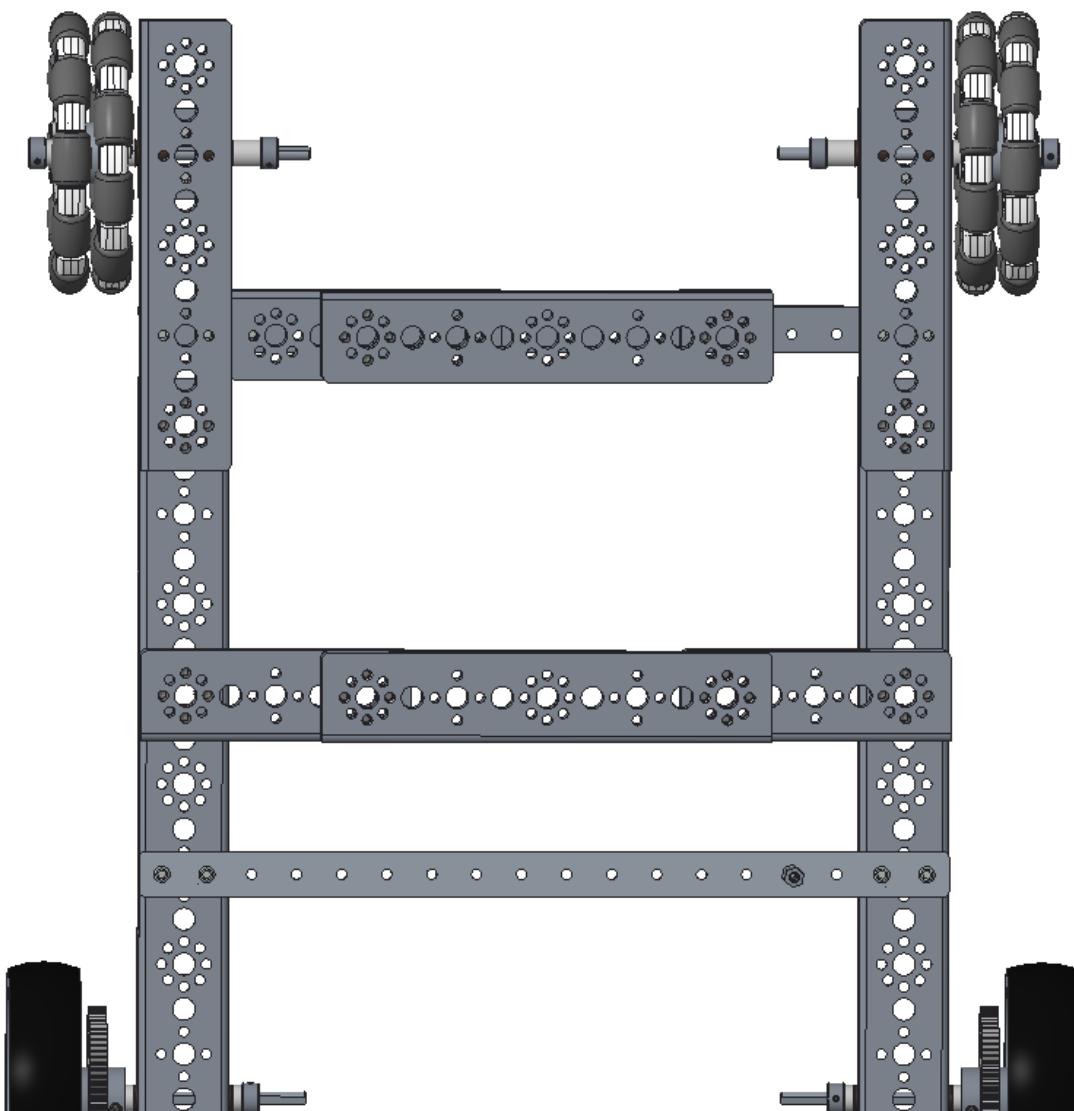
Step 3: Adding omni wheels to chassis. Parts: chassis, omni wheel assemblies, 1/8" axle spacers (4), 3/8" axle spacers (2), axle collars (4), 11mm bronze bushings (4), 100 mm axles (2). Note: assembly order is axle collar, 1/8" axle spacer, omni wheel assembly, 1/8" axle spacer, 11mm bronze bushing, channel, 11mm bronze bushing, 3/8" spacer, and axle collar. The bottom image shows the step completed for one wheel. The image on the following page shows both wheels. Note: The omni wheels are mounted to the back of the robot.



27: Parts needed in Step 3 of Adding Omni Wheels top view of back of robot.



28: Completed Step 3 of Adding Omni Wheels [close up] bottomview



29: Completed Step 3 of Adding Omni Wheels top view

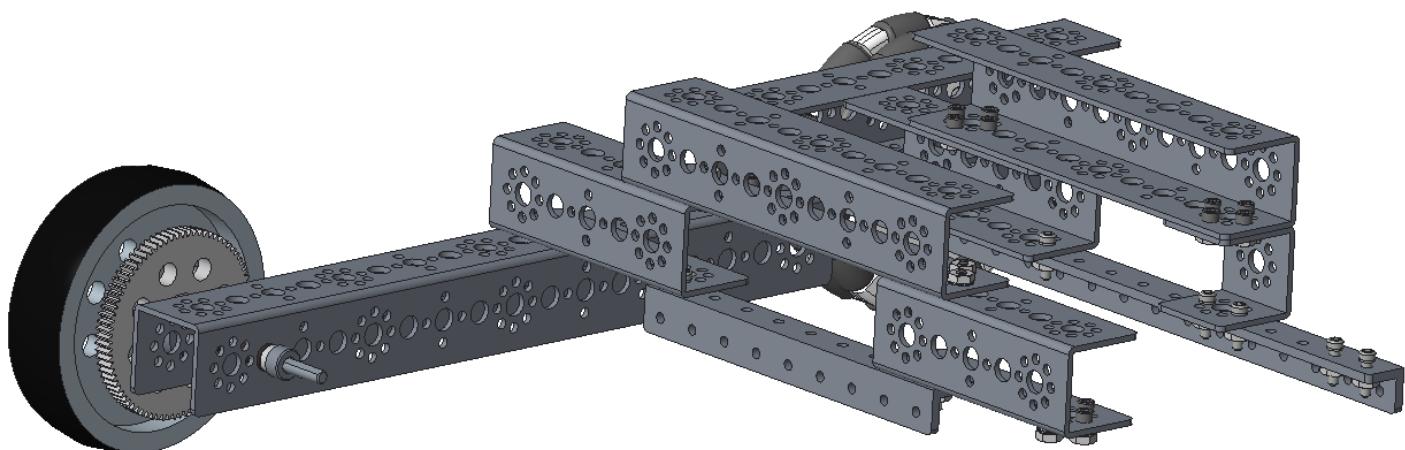
Adding the Arm Motor

The arm motor will be used to raise and lower the arm. An additional motor is included in the kit of parts. It can be used to power another mechanism on the PushBot such as a sweeper for balls and blocks.

Step 1: Parts: chassis, 144 mm angle, 1/2" socket head cap screws (2), keps nuts (2).

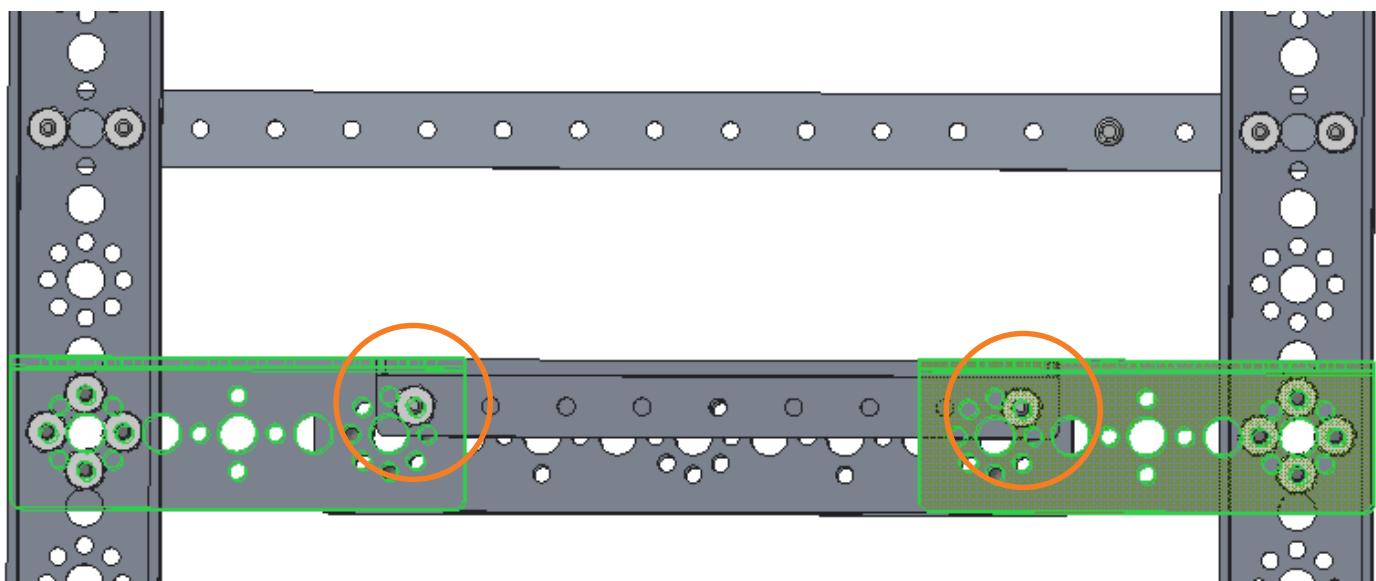


30: Parts needed in Step 1 of Adding Arm Motor



31: Completed Step 1 of Adding Arm Motor

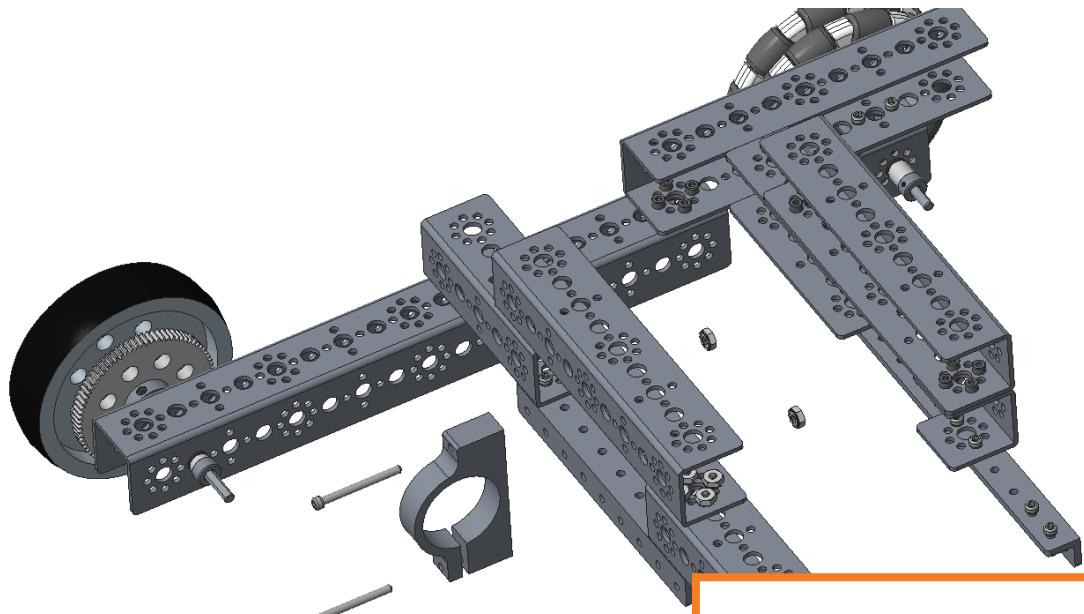
Also note that the angle is not centered. It mounts closer to the left chassis rail than the right chassis rail.



Note that the bolts are through the hole that is one down and to the right of the top hole. This allows the angle to be flush with the front face of the channel.

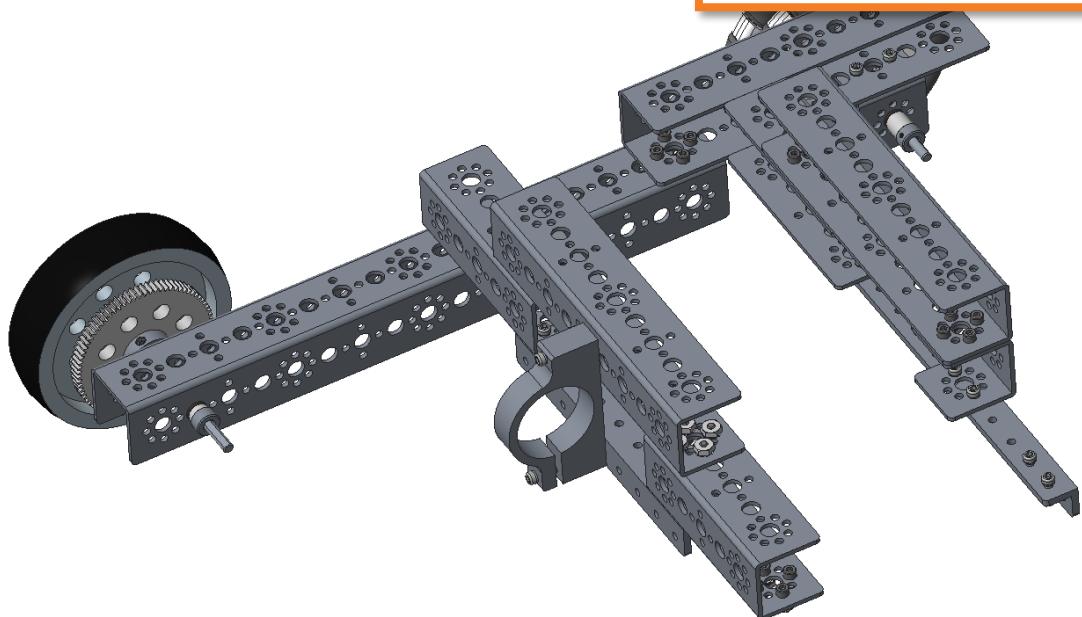
32: Completed Step 1 of Adding Arm Motor Bottom View

Step 2: Parts: chassis, motor mount (and the two bolts and nuts that come with it). Note: Do not tighten lower motor mount bolt, the side with the gap, until the motor has been inserted (see step 4). When this bolt is tightened, it clamps the motor in place.



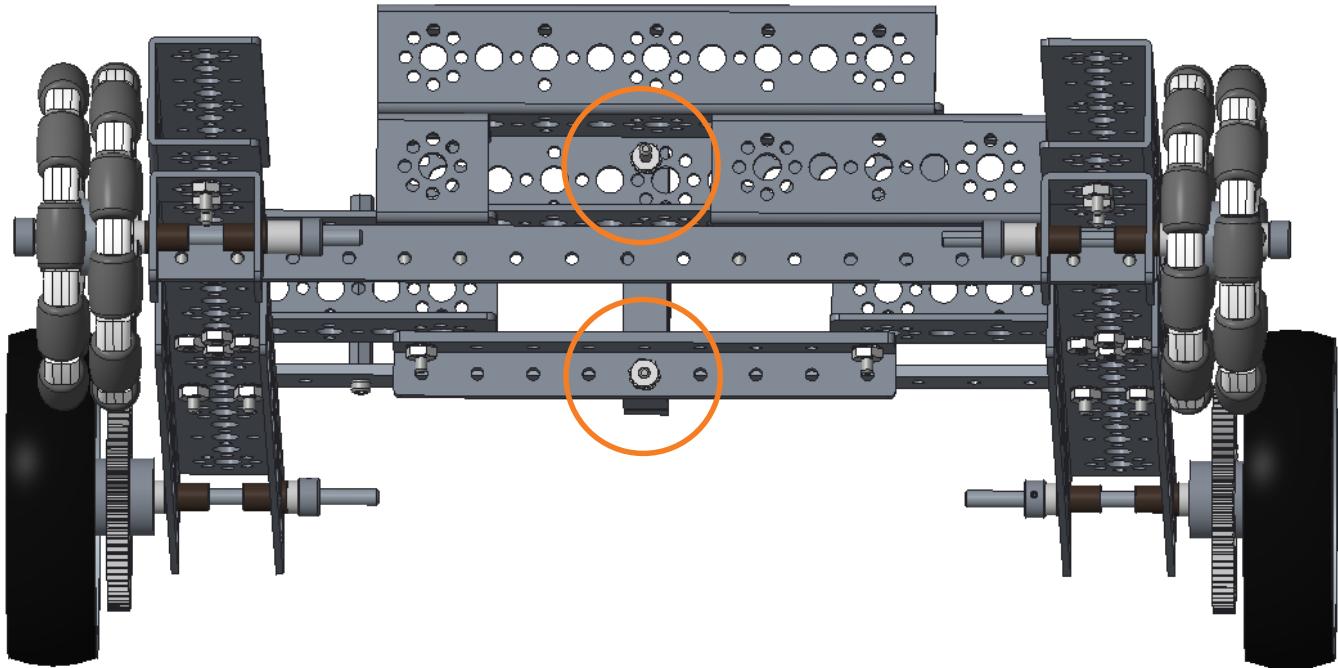
33: Parts needed in Step 2 of Adding Arm Motor

Note that parts of the chassis (the left chassis rail and the front 288mm bar) have been hidden in these images.



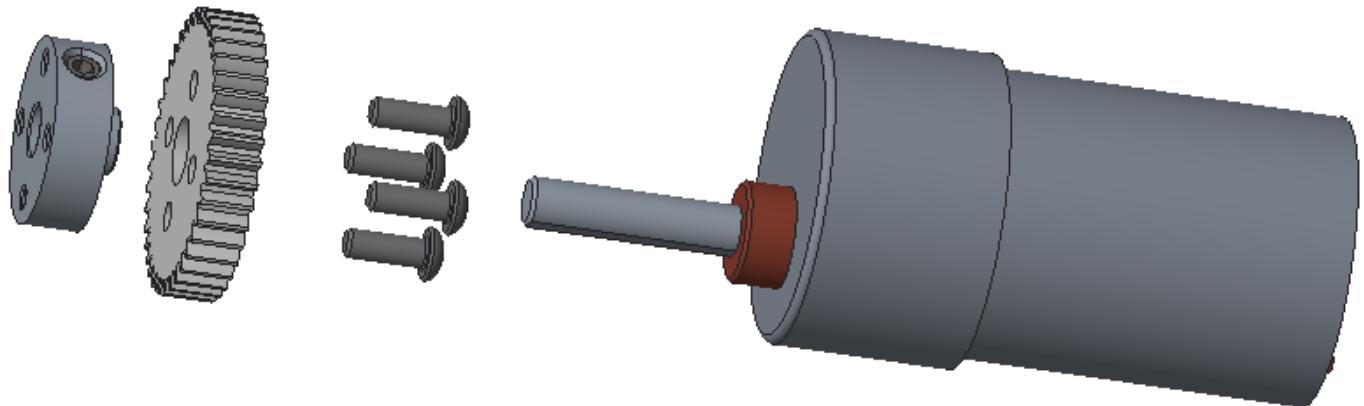
34: Completed Step 2 of Adding Arm Motor

This is a view from the “inside of the robot” (underneath the back wheels looking towards the front of the robot). Note that the motor mount is not in the top hole. From the inside it will be down and to the left. From the front of the robot it will be down and to the right.

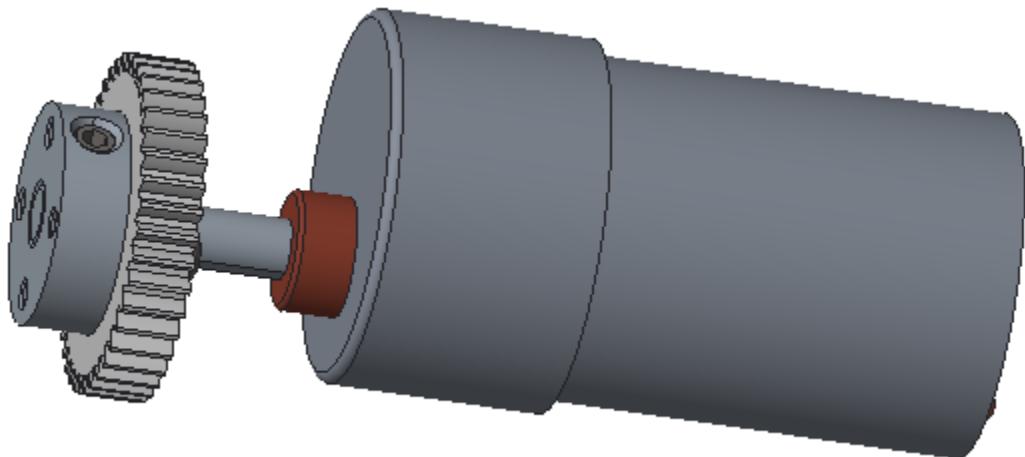


35: Completed Step 2 of Adding Arm Motor Rear View

Step 3: Adding motor hub. Parts: motor hub (with its included set screw), 40 tooth gear, DC gear motor (1), button head cap screws (4).

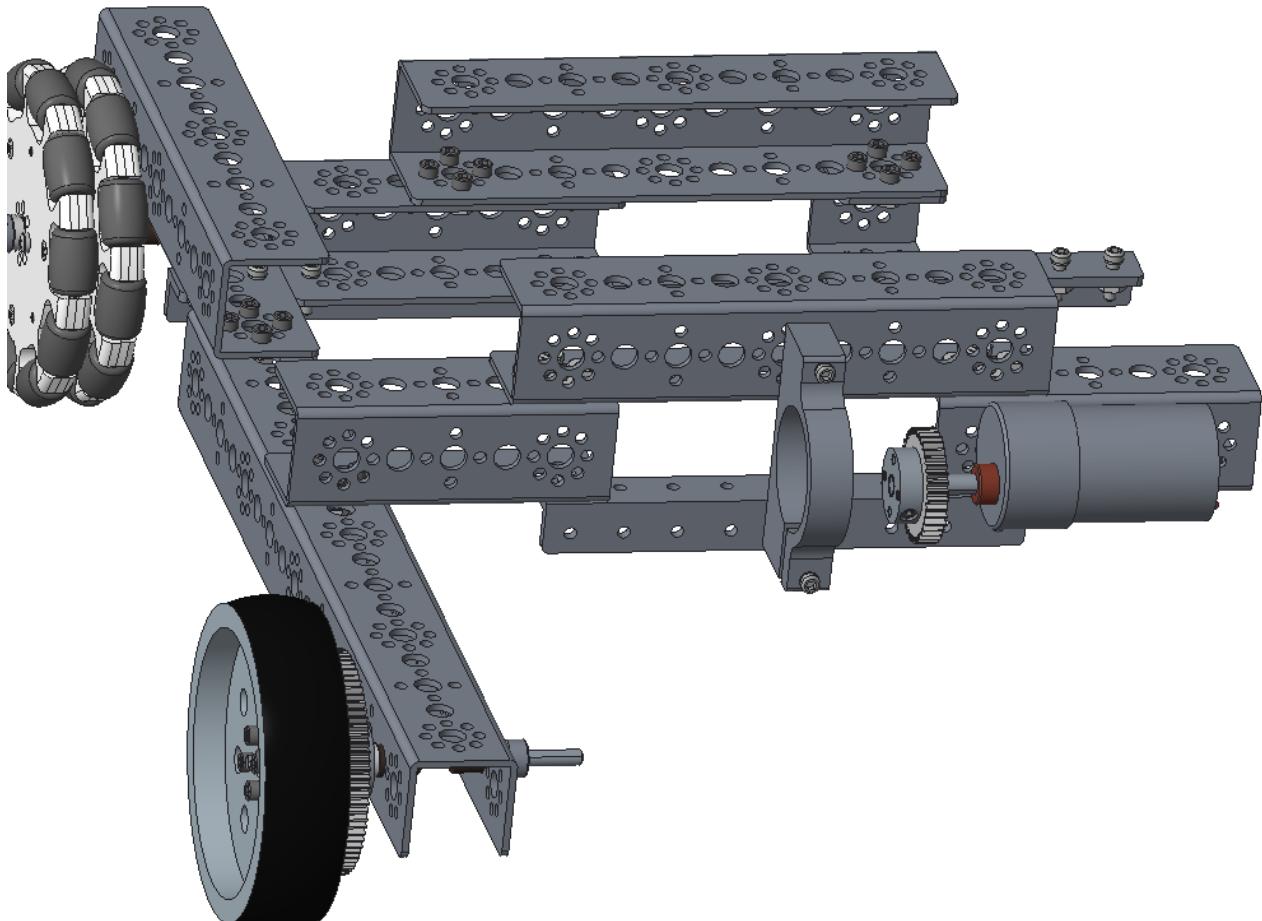


36: Parts needed in Step 3 of Adding Arm Motor

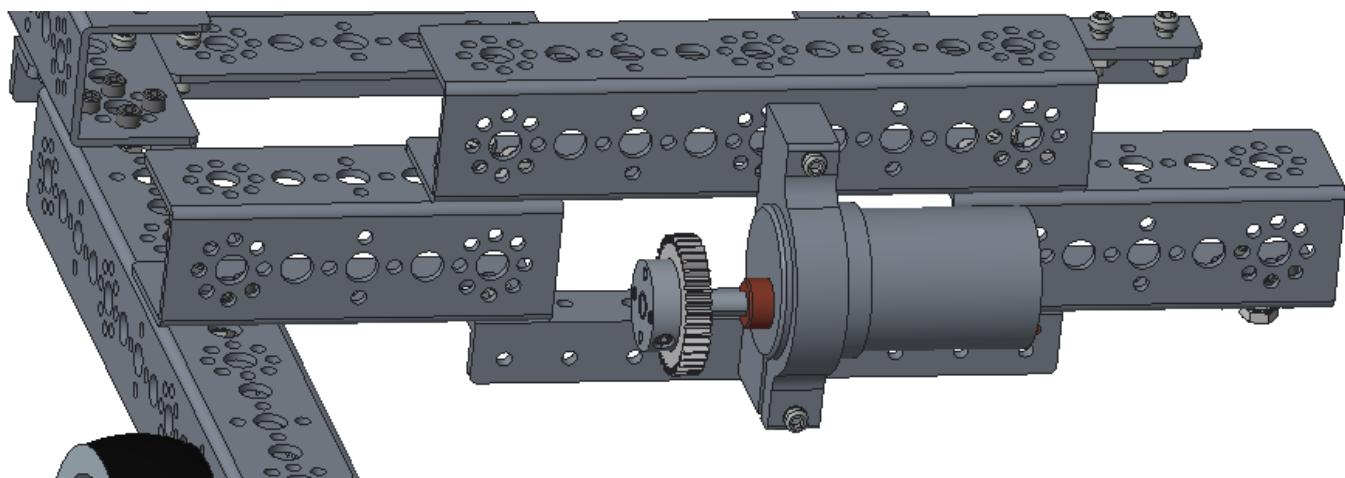


37: Completed Step 3 of Adding Arm Motor

Step 4: Parts: arm motor assembly from the previous step, chassis.



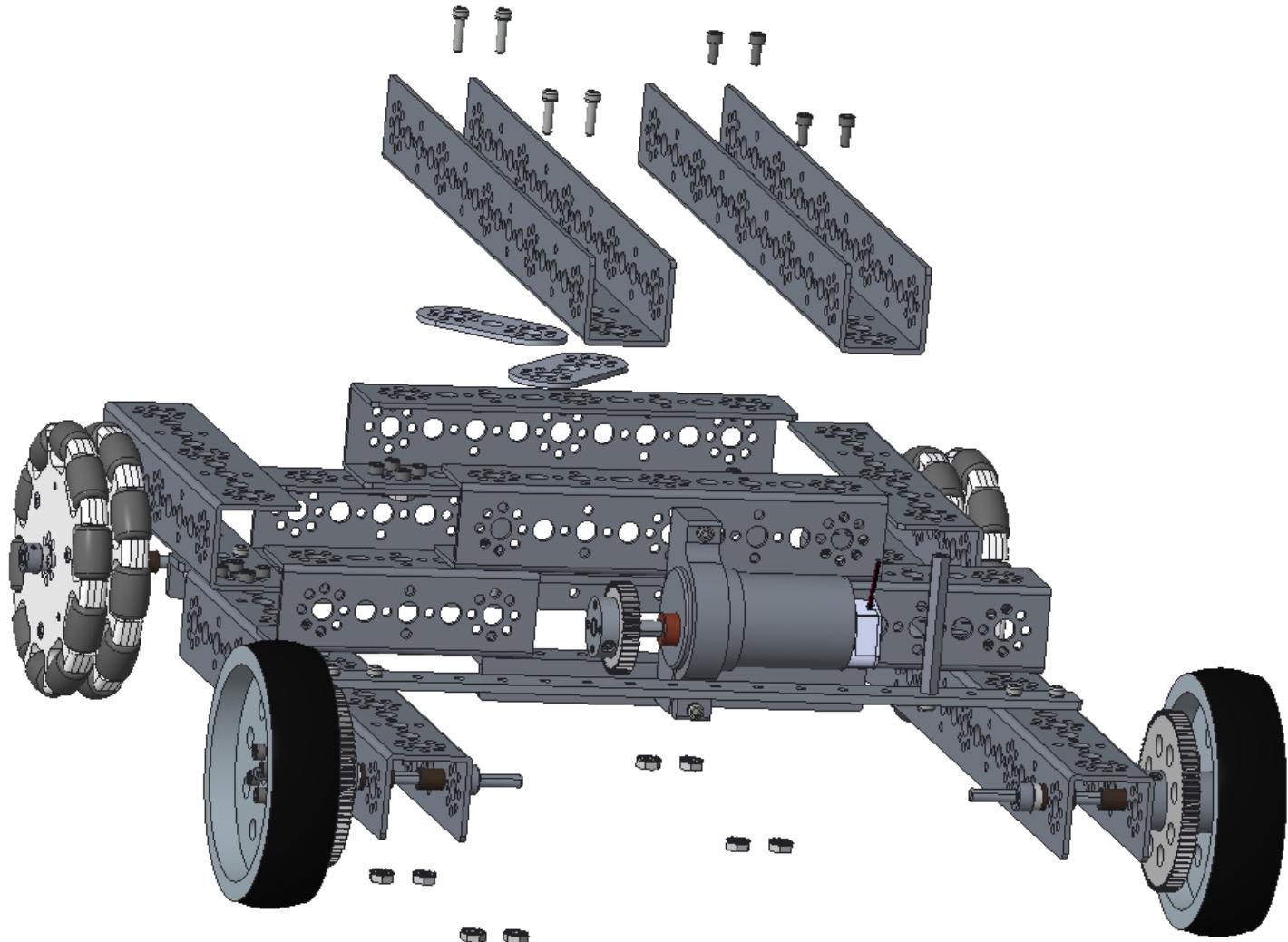
38: Parts needed in Step 4 of Adding Arm Motor



39: Completed Step 4 of Adding Arm Motor

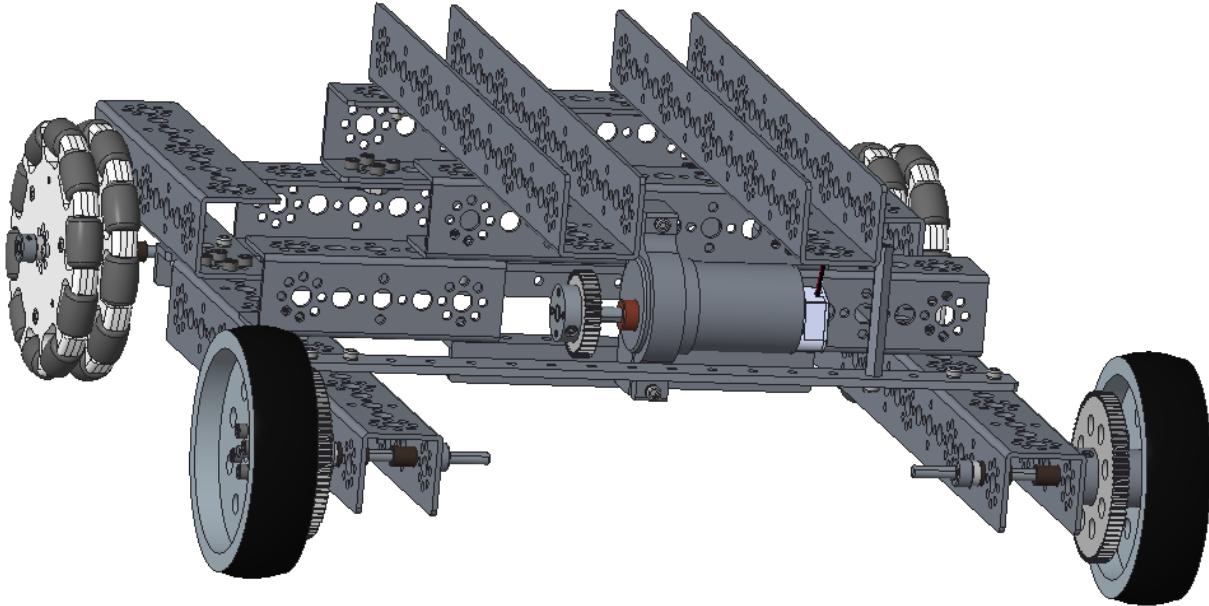
Adding the Arm Rails

Step 1: Parts: robot chassis, 288 mm channels (2), 1/2" socket head cap screws (4), 5/16" socket head cap screws (4), 64 mm flats (2), keps nuts (8).



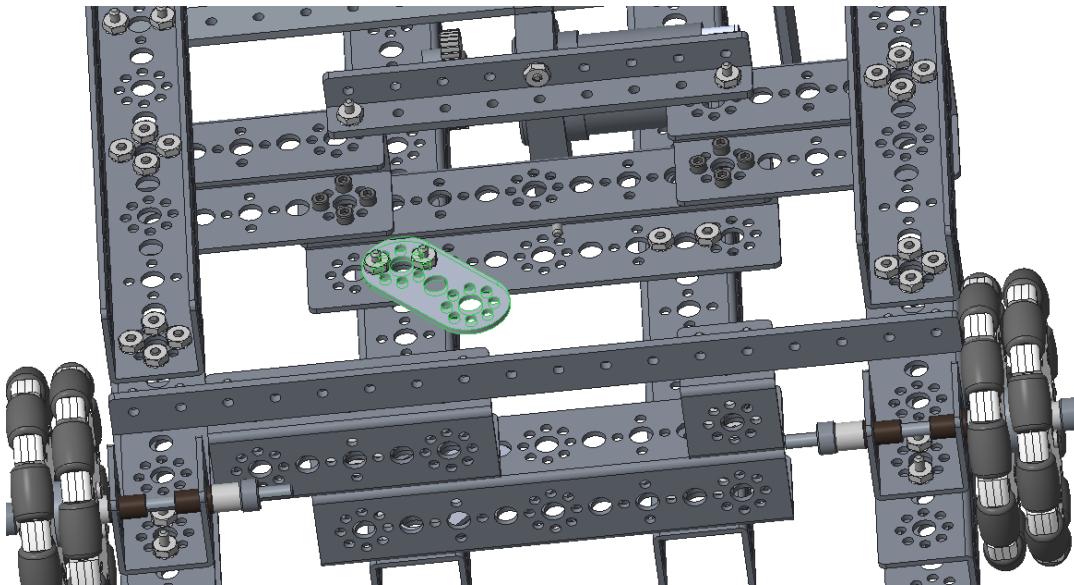
40: Parts needed in Step 1 of Adding the Arm Rails

This image shows the completed step from the front of the robot.



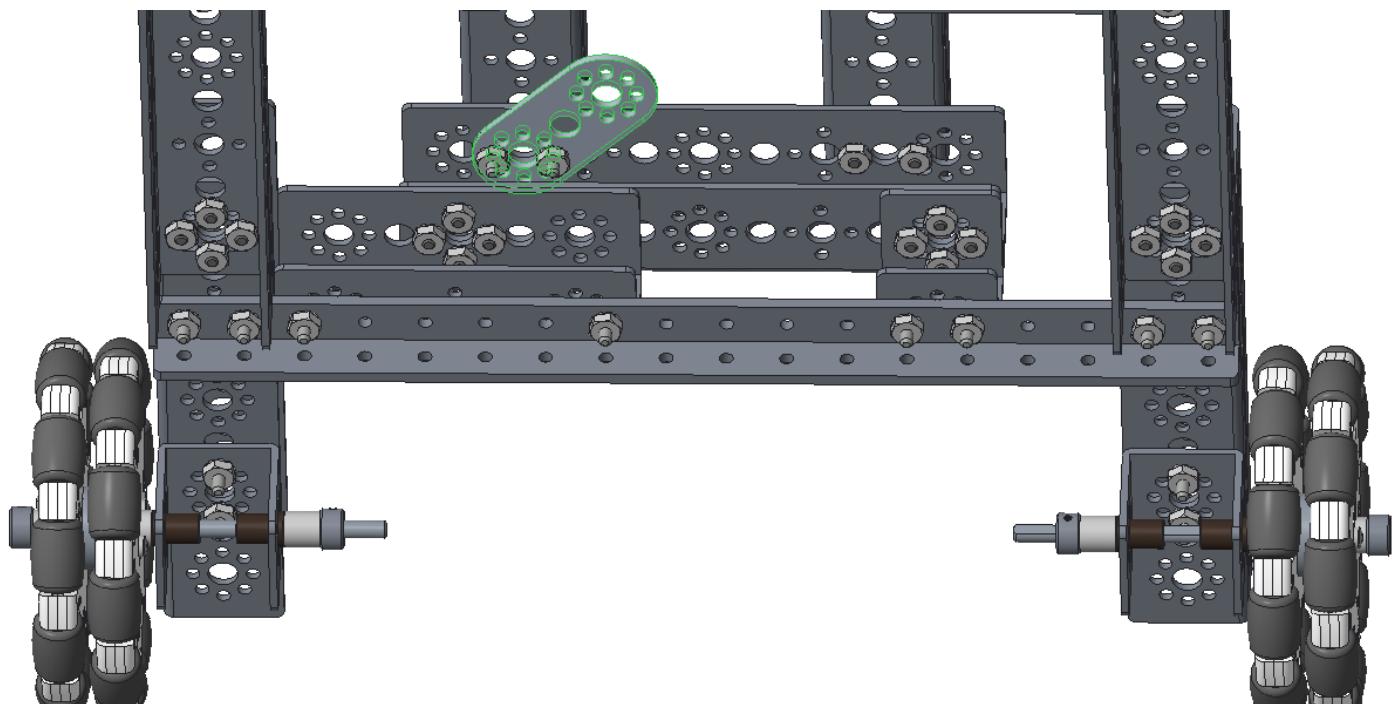
41: Completed Step 1 of Adding the Arm Rails

The image below shows the mounting of the 64 mm flat on the front cross bar. The omni-wheels are on the back of the robot and the traction wheels are on the front.



42: Completed Step 1 of Adding the Arm Rails Detail of front 64 mm flat

The image below shows the mounting of the 64 mm flat on the back cross bar.



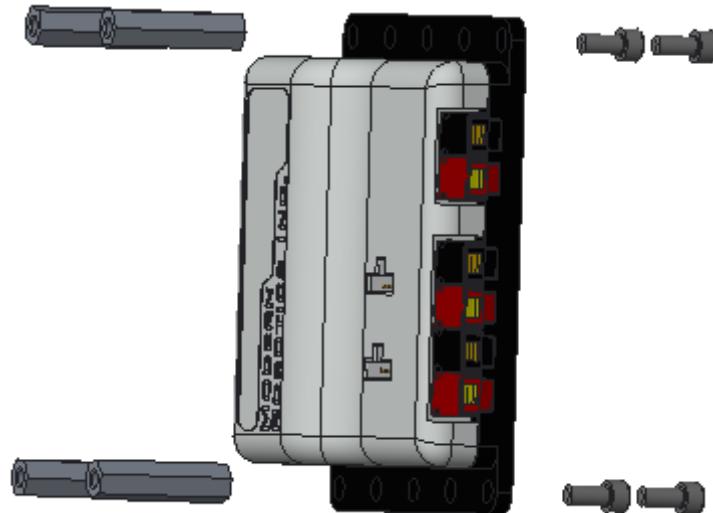
43: Completed Step 1 of Adding the Arm Rails Detail of back 64 mm flat

Adding the Controller Stack Assembly

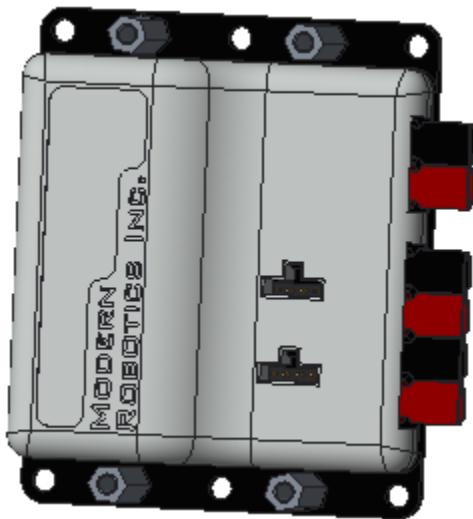
To complete this section, the servo controller and DC motor controllers will be made into an assembly and then that assembly will be added to the robot. The DC motor controllers and the servo controller translate the USB commands from the robot controller phone into voltages to run the DC motors and the servos.

The stand-offs for the top module must be in the correct positions (outside holes) to mount to the robot chassis. The stand-offs for the middle controller are one position ‘in’ from the outside holes. The stand-offs on the bottom controller are on the outside holes.

Step 1: Parts: core motor controller (1), 32 mm stand-offs (4), and 5/16" socket head cap screws (4).



44: Parts needed in Step 1 of Adding the Controller Stack Assembly

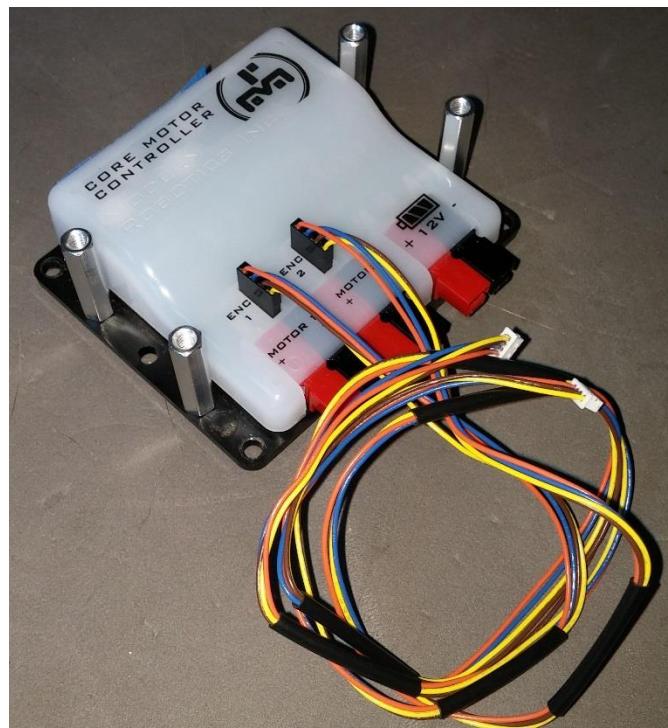


45: Completed Step 1 of Controller Stack Assembly

Step 2: Parts: assembly from previous step (this will become the core motor controller for the drive), motor encoder wires (2). Note: all of the parts for an encoder are shown. Use only the wire for this step. The encoder will measure the rotation of the drive motors to allow more precise control of the motor.

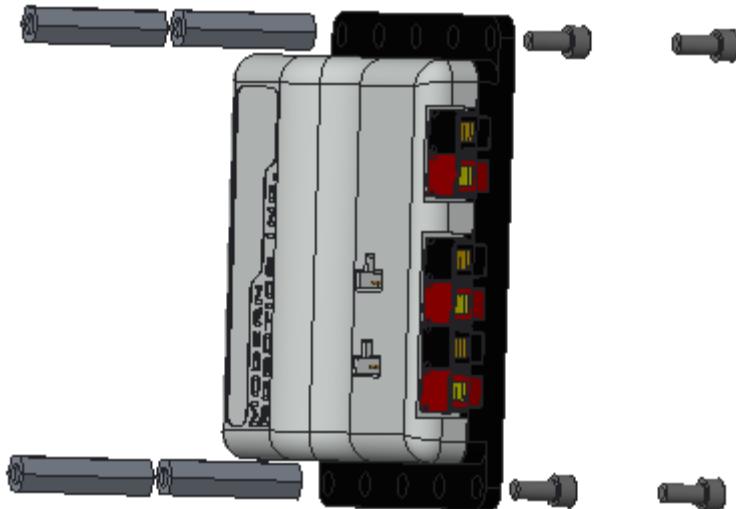


46: Parts needed in Step 2 of Adding the Controller Stack Assembly

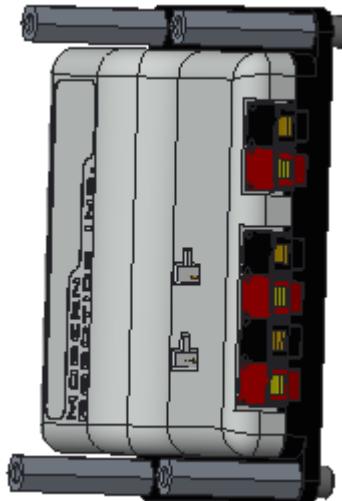


47: Completed Step 2 of Controller Stack Assembly

Step 3: Parts: core motor controller (1) (this will become the motor controller for the arm; it is not from the previous step), 32 mm stand-offs (4), and 5/16" socket head cap screws (4). The stand-offs are located at the corners of the controller.



48: Parts needed in Step 3 of Adding the Controller Stack Assembly



49: Completed Step 3 of Controller Stack Assembly

Step 4: Parts: core motor controller assemblies from the previous two steps, 5/16" socket head cap screws (4). The screws attach the left controller to the right controller using the stand-offs on the lower controller. The left controller has encoder cables (not shown) attached to it. Be careful not to harm the cables.



50: Parts needed in Step 4 of Adding the Controller Stack Assembly

The encoder cable wires are not shown in the above and below images. They are in the left controller's ports.

Make sure that the power ports are facing the same direction.



51: Completed Step 4 of Controller Stack Assembly

Step 5: Parts: core servo controller (1) (this will become the hand controller), 32 mm stand-offs (4), and 5/16" socket head cap screws (4).



52: Parts needed in Step 5 of Adding the Controller Stack Assembly



53: Completed Step 5 of Controller Stack Assembly

Step 6: Parts: core servo assembly from the previous step, double core motor controller assembly from a previous step, 1/2" socket head cap screws (4)



54: Parts needed in Step 6 of Adding the Controller Stack Assembly

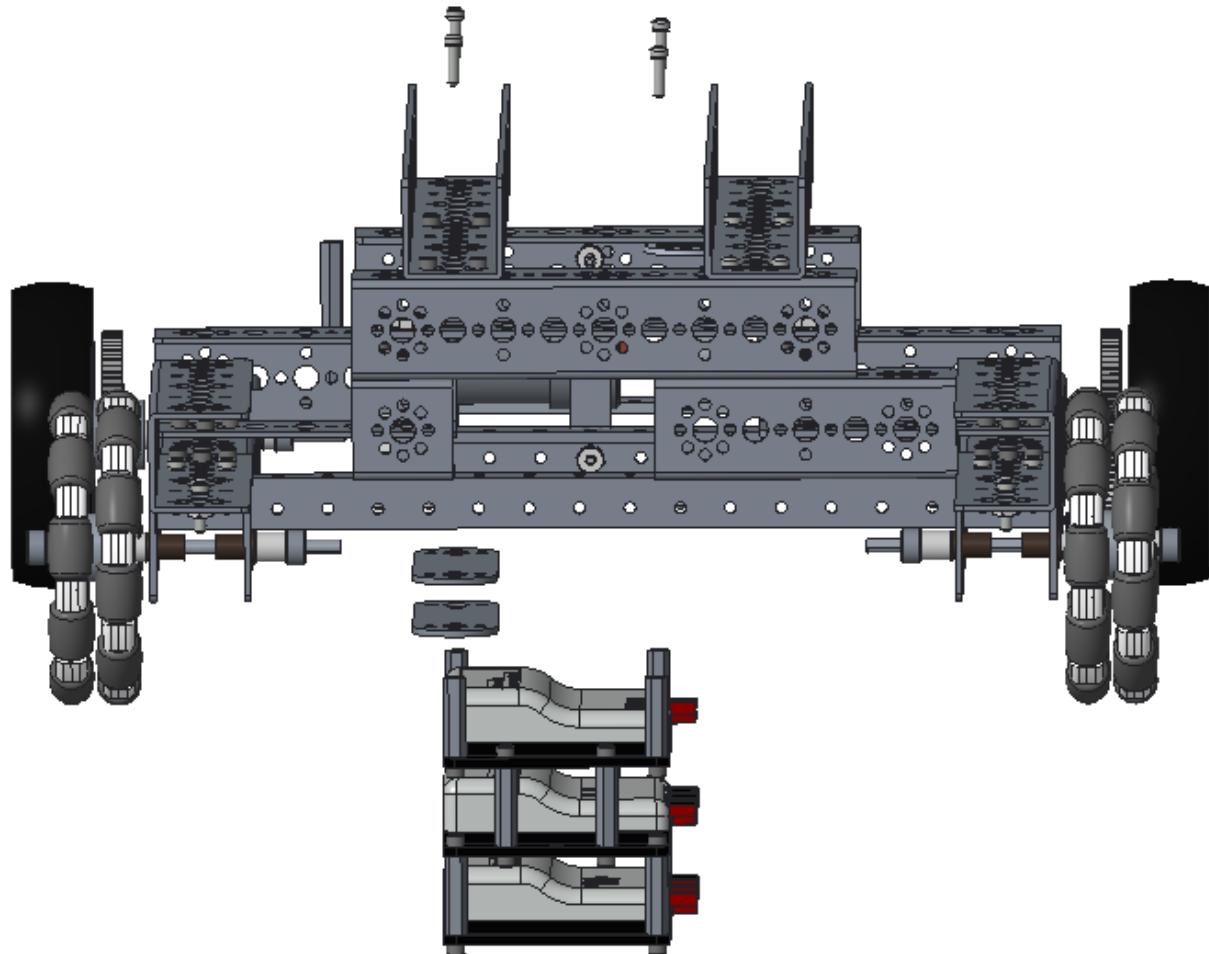
The core servo controller is on the left in the above and below images.

