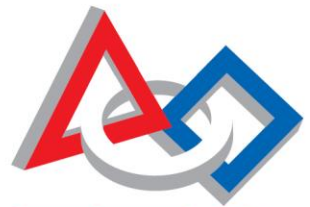


FIRST[®] Tech Challenge
PushBot Build Guide



FTC[®]

FIRST[®] Tech Challenge



Volunteer Thank You

Thank you for taking the time to volunteer for a FIRST Tech Challenge Event. FIRST and FTC 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 4,500 Teams competing annually, your dedication and commitment are paramount to the success of each Event and the FTC 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!

**Rockwell
Collins**

Official Program Sponsor

PTC®

CAD and Collaboration Sponsor

Revision History

Revision	Date	Description
1	8/15/2014	Initial Release – by FTC Team #003 Australia, The Southport School
2	9/1/2014	Replaced MATRIX with TETRIX content by former FTC Team #2843, Under the Son
3	8/6/2015	Updated using the new kit of parts and new programming environment.

Contents

Volunteer Thank You	2
Sponsor Thank You	2
Contents	3
Introduction.....	6
What is the FIRST Tech Challenge?	6
FIRST Tech Challenge (FTC) Core Values	6
What is the <i>FIRST</i> Tech Challenge PushBot Build Guide?	7
FTC Team #2843 United States – PushBot Robot.	7
Gracious Professionalism™.....	8
<i>FIRST</i> Youth Protection Program	9
Youth Protection Expectations and Guidelines.....	9
NOTICE OF NON-DISCRIMINATION	9
Building the Robot	10
Robot CAD Drawing.....	11
Cross Bar Assembly.....	12
Phone Holder Assembly.....	13
Tower Assembly	15
Arm Assembly.....	20
Drive Wheel Assembly	25
Omni Wheel Assembly.....	25
Right Chassis Rail Assembly	26
Left Chassis Rail Assembly.....	30
Robot Assembly.....	34
Install Encoders on Motors.....	47
Motor Hub and Gear Assembly	49
Drive Motor Assembly	49
Install Drive Motors on Robot.....	50
Programming the Robot.....	52

Opening Internet Explorer	53
Opening File Explorer (aka Windows Explorer)	54
Determining whether Android Studio can be installed	55
Installing the FTC SDK.....	62
Installing the Java Development Kit (JDK)	69
Installing the Android Studio Integrated Development Environment (IDE)	77
Registering the PushBot Op modes	101
Preparing the Robot Controller and Driver Station cell phones.....	104
Naming the Cell Phone	116
Enabling Debugging Mode on the Robot Controller Cell Phone	120
Installing the USB Device Driver	123
Deploying the Robot Controller	133
Deploying the Driver Station.....	135
Integrating the Op modes and Electronics	143
Building the electronics test bench	143
Configuring the USB devices	156
Assigning Gamepads	182
Pairing the Driver Station to the Robot Controller	184
Determining the drive wheel motor controller	194
Wiring the Robot.....	198
Testing the Robot	213
Running an Op mode	213
Using the gamepads	215
Adjusting the Servos	217
Final steps	218
Appendices.....	219
Appendix A: Bill of Materials List.....	220
Appendix B: References	222
Appendix C: Project Team Profiles	222
Appendix D: Resources & Support	224
FTC Game Manuals – Part I and II - http://www.usfirst.org/roboticsprograms/ftc/Game	224
FIRST Headquarters Support.....	224
USFIRST.ORG	224
FIRST Tech Challenge Social Media	224
Android-Based Technology Support.....	224
Product Support	224

Feedback 224

Introduction

What is the 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 Lexan 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 FTC and other *FIRST* Robotics competitions, visit www.usfirst.org.

FIRST Tech Challenge (FTC) Core Values

FIRST asks everyone who participates in FTC to uphold and promote 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 the *FIRST* Tech Challenge.
- We inspire others to adopt these values.

FTC is More Than Robots! While competing, students develop personal and professional skills they will be able to rely on throughout their life.



What is the *FIRST*Tech Challenge PushBot Build Guide?