The encoder cable wires (not shown in the above and below images) are in the center controller's ports.

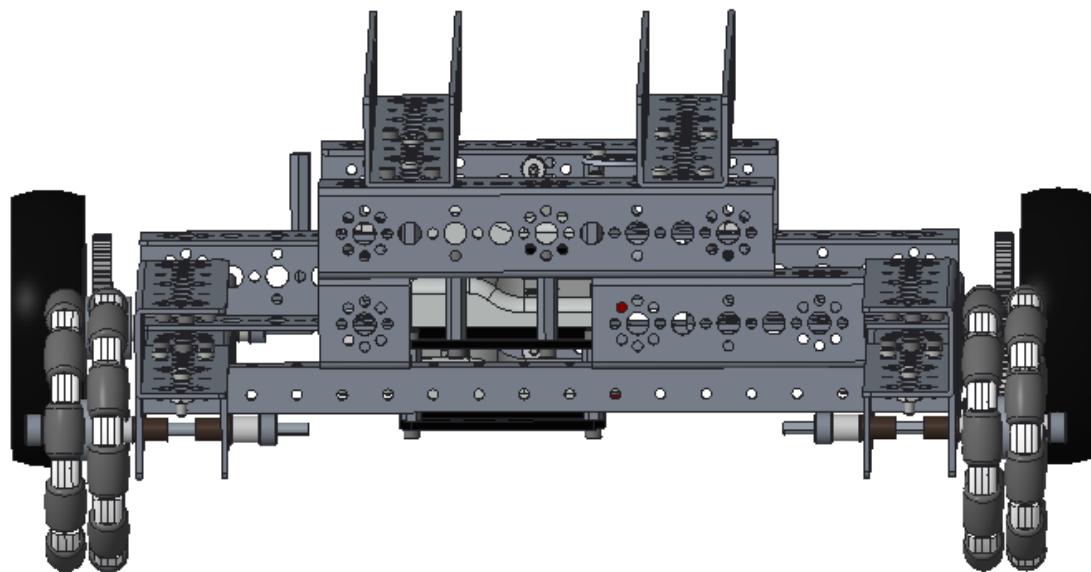


55: Completed Step 6 of Controller Stack Assembly

Step 7: Parts: chassis, controller stack assembly from the previous step, flat brackets (2), 1/2" socket head cap screws (4)

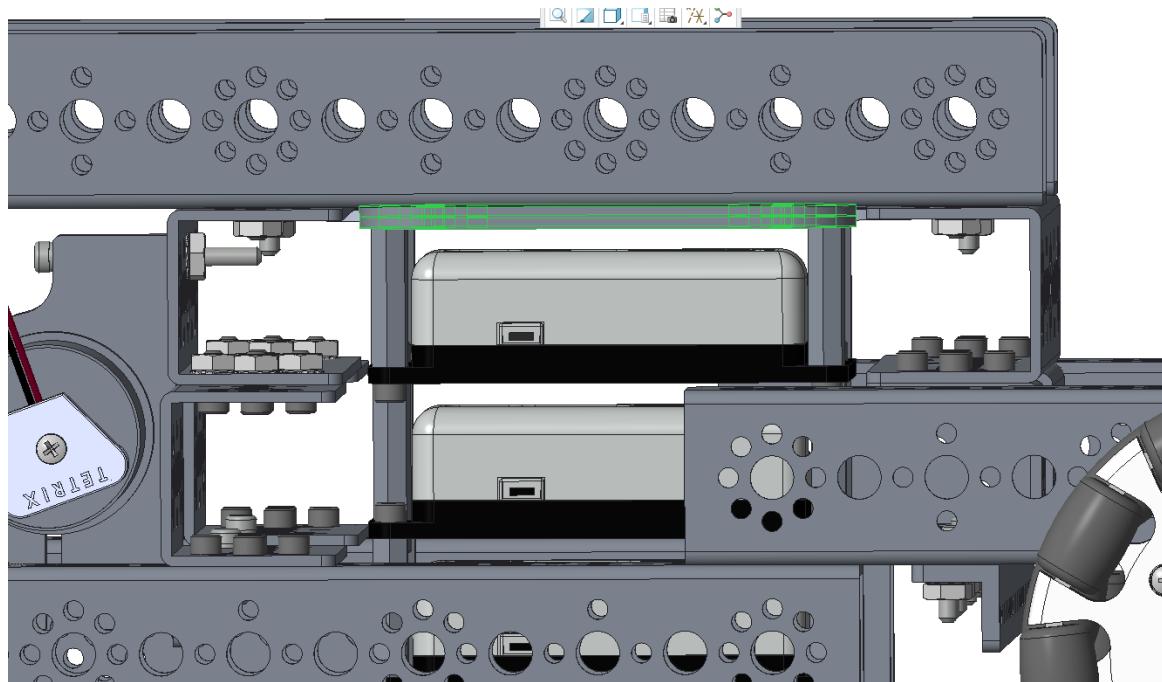


56: Parts needed in Step 7 of Adding the Controller Stack Assembly



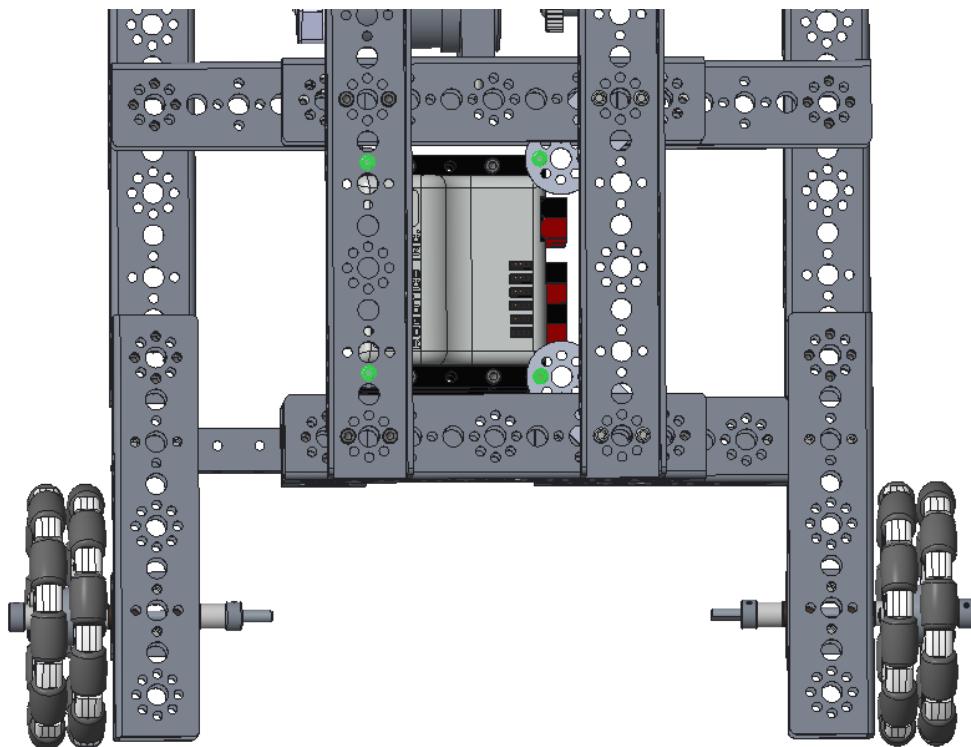
57: Completed Step 7 of Controller Stack Assembly

The two flat brackets are on the left side in the image above and the center of the image below. The image below is from the left side of the robot. They are between the channel and the stand-offs. They offset the height introduced by the mounting of the other side of the controller stack.



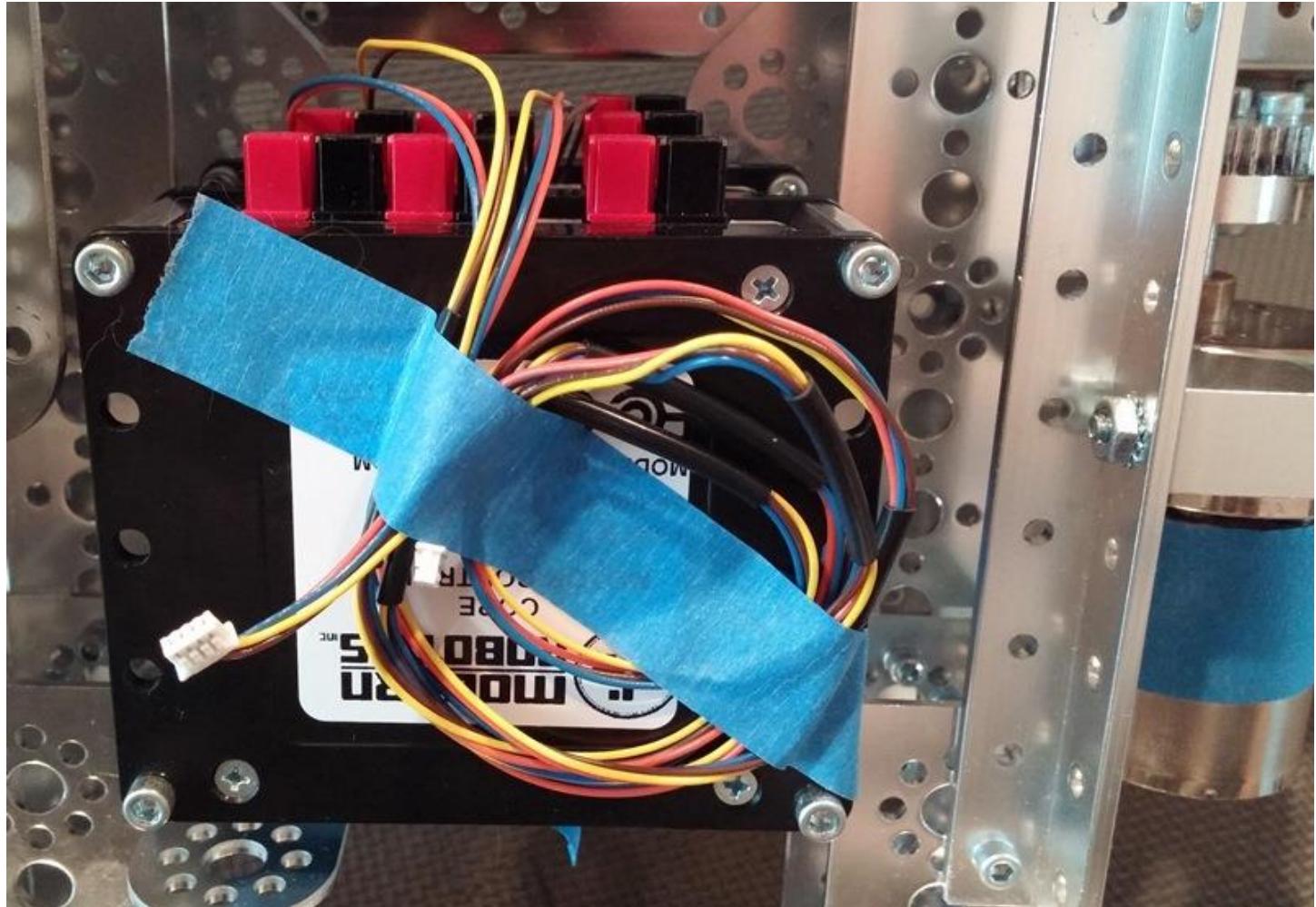
58: Completed Step 7 of Controller Stack Assembly Detail 1 showing the two flat brackets

The two flat brackets are on the left side in the image above and below. They are between the channel and the stand-offs. They offset the height introduced by the mounting of the other side of the controller stack.



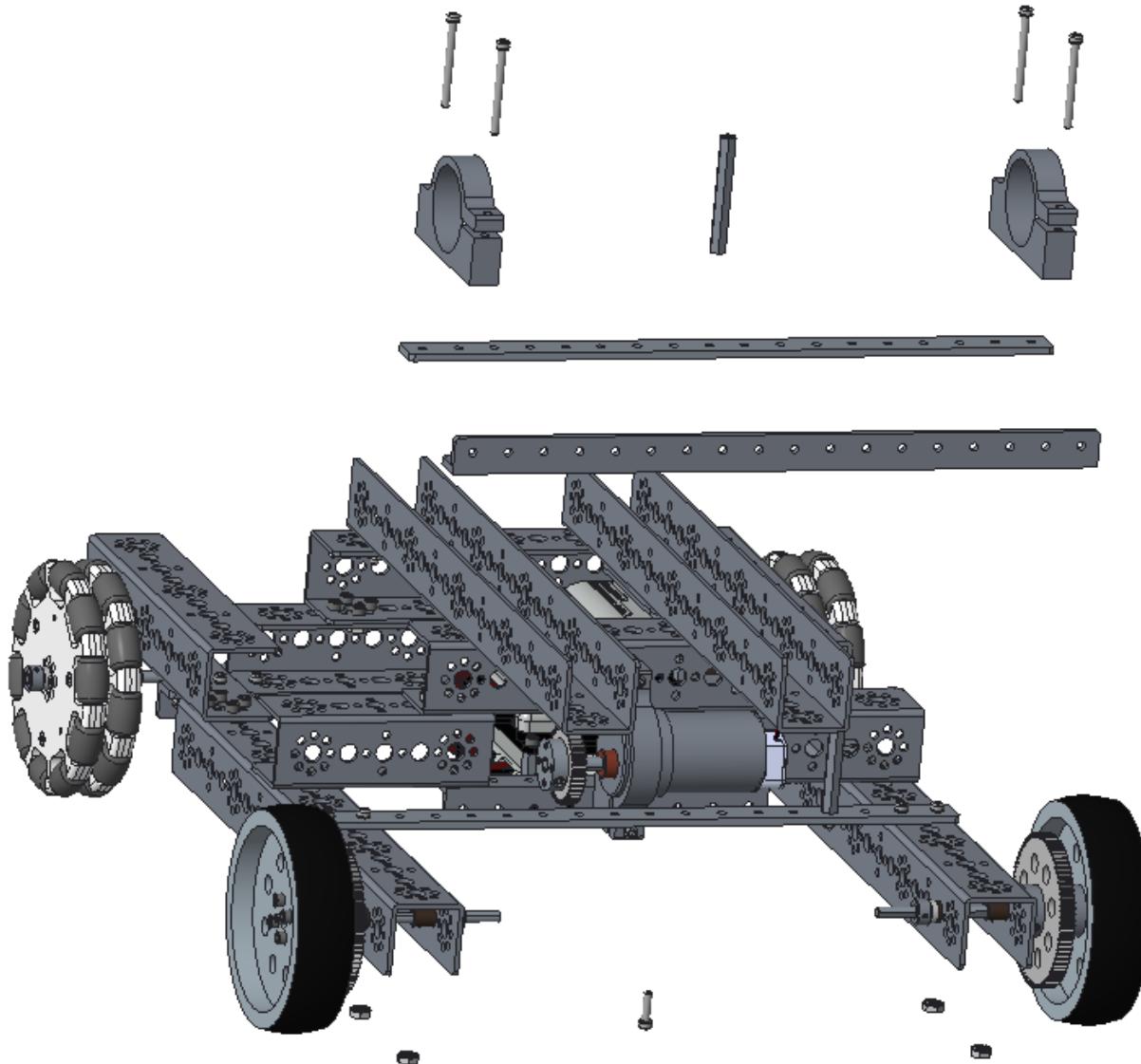
59: Completed Step 7 of Controller Stack Assembly Detail 2 showing the screw locations from above

Step 8: (optional) secure the cables to prevent damage.

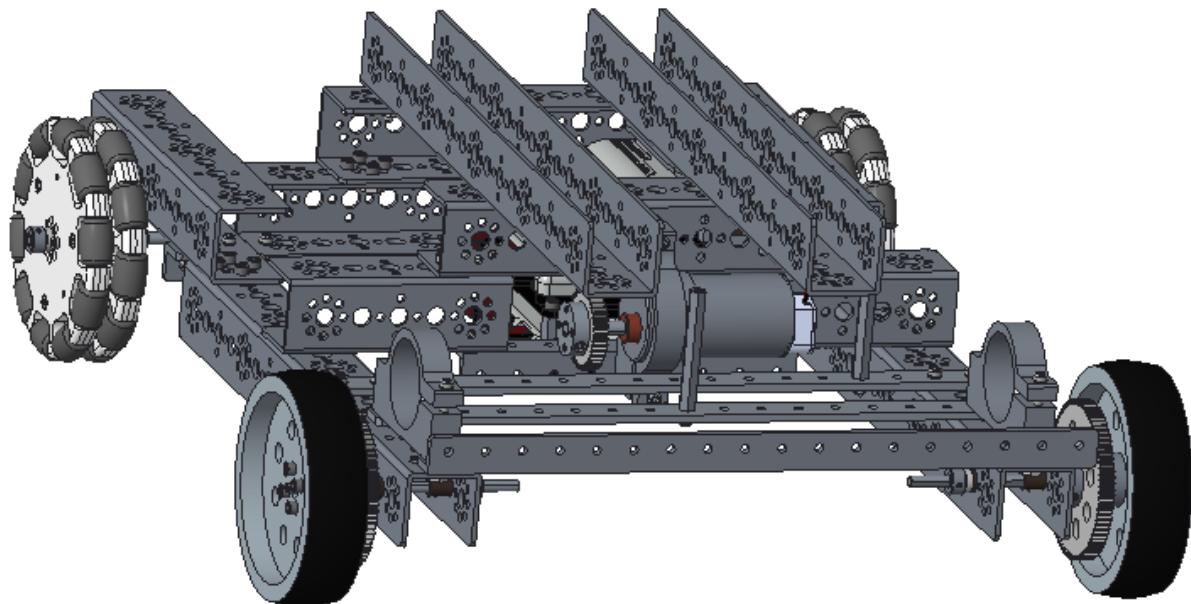


Adding the Drive Motors

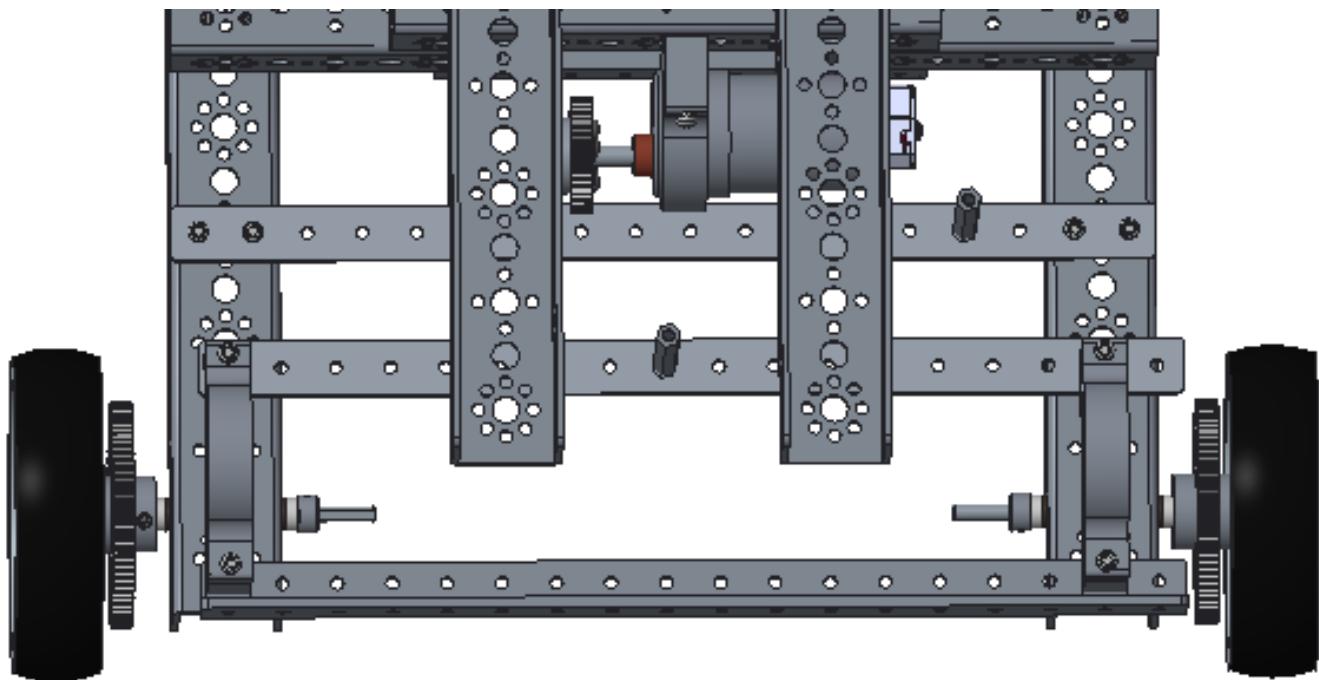
Step 1: Parts: chassis, motor mounts (and the two bolts and nuts that come with them) (2), 288 mm angle (1), 288 mm flat bar (1). Note: Do not tighten the rear-most motor mount bolts, the gap in the bracket, until the motors have been inserted, which happens at the end of this section. When this bolt is tightened, the motor is clamped in place.



60: Parts needed in Step 1 of Adding the Drive Motors

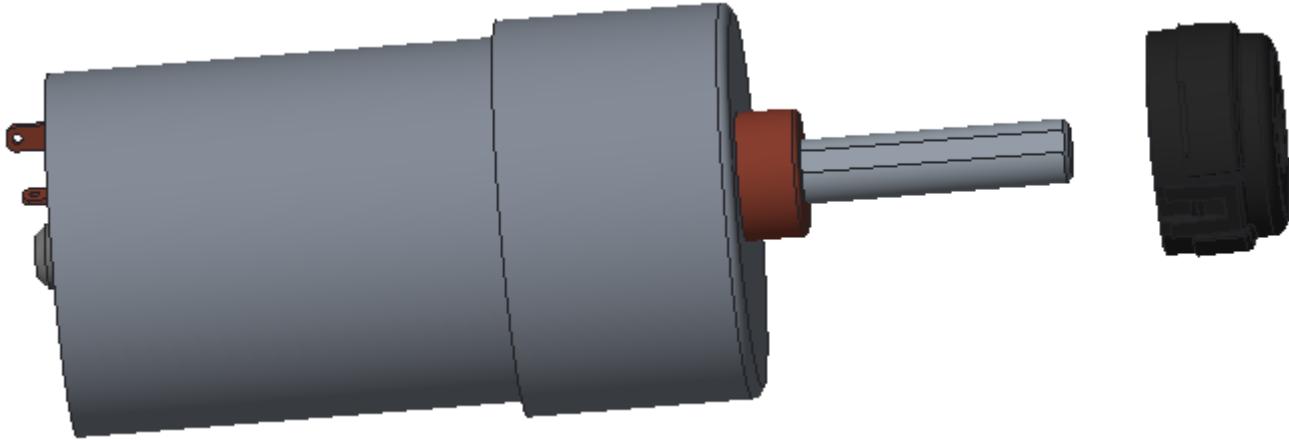


61: Completed Step 1 of Adding the Drive Motors

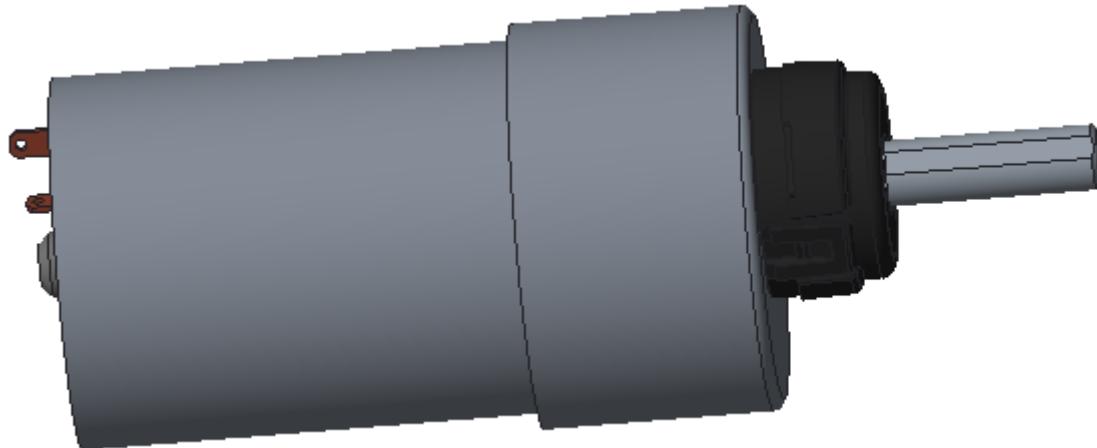


62: Completed Step 1 of Adding the Drive Motors detail view from above

Step 2: [make two] Parts: DC gear motor (1), encoder (1). Refer to the instructions from the encoder package to install the encoder onto the motor. Encoder styles vary, so the one in the kit of parts may not look like the one in this document.

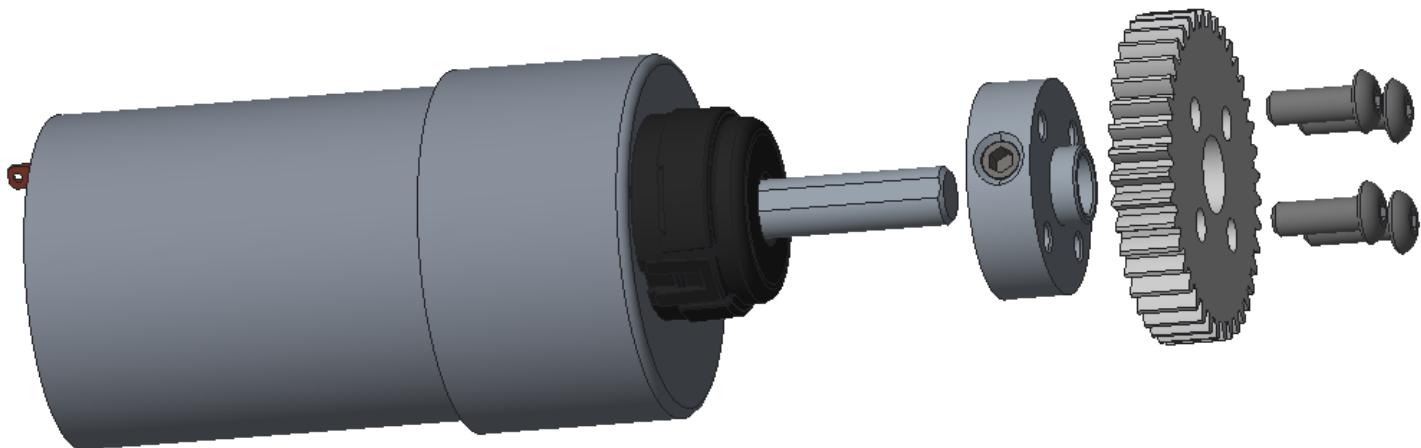


63: Parts needed in Step 2 of Adding the Drive Motors

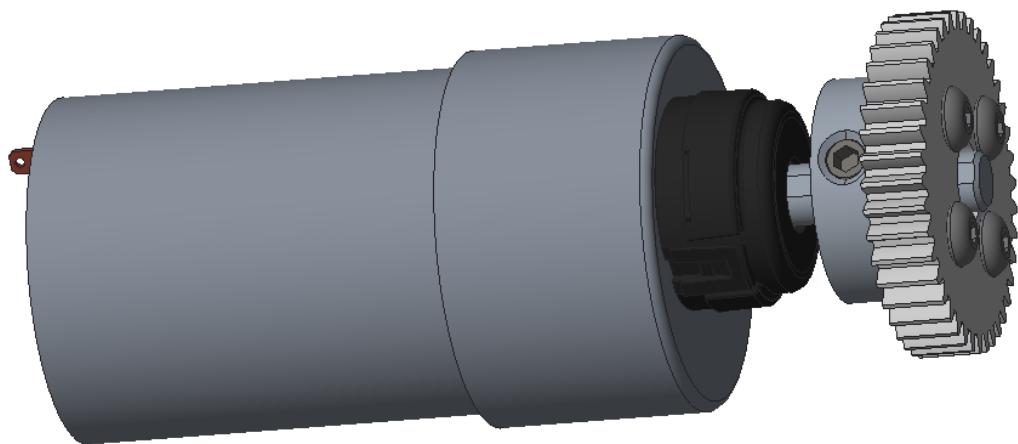


64: Completed Step 2 of Adding the Drive Motors

Step 3: [make two] Parts: motor assembly from the previous step, motor shaft hub (and its included set screw) (1), 40-tooth gear (1), button head cap screws (4)

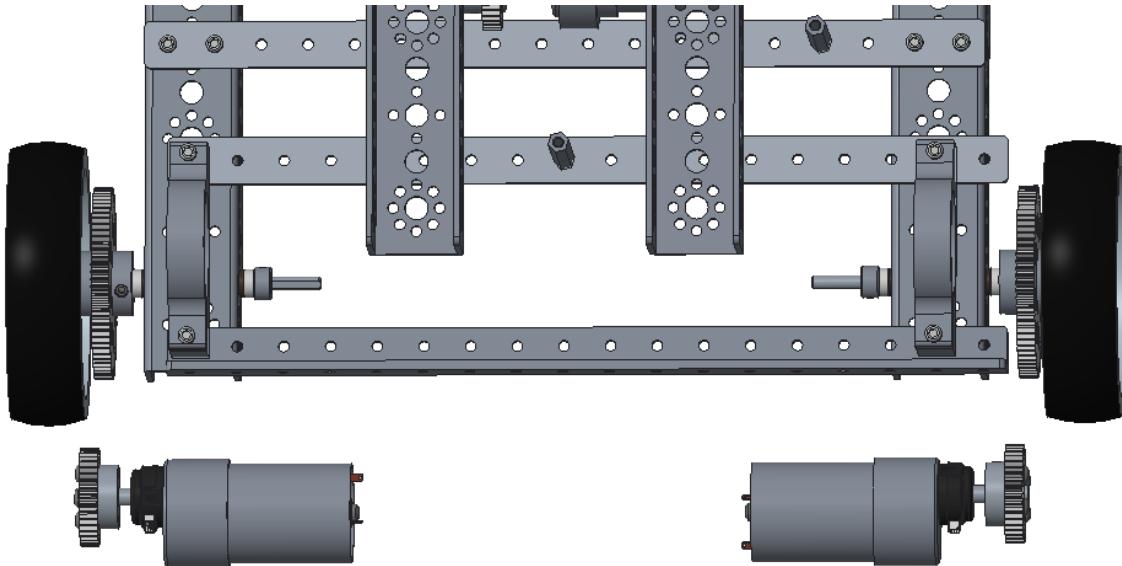


65: Parts needed in Step 3 of Adding the Drive

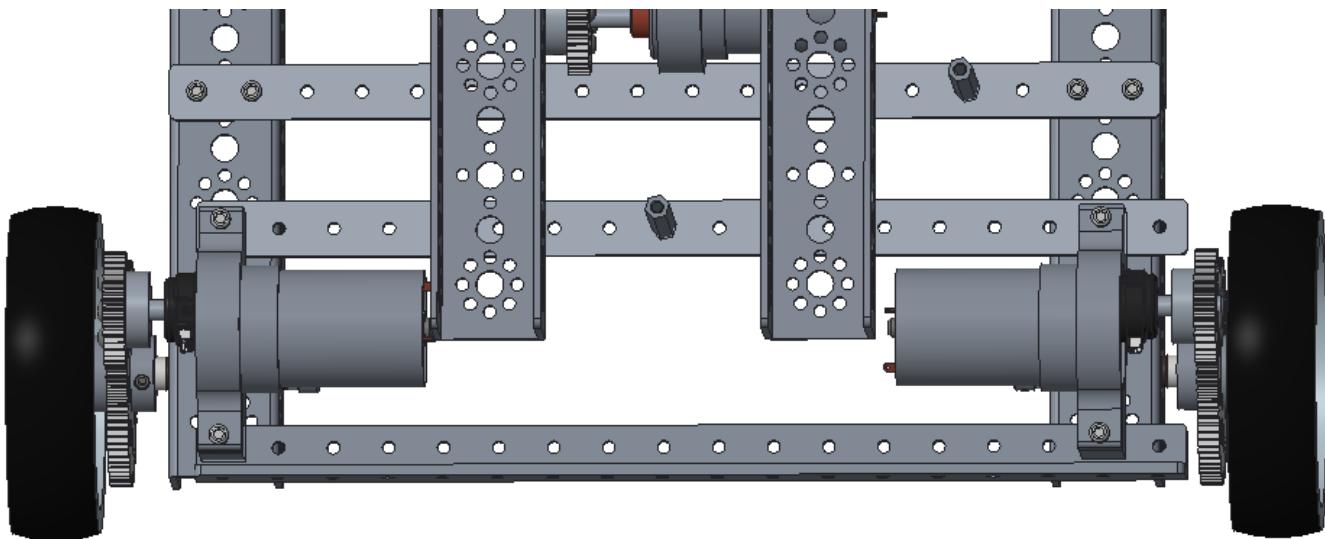


66: Completed Step 3 of Adding the Drive Motors

Step 4: Parts: chassis, motor assemblies from the previous steps (2). Insert the motors into the motor mounts, aligning the small gears on the motors with the medium gears on the wheels. Do not make the gears too tight or too loose or this will degrade the performance of the drive system. Experimentation with tightness can be done during the testing phase of this document. Tighten the motor mount bolts so the motor will not rotate.



67: Parts needed in Step 4 of Adding the Drive

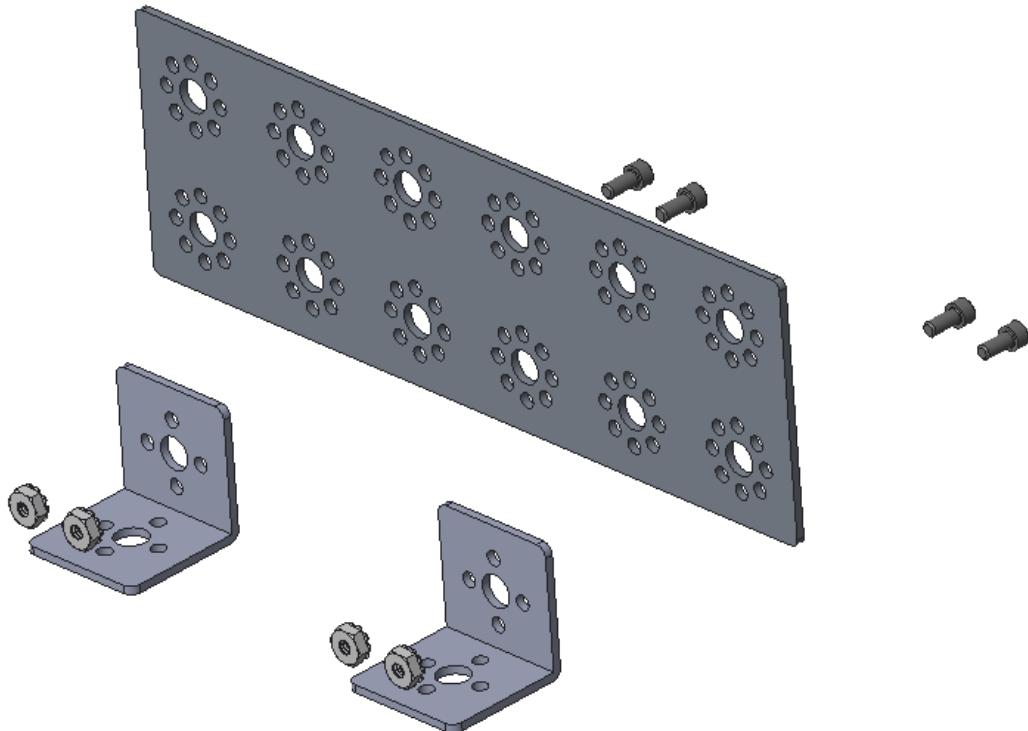


68: Completed Step 4 of Adding the Drive Motors

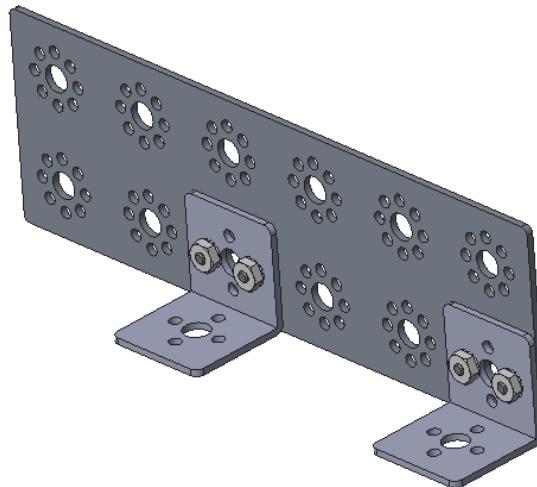
Adding the Power Distribution Module Assembly

The power distribution module supplies power to all of the controllers on the robot and also includes a USB hub to allow the phone to talk to each controller.

Step 1: Parts: flat building plate (1), inside corner brackets (2), 5/16" socket head cap screws (4), keps nuts (4). The button head screws are on the bottom plate.

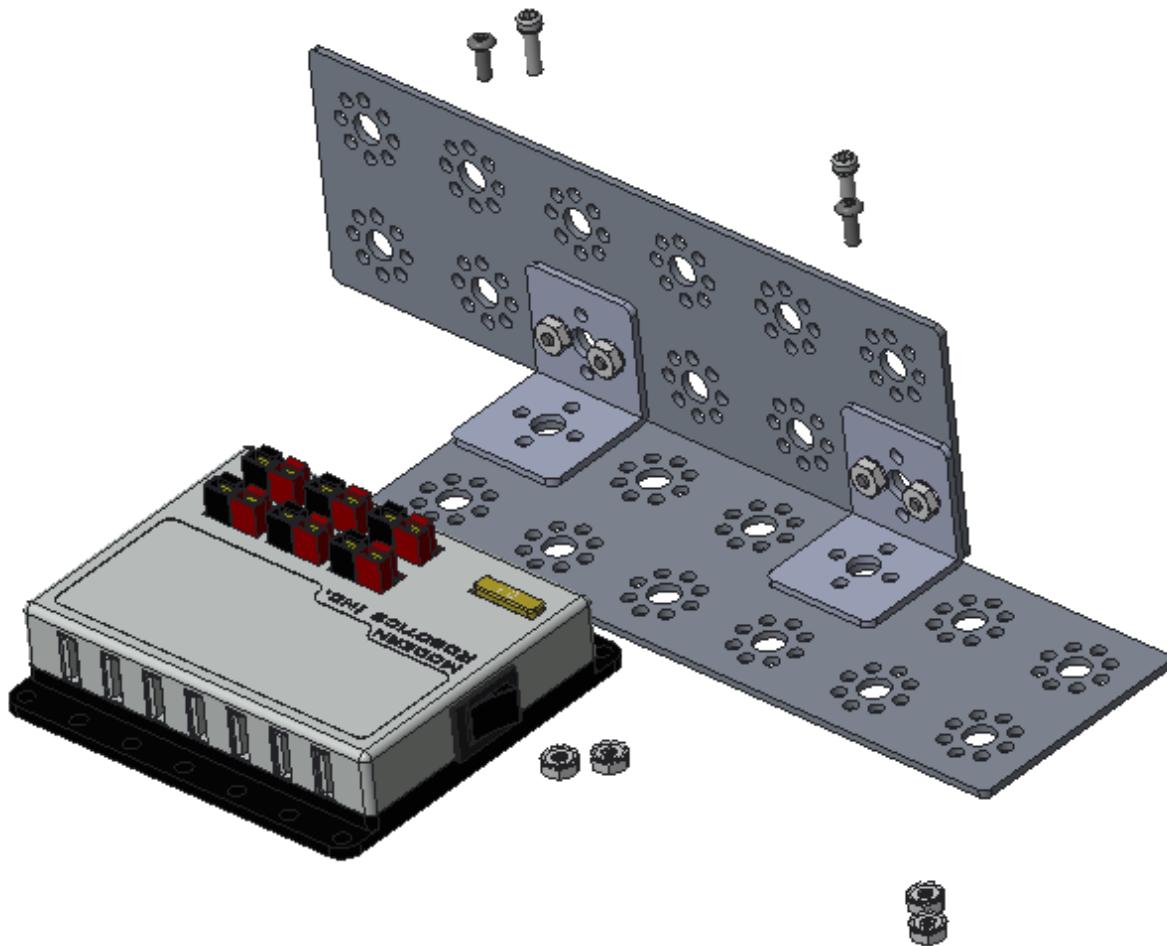


69: Parts needed in Step 1 of Adding the Power Distribution Module Assembly



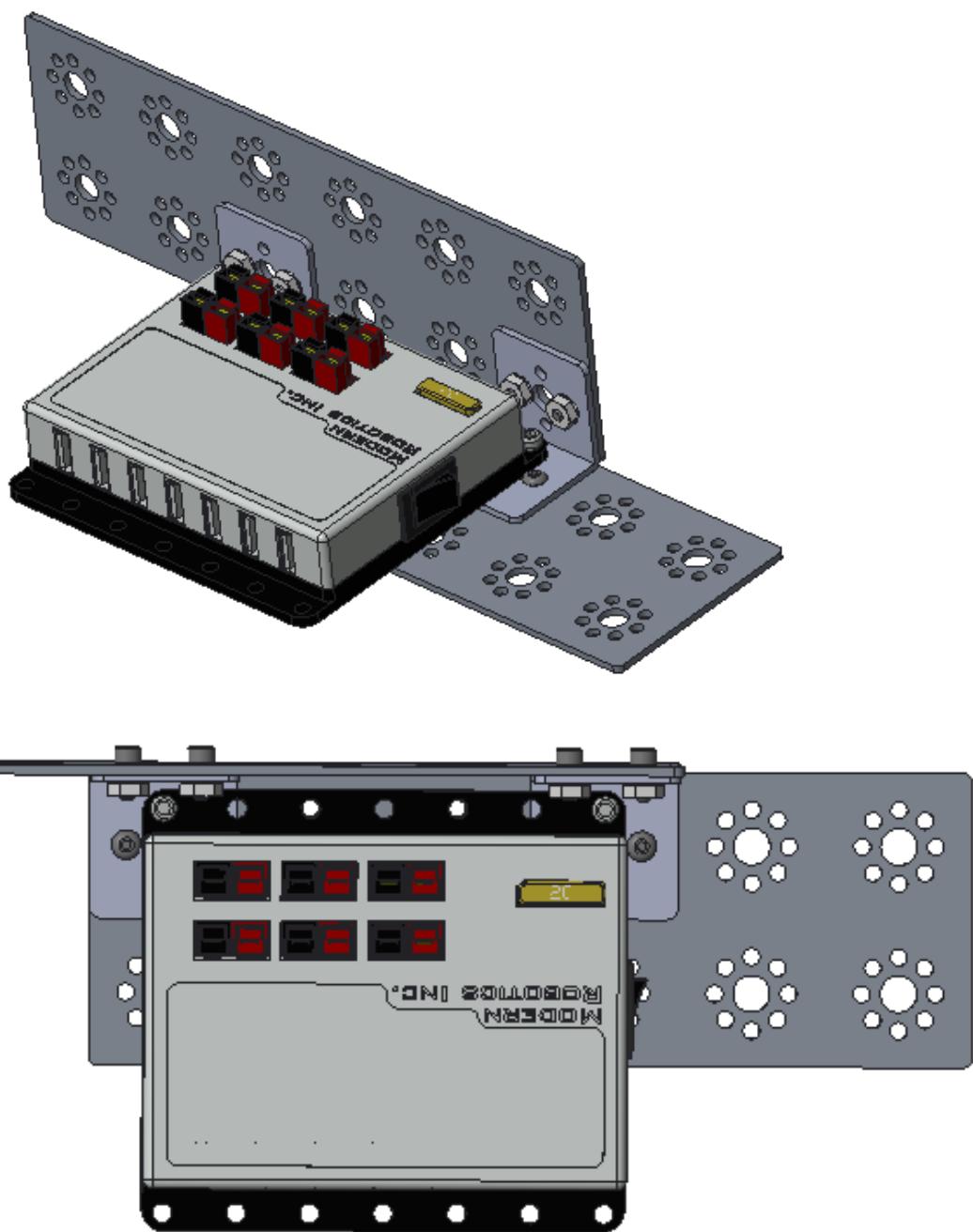
70: Completed Step 1 of Adding the Power Distribution Module Assembly

Step 2: Parts: assembly from previous step, flat building plate (1), core power distribution module (1), button head cap screws (2), 1/2" socket head cap screws (2), keps nuts (4). The button head screws are on the outside edges of the angle brackets.



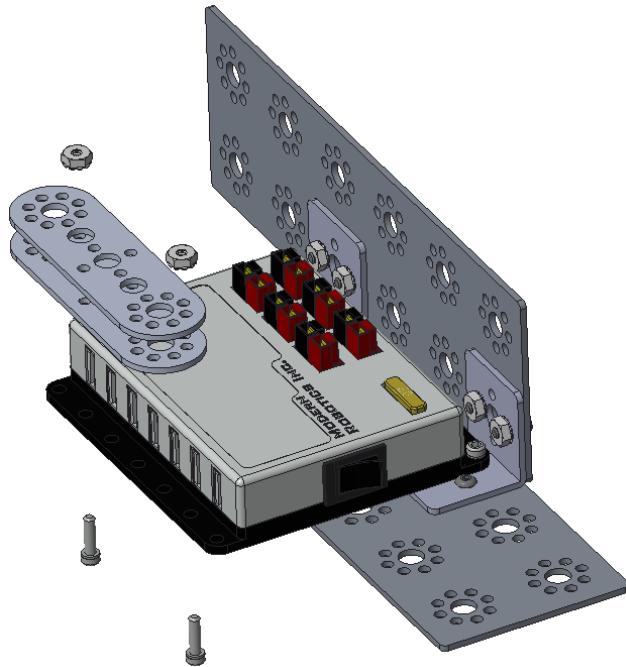
71: Parts needed in Step 2 of Adding the Power Distribution Module Assembly

The completed assembly is shown on the next page.

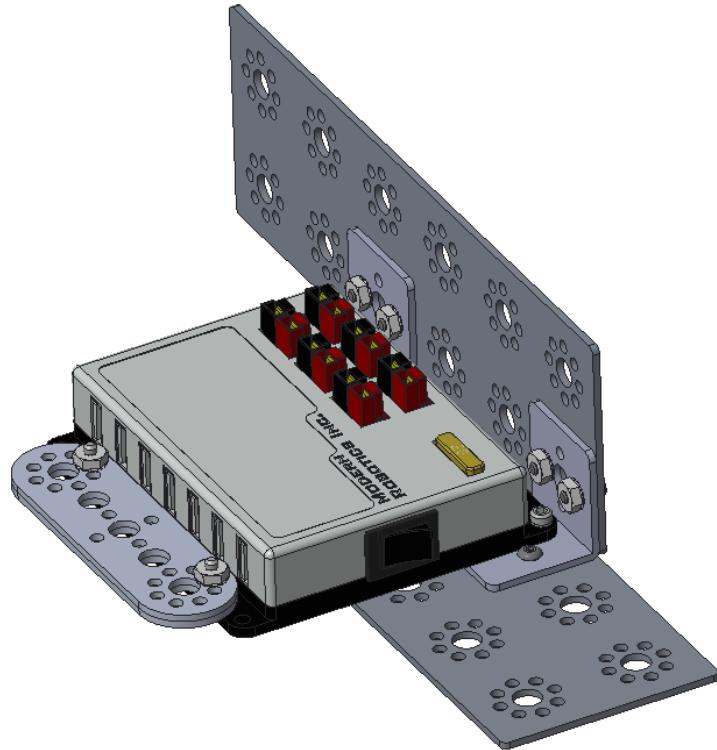


72: Completed Step 2 of Adding the Power Distribution Module Assembly

Step 3: Parts: assembly from the previous step, 96 mm flat brackets (2), 1/2" socket head cap screws (2), keps nuts (2)

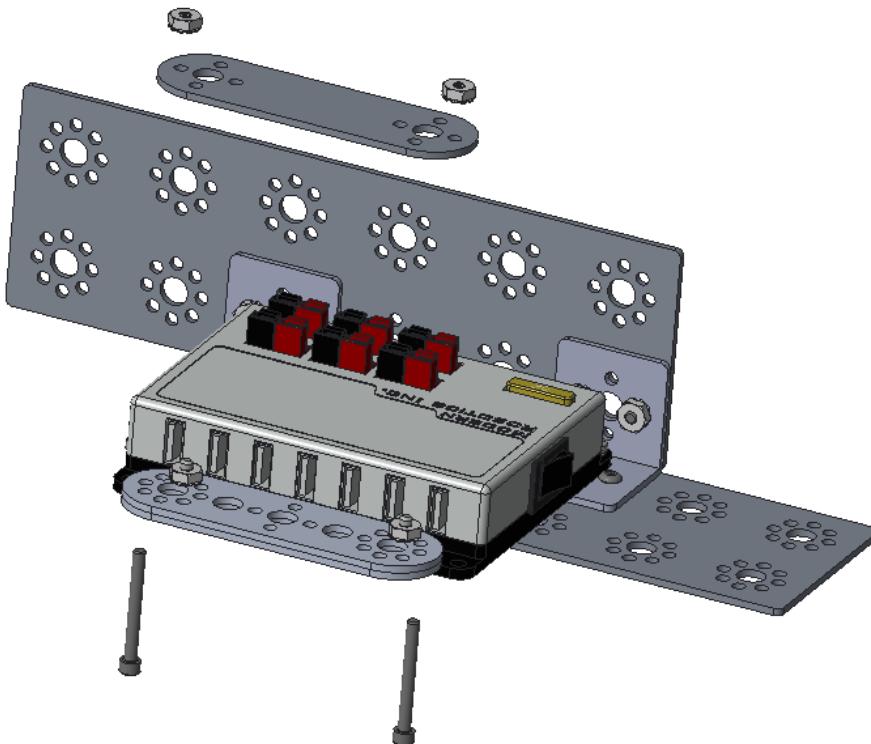


73: Parts needed in Step 3 of Adding the Power Distribution Module Assembly

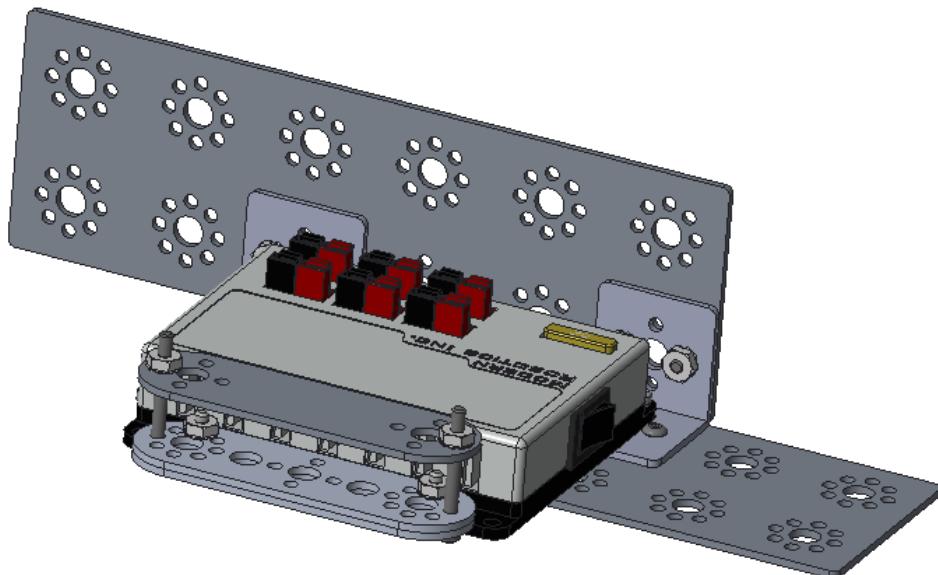


74: Completed Step 3 of Adding the Power Distribution Module Assembly

Step 4: Parts: assembly from the previous step, 96 mm flat brackets (1), bolts from a single standard-scale servo motor mounting bracket (2), keps nuts (2). The top bracket will not be tight. The USB A to mini-B cables from the PDM to the controller stack will be secured using this device (shown in the wiring section later).