In early 2014, a design challenge was set by *FIRST* for experienced FTC Teams to design a basic Robot and manual for Rookie FTC 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.

PushBot 2 (PB2) needs to meet all FTC 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 needed to have basic programming for autonomous mode; as well as a program for manual mode. The code was to be developed in the Java programming language and deployed using the Android Studio IDE.

Depending on the 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.

FTC Team #2843 United States – PushBot Robot.

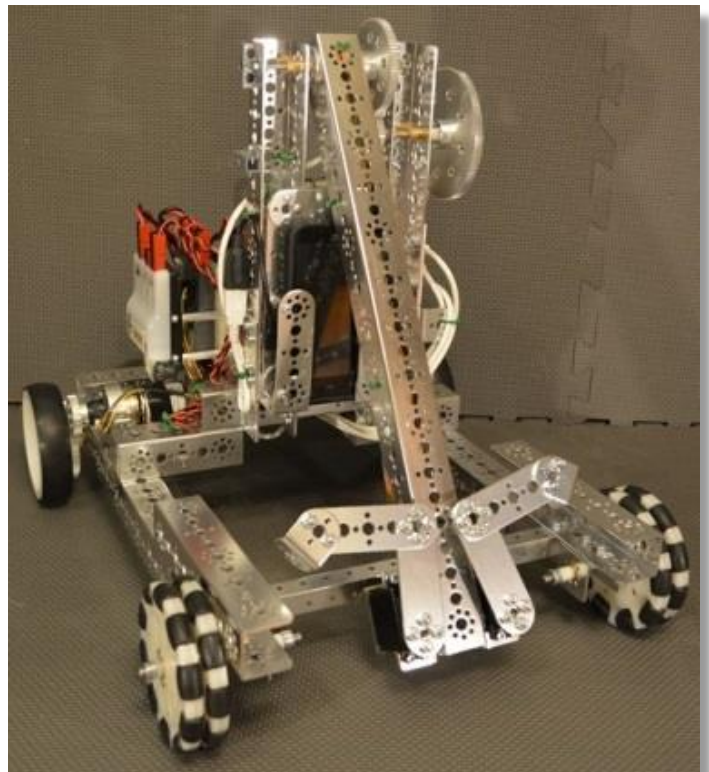
2015 – 2016 Robot Name: PB2 (PushBot 2)

Purpose: Assist new FTC competitors to develop a sense of how an FTC 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. PB2 can push objects using its gripper or a front panel can be installed to 'push' objects. This Guide will assist rookie FTC 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 2015-2016 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 [FTC Team Resources webpage](#).



Gracious Professionalism™

FIRST uses this term to describe the program's 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 everyone 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.

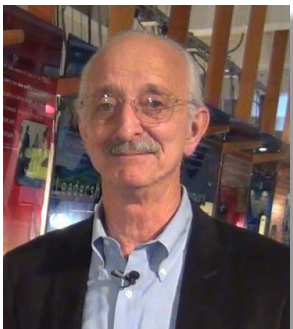
An example of **Gracious Professionalism** is sharing information and resources with other Teams whenever possible.

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***

FIRST 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.



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

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.usfirst.org/aboutus/youth-protection-program>

Information on the US Screening process is available here: http://www.usfirst.org/sites/default/files/uploadedFiles/About_Us/U

[S-Youth-Protection-Clearance-Process.pdf](#)

Information on the Canadian Screening process is available here:

http://www.usfirst.org/sites/default/files/uploadedFiles/About_Us/Canadian-Youth-Protection-Clearance.pdf