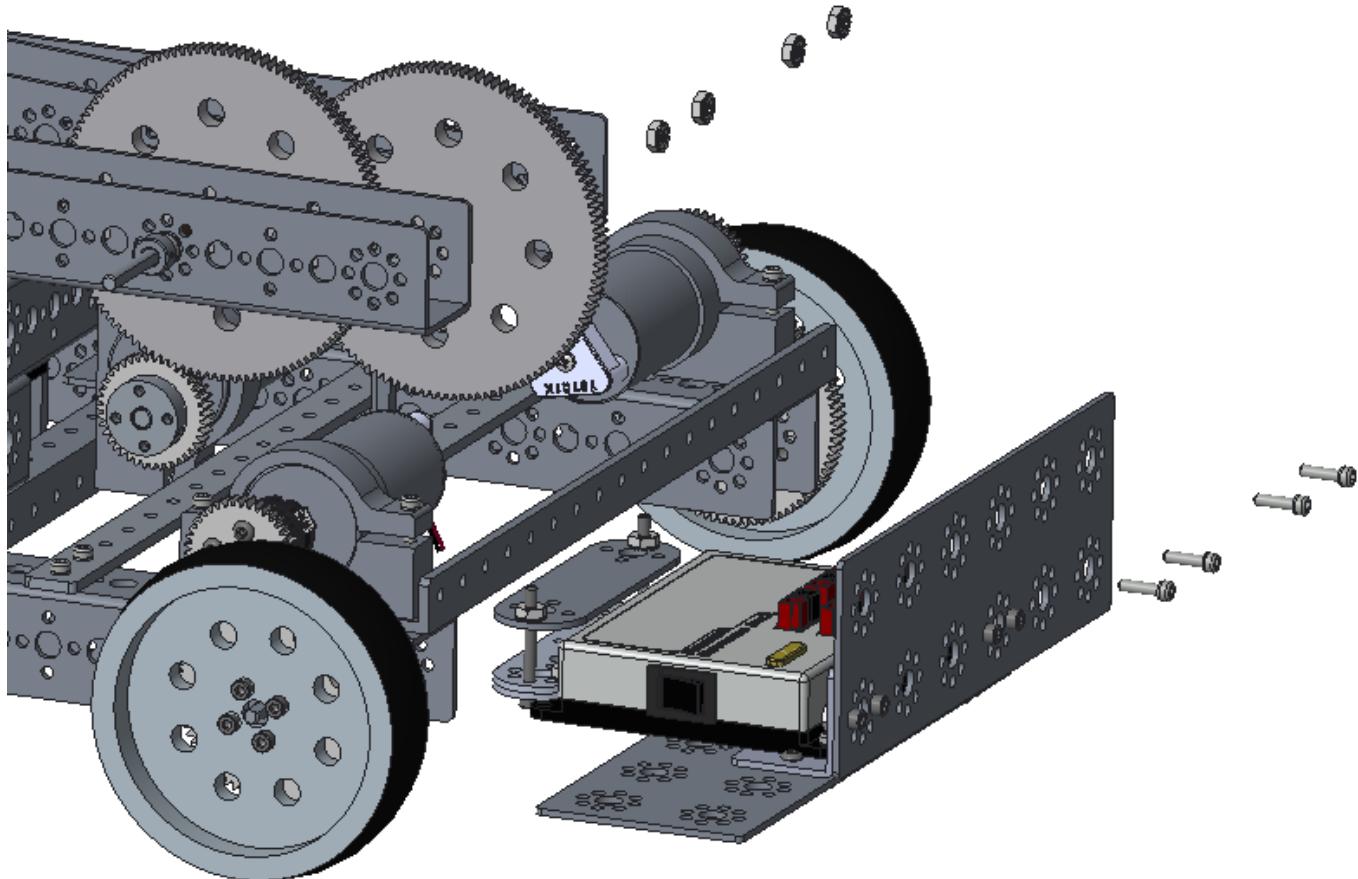


75: Parts needed in Step 4 of Adding the Power Distribution Module Assembly



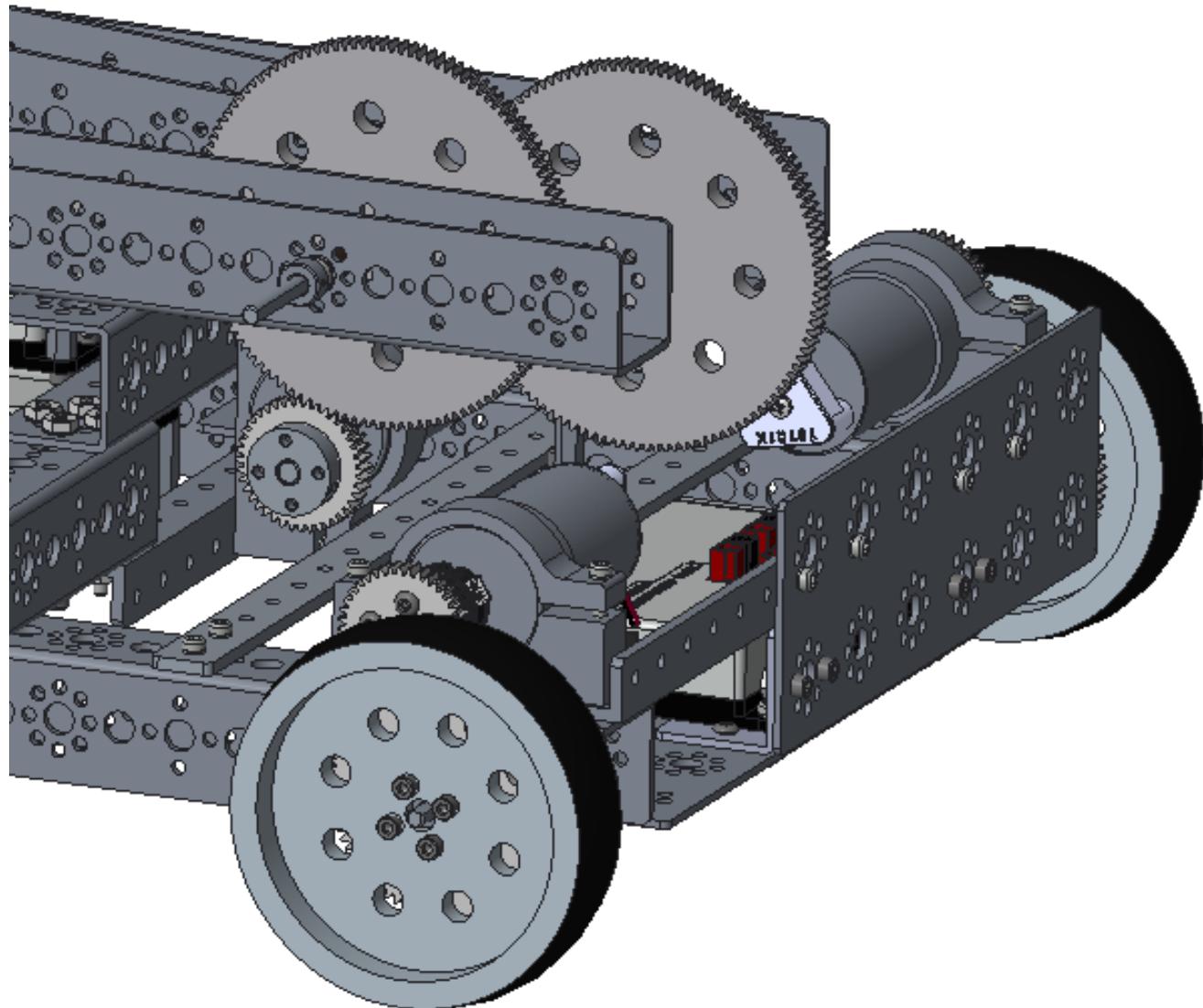
76: Completed Step 4 of Adding the Power Distribution Module Assembly

Step 5: Parts: chassis, assembly from the previous step, 1/2" socket head cap screws (4), keps nuts (4)



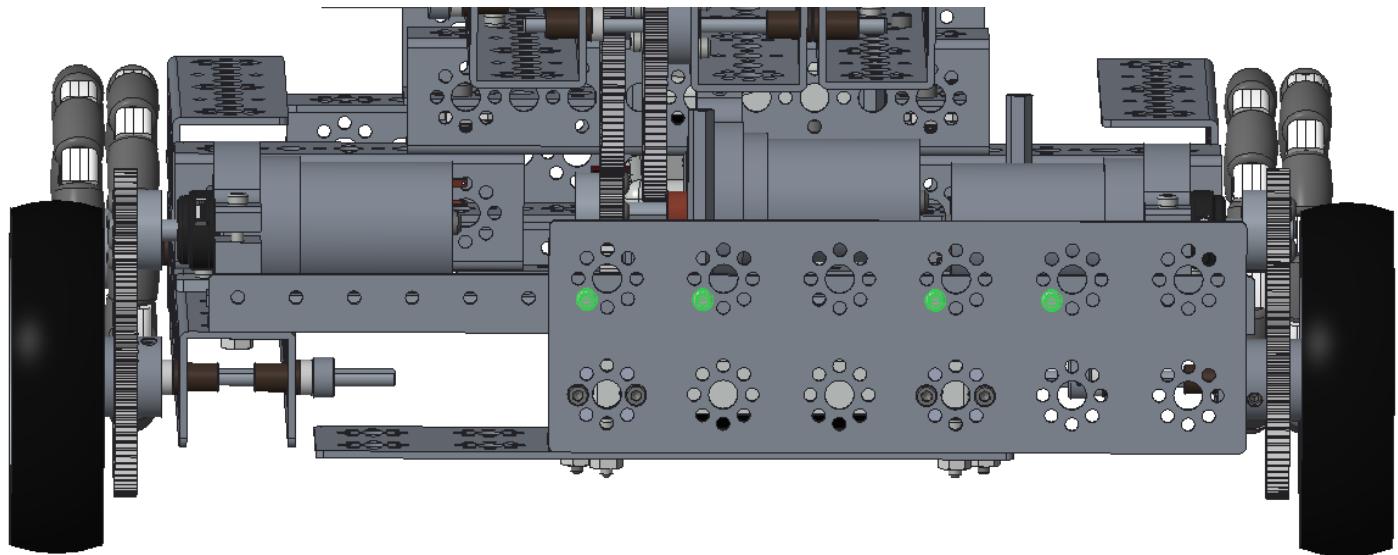
77: Parts needed in Step 5 of Adding the Power Distribution Module Assembly

The completed assembly is shown on the next two pages.



78: Completed Step 5 of Adding the Power Distribution Module Assembly

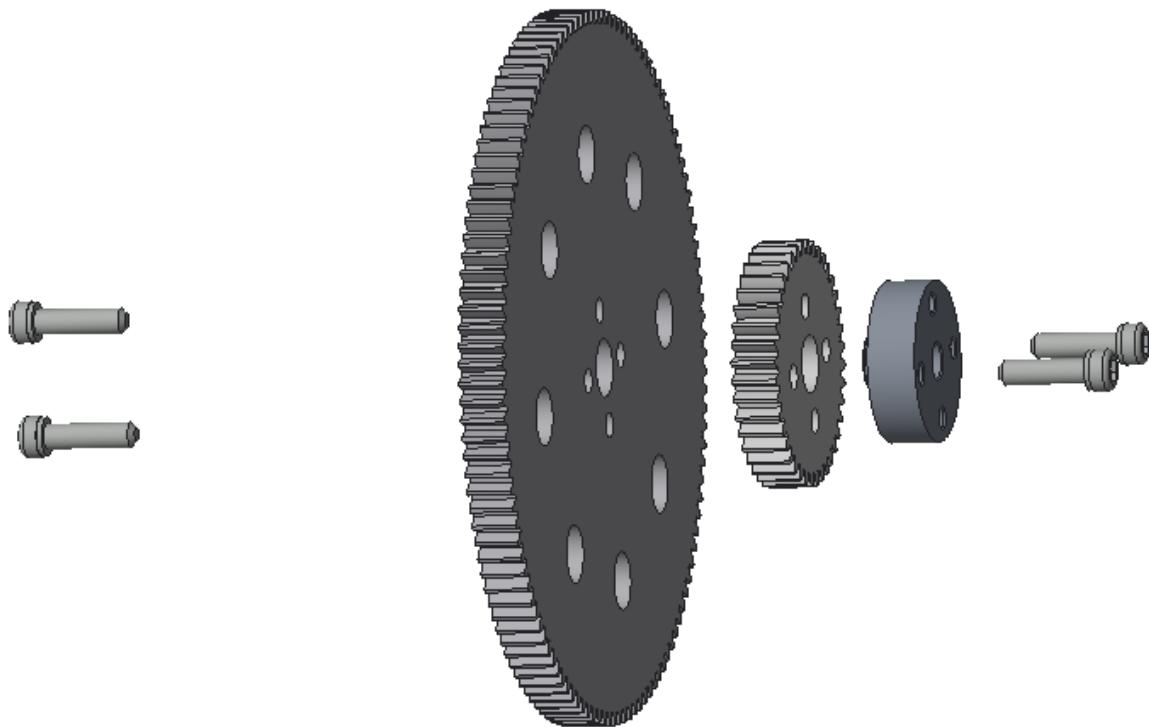
Notice on this rear view of the robot that the bolts are one up and to the left of the hole pattern. This brings the lower plate higher on the robot to allow for a higher drive clearance.



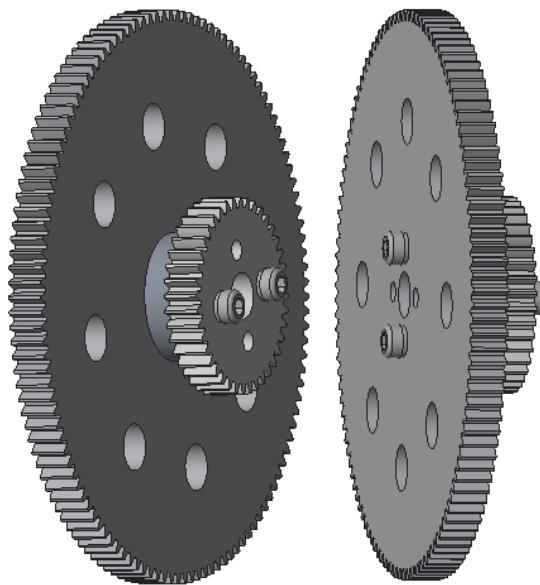
79: Completed Step 5 of Adding the Power Distribution Module Assembly detail view of the back of the robot

Adding the Arm Motor Gears

Step 1: Parts: 40-tooth gear (1), 120-tooth gear (1), axle hub (with its set screw) (1), 1/2" socket head cap screws (4). Note that the bolts are inserted from opposite sides. The two bolts on each side are in opposite holes. The bolts on one side will be 90 degrees rotated from the bolts on the other side.

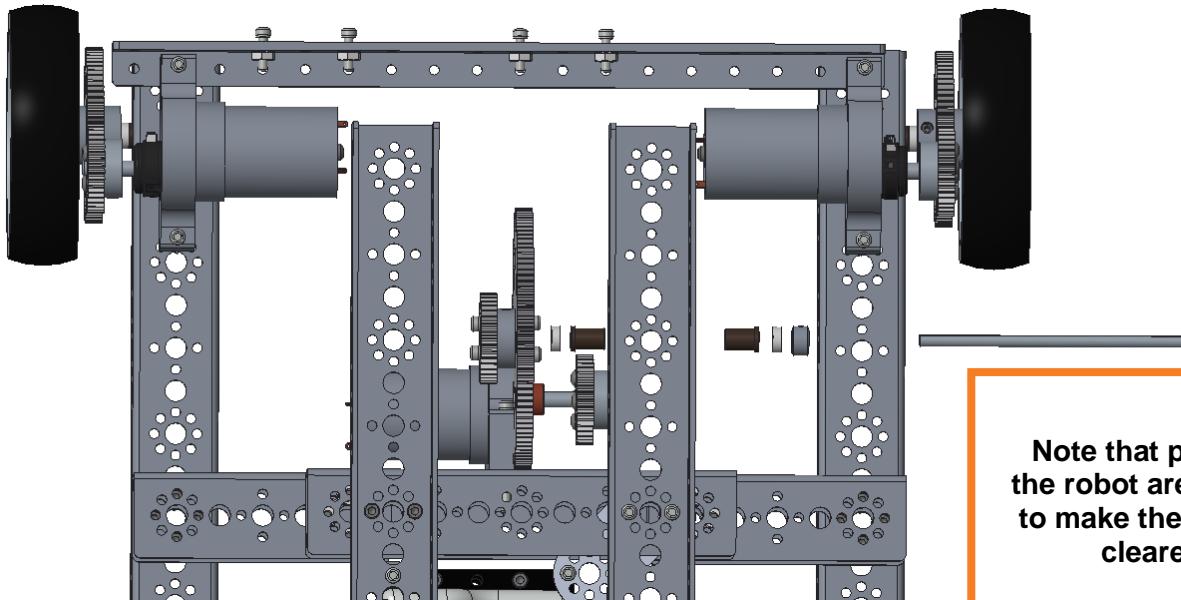


80: Parts needed in Step 1 of Adding the Arm Motor Gears



81: Completed Step 1 of Adding the Arm Motor Gears

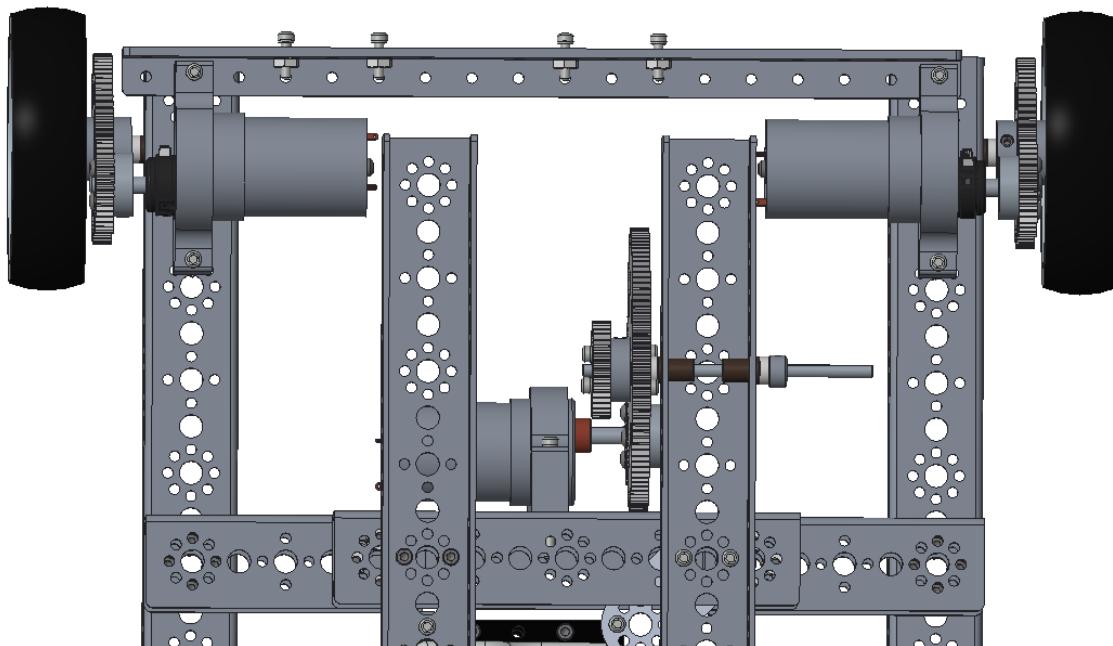
Step 2: Parts: chassis, assembly from the previous step, 1/8" axle spacers (2), 11 mm bronze bushings (2), axle set collar (with its set screw) (1), 100 mm axle (1). Tighten the axle collar and axle clamp set screws, making sure the set screws face the flat side of the axle.



Note that parts of the robot are hidden to make the picture clearer.

82: Parts needed in Step 2 of Adding the Arm Motor Gears

Do not yet tighten the motor screw (the lower one through the gap in the motor mount) - the motor should remain disengaged from the gearing so that the arm gears will be free to spin as the arm is installed.

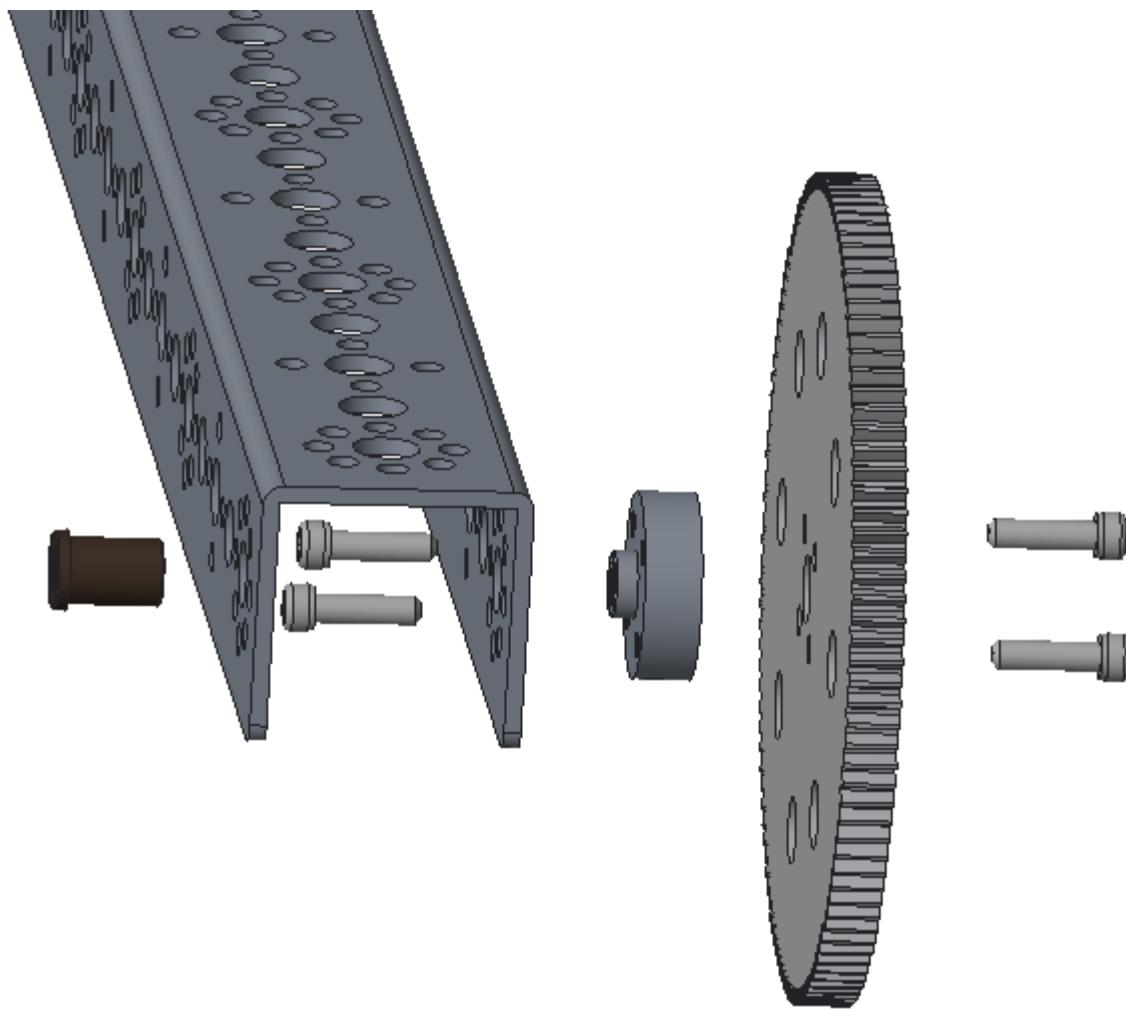


83: Completed Step 2 of Adding the Arm Motor Gears

Adding the Arm

The arm can be used to manipulate various game elements.

Step 1: Parts: 416 mm channel (1), 1/2" socket head cap screws (4), 11 mm bronze bushing (1), axle set clamp (with its set screw) (1).

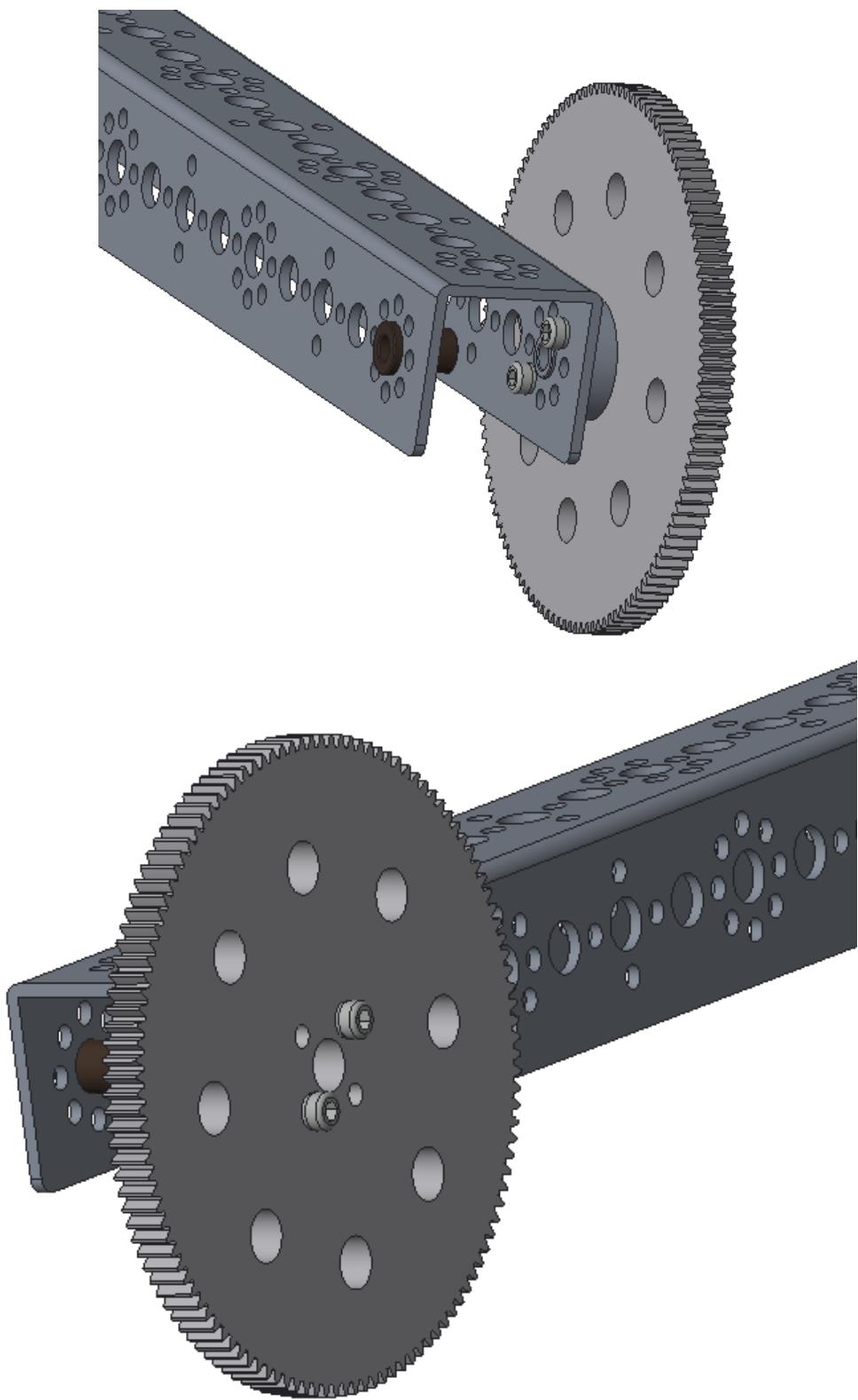


84: Parts needed in Step 1 of Adding the Arm

Notice that the two bolts on one side are in opposite holes from the bolt on the other side.

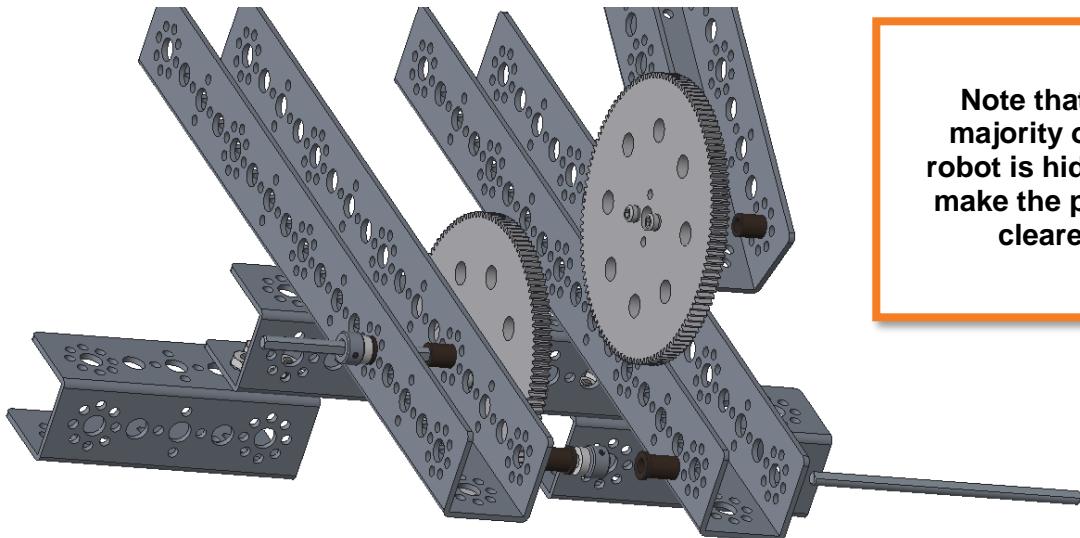
Also take note of placing the axle clamp so its set screw faces the open end of the channel. This will make it easier to tighten/loosen with the arm in its final position.

The completed step is shown on the next page from two different angles.



85: Completed Step 1 of Adding the Arm Motor Gears

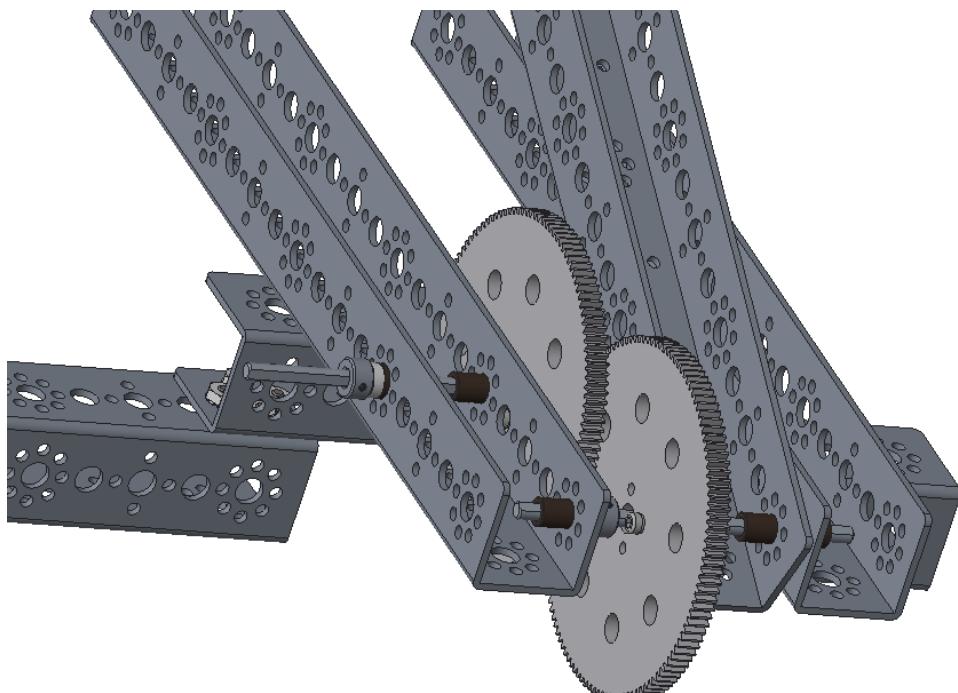
Step 2: Parts: chassis, arm assembly from the previous step, axle collar (with its set screw), 11 mm bronze bushings (3), 100 mm axle, 1/8" axle spacer (1).



Note that the majority of the robot is hidden to make the picture clearer.

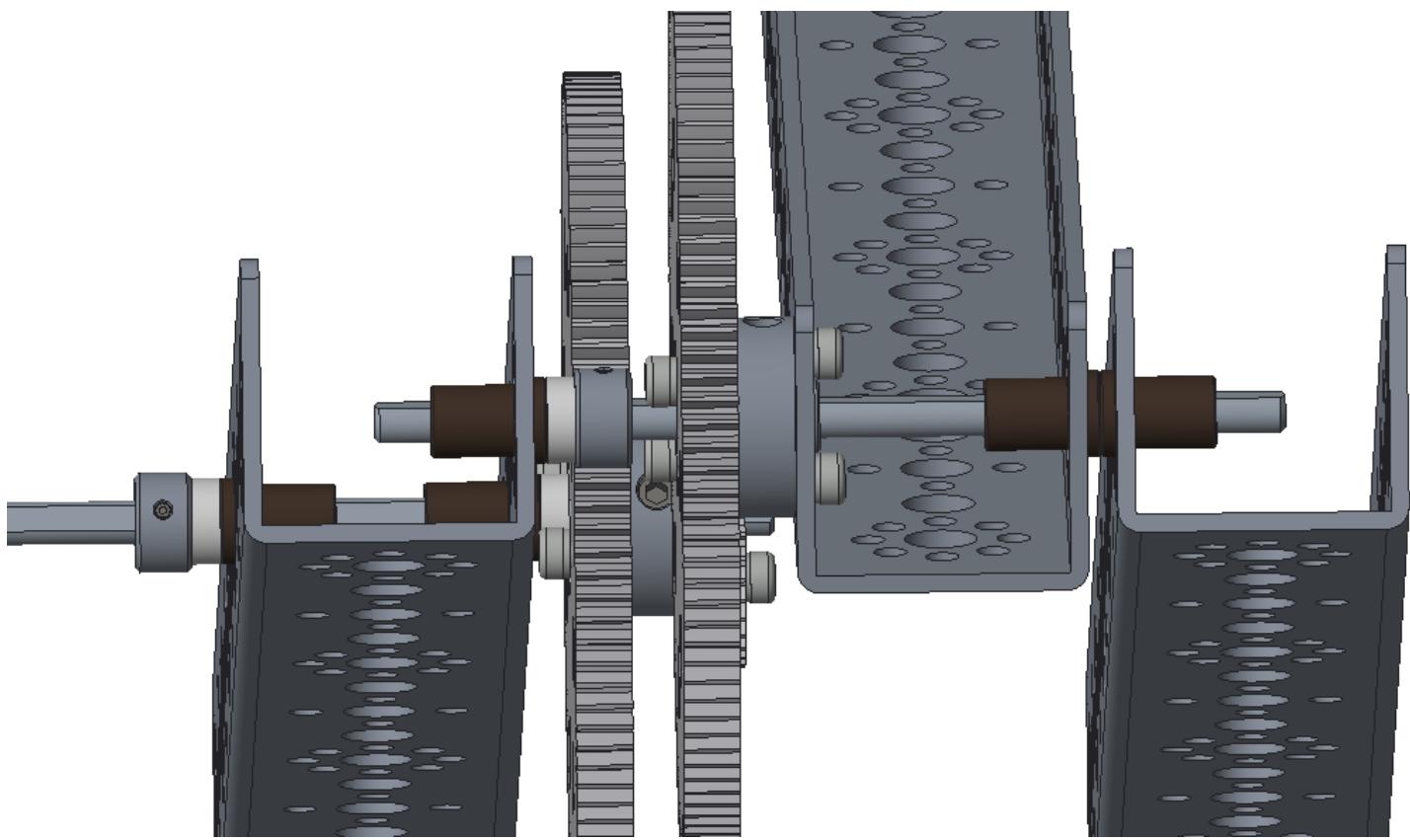
86: Parts needed in Step 2 of Adding the Arm

Make sure the set screws of the collar and clamp are facing the flat side of the axle. The order from left to right in the image below is bushing, channel, spacer, collar, [space], arm, bushing, bushing, channel. Align the large arm gear with the small gear on the chassis.



87: Completed Step 2 of Adding the Arm Motor Gears

A detail of the axel is shown on the next page.



88: Completed Step 2 of Adding the Arm Motor Gears detail view

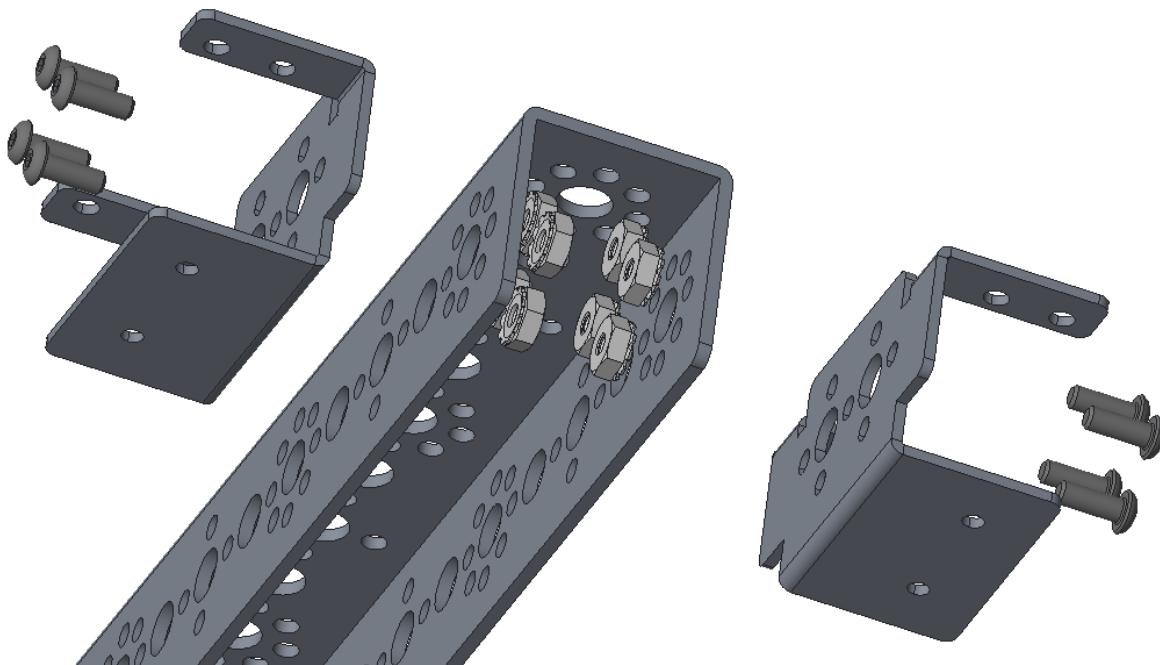
This detail of the completed step (shown on the previous page) shows the order.

The order from left to right is bushing, channel, spacer, collar, [space], arm, bushing, bushing, channel.

Adding the Hands

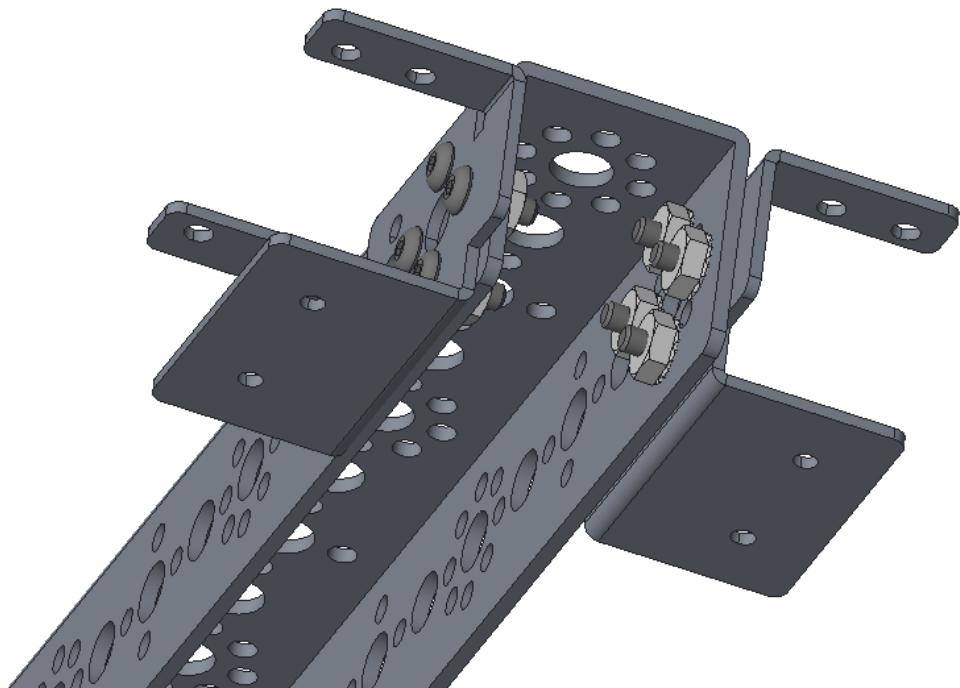
The hand can be used to pick up game elements.

Step 1: Parts: chassis, single standard-scale servo motor mounting brackets (2), button head cap screws (8), keps nuts (8).



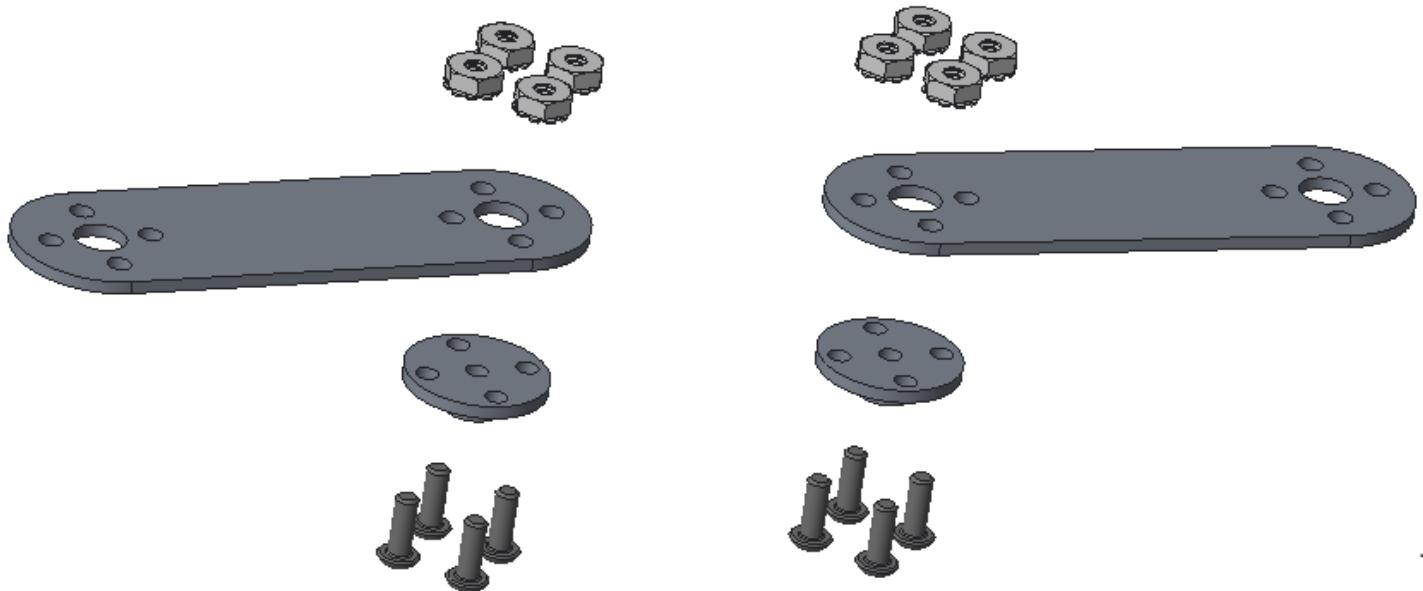
89: Parts needed in Step 1 of Adding the Hands

The arm motor is not yet engaged with the gear system, so rotate the arm so it is shown in the above orientation.



90: Completed Step 1 of Adding the Hands

Step 2: Parts: flat brackets (2), quarter-scale servo horns (2), button head cap screws (4), keps nuts (8)



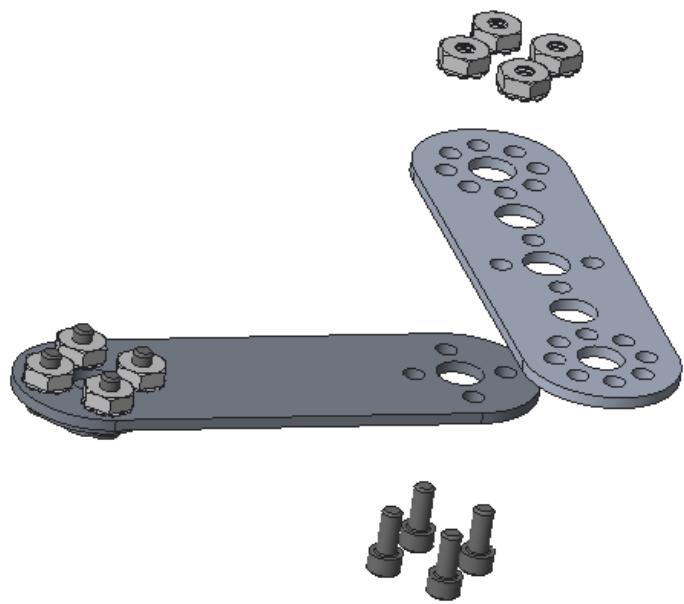
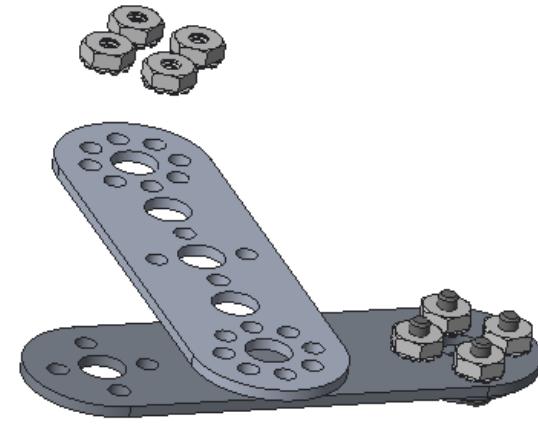
91: Parts needed in Step 2 of Adding the Hands

This step shows the making of both hands. Do not duplicate this step.



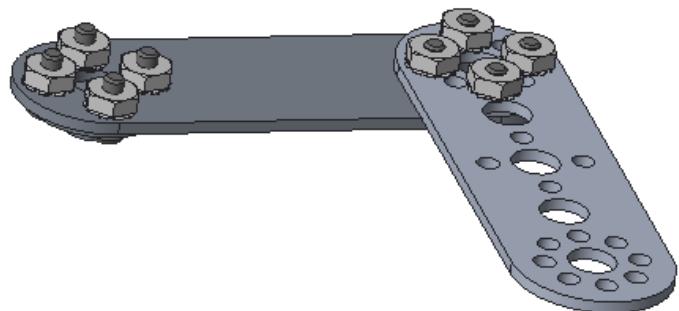
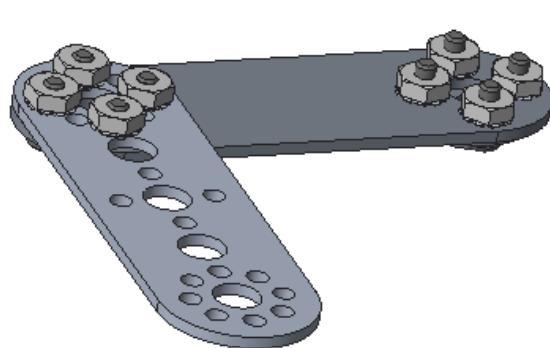
92: Completed Step 2 of Adding the Hands

Step 3: Parts: 96 mm flats (2), 5/16" socket head cap screws (8), keps nuts (8).



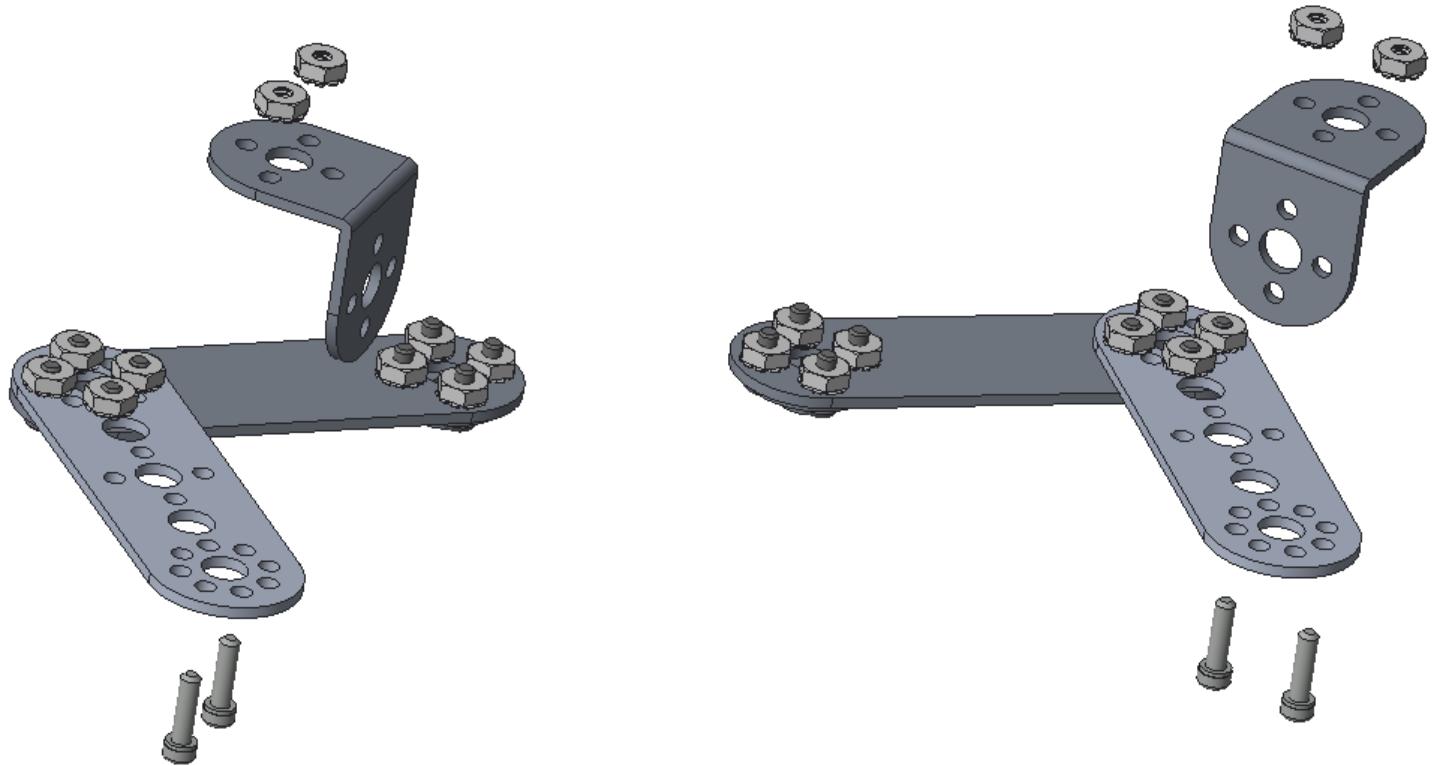
93: Parts needed in Step 3 of Adding the Hands

This step shows the making of both hands. Do not duplicate this step.



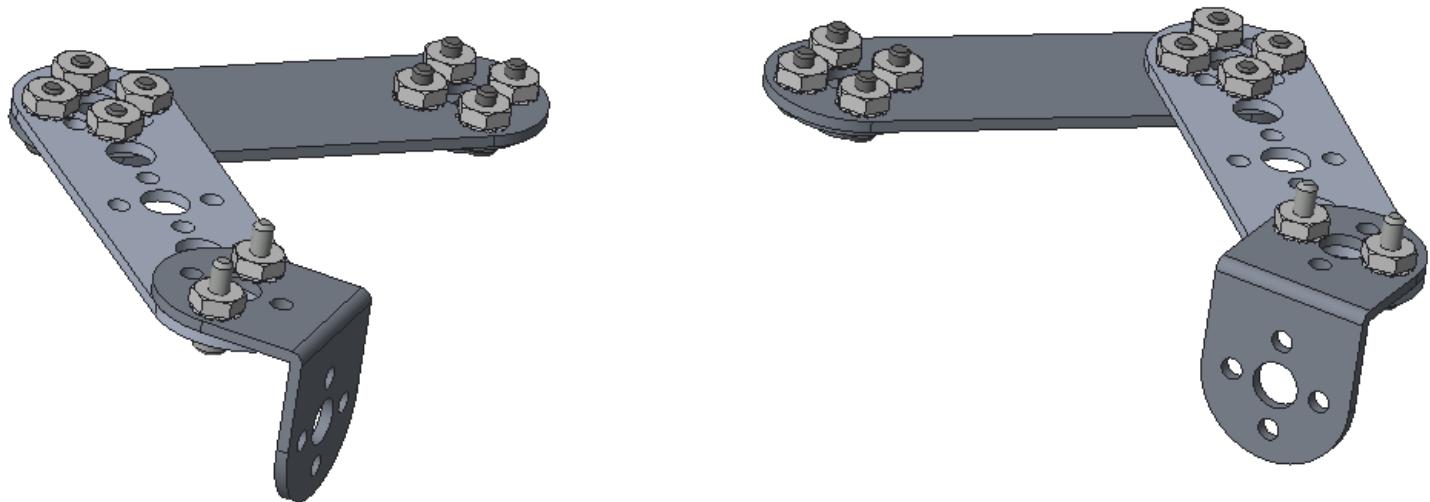
94: Completed Step 3 of Adding the Hands

Step 4: Parts: L-Brackets (2), 1/2" socket head cap screws (4), keps nuts (4)



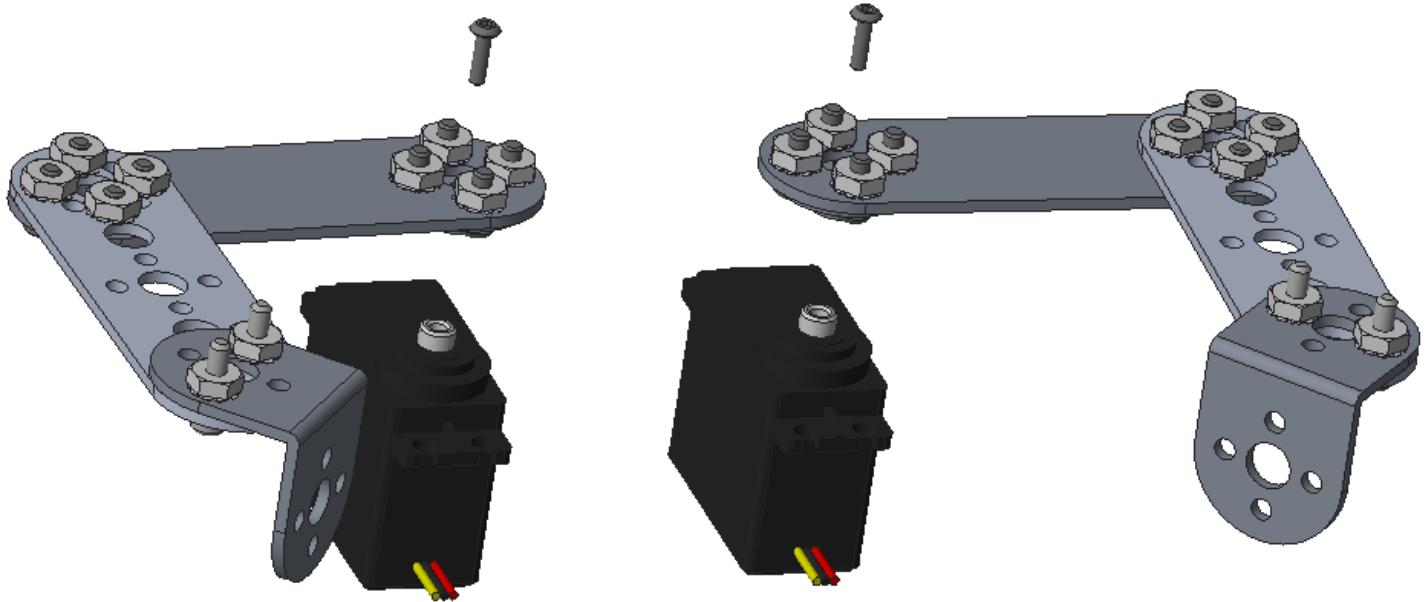
95: Parts needed in Step 4 of Adding the Hands

This step shows the making of both hands. Do not duplicate this step.



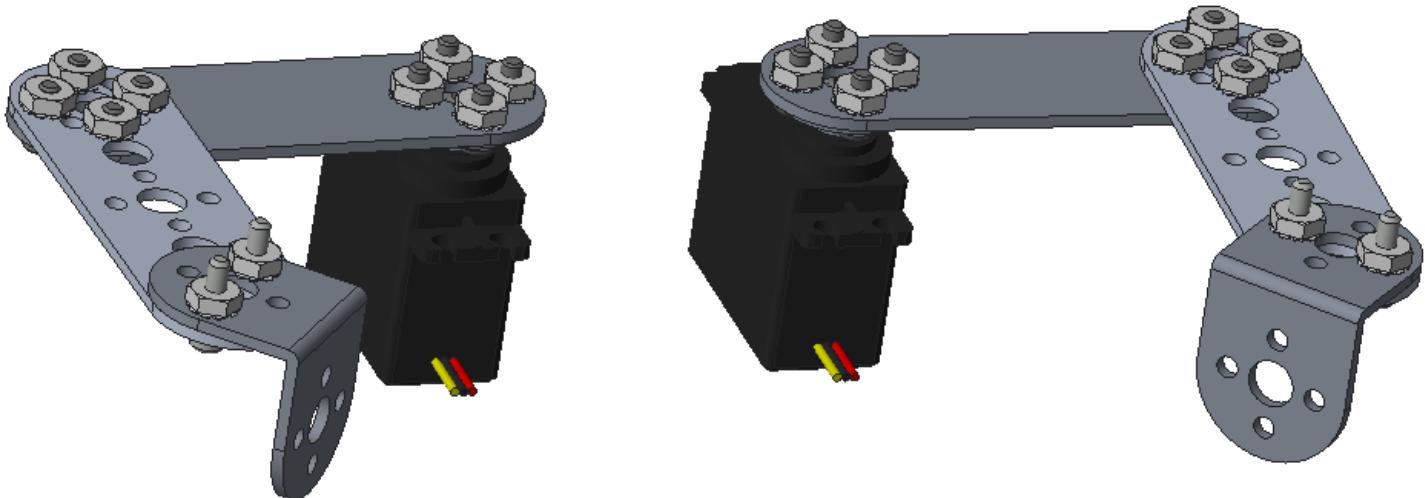
96: Completed Step 4 of Adding the Hands

Step 5: Parts: standard-scale servo motors (with the screws that come in the package) (2)



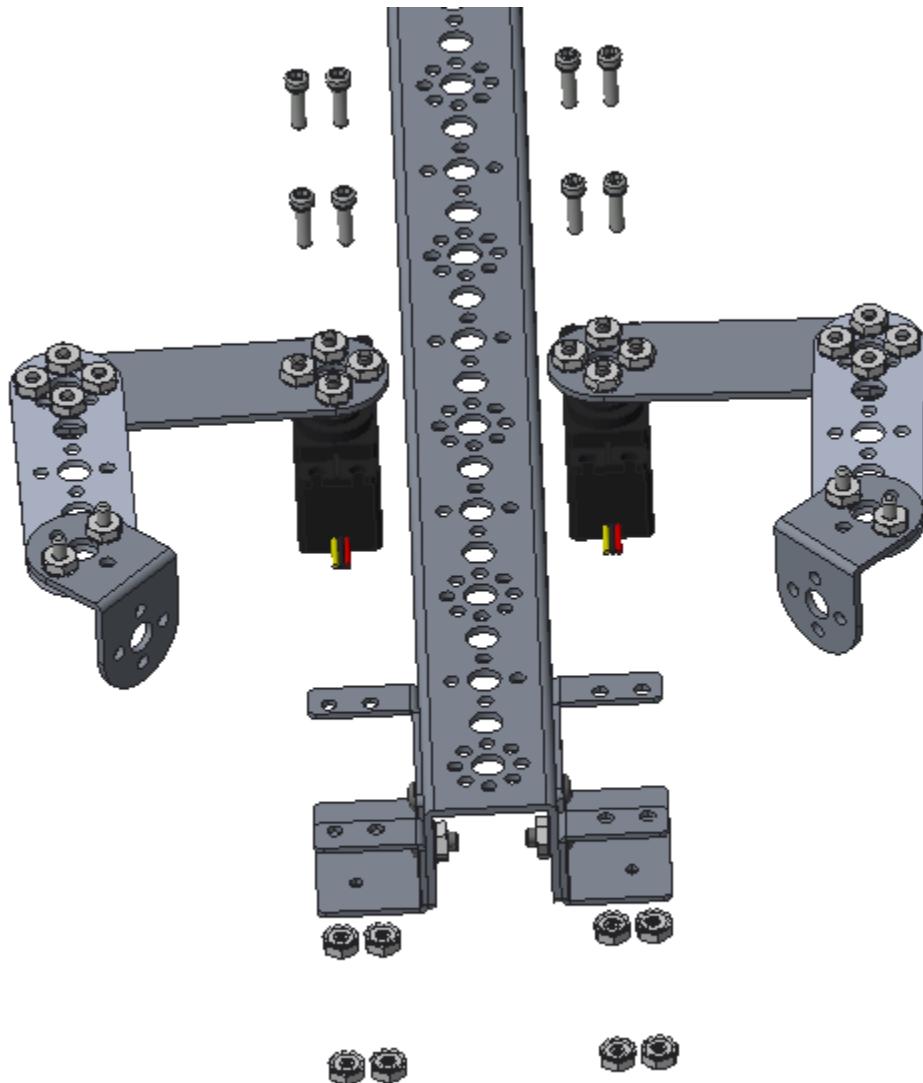
97: Parts needed in Step 5 of Adding the Hands

This step shows the making of both hands. Do not duplicate this step.



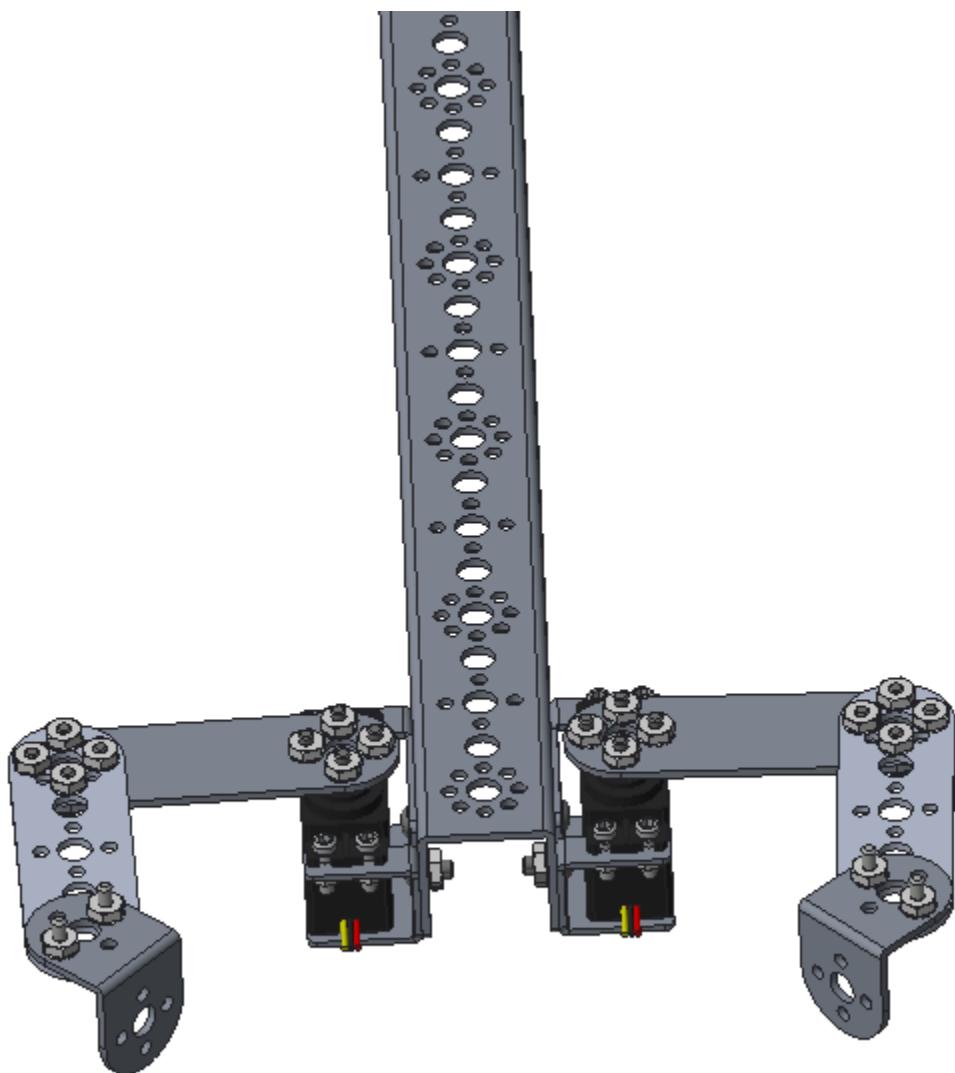
98: Completed Step 5 of Adding the Hands

Step 6: Parts: chassis, hand assemblies from previous steps, 1/2" socket head cap screws (8), keps nuts (8)



99: Parts needed in Step 6 of Adding the Hands

The completed step is shown on the next page.

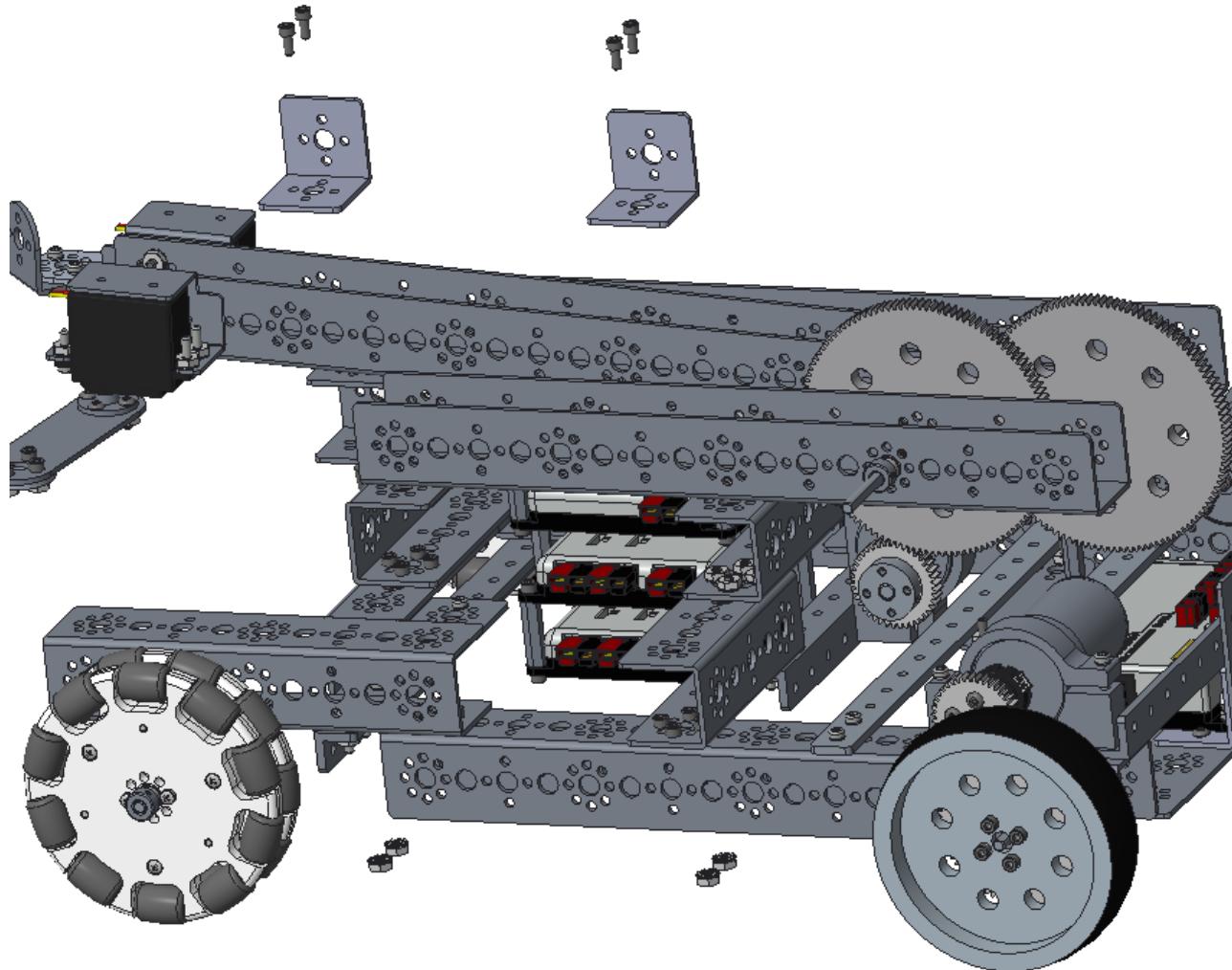


100: Completed Step 6 of Adding the Hands

Rotate the arm to a convenient position, align the arm motor gear with the large gear, and then tighten the arm motor screw, so the motor will no longer rotate. Do not mesh the motor gear's teeth too tightly with the large gear's teeth. Experimentation later (after the wiring phase) can be done to optimize the meshing of the gears.

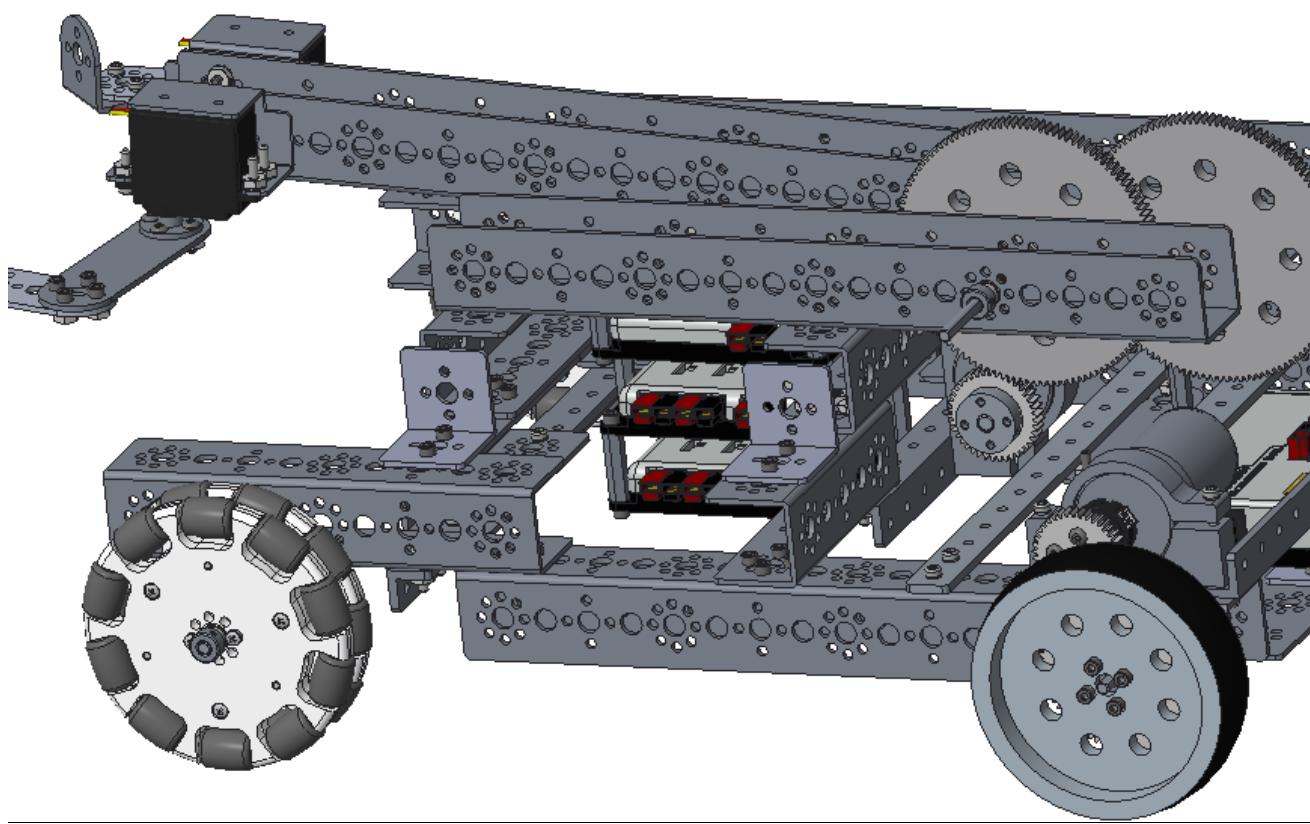
Adding the Phone and Flag Holder Assembly

Step 1: Parts: chassis, inside corner brackets (2), 5/16" socket head cap screws (4), keps nuts (4)

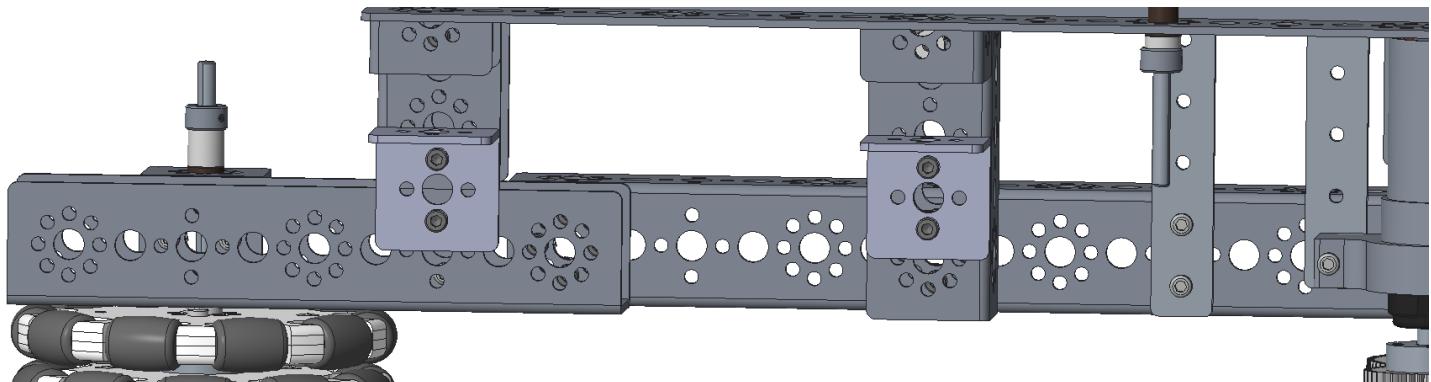


101: Parts needed in Step 1 of Adding the Phone and Flag Holder Assembly

The next page shows the completed step from two different vantage points.

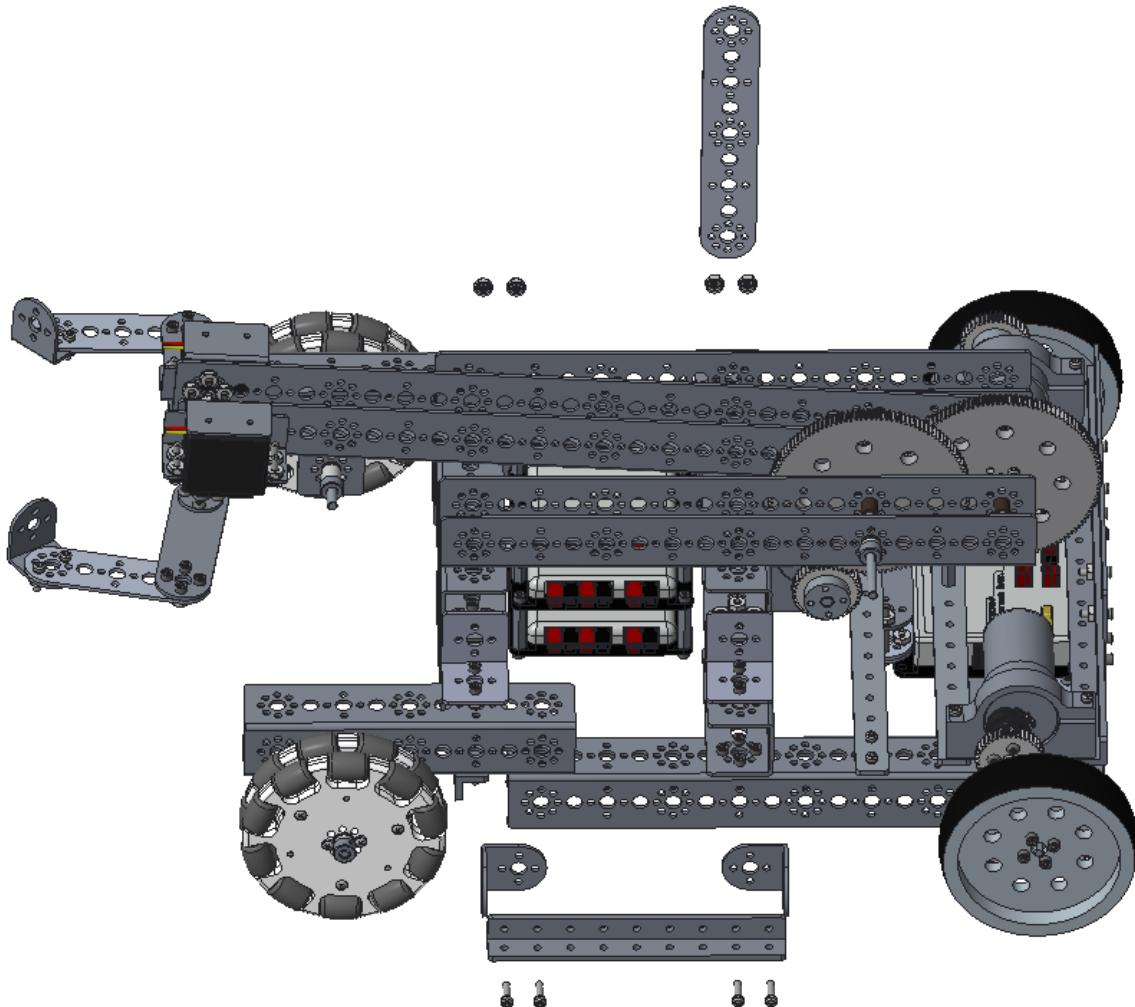


Note that wiring has been hidden in these images.



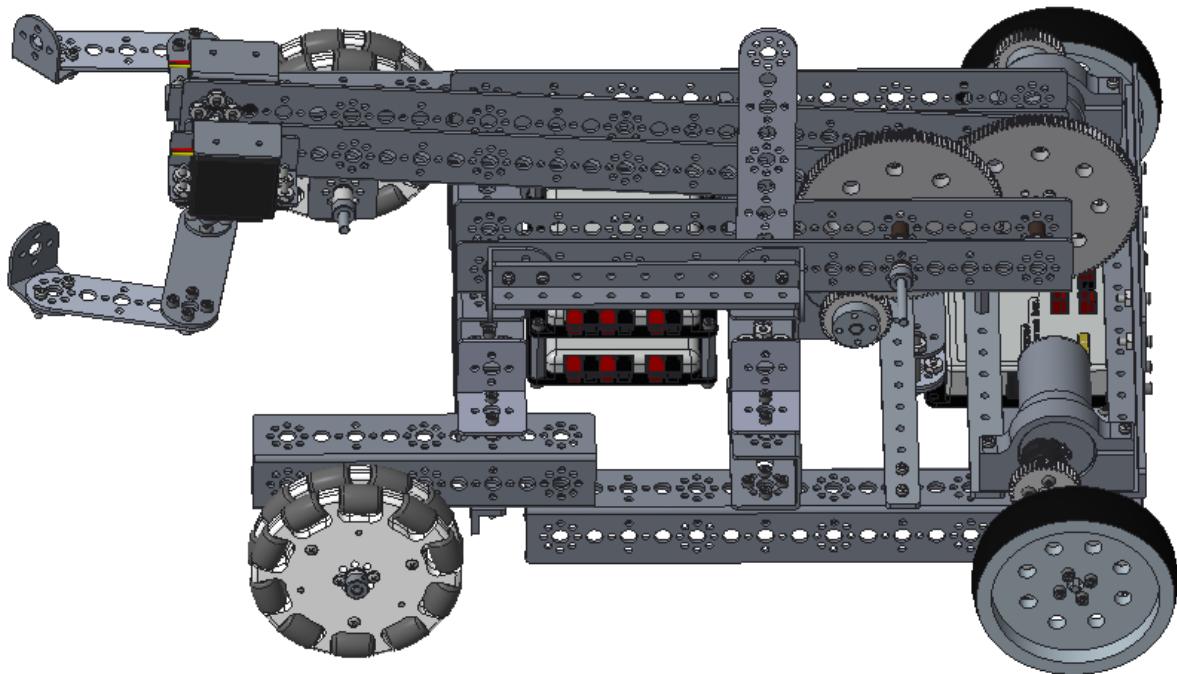
103: Completed Step 1 of Adding the Phone and Flag Holder Assembly right side view top image and top view of right chassis rail in lower image

Step 2: Parts: 144 mm angle (1), L-brackets (2), 160 mm flat (1), 1/2" socket head cap screws (4), keps nuts (4)

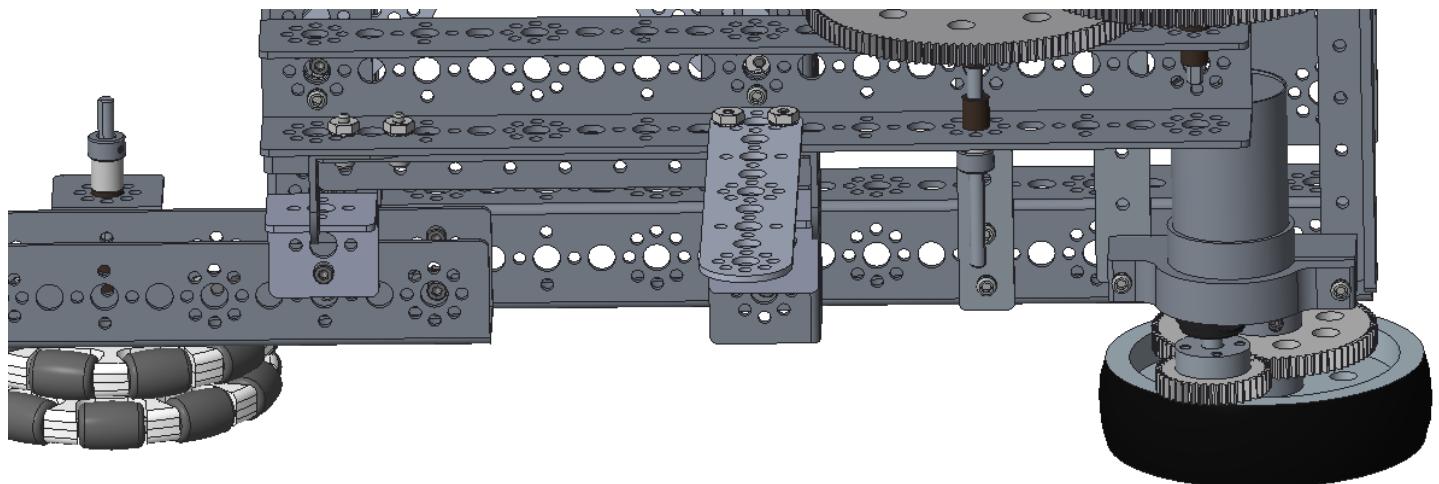


104: Parts needed in Step 2 of Adding the Phone and Flag Holder Assembly

The next page shows the completed step from two different vantage points.

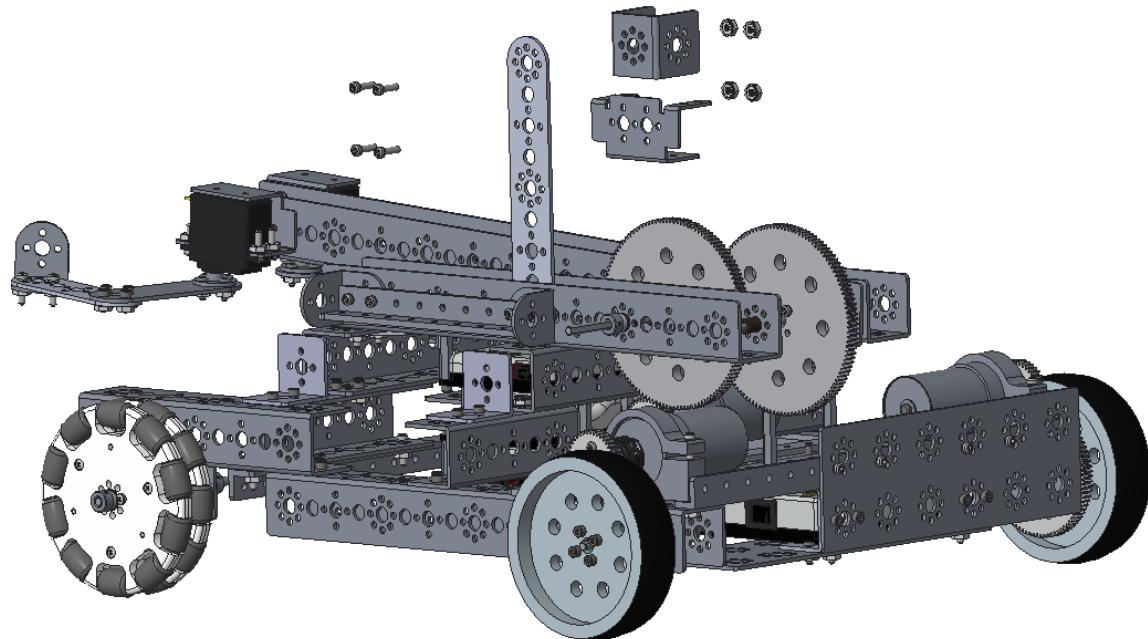


Note that the bolts on the right side hold the angle and L-bracket onto the outside of the channel and the flat on the inside of the channel.

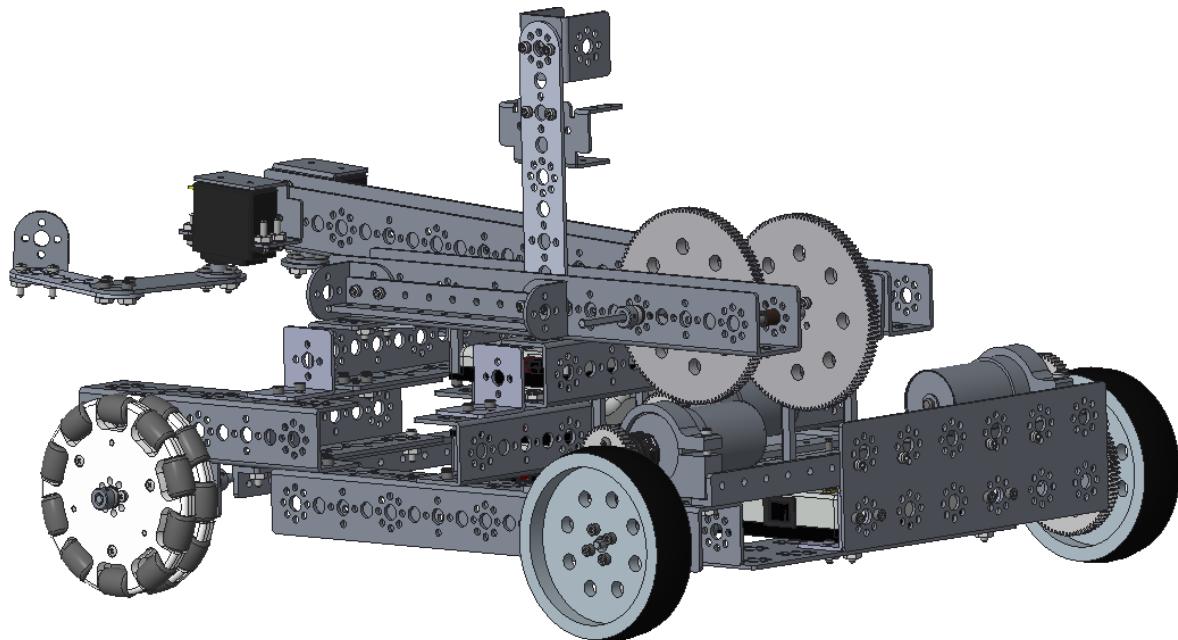


105: Completed Step 2 of Adding the Phone and Flag Holder Assembly right side view in top image and top view of right channel in lower image

Step 3: Parts: single standard-scale servo motor mounting brackets (1), 32 mm channel (1) 1/2" socket head cap screws (4), keps nuts (4)



106: Parts needed in Step 3 of Adding the Phone and Flag Holder Assembly

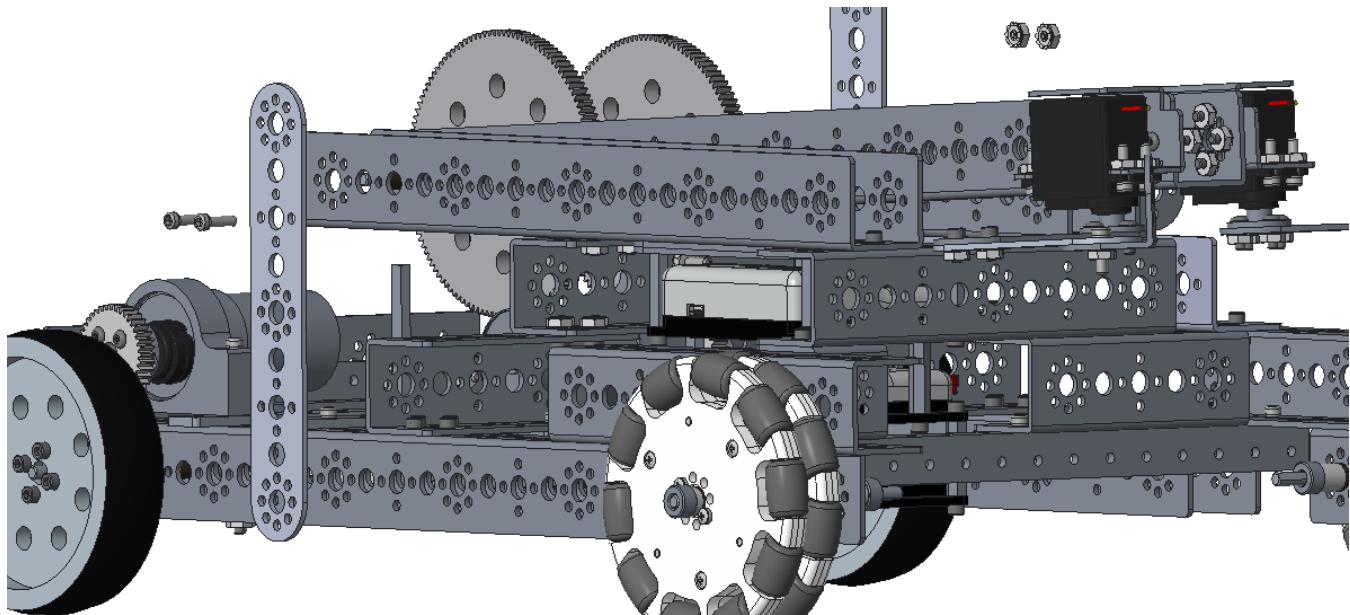


107: Completed Step 3 of Adding the Phone and Flag Holder Assembly

Adding the Cable Holder

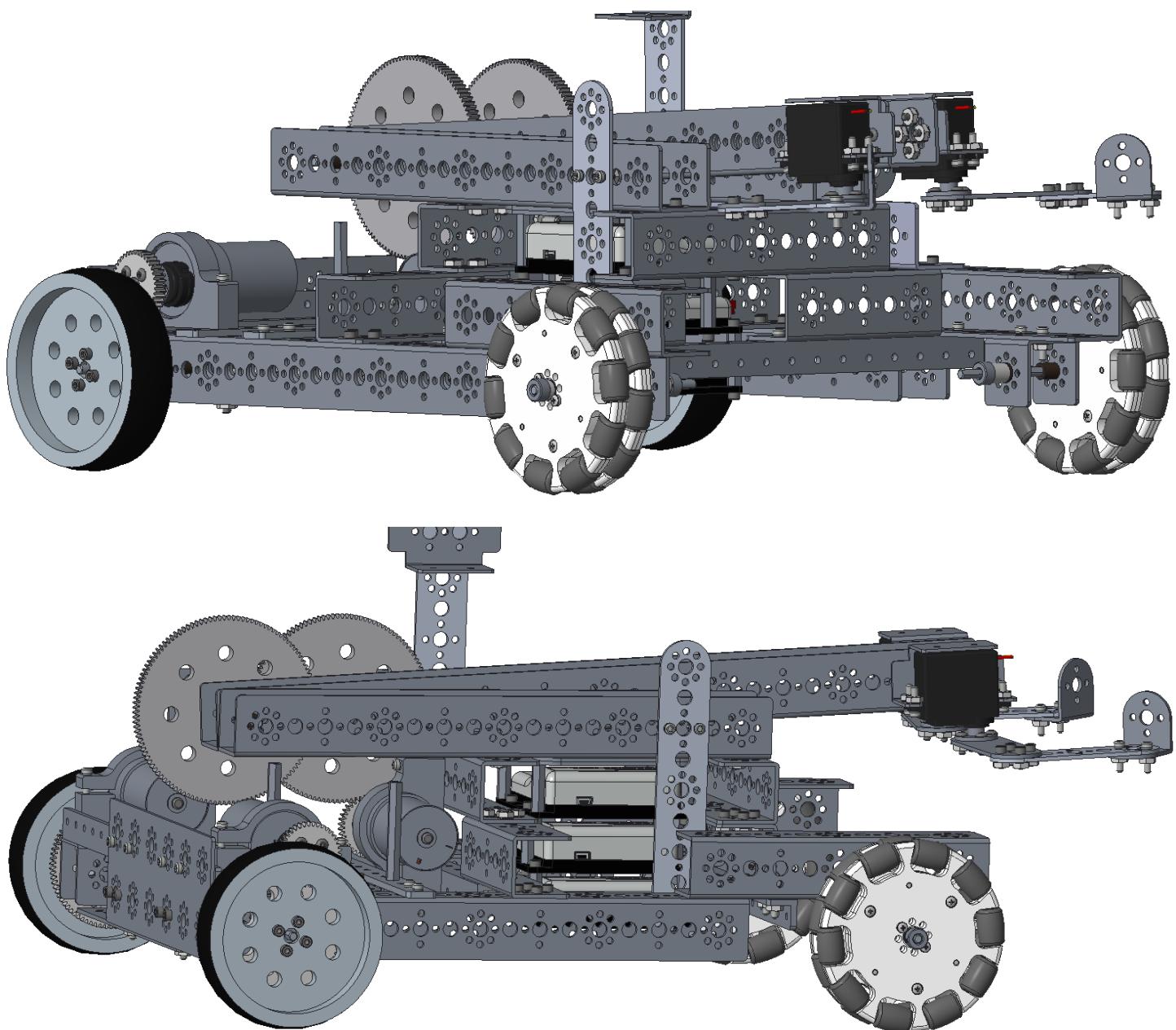
The cable holder provides a convenient location to secure the USB cables from the control modules. By zip tying them to the holder, they are less likely to come loose as the robot is driving.