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

NOTICE OF NON-DISCRIMINATION

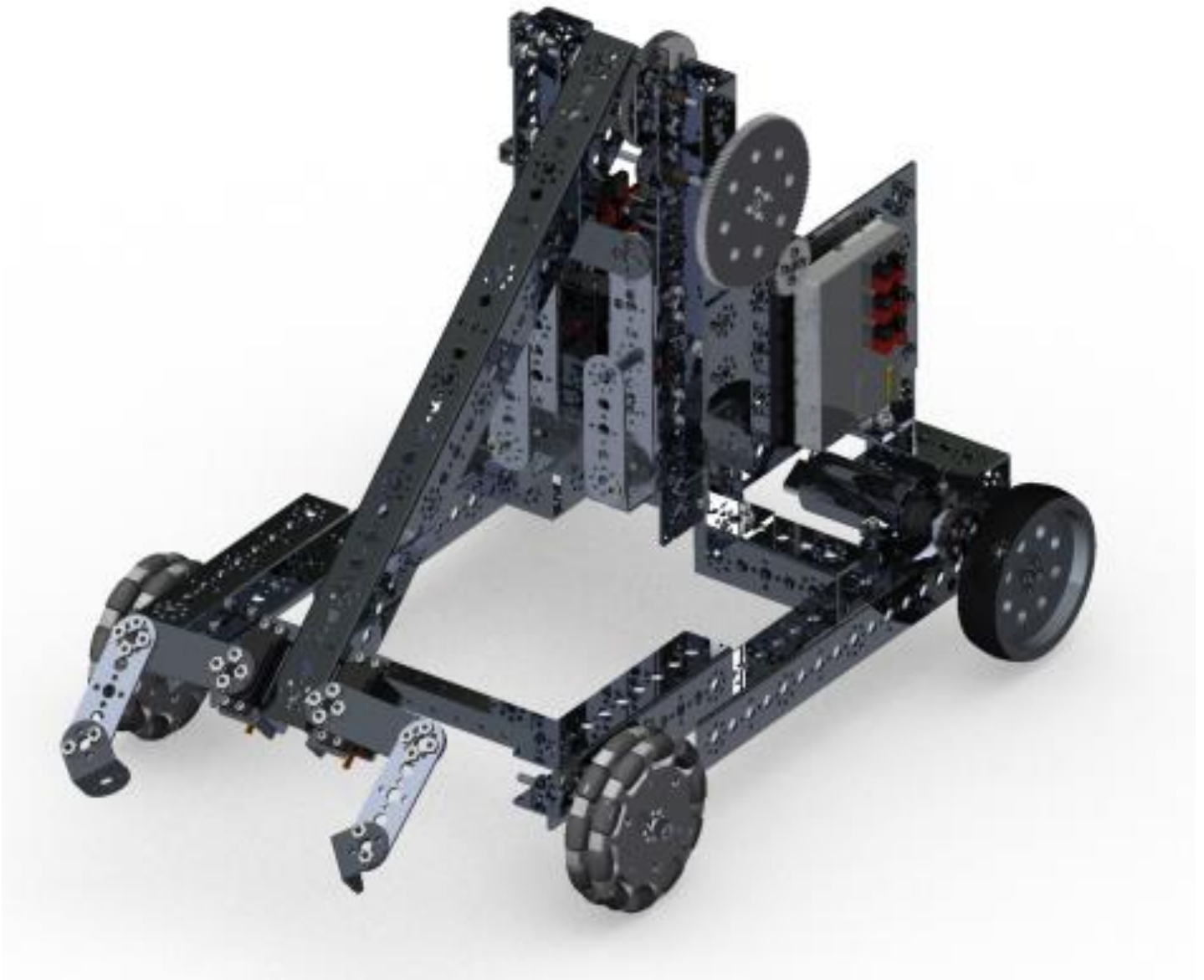
United States Foundation for Inspiration and Recognition of Science and Technology (*FIRST*®) does not discriminate on the basis of race, color, national origin, sex, disability, or age in its programs and activities. The following person has been designated to handle inquiries regarding the non-discrimination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

Building the Robot

This section will outline the construction of the chassis, sensors, and electronics. The chassis is constructed using the TETRIX Kit of Parts. The standard tools included in the Kit of Parts set will be needed to build the chassis. All parts needed for the construction are listed in the instructions (most are shown below). Make sure that when using an axle hub, motor hub, or axle collar that the set screw is installed. Also, make sure that all assemblies are square. It is hard to drive a crooked Robot straight!