Step 1: Parts: 160 mm flat (1), 1/2" socket head cap screws (2), keps nuts (2)



108: Parts needed in Step 1 of Adding the Cable Holder

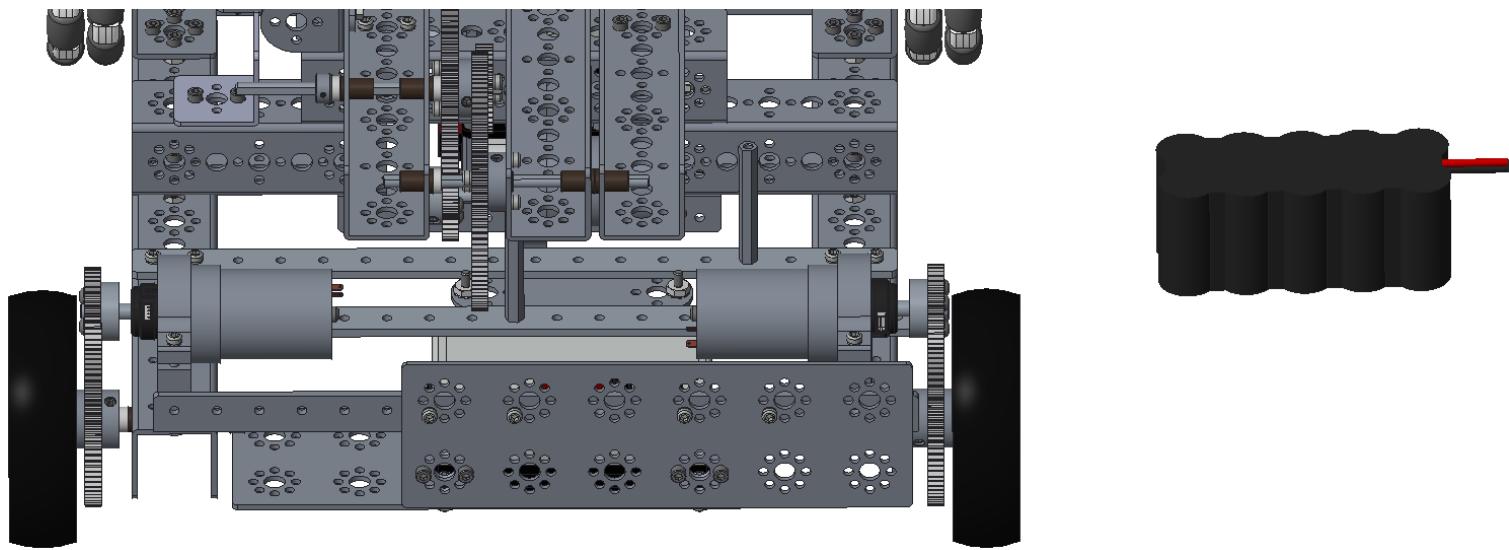
The next page shows the completed step from two different vantage points.



109: Completed Step 1 of Adding the Cable Holder top image is a view from the rear left of the robot. The lower image is a view of the left side.

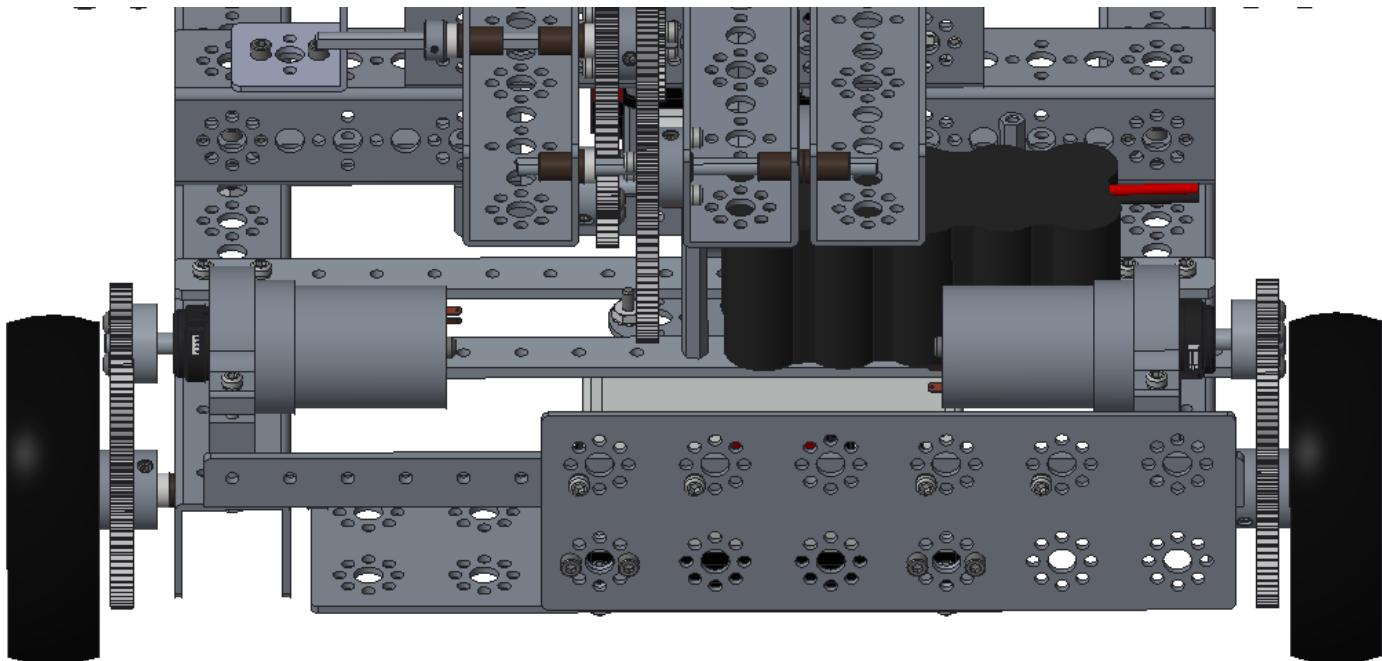
Adding the Battery

Step 1: Parts: chassis, 12-volt Rechargeable NiMH Battery



110: Parts needed in Step 1 of Adding the Battery

Place the battery on the 288 mm flat bars. The two inch standoffs will help keep the battery in place. Use a zip tie (not shown) around the center of the battery and the two flat bars to secure the battery to the bars.



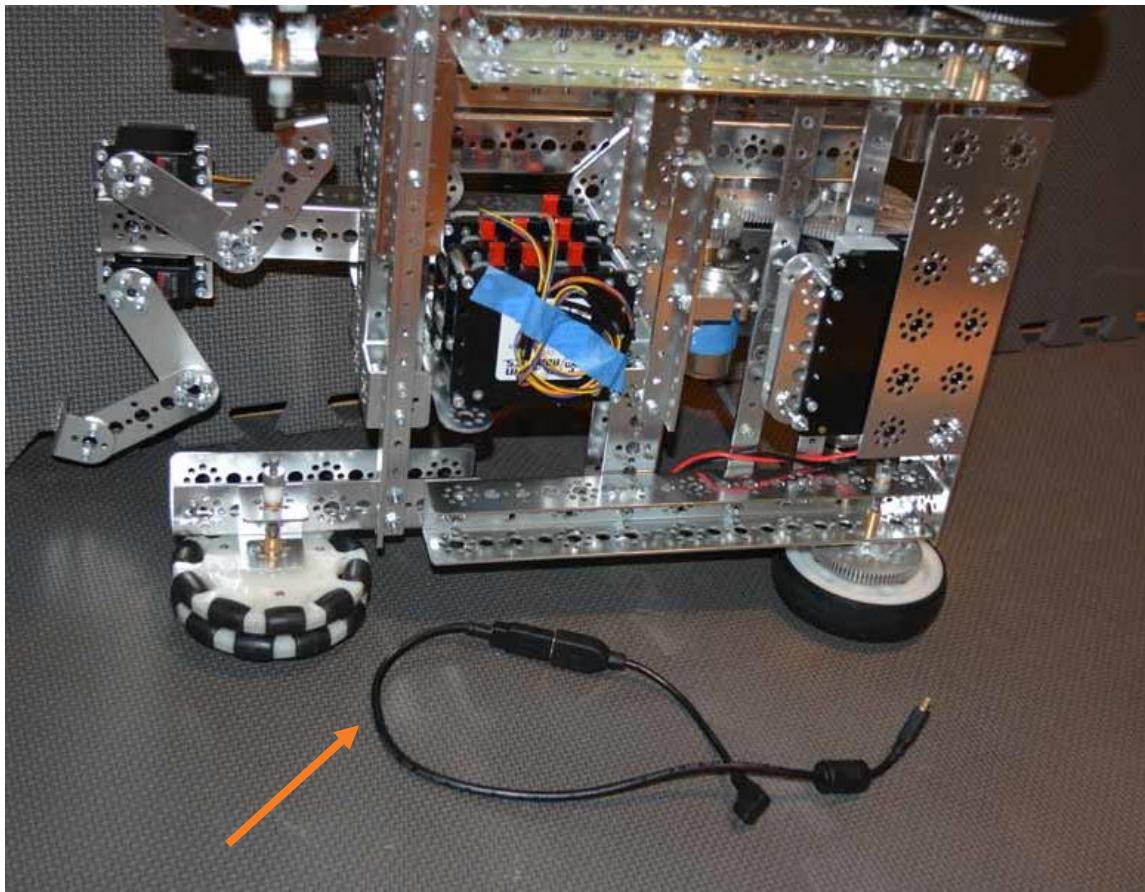
Wiring the Robot

A Robot is made up of more than just the metal pieces you have now assembled. In order for your Robot to function, you must properly wire (creating connection between power sources, motors, servos, and moving parts).

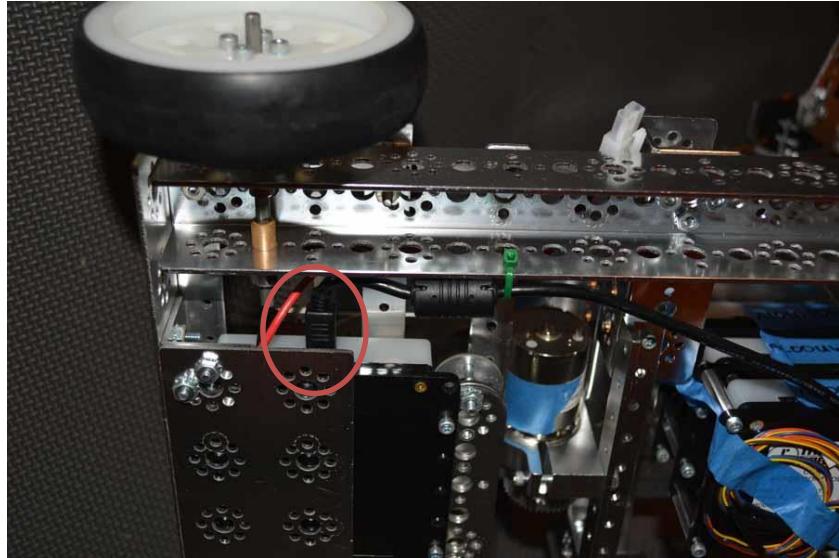
Securing loose wires is strongly recommended. Securing the wires as the wires are being routed or securing all of the wires at the end of this section are both good options. Secure wires are less prone to be caught in gearing, less prone to be entangled with another robot during a match, and make the robot look more organized. Properly secured wires also support the connectivity of the Robot. Poor Robot performance is often due to unsecured wires. This document shows the use of zip ties. Find more tips through our [5 Tips for Great Events](#) Video.

Wiring between the Phone and the Power Distribution Module (PDM)

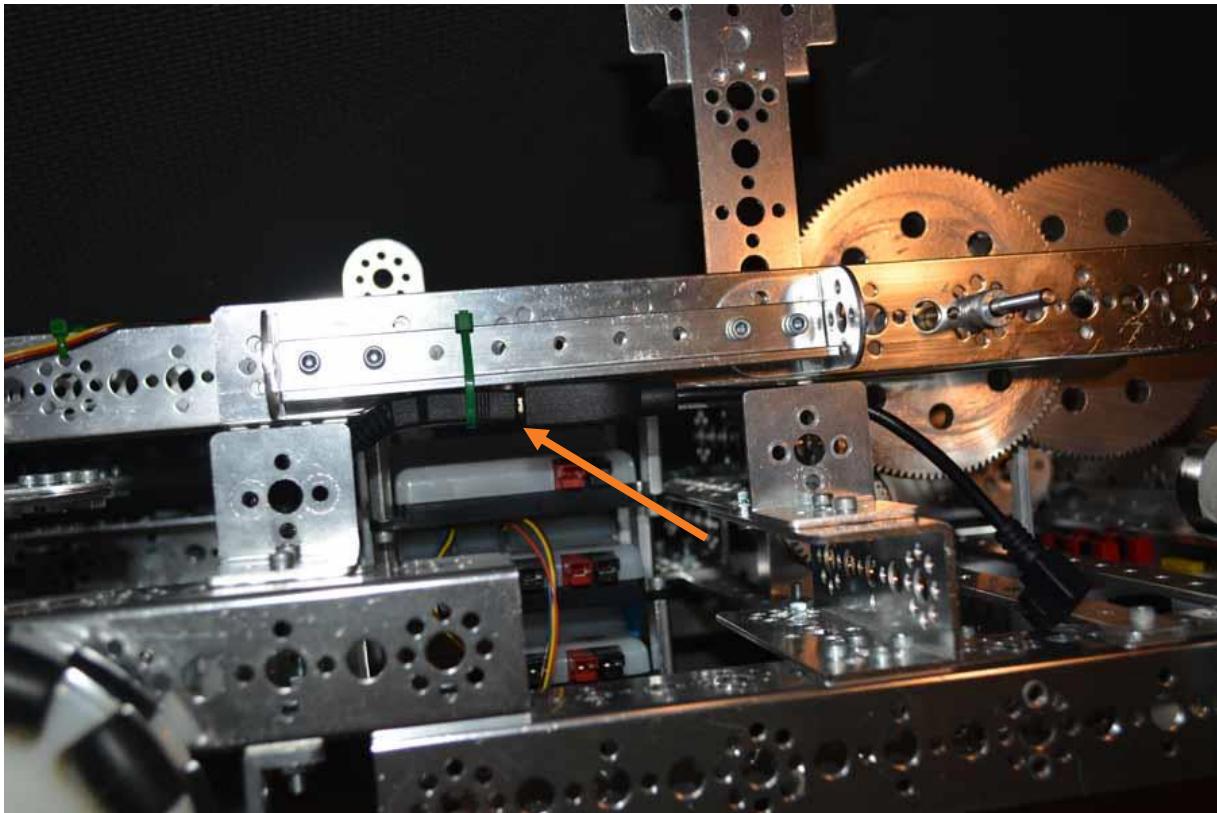
Step 1: Locate the cable(s) that will connect the PDM to the cell phone. Cables shown in the images in this section may vary from those in the Kit of Parts.



Step 2: Connect one end of the cable to the PDM.

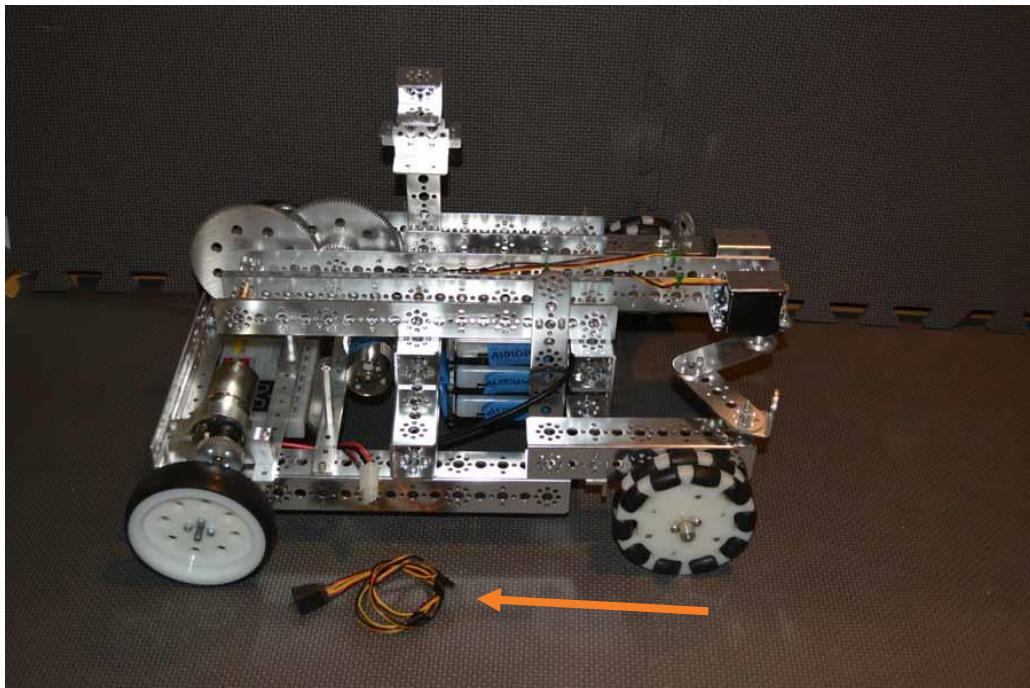


Step 3: Route the cable to behind the phone holder. Secure it to the 144 mm angle.

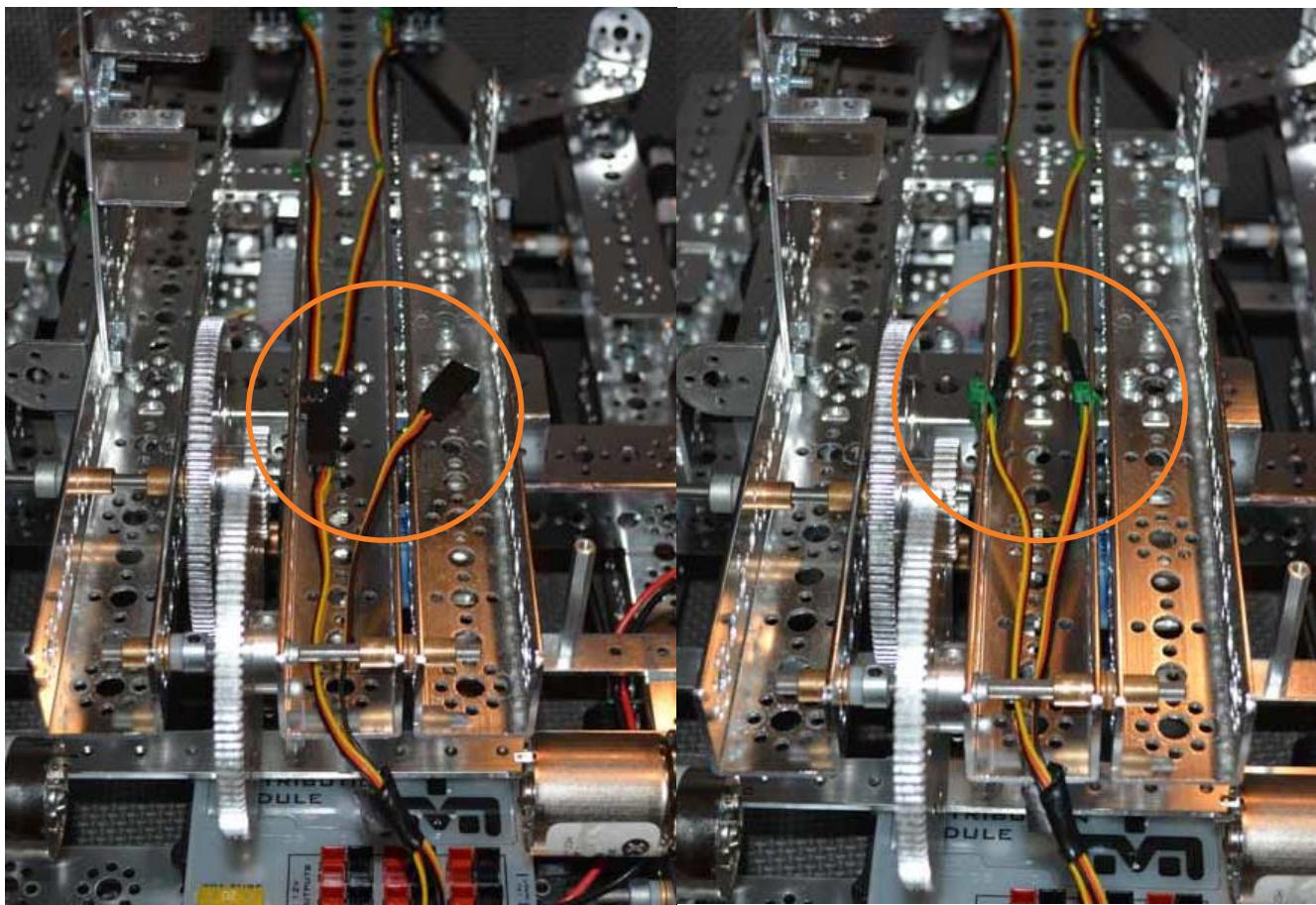


Wiring the Hands to the Servo Controller

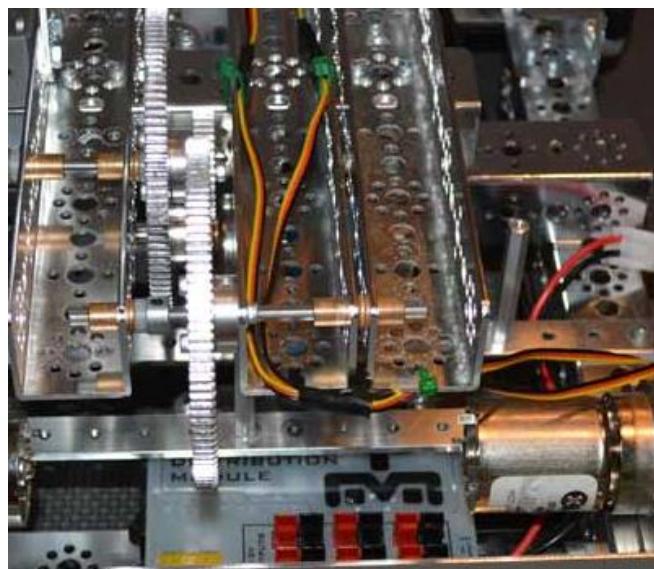
Step 1: Locate the servo extension wires.



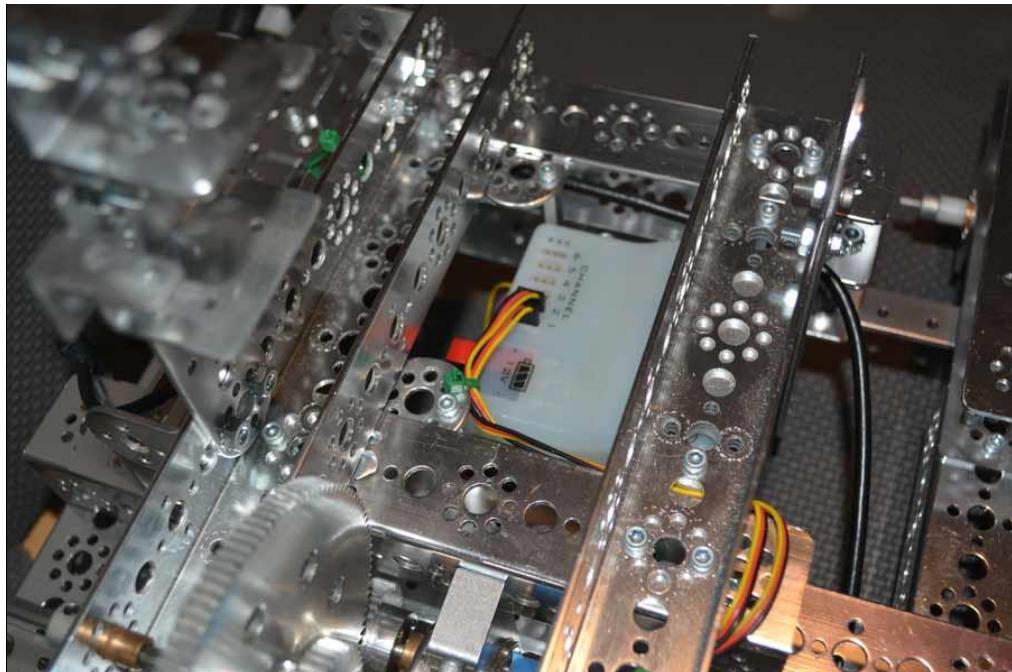
Step 2: Connect the extensions to the cables coming from the servos.



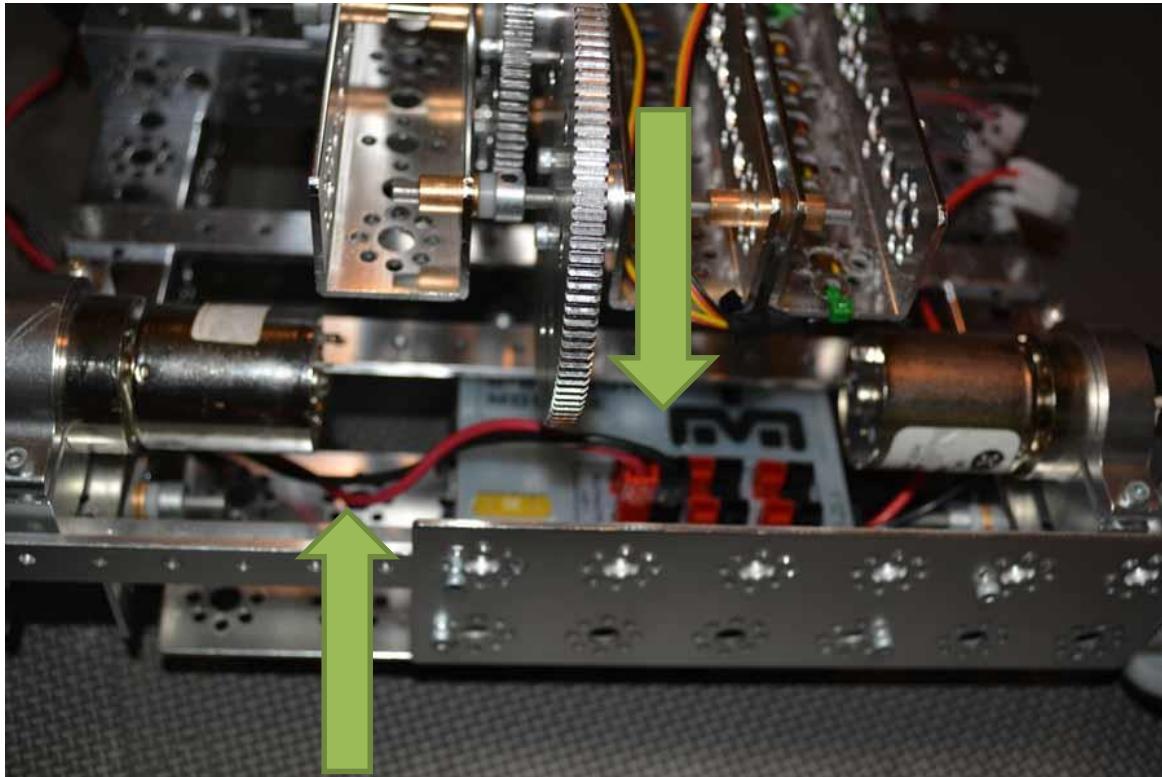
Step 3: Route the cables from the movable portion of the arm to under the stationary channel of the arm. Wrapping electrical tape around the cables will protect them from the damaging effects of rubbing the channel as the arm is being extended and retracted. Leave enough slack so the cables will not be stretched when the arm is retracted and pull the tightest against the wires.



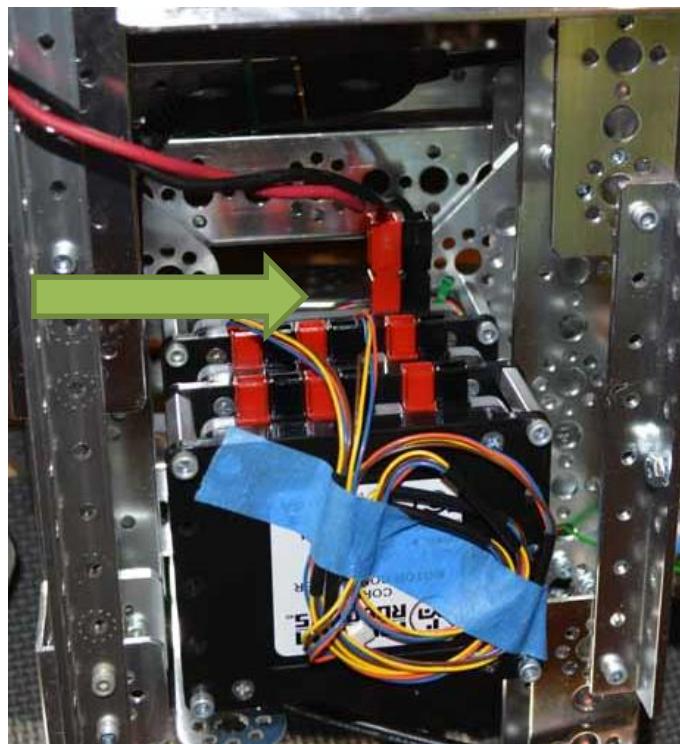
Step 4: Run the cables to the top of the controller stack and plug the cable for the left hand (as viewed from the top of the robot when the traction tires are furthest away and omni-wheels are closer) into port 1. Plug the other cable into port 2.



Step 5: Plug one end of a power cable into the PDM and run the other under the flat beam.

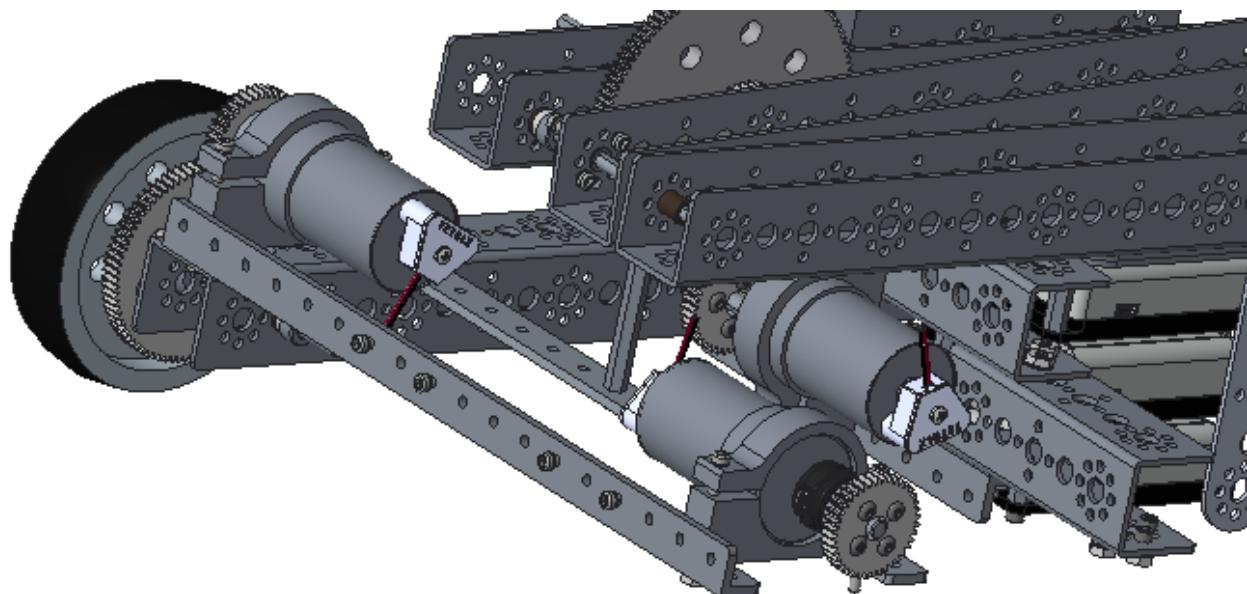
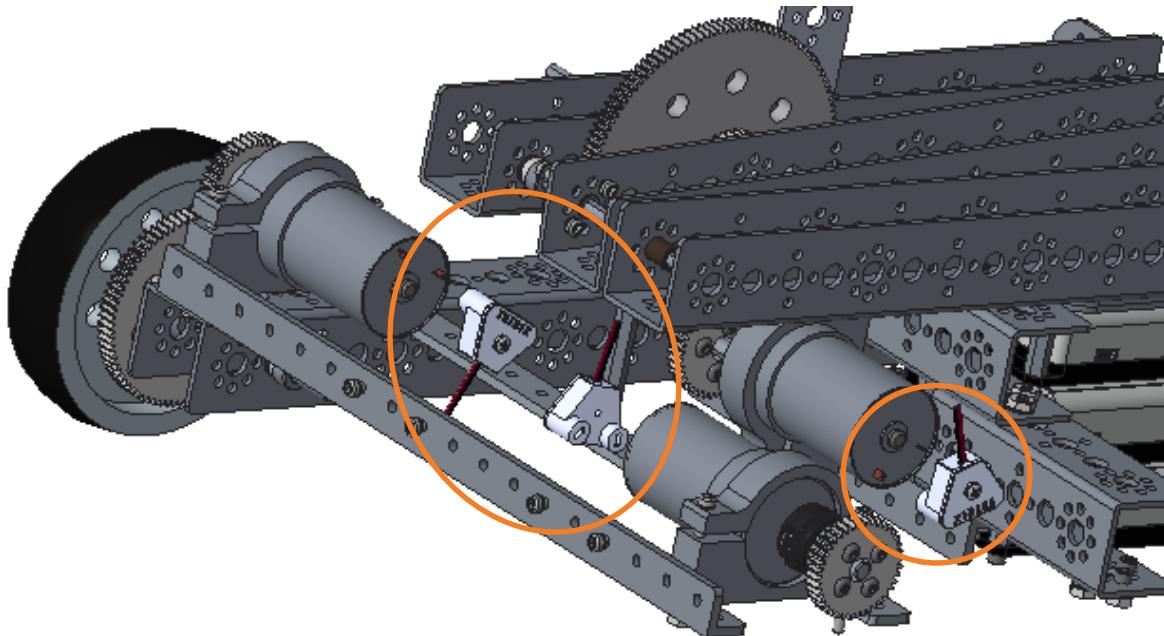


Step 6: Attach the other end into the power port of the servo controller.

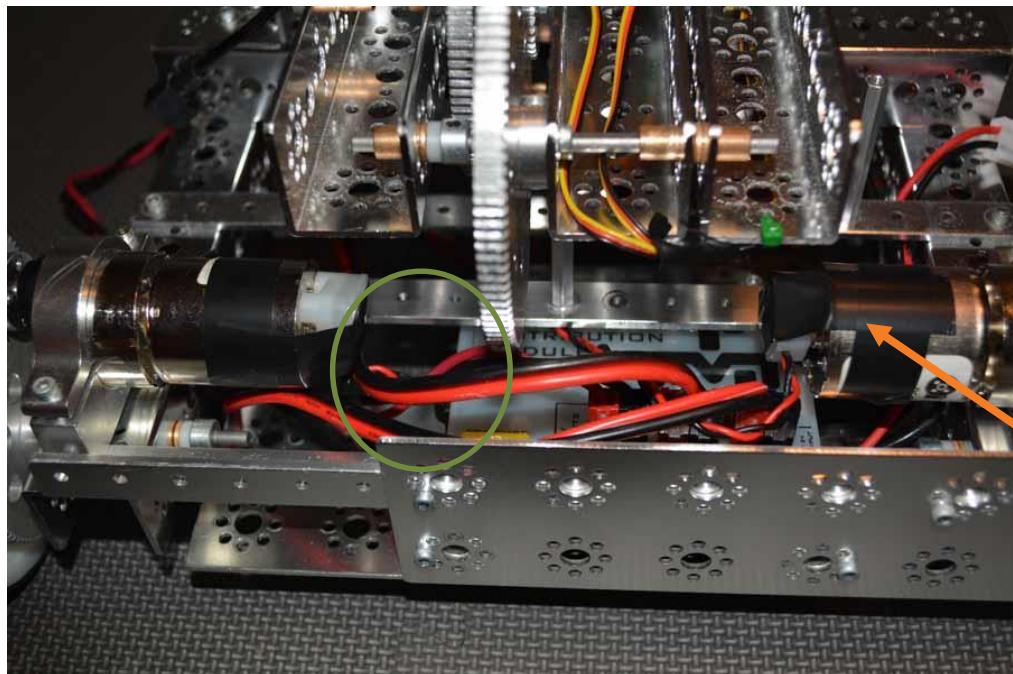


Wiring the DC Motor Cables

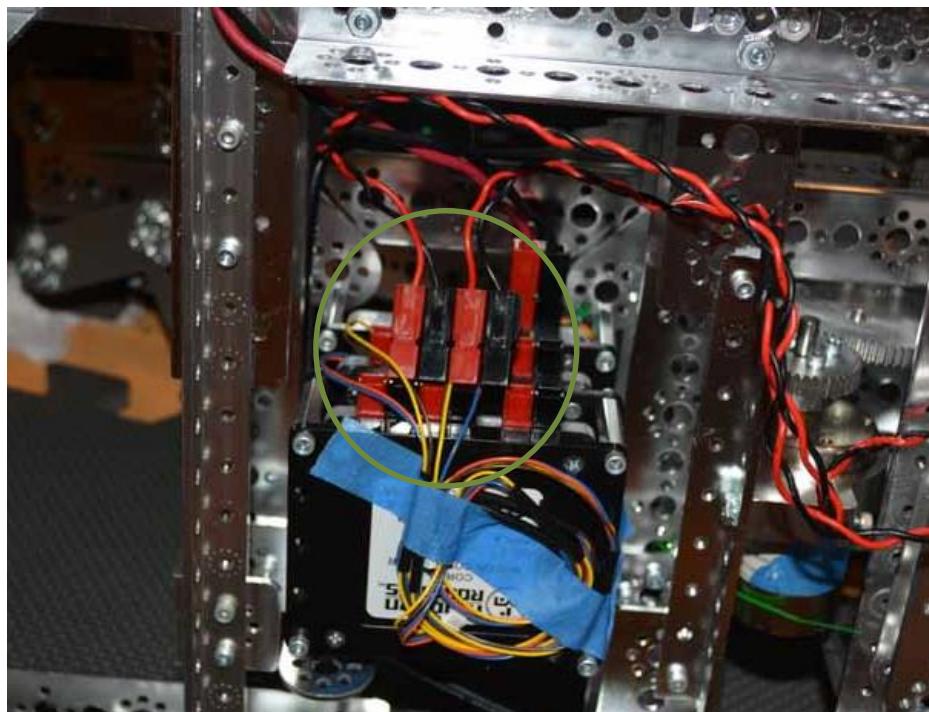
Step 1: Locate and attach the power cables for all three DC motors.



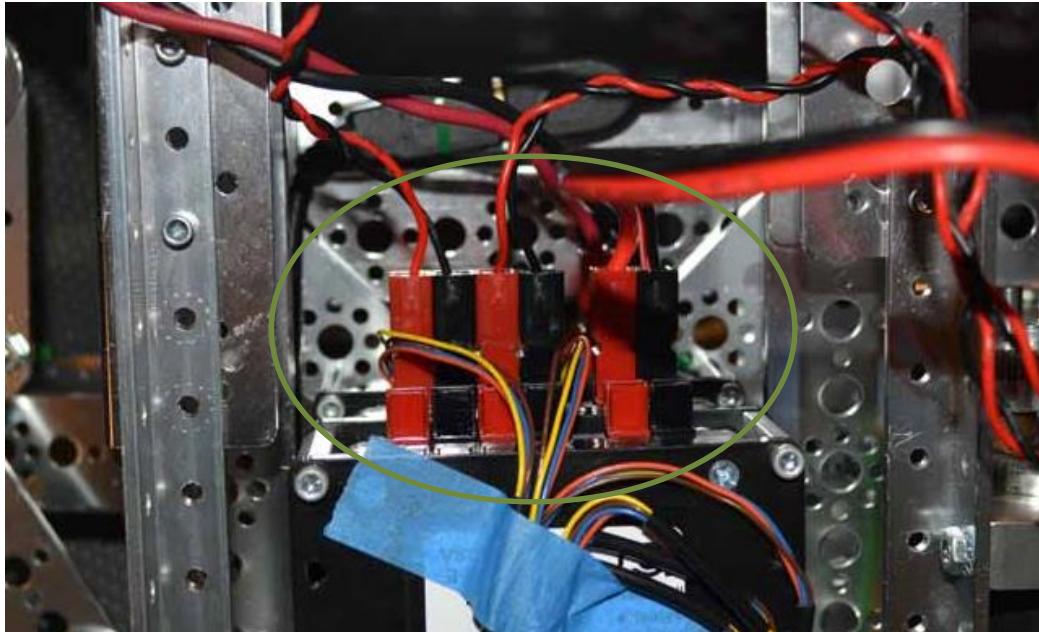
Step 2: Secure the DC motor cables to the motors using electrical tape, if some is available. Plug one end of a power cable into the PDM (shown below into the right bottom power port). Run the other ends under the flat beam (green circle).



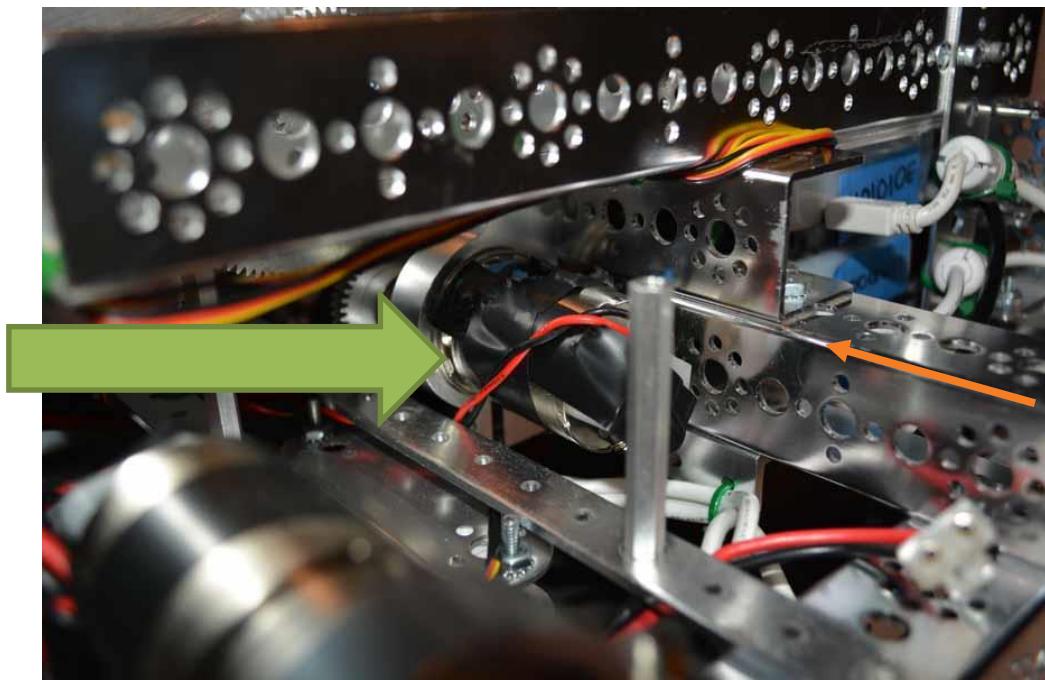
Step 3: Connect the left and right drive motor power cables to each other. Make sure that the left motor's cable is on the left and the right motor's cable is on the right.



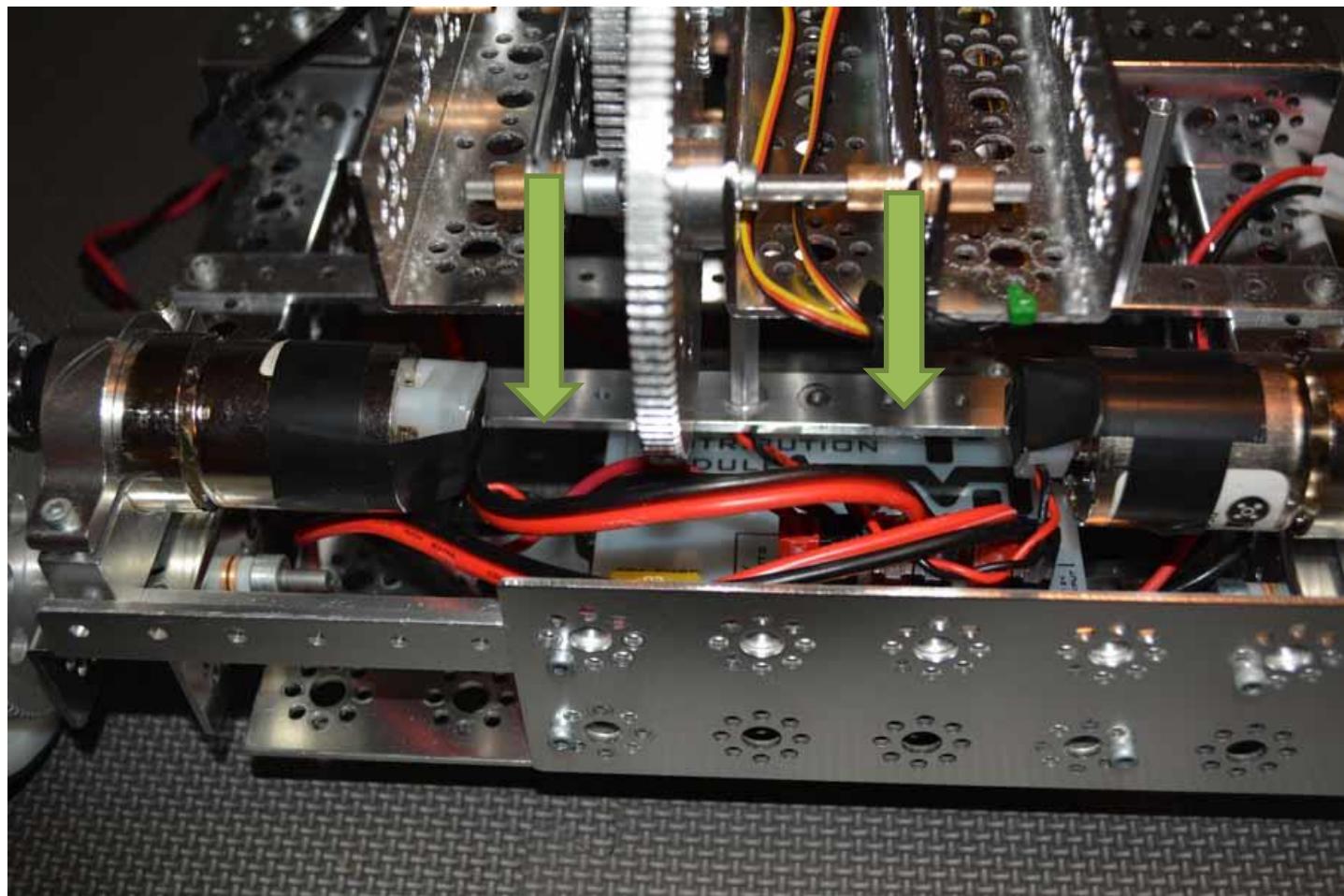
Step 4: Working with the center controller of the controller stack (the one with the encoders) connect the power cables from the motors into the ports on the left. Connect the power cable coming from the PDM to the port on the right.



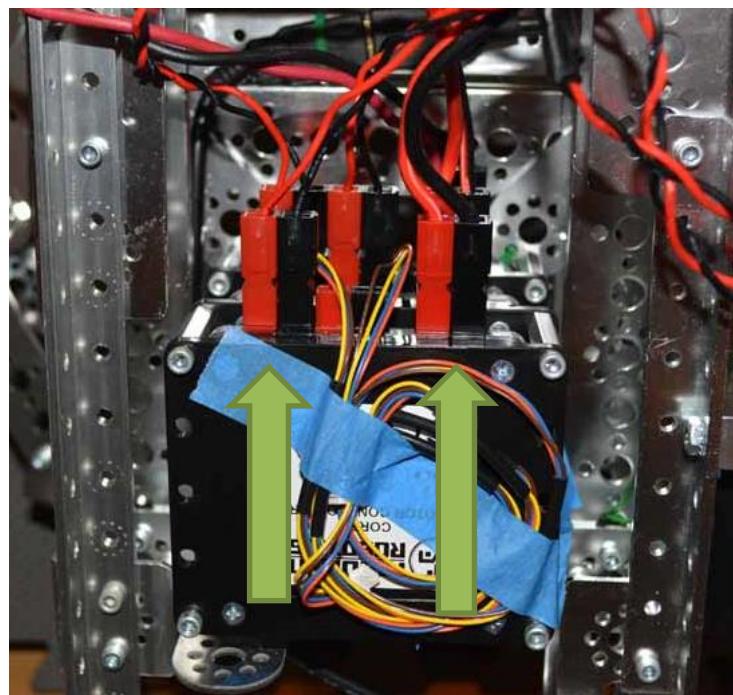
Step 5: Secure a DC motor cable to the arm motor using electrical tape, if some is available.



Step 6: Plug one end of a power cable from the DC motor into the PDM. Run the other end under the flat bar.

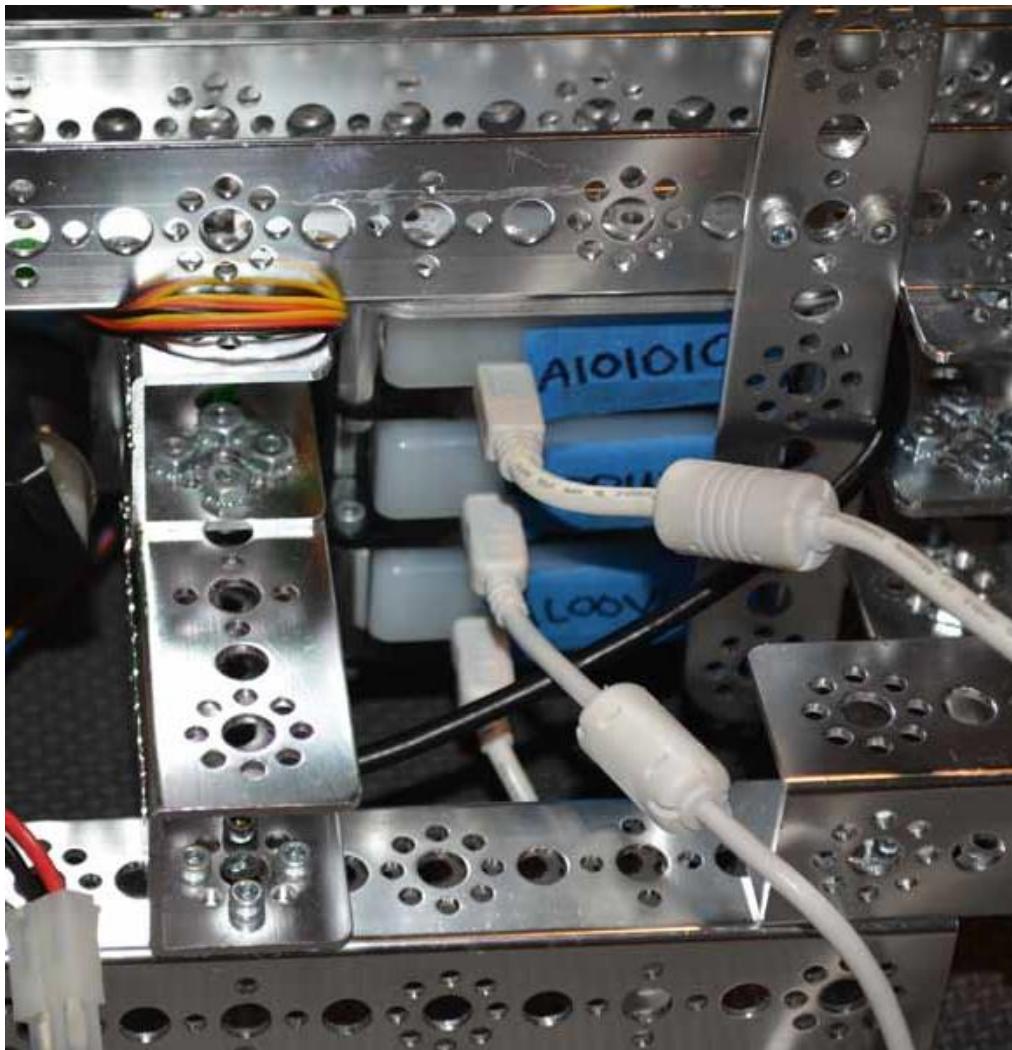
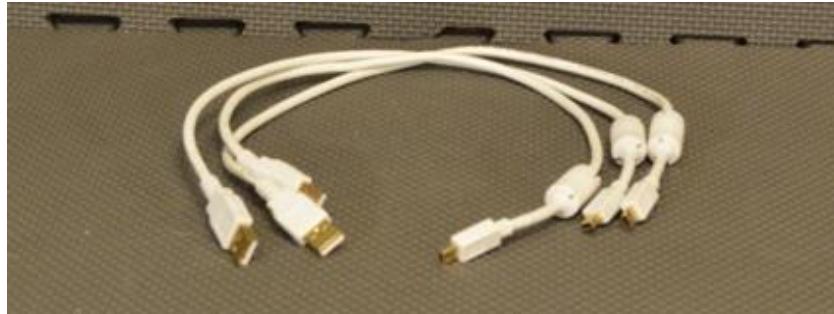


Step 6: Working with the arm controller, plug the DC motor cable into the left port. Plug the power cable from the PDM into the right port.



Wiring between the Controller Stack and PDM

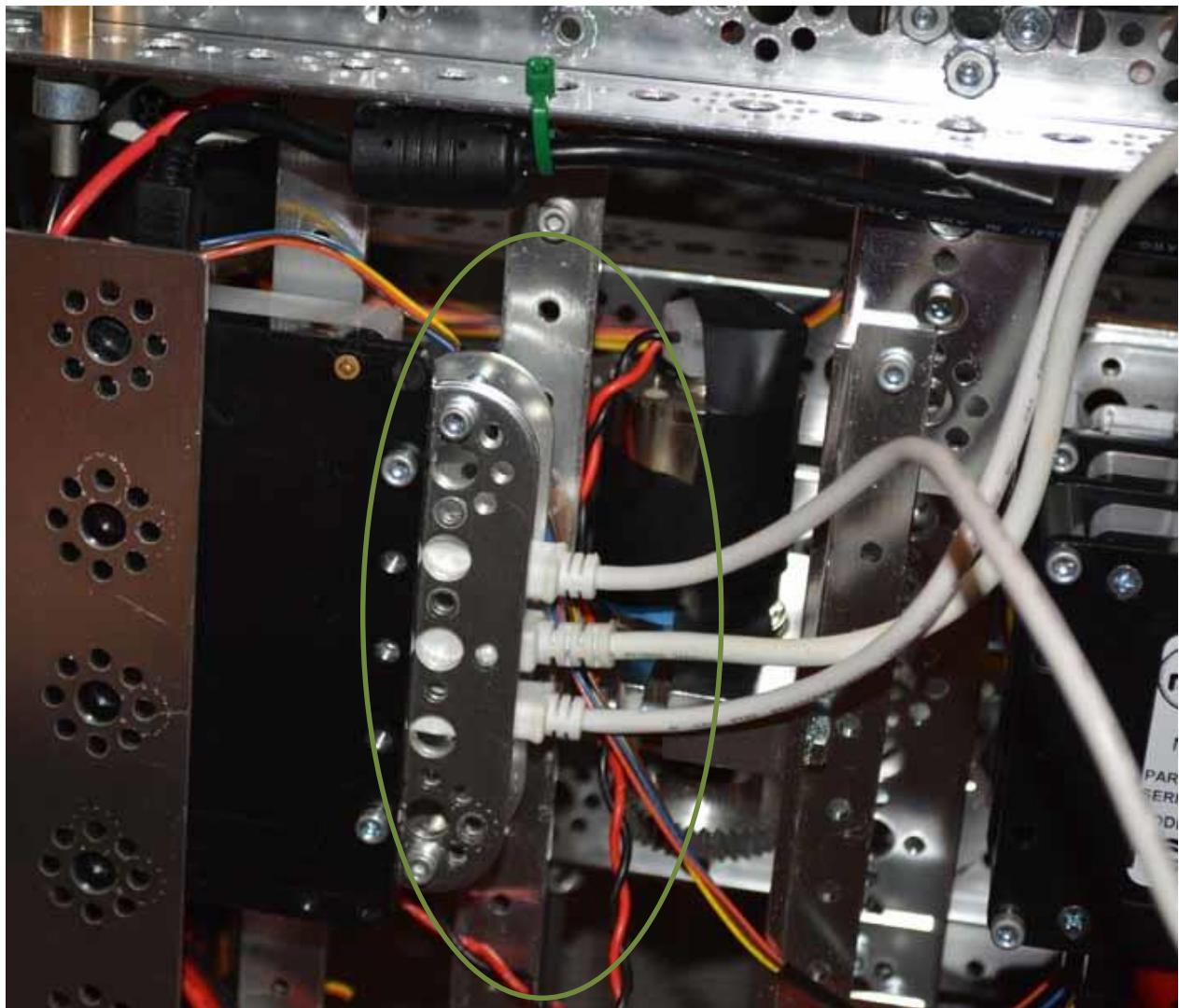
Step 1: Locate the three data cables that will connect each controller to the PDM. Plug the micro end into the controllers.



Step 2: Secure the ferrite choke onto the 160 mm flat.

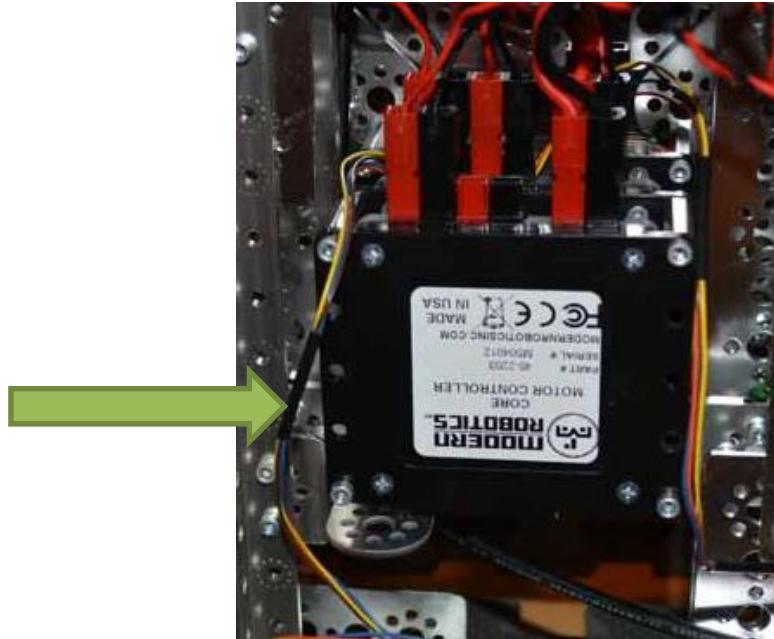


Step 3: Secure the other ends of the data cables to the PDM. Tighten the screws on the 96 mm flat brackets alternately until the screws are firm, but do not over tighten or the middle cable will become loose.

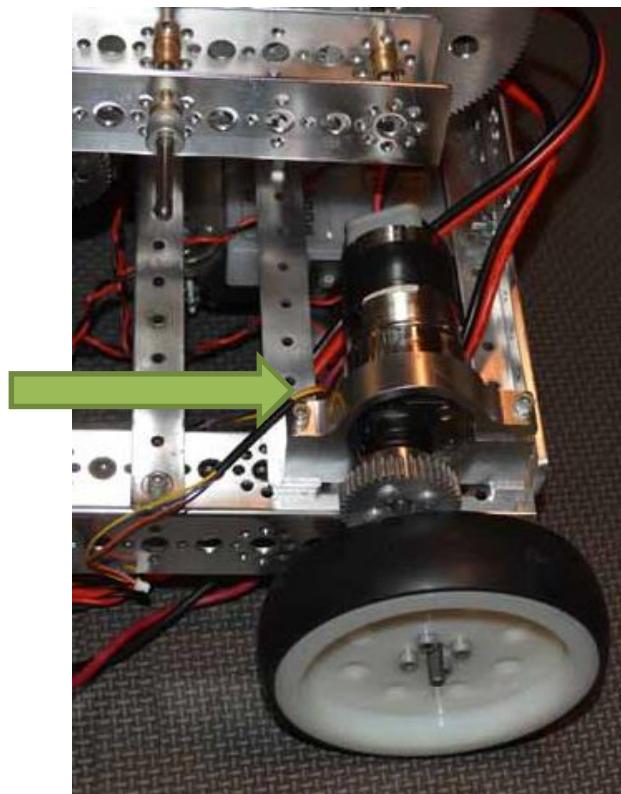


Wiring the Drive Encoders

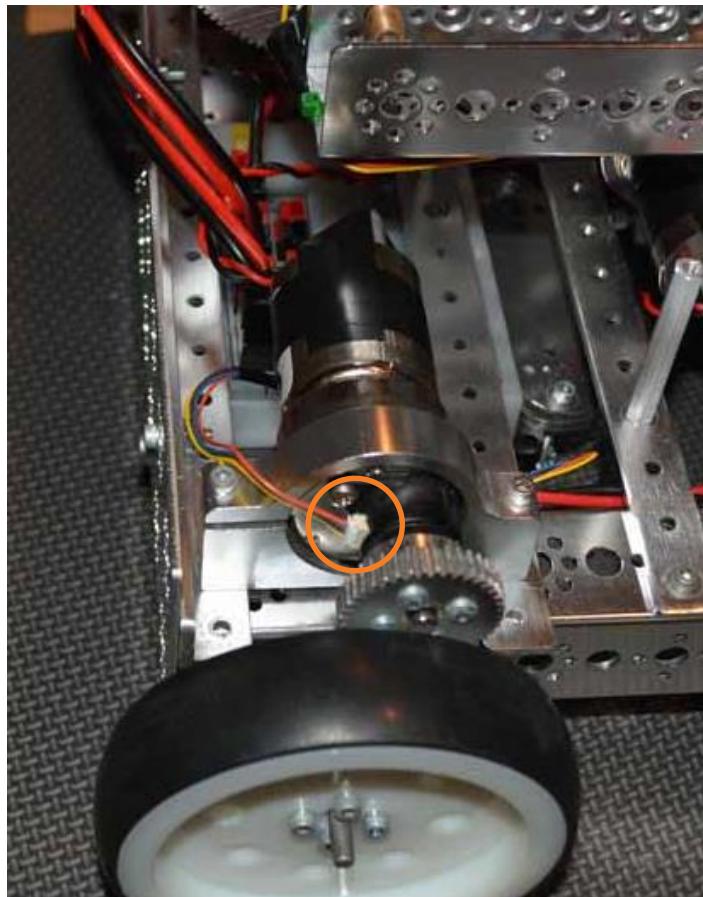
Step 1: Find the encoder wire attached to port 1. From the bottom of the robot, it will be the one on the left. It will be attached to the left drive motor in subsequent steps.



Step 2: Run the wire between the flat bar and the motor.



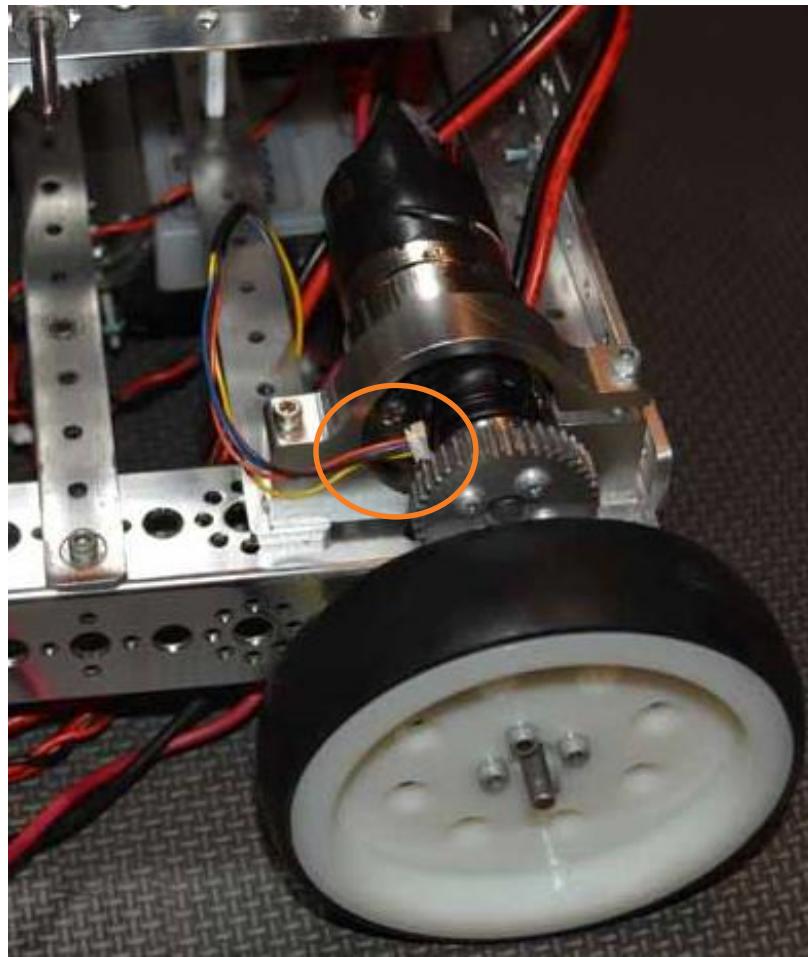
Step 3: Attach the encoder to the motor.



Step 4: Find the encoder attached to port 2. From the bottom of the robot, it will be the one on the right. It will be attached to the right drive motor in subsequent steps.

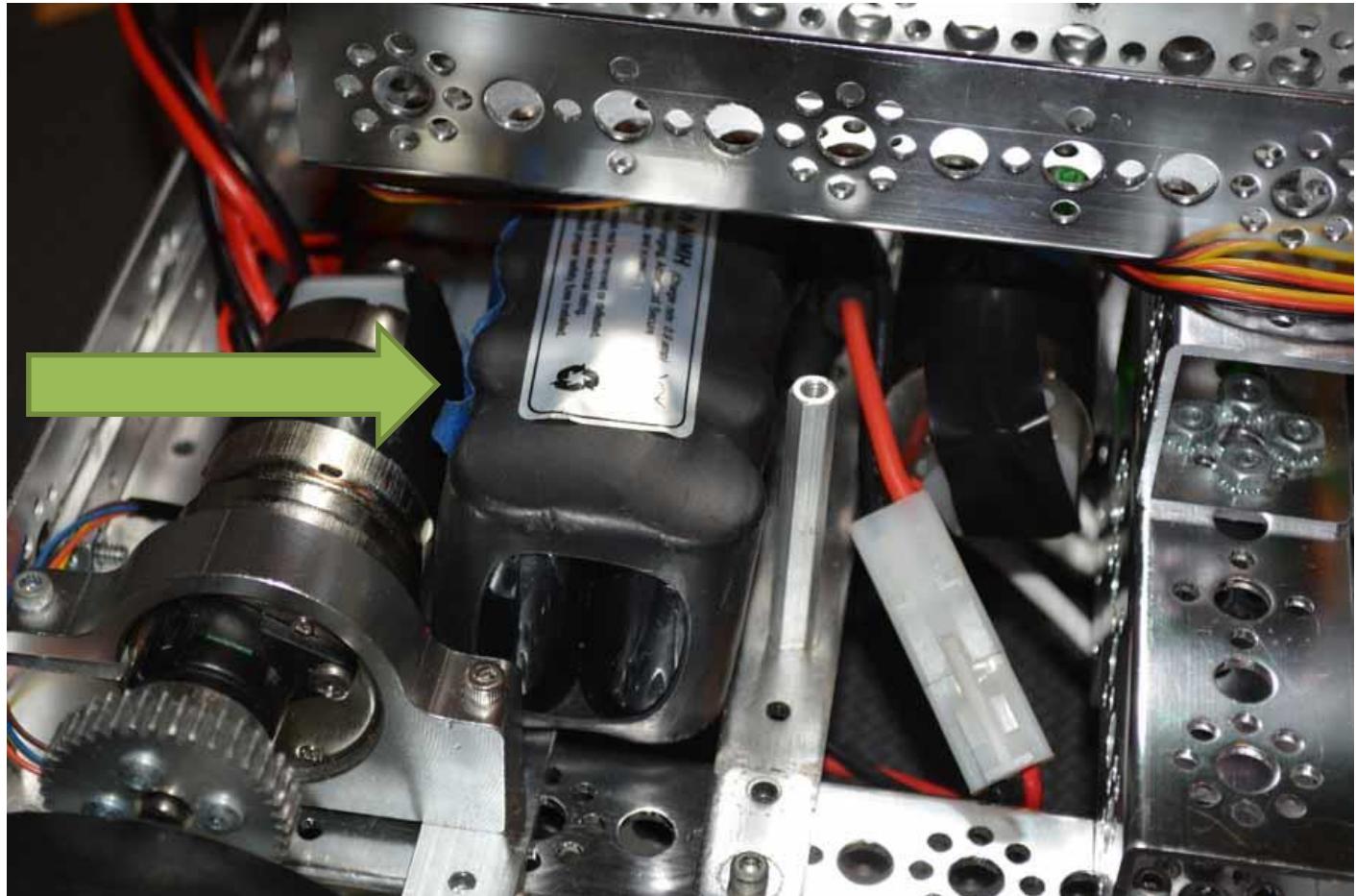


Step 5: Run the wire between the flat bar and the motor; attach the encoder to the motor.

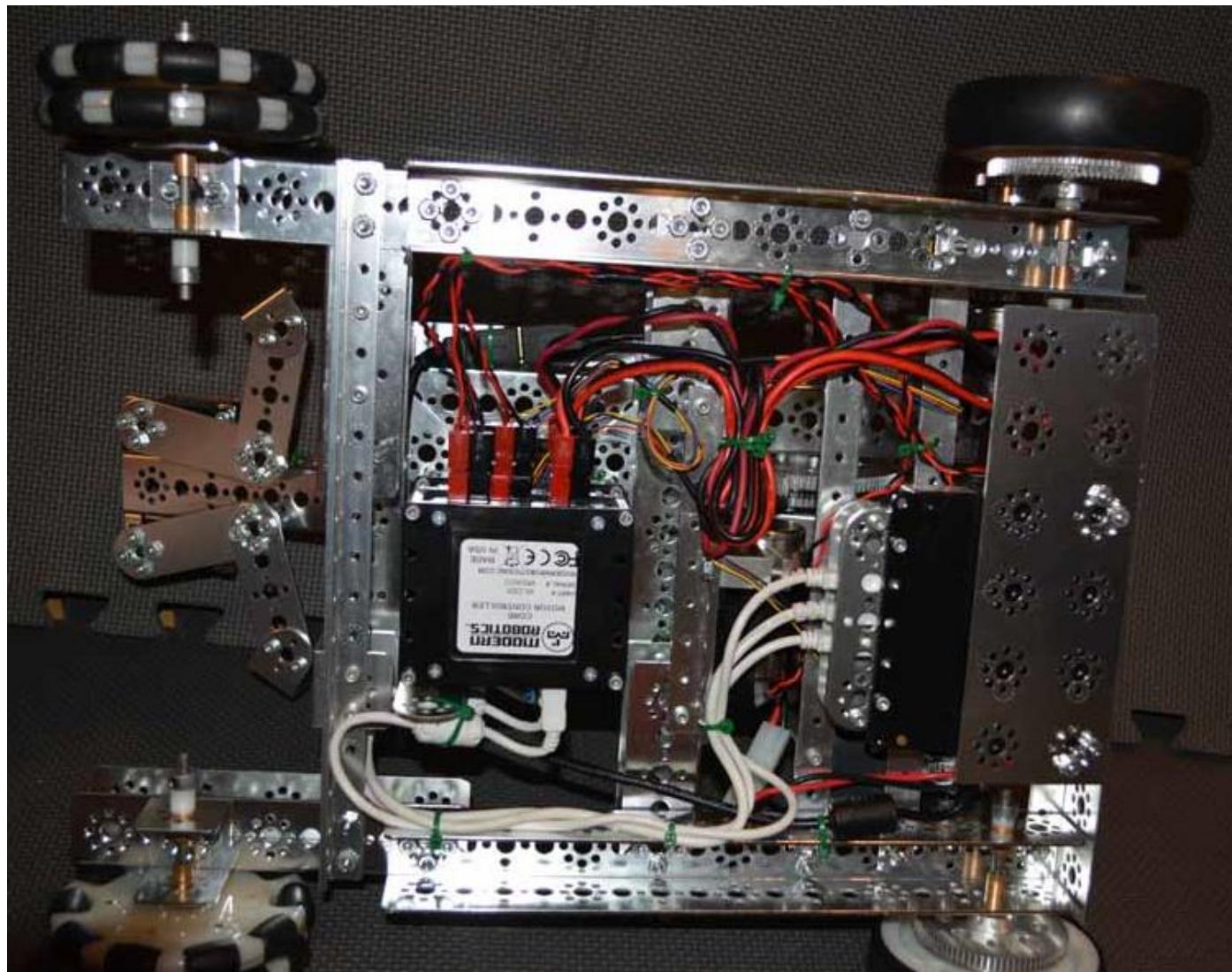


Installing the Battery and cleaning up wiring

Step 1: Place the battery on the two flat beams by the left motor. The 2" stand offs will help hold the motor in place. If available, use a zip tie to secure the battery to the two flat beams.



Step 2: Secure the wires to the chassis using zip ties, if some are available. Tie wires down to the robot structure to prevent them from getting caught in gearing or mechanisms. Also to prevent them from snagging on field elements or other robots.



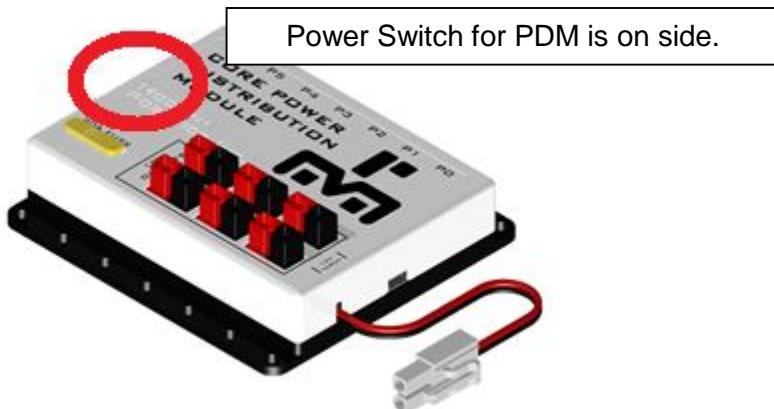
Configuring the Robot

This section will list the steps necessary to name the motors attached to the controllers.

Creating a configuration file

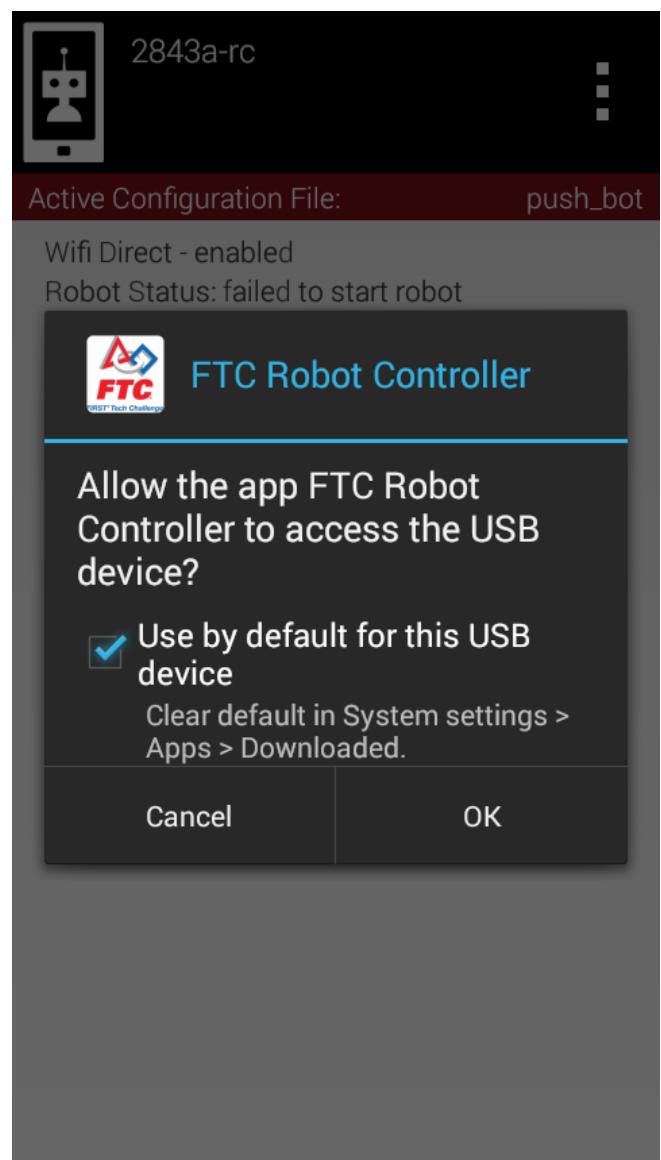
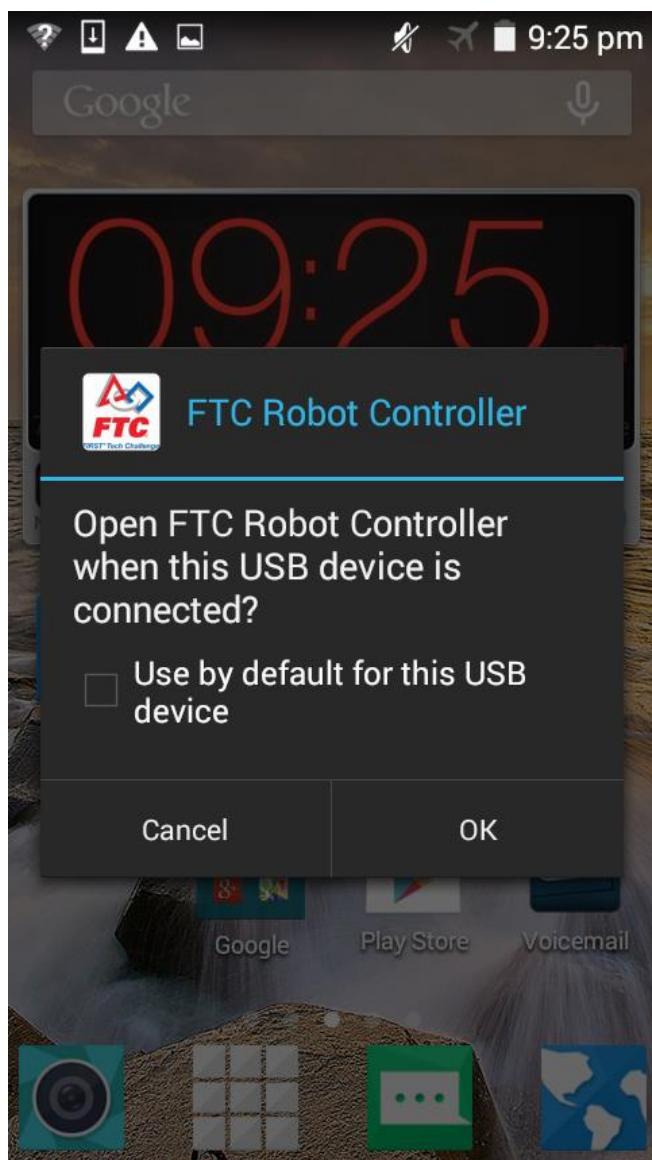
From [previous steps](#), the Driver Station and Robot Controller apps have been installed on the cell phones. The cell phones should be powered on and the FIRST Tech Challenge apps have been started. When the Robot Controller app configures the USB devices, the devices are assigned randomly – the order depends on which controller responds to the query first. This means that sometimes when the devices are first scanned, motor ‘A’ is named Motor Controller 1, Motor Controller 2 or Motor Controller 3 - the numbers are assigned 1 to the number of USB devices, not the number of motor controllers. This section will determine which motor controller operates the wheels and which operates the arm. Once the USB devices are named the serial number of the device is used to control communication, not the order in which the devices are discovered. The serial numbers should be recorded at this point (see Recording the Controller Serial Numbers). This section will list the steps necessary to name the motors attached to the controllers.

Step 1: Plug a fully charged battery into the Core Power Distribution Module (PDM). Set the PDM power switch to on. The switch is on the side away from the power cable.



Step 2: When the Core Power Distribution Module is powered on, the Robot Controller app should auto-start (the home screen is replaced by the FIRST Tech Challenge Robot Controller activity) and may prompt the user to “remember” whether it can auto-start when the core power distribution module is detected in the future (left image). Check the box to auto-start. Select ‘OK’. If this page is not being shown, then continue to the next step.

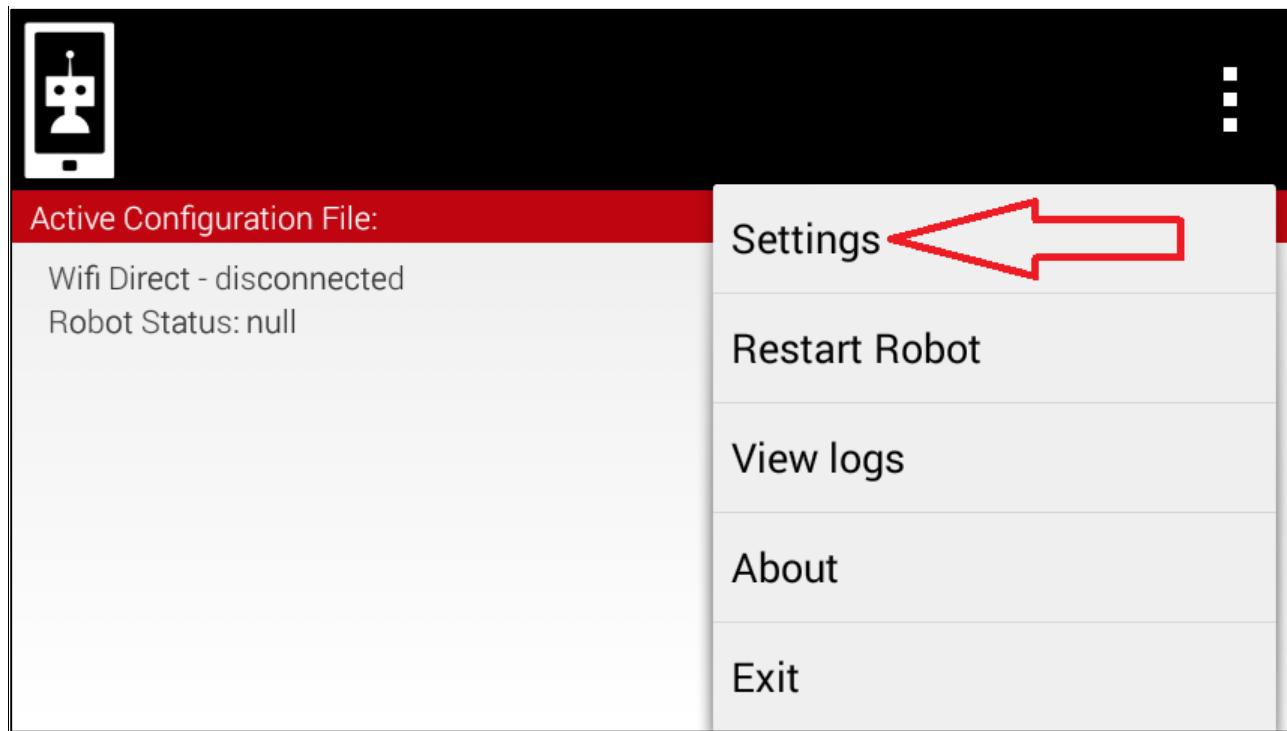
Sometimes the app also requests access to a USB device. If this screen is shown then select “Use by default” and then OK.



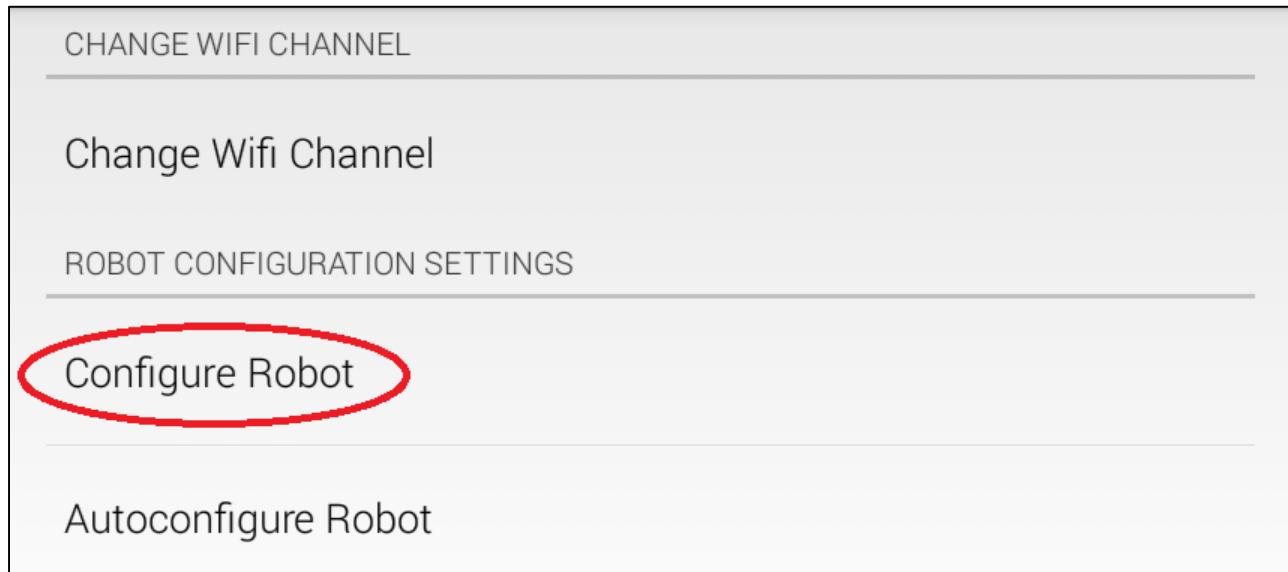
Step 3: Touch the overflow settings icon (three dots in the top right corner) to open a menu.



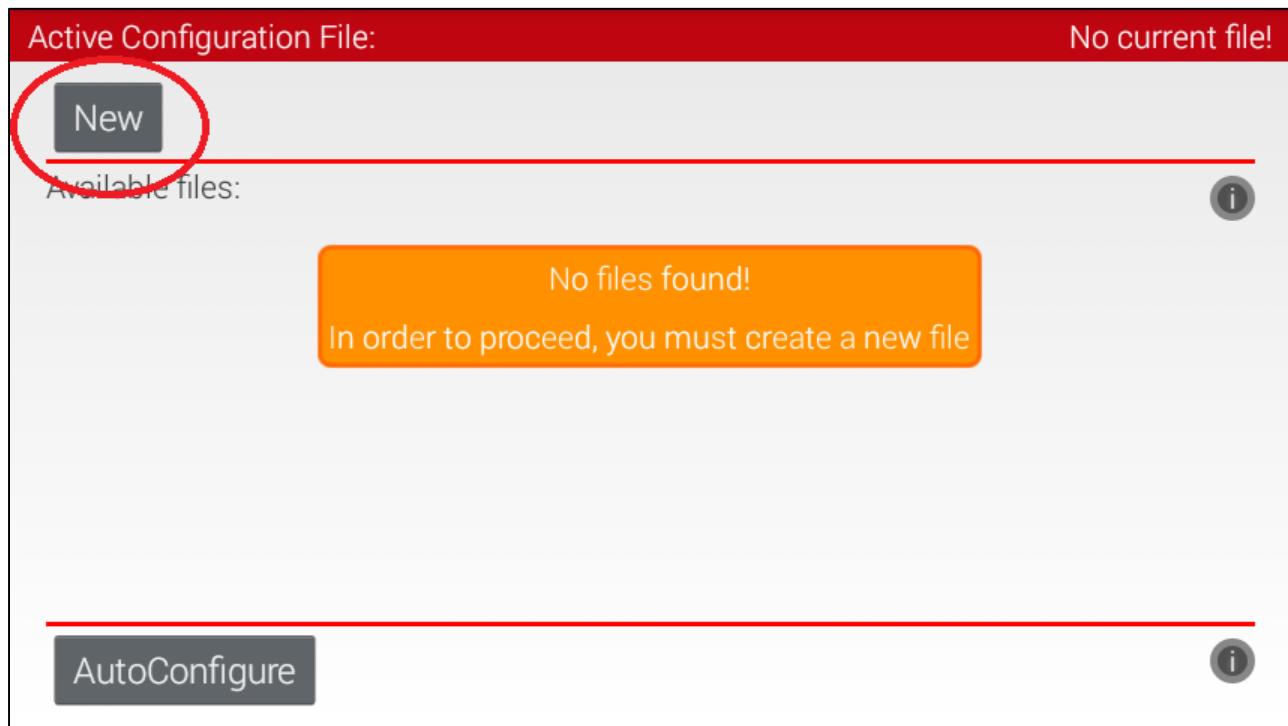
Step 4: Select 'Settings'.



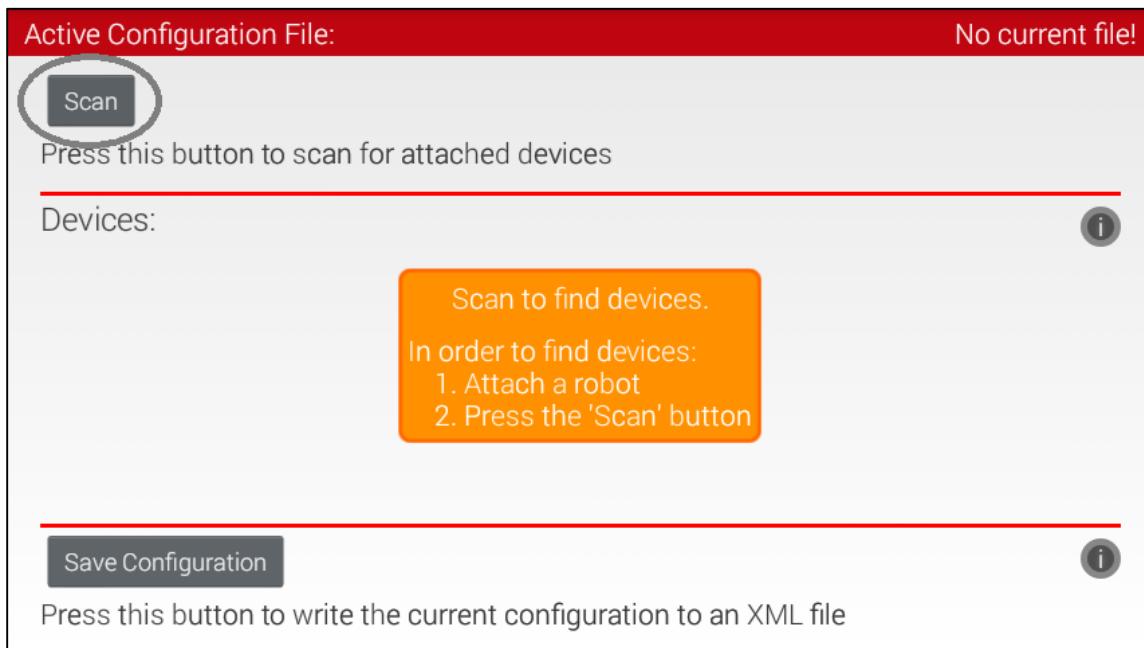
Step 5: Select 'Configure Robot'.



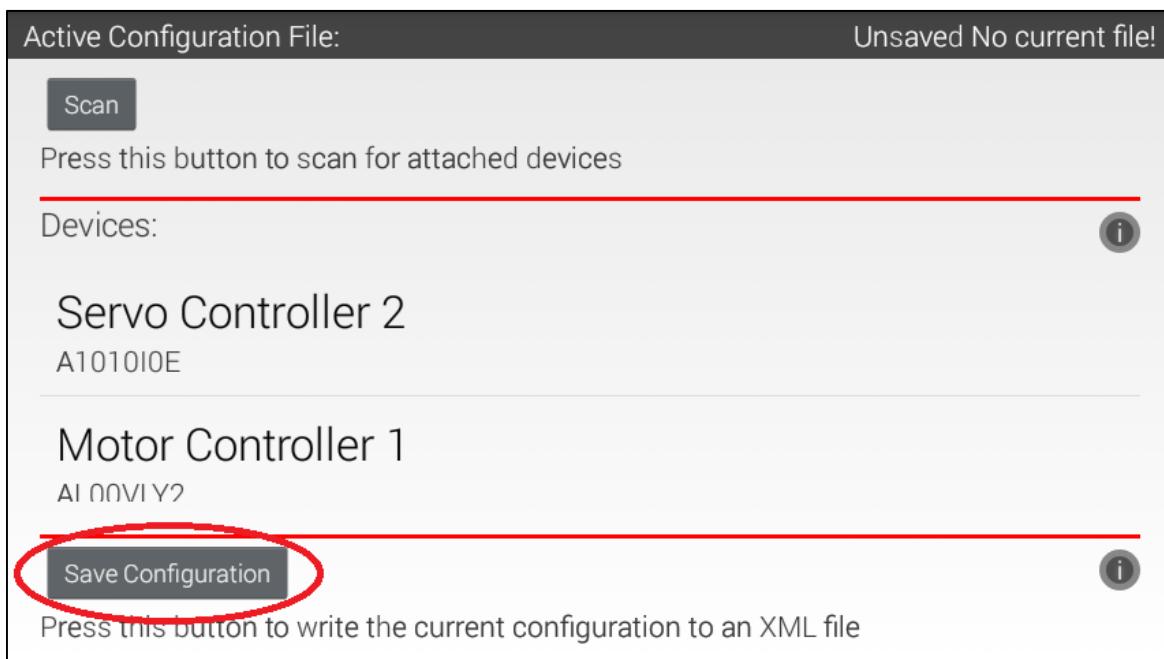
Step 6: Select 'New'.



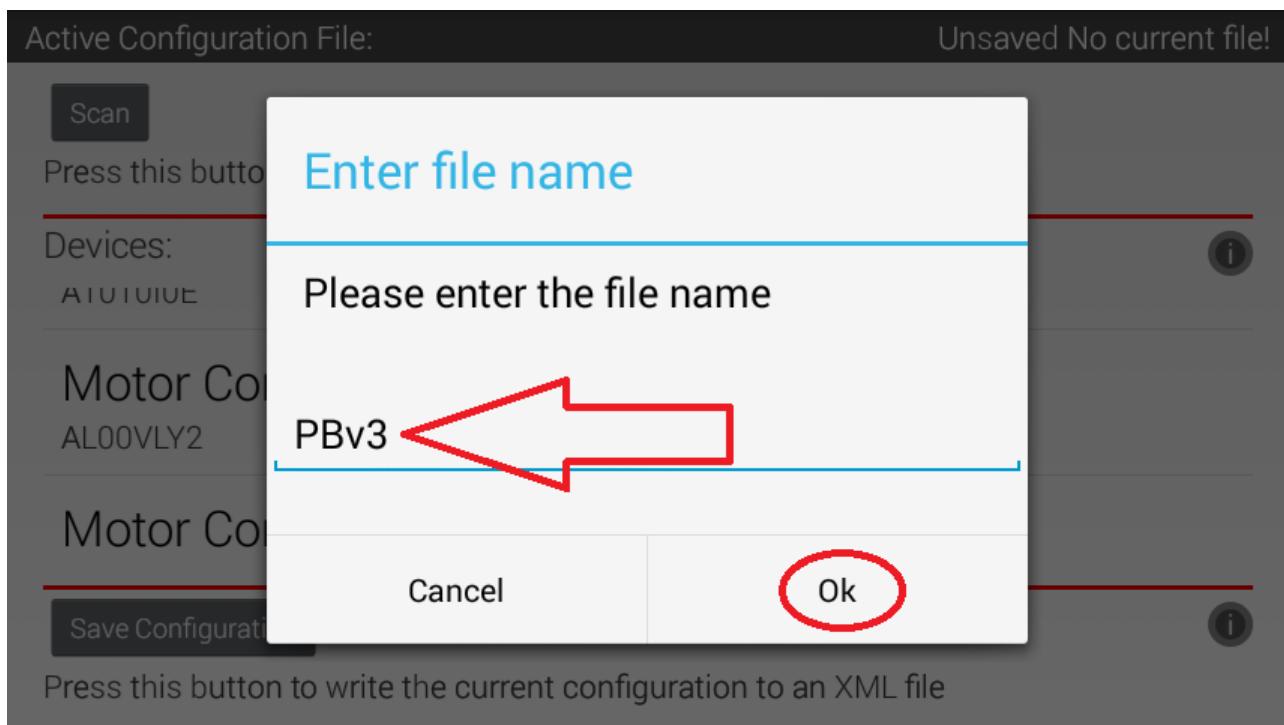
Step 7: Select ‘Scan’. Several presses may be necessary for all devices to display on the screen. Scrolling the screen up and down may also be necessary to see all of the controllers. There should be three and those three should have the same serial numbers listed as was recorded before (see [Recording the Controller Serial Numbers](#)). If this dialog is shown (below), then check the box to allow the app to access the USB device by default and select ‘OK’. If it is not shown, then continue to the next step.



Step 8: Click “Save Configuration”. This step is not strictly necessary at this time, but it is always a good idea to save often.

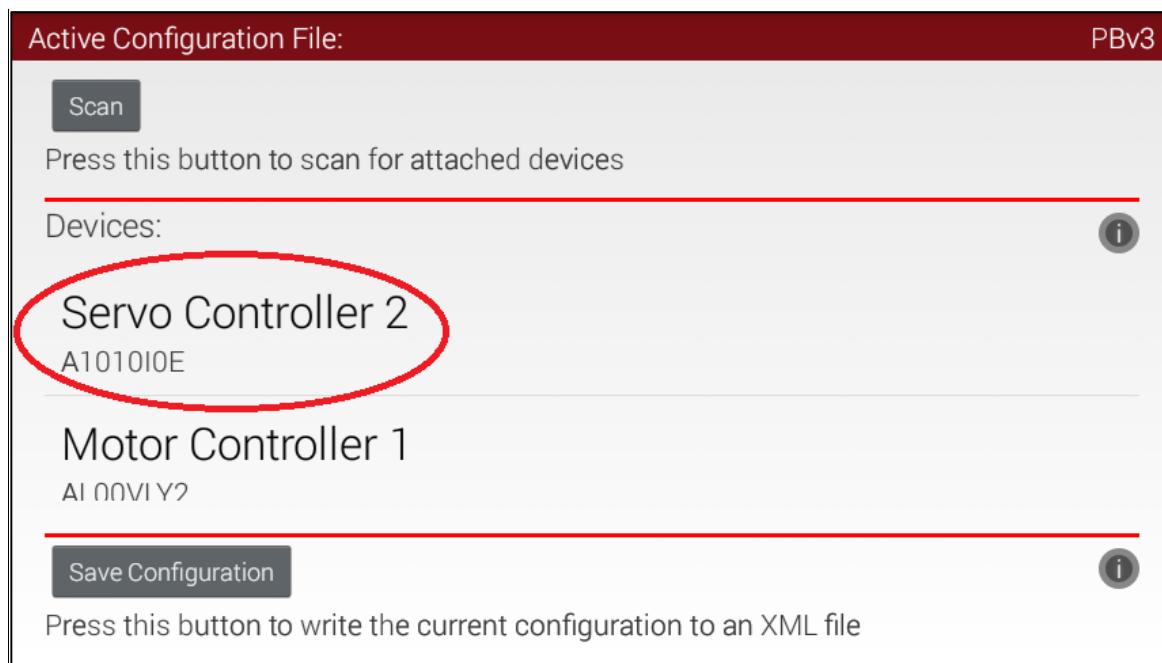


Step 9: Enter a file name. In this example, the file will be called PBv3. (Tip: Long holding the 'e' key will insert a '3'.) Click 'Done'. Click 'Ok'. The text PBv3 will show in the top right corner.