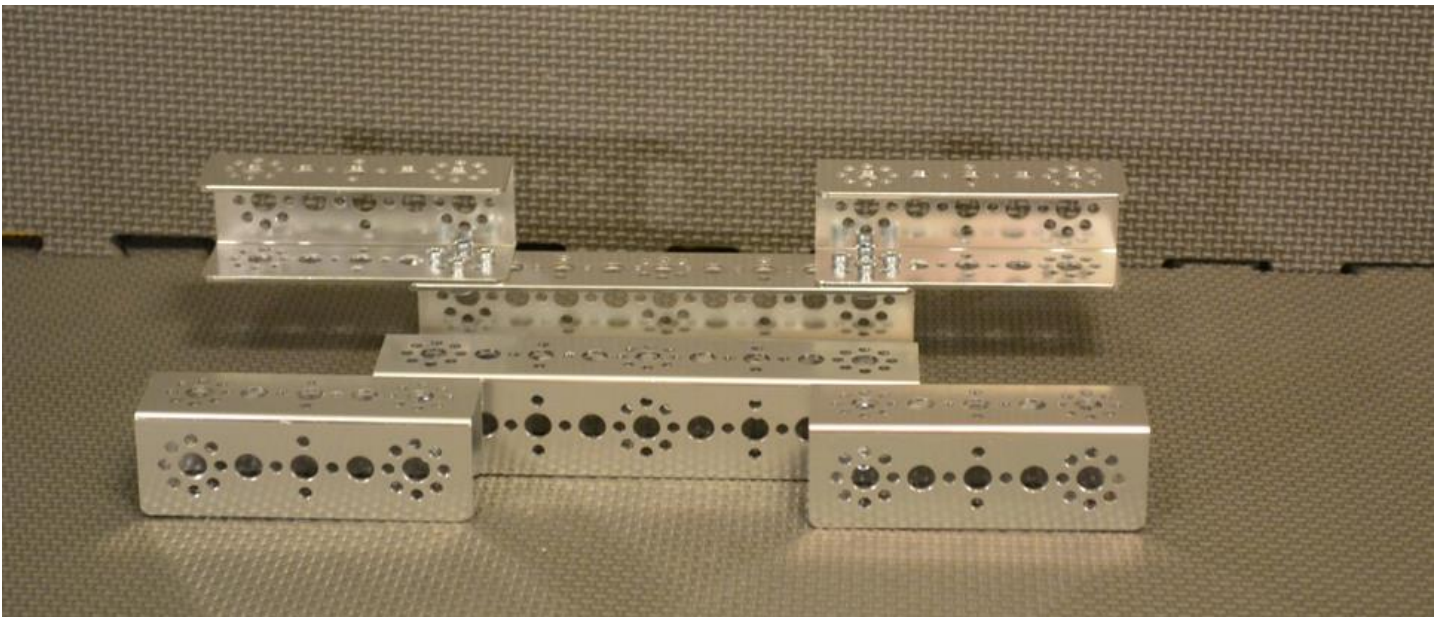
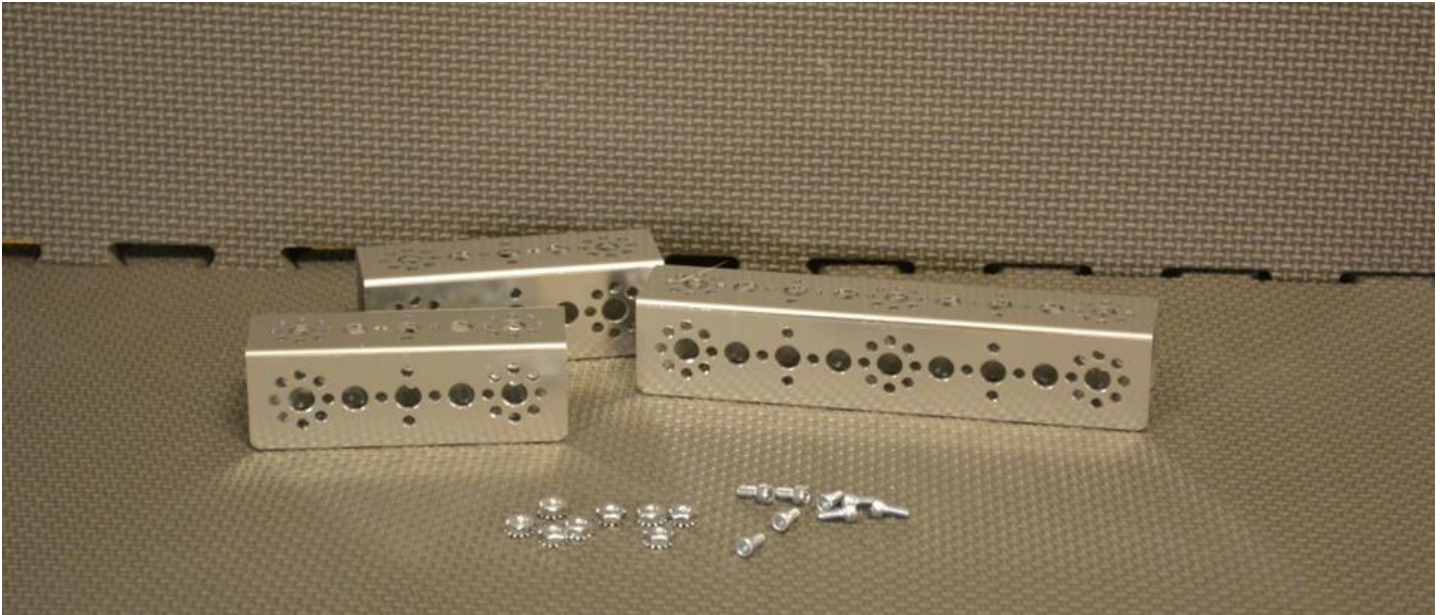
Robot CAD Drawing