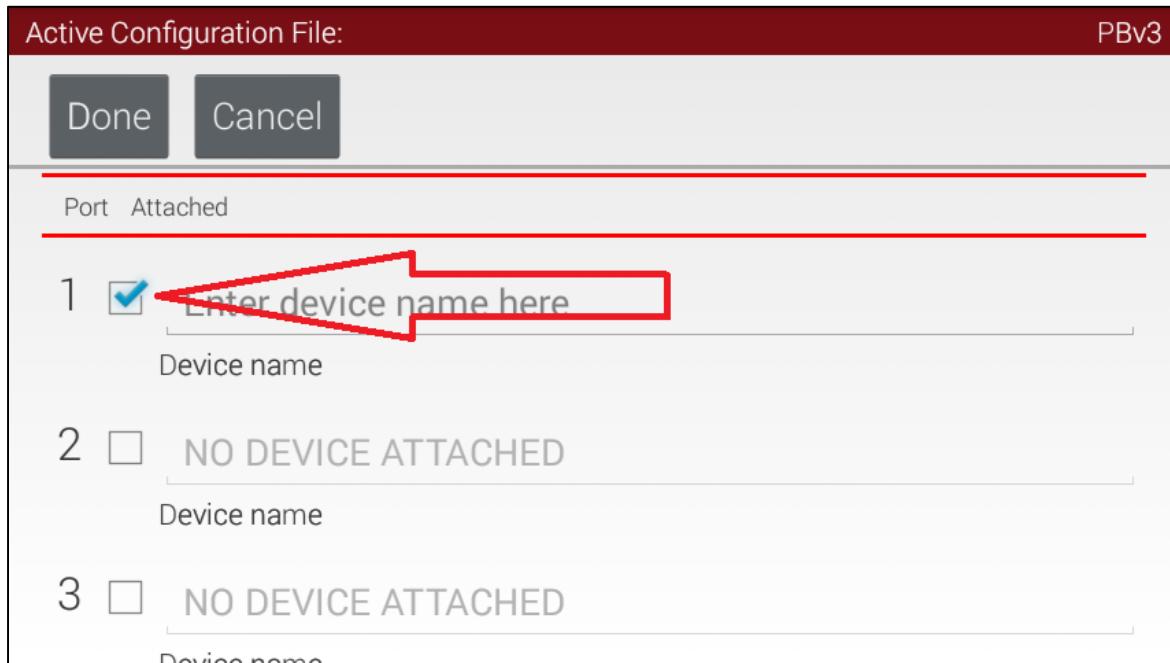


Configuring the Servo Controller

Step 1: Click on the Servo Controller. From our example, its serial number is A1010I0E and it is displayed on the screen as Servo Controller 2. Remember that there is only one servo controller, even though it shows a '2'



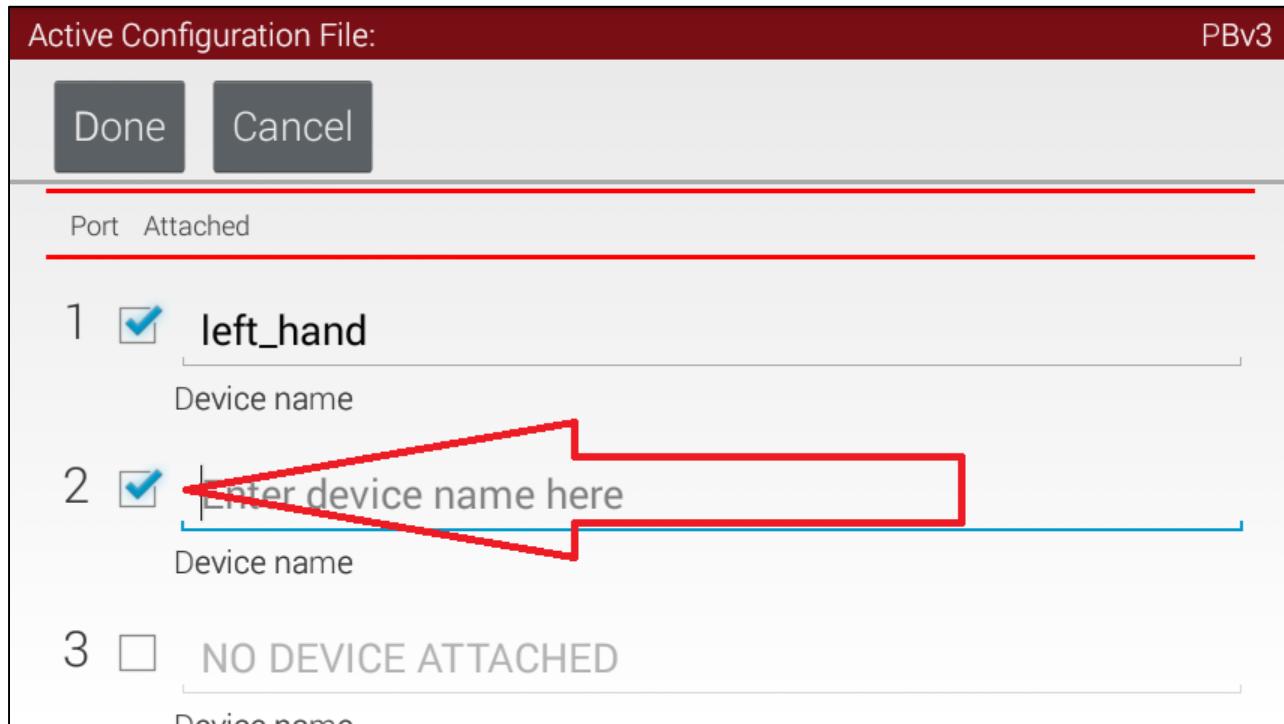
Step 2: Place a check mark next to port 1 and touch ‘Enter device name here’.



Step 3: Enter left_hand. (Tip: Long holding the ‘f’ key will produce an underscore.) Click ‘Next’.



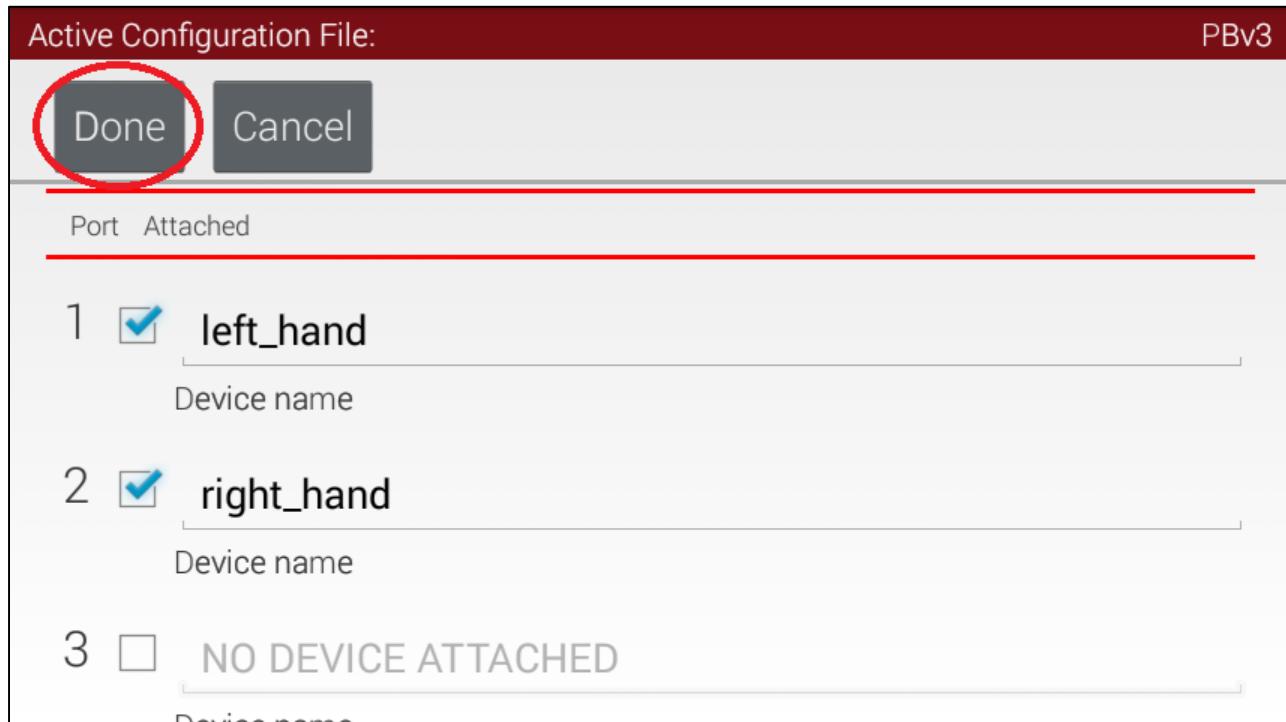
Step 4: Place a check mark next to port 2 and touch ‘Enter device name here’.



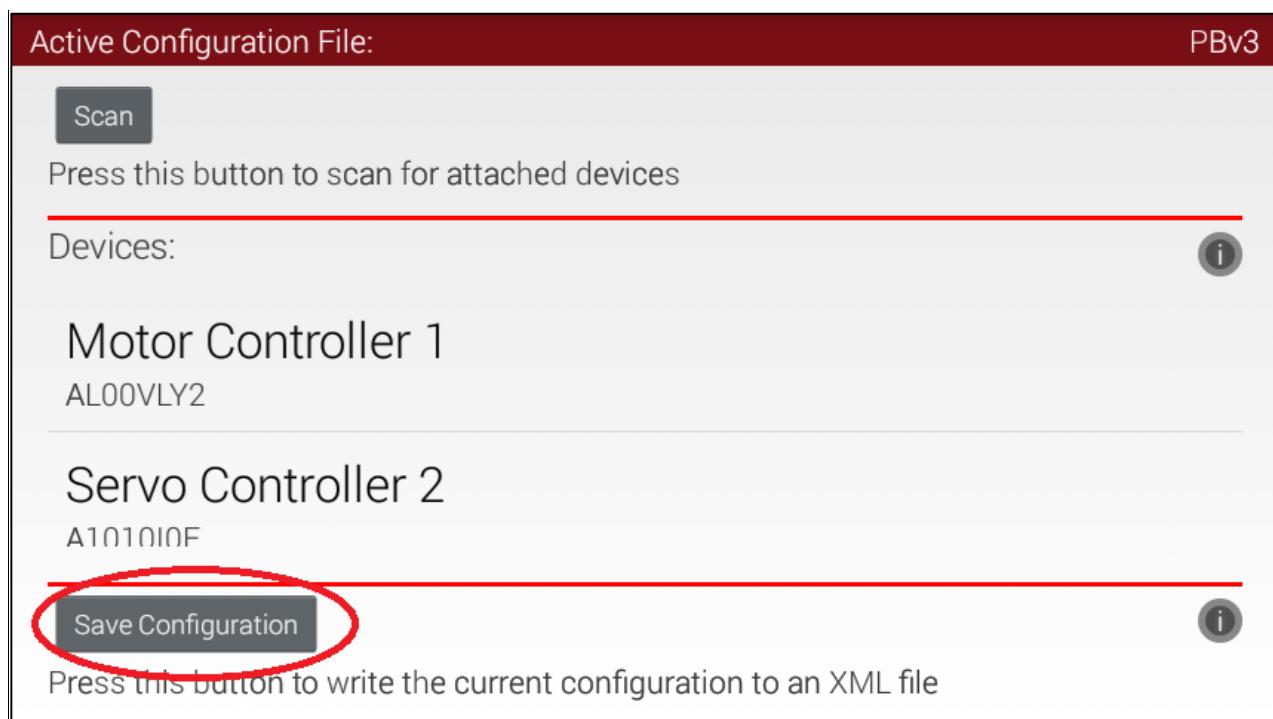
Step 5: Enter right_hand. (Tip: Long holding the ‘f’ key will produce an underscore.) Click ‘Next’.

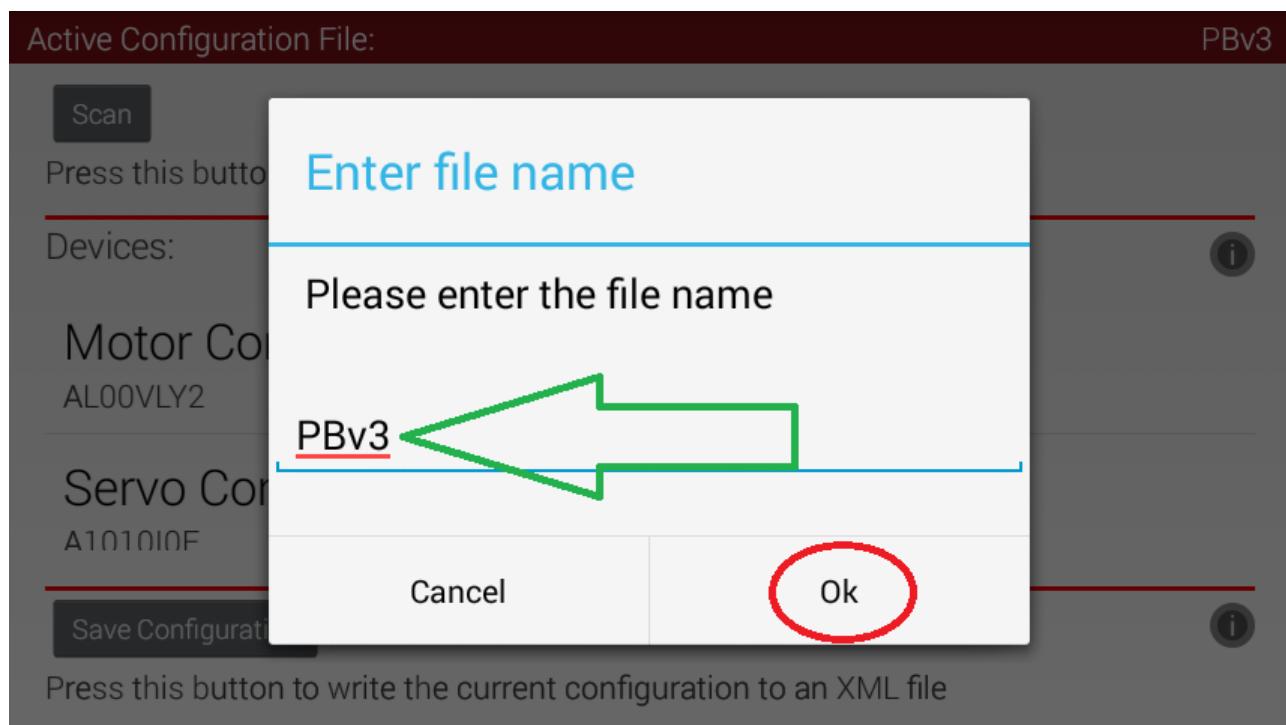


Step 6: Click ‘Done’.



Step 7: Click ‘Save Configuration’. Saving often is always a good idea. The dialog will already show PBv3; click ‘Ok’.





Configuring the Drive Controller

Step 1: Click on the Motor Controller that has the proper serial number for the drive controller. From our example, its serial number is AL00UWI7 and it is displayed on the screen as Motor Controller 3. Remember that there are two motor controllers. IGNORE the number next to Motor Controller. USE the serial number instead.

Active Configuration File:

Scan

Press this button to scan for attached devices

Devices:

Servo Controller 2	(i)
A1010IE	
Motor Controller 3	(i)
AL00UWI7	



Save Configuration

Press this button to write the current configuration to an XML file

Note: Always use the serial number as reference. Properly labeling your Robot components' serial numbers will make this easier to identify what is connected to what.

Step 2: Place a check mark next to port 1 and touch 'Enter motor name here'.

Active Configuration File: PBv3

Done **Cancel**

Motor Controller 3

AL00UWI7

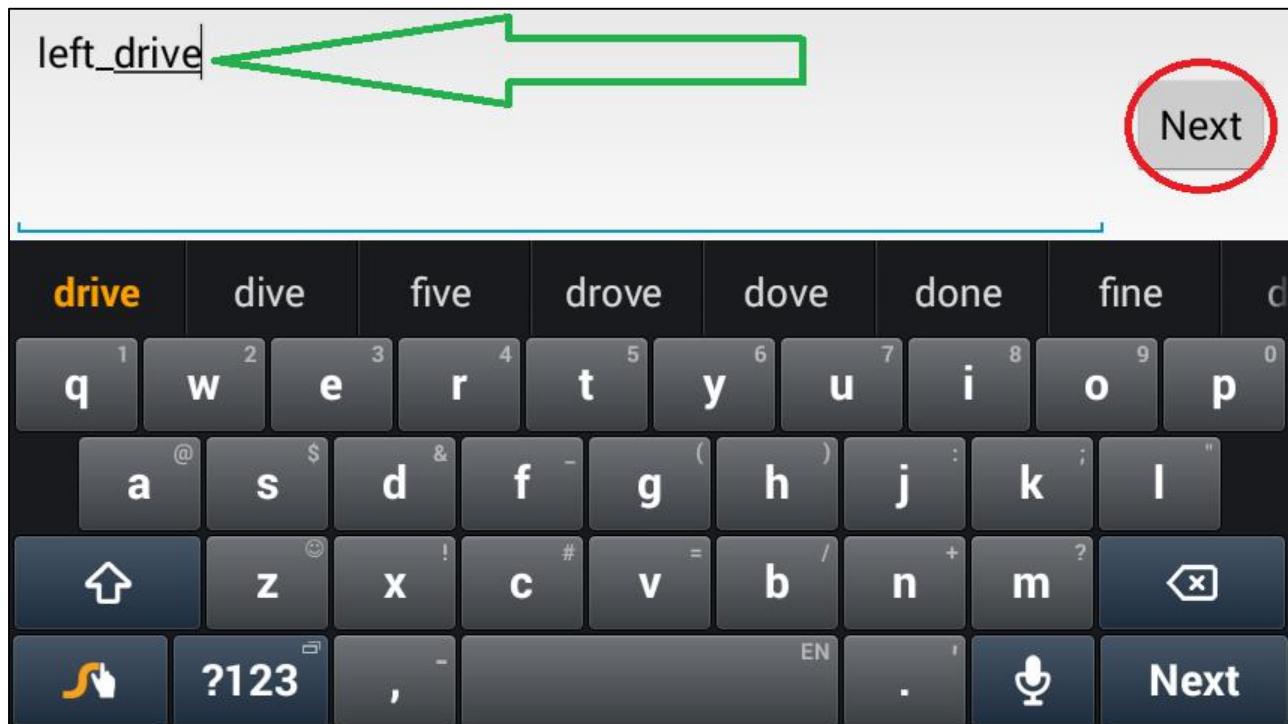
Enter the name for this motor controller here

Port Attached

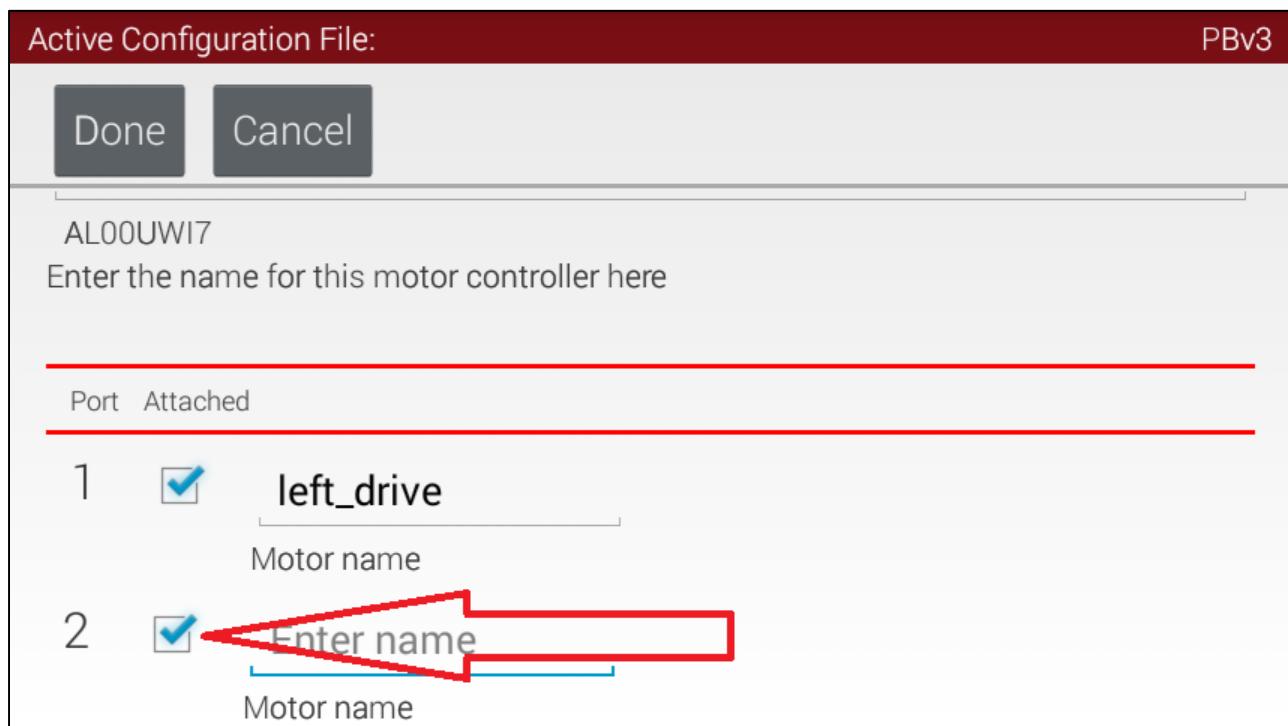
1	<input checked="" type="checkbox"/>	<input type="text" value="Enter motor name"/>
Motor name		
2	<input type="checkbox"/>	NO DEVICE ATTACHED



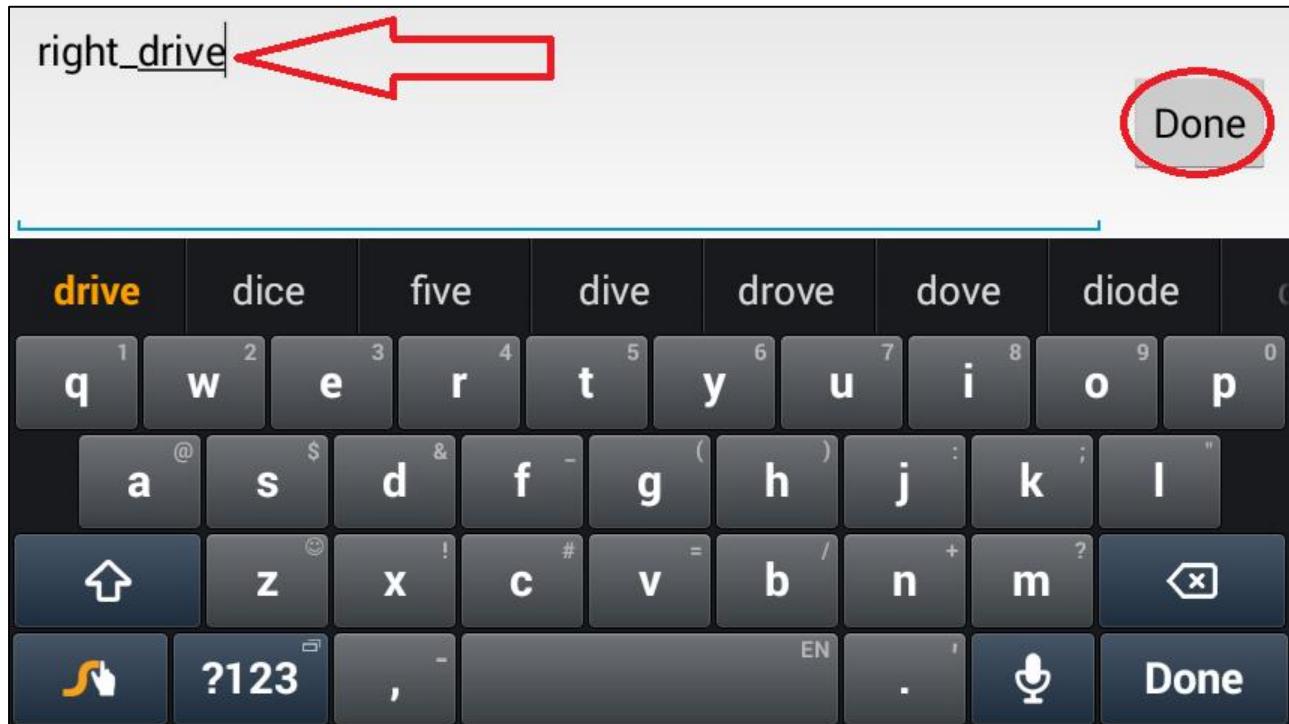
Step 3: Enter left_drive. (Tip: Long holding the 'f' key will produce an underscore.) Click 'Next'.



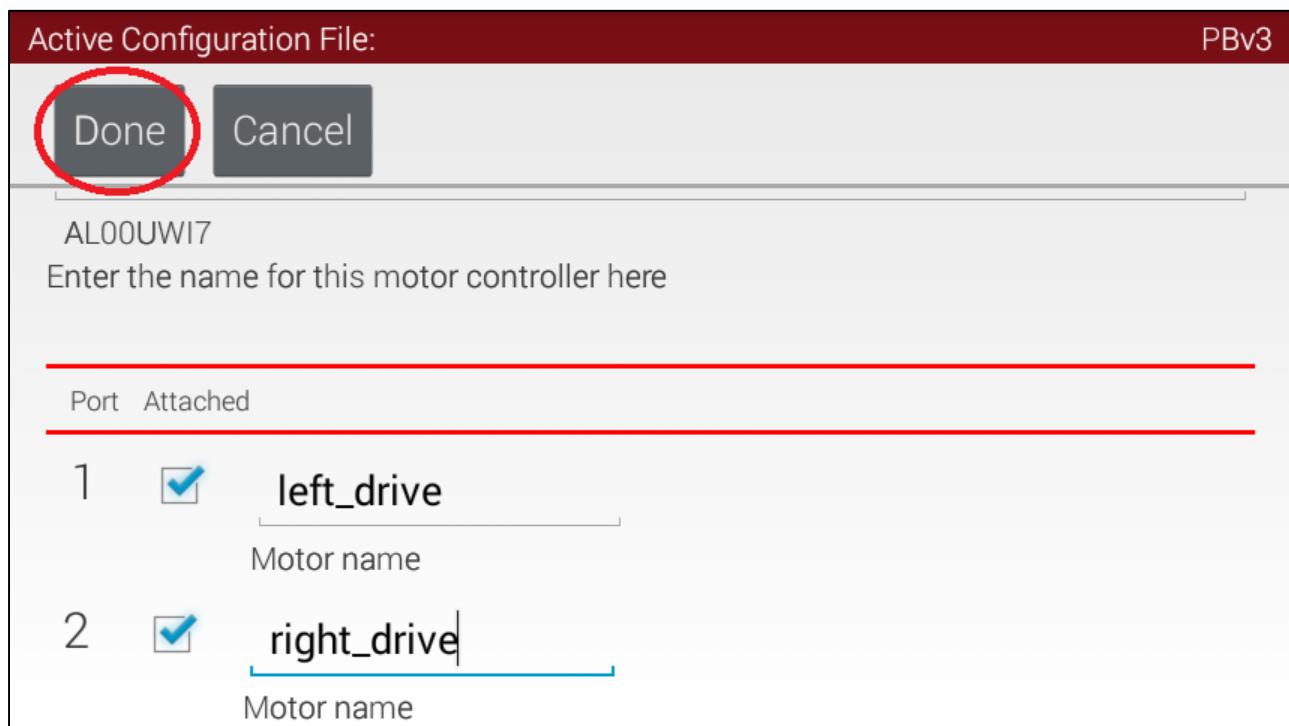
Step 4: Place a check mark next to port 2 and touch 'Enter name'.



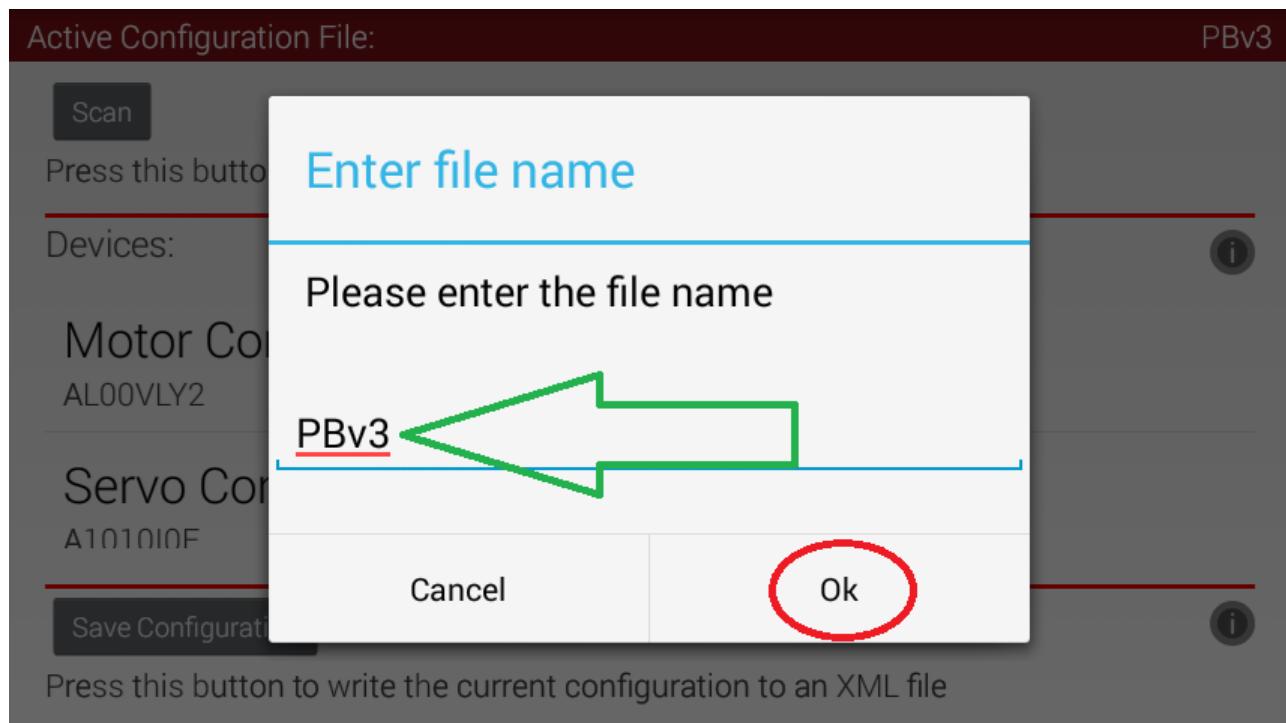
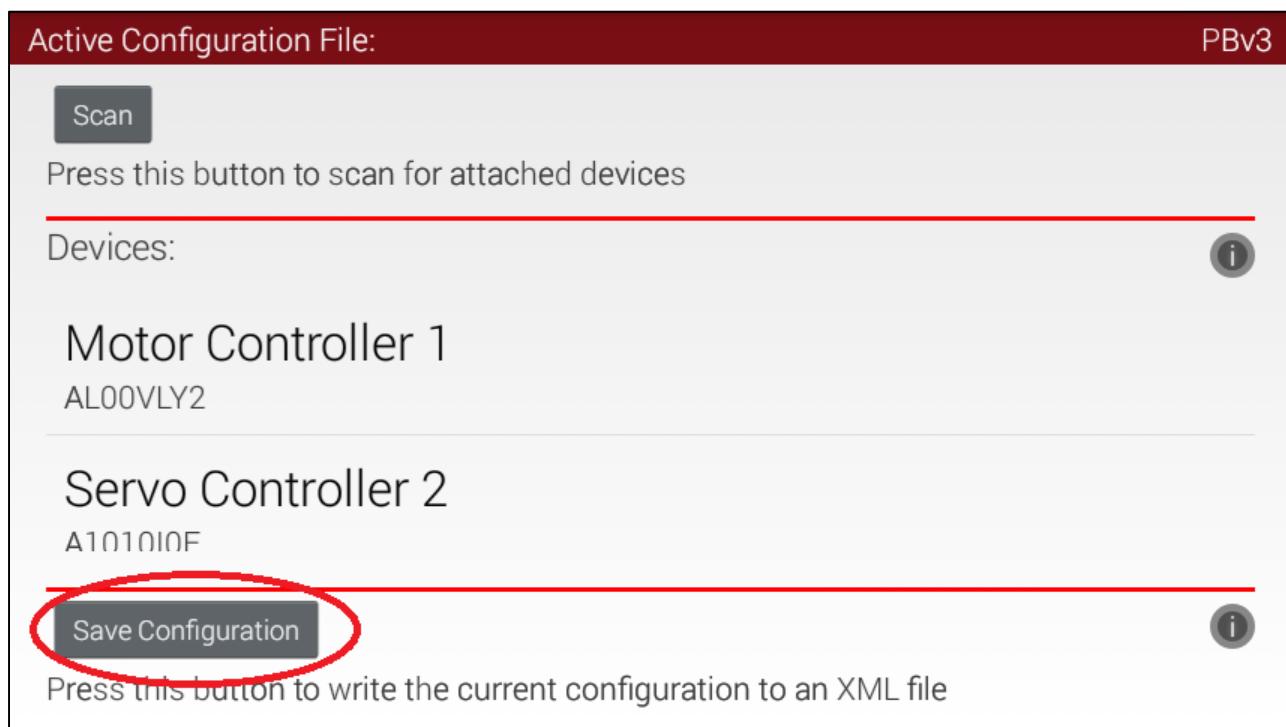
Step 5: Enter right_drive. (Tip: Long holding the 'f' key will produce an underscore.) Click 'Next'.



Step 6: Click 'Done'.



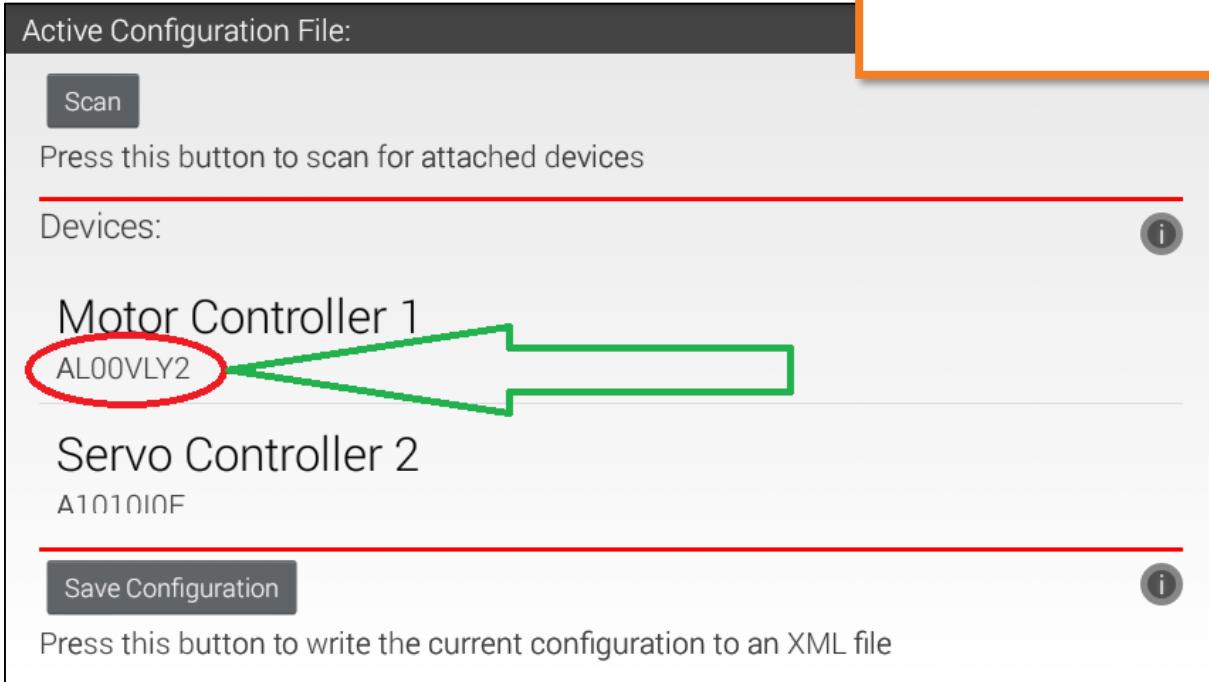
Step 7: Click 'Save Configuration'. Saving often is always a good idea. The dialog will already show PBv3; click 'Ok'.



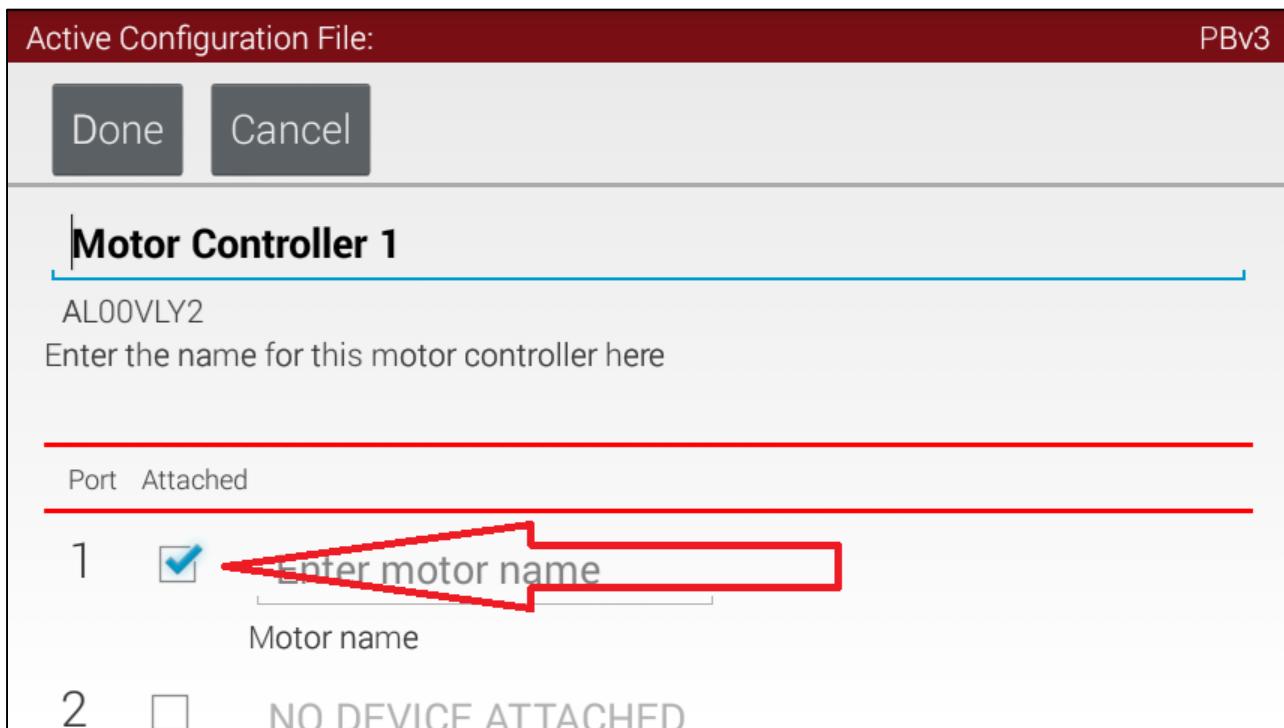
Configuring the Auxiliary (Arm) Controller

Step 1: Click on the Motor Controller that has the proper serial number for the auxiliary (aka arm) controller. From our example, its serial number is AL00VLY2 and it is displayed on the screen as Motor Controller 1. Remember that there are two motor controllers. IGNORE the number next to Motor Controller. USE the serial number instead.

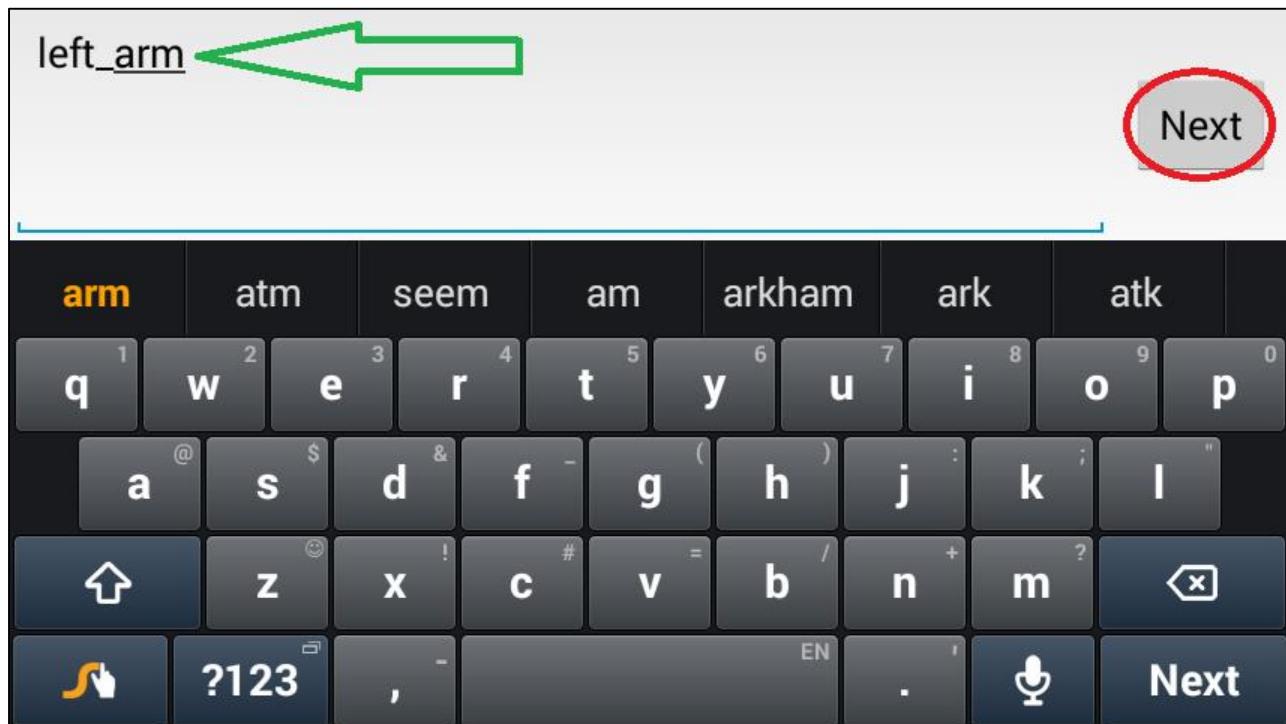
Note: Always use the serial number as reference. Properly labeling your Robot components' serial numbers will make this easier to identify what is connected to what.



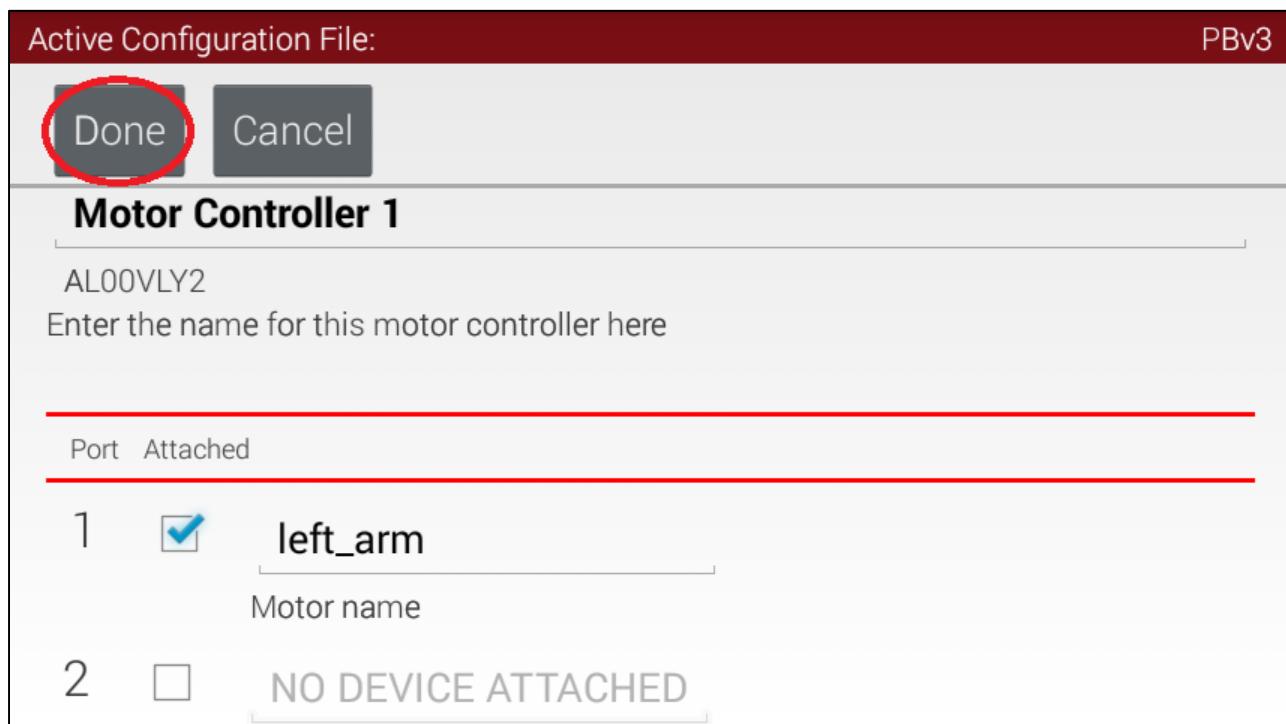
Step 2: Place a check mark next to port 1 and touch 'Enter motor name here'.



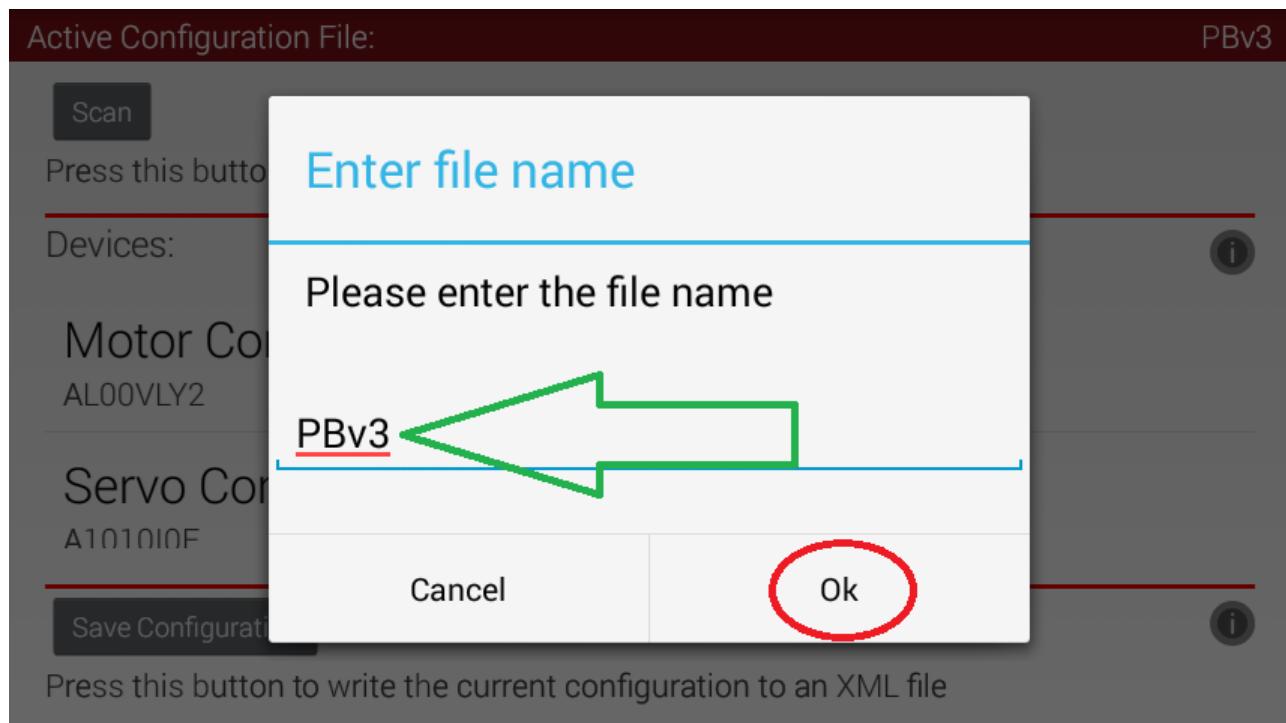
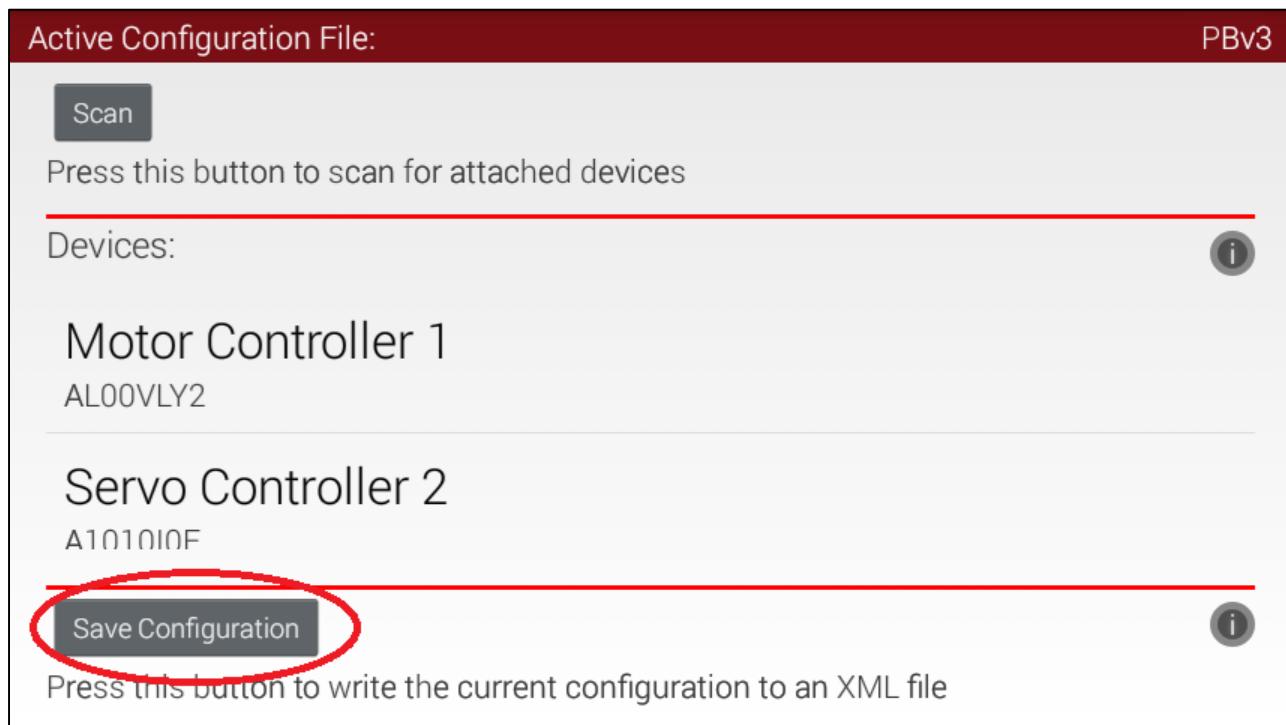
Step 3: Enter left_arm. (Tip: Long holding the 'f' key will produce an underscore.) Click 'Next'.



Step 4: Click 'Done'.



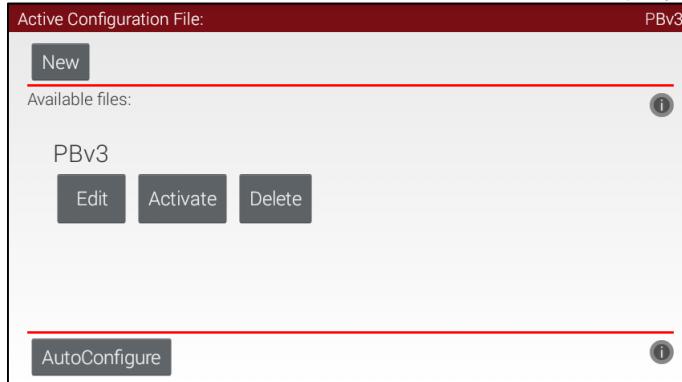
Step 11: Click 'Save Configuration'. This step IS necessary at this time - all of the controllers have been named. The dialog will already show PBv3; click 'Ok'.



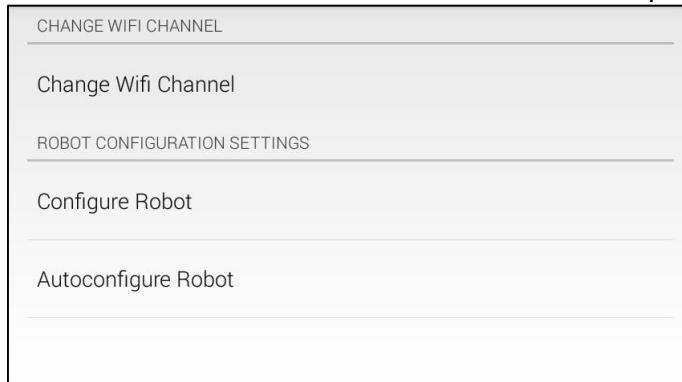
Testing the Configuration

Step 1: Click the back button three times (one of the three buttons on the lower portion of the phone).

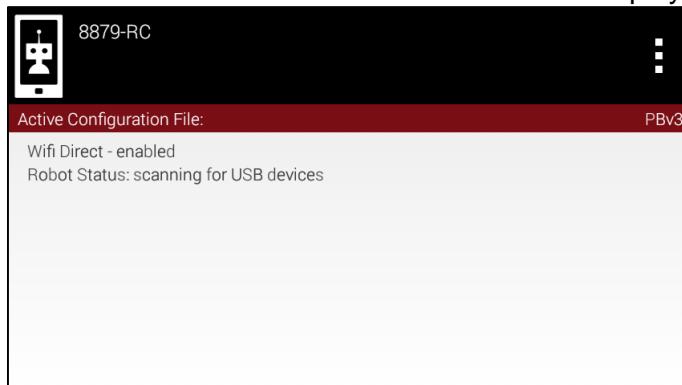
After the first click the below screen should be displayed.



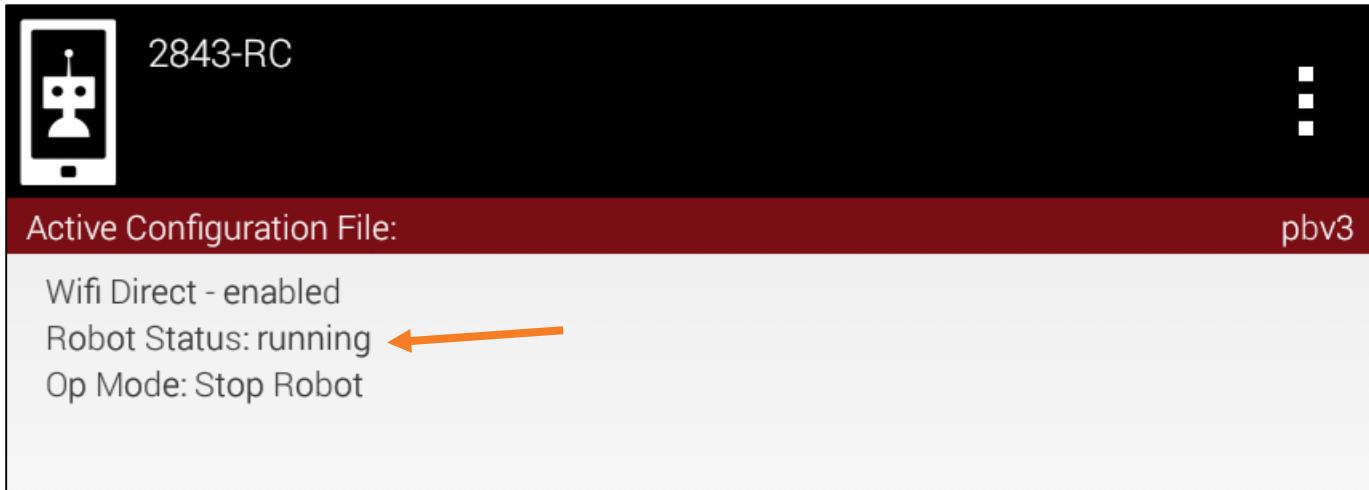
After the second click the below screen should be displayed.



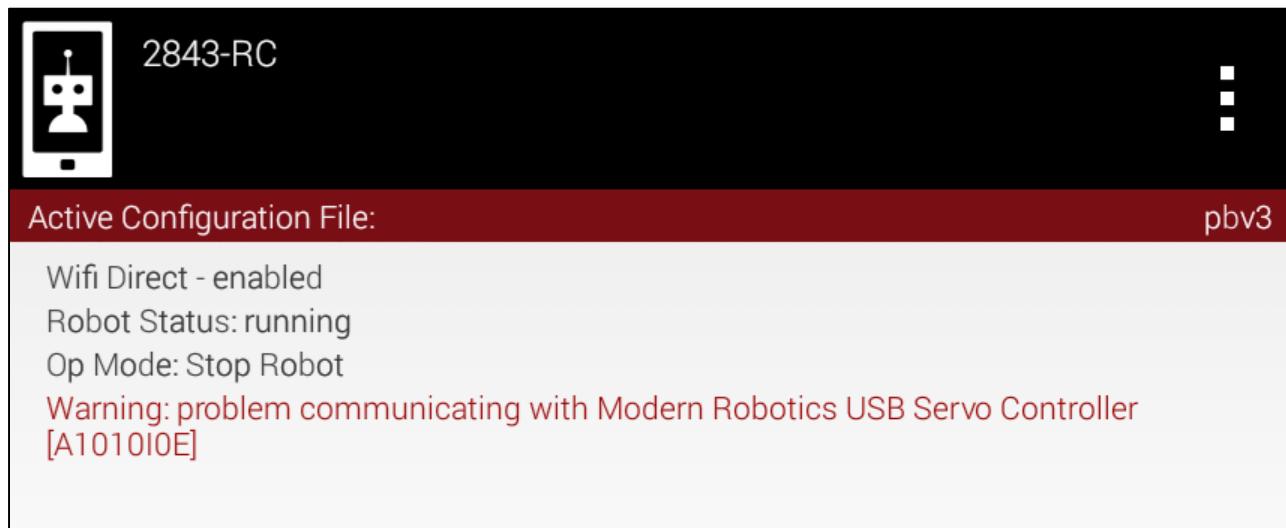
After the third click the below screen should be displayed.



Step 2: If all of the information is valid, the screen should show the active configuration file in the top right corner of the screen and the Op Mode status will be Stop Robot. The Robot Status will rapidly change from “scanning for USB devices”, “starting robot”, and finally to “running”.



If the following dialog is shown, then check to make sure all of the controllers have power and are connected via USB to the Core Power Distribution Module. Also make sure that all your wires have been properly secured.



Testing the Robot

After the Robot is built, the programming is complete, and the electronics are installed, testing must be done to ensure everything came together as planned. This section assumes that the Core Power Distribution Module (PDM) is powered on, the cell phones are powered on and the *FIRST* Tech Challenge apps are running.

Assigning Gamepads

Two Logitech F310 gamepads will be connected to the Driver Station cell phone. These gamepads will be used by students to control a Robot's functionality. The Driver Station app must configure the gamepads before they can be used.



Step 1: Set the switch on the bottom of both gamepads to the “X” position.

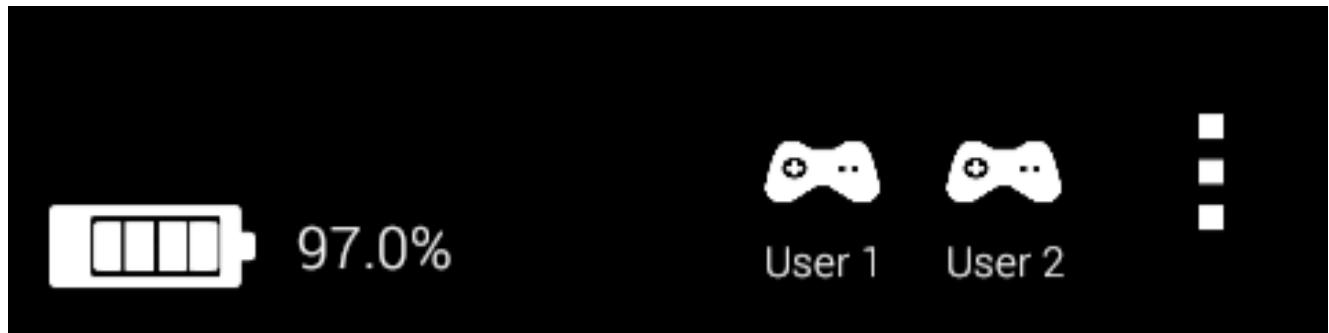
Step 2: Select a gamepad that will be used as driver 1. Push the **Start** key on the gamepad while simultaneously pressing the **A** button.



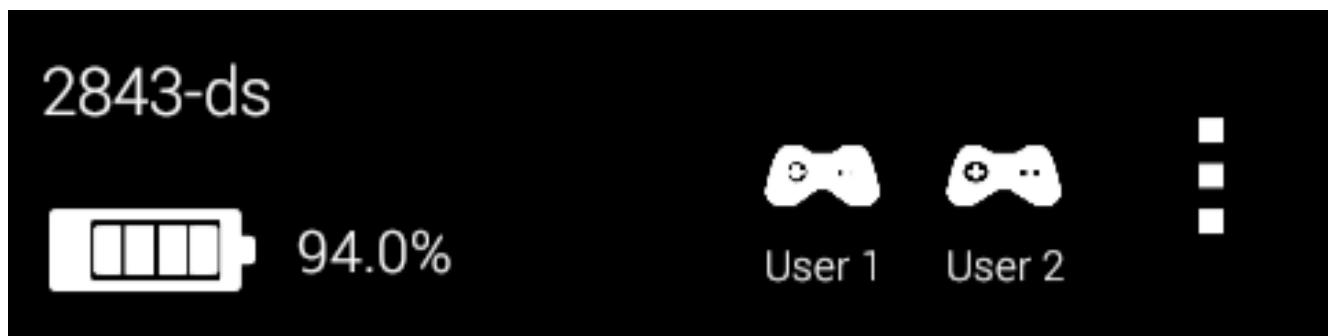
Step 3: Select the other gamepad. It will be used as driver 2. Push the **Start** key on the gamepad while simultaneously pressing the **B** button.



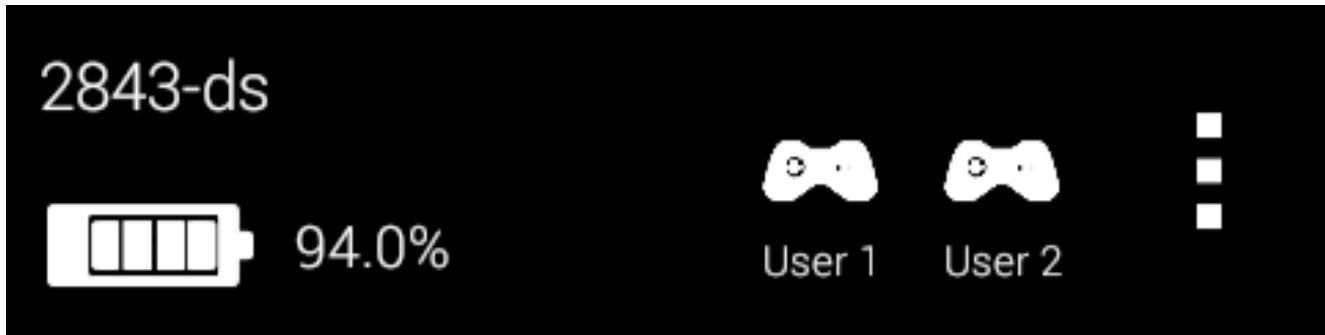
Step 4: Observe that small gamepad icons appear at the top of the screen. User 1's information should be on the left. User 2's information should be on the right. This image shows that neither gamepad is assigned.



For example, when the gamepad that represents driver #1 is active, then the left icon will change to show that the gamepad is active.



If the gamepad that represents driver #2 is active, then the right icon will change.



Note that for the joysticks on the gamepads, the values range from -1 to +1. For the x axis, -1 represents the far left range of motion and +1 represents the far right range. For the y axis -1 represents the topmost position and +1 represents the bottommost position.

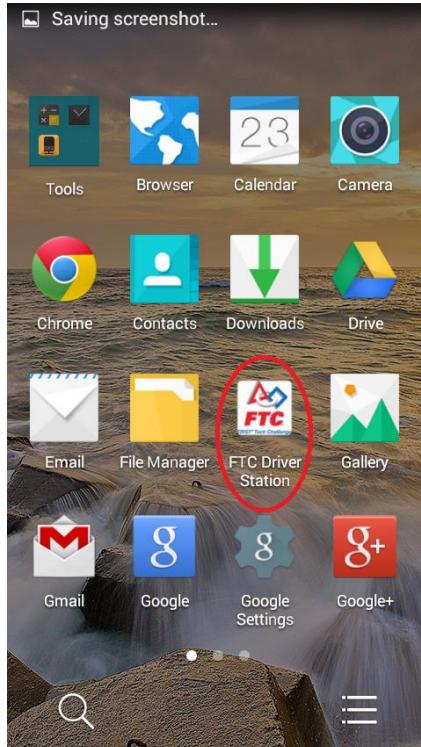
Pairing the Driver Station to the Robot Controller

The Driver Station app must first “pair” to the Robot Controller to provide communication between the Robot and the gamepads.

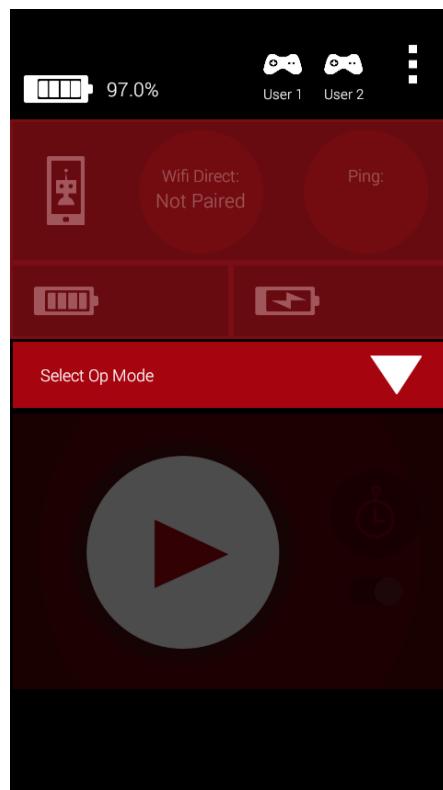
Step 1: Power both cell phones on. Open the app drawer if the FIRST Tech Challenge Driver Station icon is not present on the main screen.



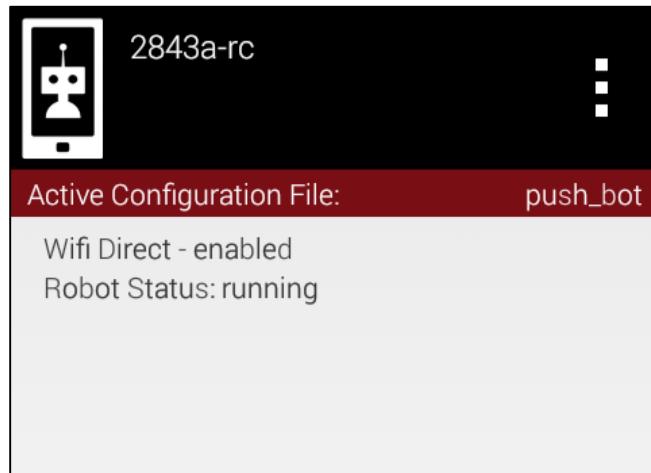
Step 2: Start the Driver Station app on the Driver Station cell phone.



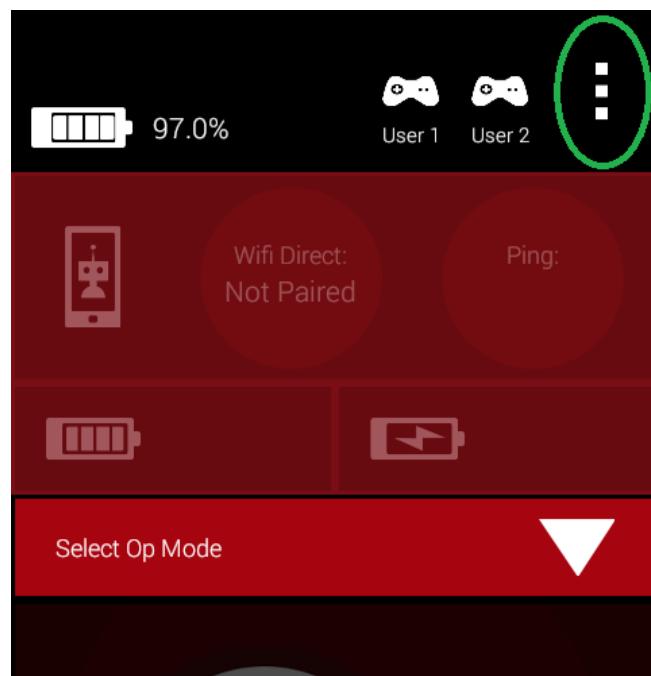
Step 3: The screen below should be displayed on the Driver Station cell phone.



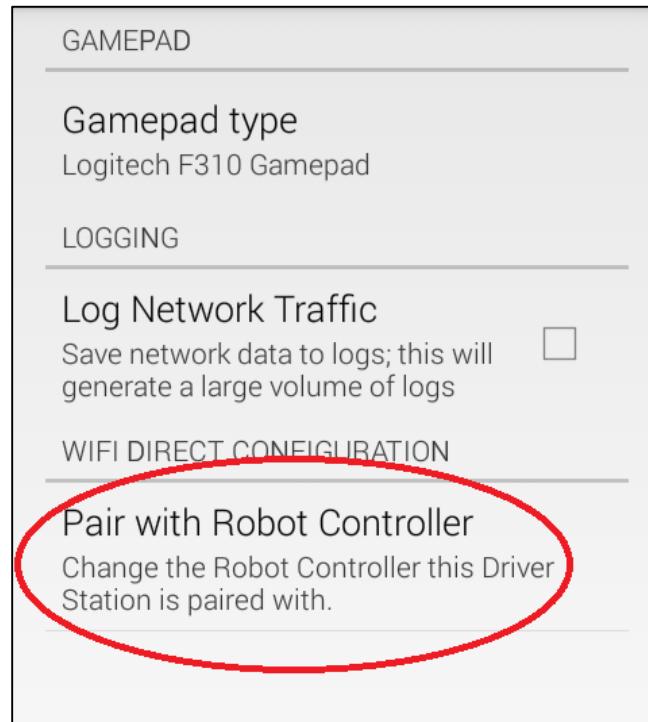
Step 4: Make sure the Robot Controller app is running on the Robot Controller cell phone.



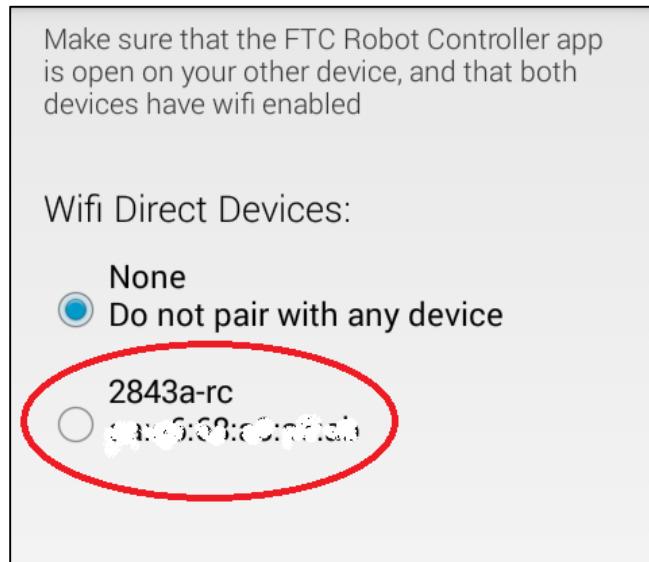
Step 4: On the Driver Station app touch the three vertical dots on the upper right hand corner and select 'Settings' from the pop up menu.



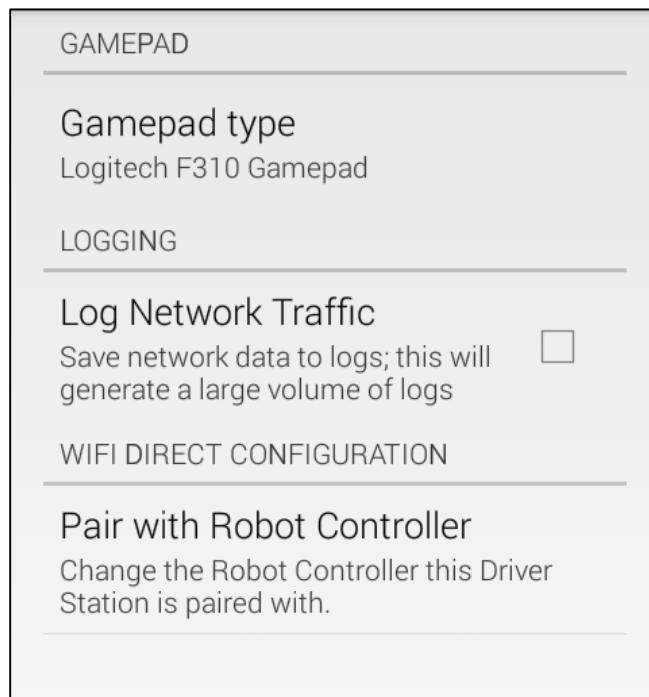
Step 5: Press the "Pair with Robot Controller" option to start the pairing process.



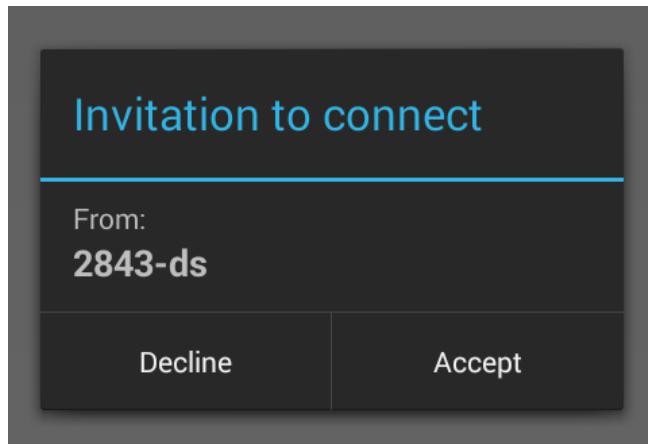
Step 6: After the previous step, the Driver Station will display the following screen. Select the device.



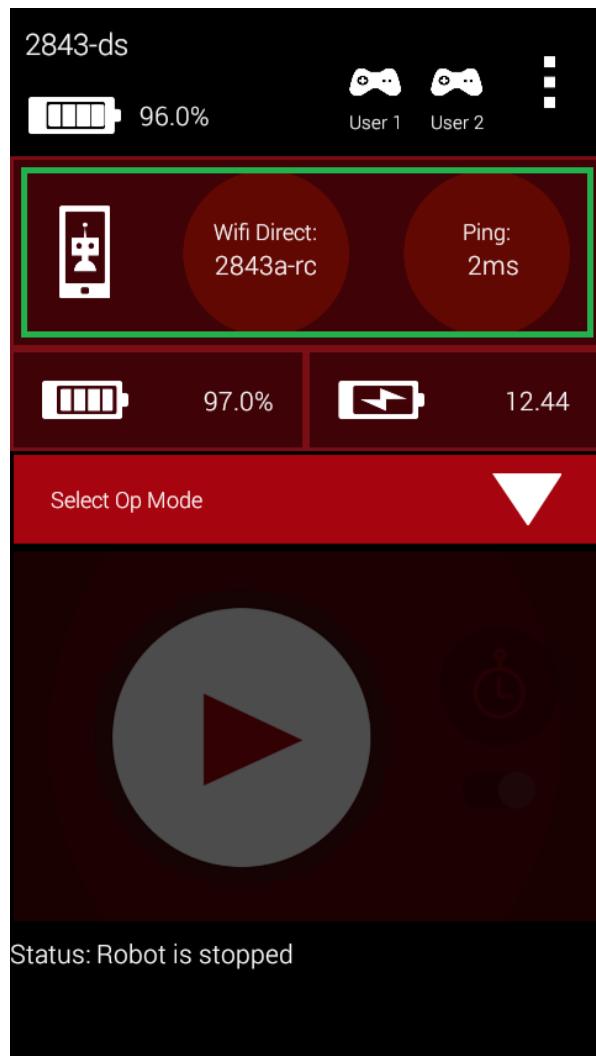
Step 7: Use the back button to send a connection request to the Robot Controller.



Step 8: If this is the first pairing attempt, the Robot Controller device might display a prompt asking if it is OK to allow an Android device to establish a Wi-Fi Direct connection. Select ‘Accept’.



Step 9: Observe that some of the controls on the display have changed indicating that the Driver Station is now connected to a Robot Controller (green rectangle).



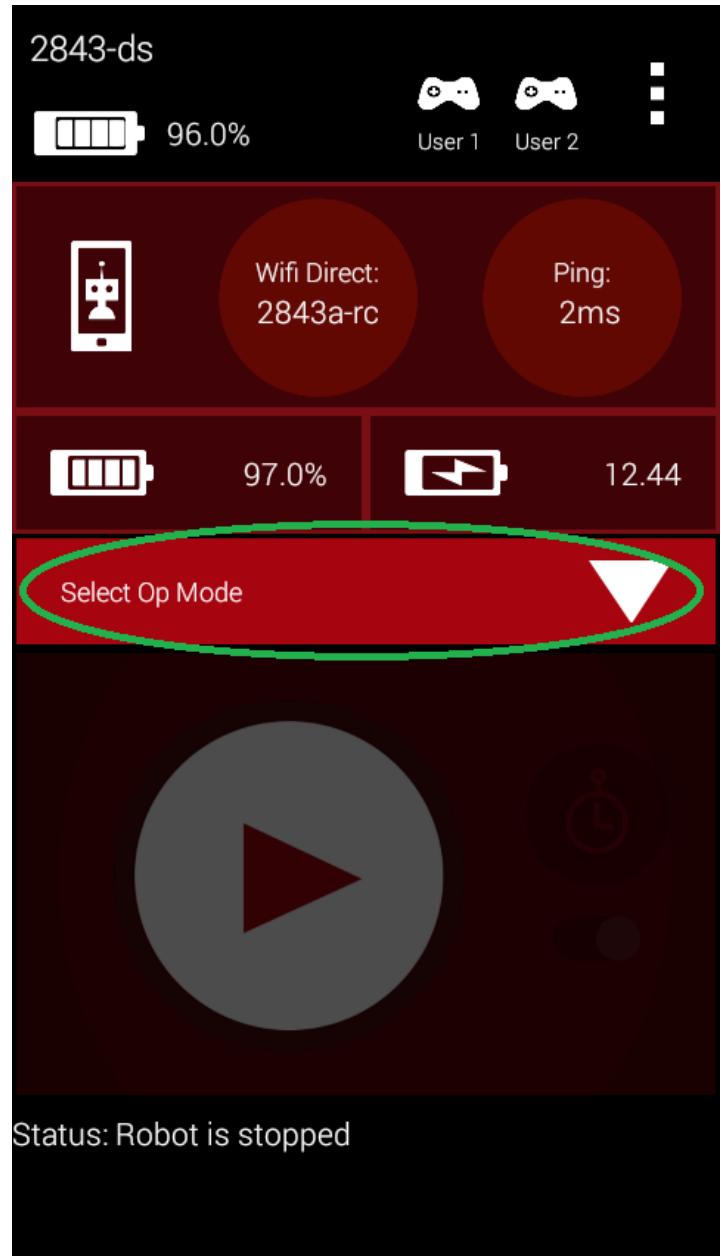
Running an Op mode

Step 1: Operational Modes or Op Modes are program modules that cause the Robot to perform specific behaviors. The Robot Controller app should include some pre-installed op modes. If the Robot Controller has been successfully paired to the driver station, and if the Robot is running, then the Op mode buttons on the FIRST Tech Challenge Driver Station app should now be enabled.

Note that if the Driver Station app is connected successfully to the Robot Controller app, then some ping statistics should be visible at the top right hand corner of the Driver Station screen. In the image to the right the ping value is “2ms”. This means it took 2 milliseconds for the driver station to send a packet to the Robot controller and for the Robot controller to acknowledge the packet and send its acknowledgement back to the driver station.

The ping round trip time is a rough measure of network quality. As the round trip time increases, the network performance usually degrades. In our testing, an ideal round trip time is anything on the order of 10 milliseconds or lower. Sometimes the ping time period increases if there is a surge in network traffic or interference. If the period starts to increase to the order of or greater than 100 milliseconds, then there might be excessive network traffic or noise on the wireless channel that cell phones are using.

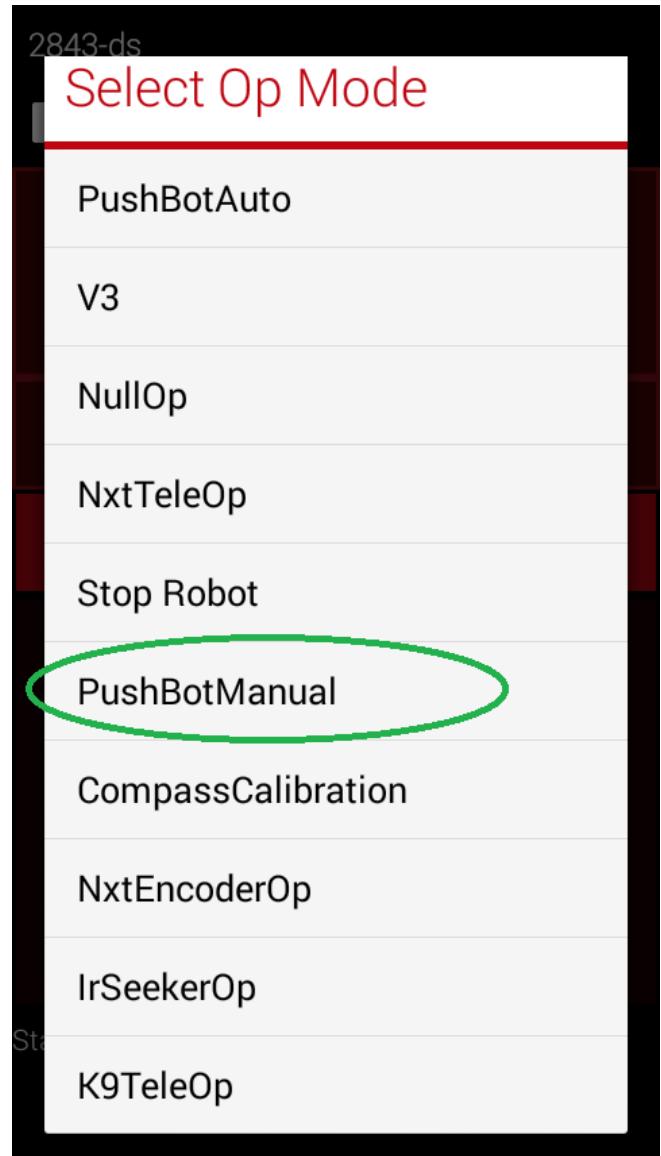
Step 2: Touch the **Select Op Mode** button to display a list of the available Op Modes sent to the Driver Station app from the Robot Controller app. Select ‘PushBotManual’ from the list and the round play button should become brighter.



Note that any of the other Op modes will fail to work as the names and number of controllers are different from the PushBot.

Step 3: Once the PushBotManual Op mode is selected, the FIRST Tech Challenge Driver Station screen should change. The name of the selected Op mode should appear where the words ‘Select Op Mode’ once was. Also, the **Start** button should be enabled.

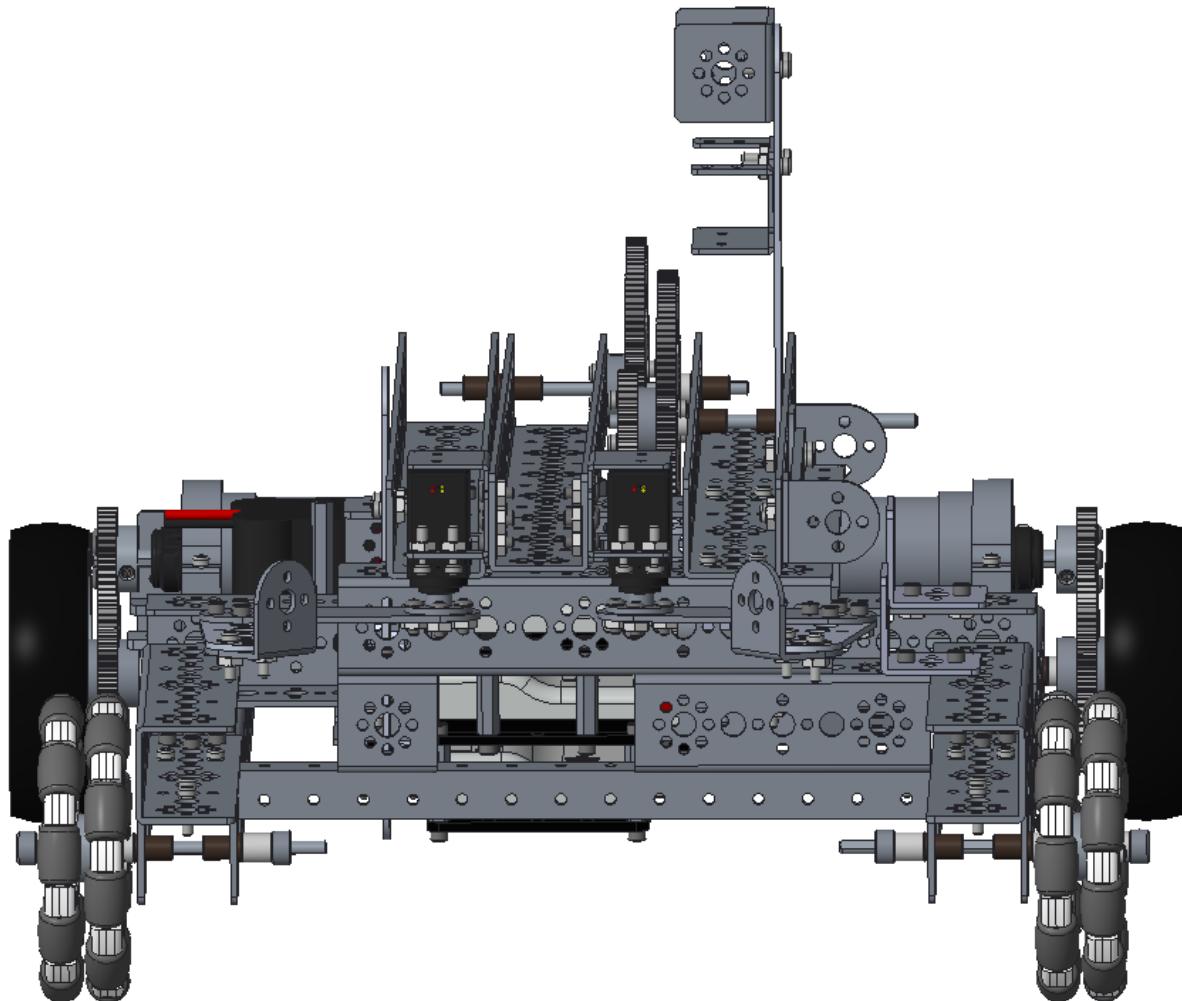
If the **Start** button is pressed, then the Op mode will run indefinitely until the program terminates or until the **Stop** button is pressed. Press the start button and proceed to the next section.



Using the gamepads

This section describes how the two gamepad controls control the function of the Robot – make the Robot drive forward/backward, lift/lower the arm and open/close the hand.

The first gamepad will be used to drive the robot. The left joystick controls the ‘left’ drive motor and the right joystick controls the ‘right’ drive motor. The image below shows the back of the Robot.





Push both joysticks away from your body and the Robot drives forwards. Pull both joysticks toward your body and the Robot drives backwards. Push one joystick up and the other down and the Robot turns.

The programming sections show the programming necessary to use a second Logitech controller (F310) to raise and lower the arm and open and close the 'hand'. This section describes how to use the controls on the gamepad to move the arm up and down and how to open and close the hand.

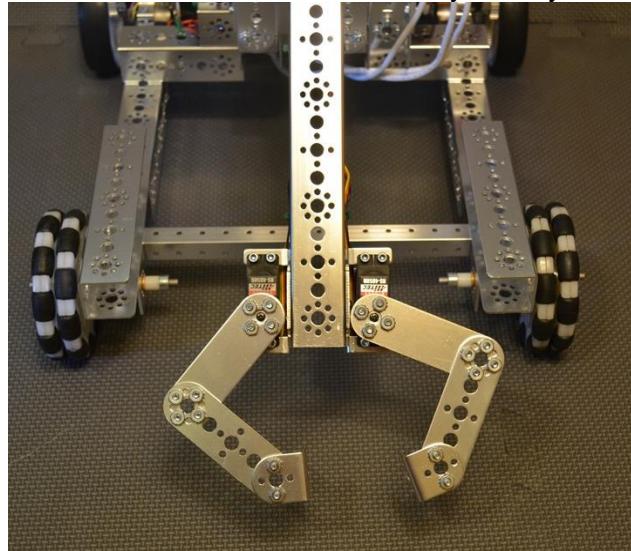
The 'arm' allows the Robot to lift items from the floor to at least 84 cm (33 in) above the ground. The arm uses a 1:9 gear-ratio to enable the arm to lift moderately heavy items. Push the left joystick away from the body to move the arm from the stowed position (folded onto the robot); toward the body to move the arm into the stowed position.

The 'hand' is capable of opening and closing gradually. Button 'X' opens the hand; button 'B' closes the hand. The term gradually is used, because when the button is released, the servo stops at its last position – it doesn't close or open all the way when the button is pressed.



Adjusting the Servos

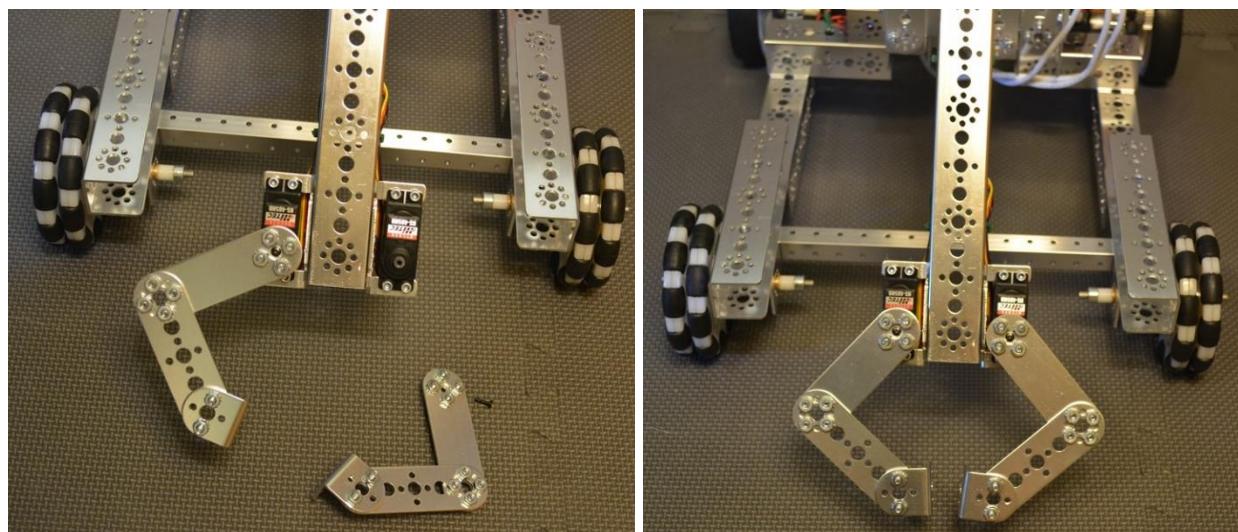
If the servos do not close all of way or they chatter, perform the following steps to correct them.



Step 1: Unplug one servo.

Step 2: Run servos to fully closed position using button X on gamepad 2.

Step 3: Remove the black screw from the servo. Remove the hand assembly. Replace the hand assembly such that the L bracket is as close to parallel to the arm as possible. Replace the black screw. See the pictures for an example. First picture is closed by software. Second picture is hand removed. Third picture is hand reattached in closed position.



Note: These images are from PBv2, but the same steps can be used for PBv3. The hands remain the same between the two versions.

Repeat step 1-3 for the other servo.

Making the Robot Competition Ready

- Now that testing has shown that all of the wires are connected properly, use zip ties or some other approved material to secure loose wires. Check out Legal and Illegal Parts List on the [Game Materials Page](#).
- It is recommended that an approved material such as non-skid be layered in the back of the phone holder to prevent damage to the phone.
- It is recommended that the phone be secured in the holder using a zip tie or some other mechanism to prevent it from being separated from the Robot during competition.
- Place non-skid around the gripper to provide extra grip...so hockey pucks, wiffle balls, pipes, racquetballs, crates, rings, blocks or practice golf balls, or red herrings can be collected with ease!
- Add your team numbers as directed in [Game Manual Part 1](#).

Adding Team Numbers

It is important for your Robot to identify itself. Field personnel use numbers to help you during matches and is required to pass Robot Inspection for any Official FIRST Tech Challenge Event. Other Teams can scout your Team throughout a competition and perhaps you will be chosen to form an alliance, if you are not already a Team captain! Announcers use it to engage the audience. **It is also part of the rules.** The [Game Manual Part 1](#) is the official source of guidance in this area and should be read every year., but from many previous years' experience there are some things that are consistent: make the numbers legible, large and conspicuous! Remember to read the Game Manuals for each season and refer back to the rules, so your Robot will pass inspection with flair!

Here is one suggestion for mounting your team's numbers. Laminate them and place them on opposite sides of the robot. An allen wrench can be used to punch holes and then use bolts to secure them.



2016-2017 FIRST® Tech Challenge PushBot Build Guide

Appendices

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. In order to submit a new question, you must have a unique Q&A System User Name and Password for your team.

FIRST Tech Challenge Game Manuals

Part 1 and 2 - <http://www.firstinspires.org/node/4271>

FIRST® Headquarters Support

Phone: 603-666-3906

Email: FTCTeams@firstinspires.org

FIRST Website: firstinspires.org

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

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

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

FIRST Tech Challenge Social Media

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

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

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

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

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

[FIRST Tech Challenge Google+](#) community - If you are on Google+, follow the FIRST Tech Challenge community for news updates.

Feedback

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

Appendix B: Bill of Material List

This list does not include the cell phones and the cables that connect the electronics.

Quantity	Name	Common Name
2	CORE_MOTOR_CONTROLLER	Core Motor Controller
1	CORE_POWER_DISTRIBUTION_MODULE	Core Power Distribution Module
1	CORE_SERVO_CONTROLLER	Core Servo Controller
10	SCREW_SHCS_6-32_X_1_25_2011	Gear Hub Screw
2	TETRIX_731901_120_2013	120-Tooth Gear
3	TETRIX_731903_2012	Motor Power Wire
2	TETRIX_736466_2012	4" Dual Omni Wheel Assembly
2	TETRIX_738000_2012	Motor Encoder
4	TETRIX_739028_2012	40-Tooth Gear
2	TETRIX_739055_2012	4" Wheel
1	TETRIX_739057_2012	Battery Pack
3	TETRIX_739060_2012	Single Standard Scale Servo Bracket
5	TETRIX_739061_2012	Flat Bracket
4	TETRIX_739062_2012	L Bracket
4	TETRIX_739065_2012	32mm Channel
3	TETRIX_739066_2012	96mm Channel
4	TETRIX_739067_2012	160m Channel
4	TETRIX_739068_2012	288mm Channel
1	TETRIX_739069_2012	416mm Channel
2	TETRIX_739070_2012	288mm Flat Bar
2	TETRIX_739071_2012	288mm Angle
2	TETRIX_739072_2012	144mm Angle
2	TETRIX_739073_2012	Flat Building Plate
3	TETRIX_739079_2012	Motor Hub
3	TETRIX_739083_2012	TETRIX DC Motor
2	TETRIX_739086_2012	80-Tooth Gear
6	TETRIX_739088_2013	Axle
3	TETRIX_739089_2012	DC Motor Mount
2	TETRIX_739090_2012	Gear Hub Spacer
17	TETRIX_739091_2013	Bronze Bushing
8	TETRIX_739092_2012	Axle Collar
130	TETRIX_739094_2013	Kep Nut
66	TETRIX_739097_2013	1/2" Socket Head Cap Screw
72	TETRIX_739098_2012	5/16" Socket Head Cap Screw
11	TETRIX_739100_2012	1/8" Axle Spacer
2	TETRIX_739101_2012	3/8" Axle Spacer
12	TETRIX_739102_2012	1" Stand-off
2	TETRIX_739103_2012	2" Stand-off
30	TETRIX_739111_2012	Button Head Cap Screw
4	TETRIX_739172_2012	Axle Hub
6	TETRIX_739195_2012	Motor Mount Screw

2	TETRIX_739197_2012	Standard-Scale HS-485HB Servo Motor
2	TETRIX_739272_2013	160mm Flat
4	TETRIX_739273_2013	96mm Flat
2	TETRIX_739274_2013	64mm Flat
4	TETRIX_739281_CORNER_BRA_2013	Inside Corner Bracket
2	39081	Servo Extension Wire

Appendix C: References

The right to use any unnumbered quoted text in this document has been graciously given by FIRST® for use in this document. It may not be used outside of this document without the consent of FIRST.

Appendix C: Project Team Profiles

	Isa Year 10 student at Great Mills High School. Student for 1 year on Team #2843, Under the Son
	Alanis Year 10 student at Great Mills High School. Student for 1 year on Team #2843, Under the Son
	Laura Year 10 student at Leonardtown High School. Student for 8 years on Team #2843, Under the Son
	Ian Year 12 student at Leonardtown High School. Student for 2 years on Team #2843, Under the Son

	Claudia Mentor for 2 years on Team #2843, Under the Son
	Lydean Coach for 9 years on Team #2843, Under the Son Volunteer for 9 years, MD FIRST ssi@the-spanglers.net
	David Mentor for 9 years on Team #2843, Under the Son Volunteer for 9 years, MD FIRST ssi@the-spanglers.net