Cross Bar Assembly

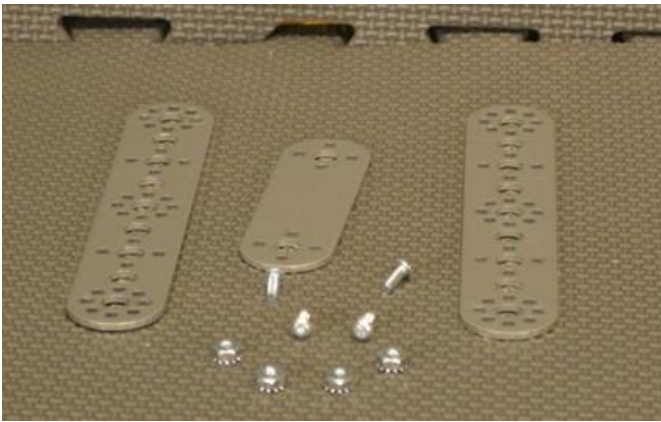
Make 2 copies

Step 1: 160mm channel (1), 96mm channels (2), 5/16" socket head cap screws (8) keps nuts (8). The second picture shows BOTH completed assemblies.

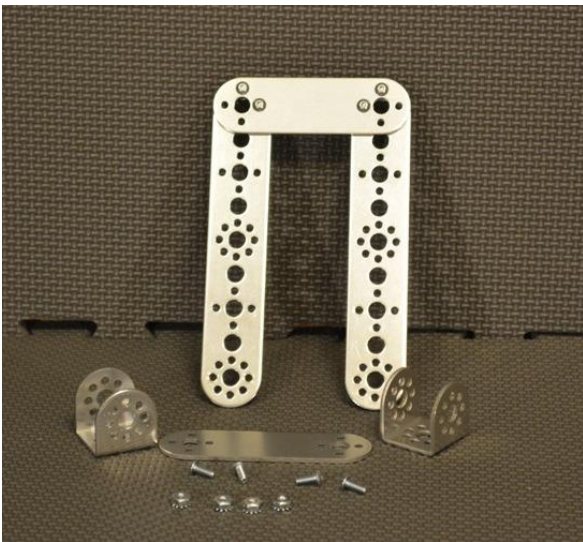


Phone Holder Assembly

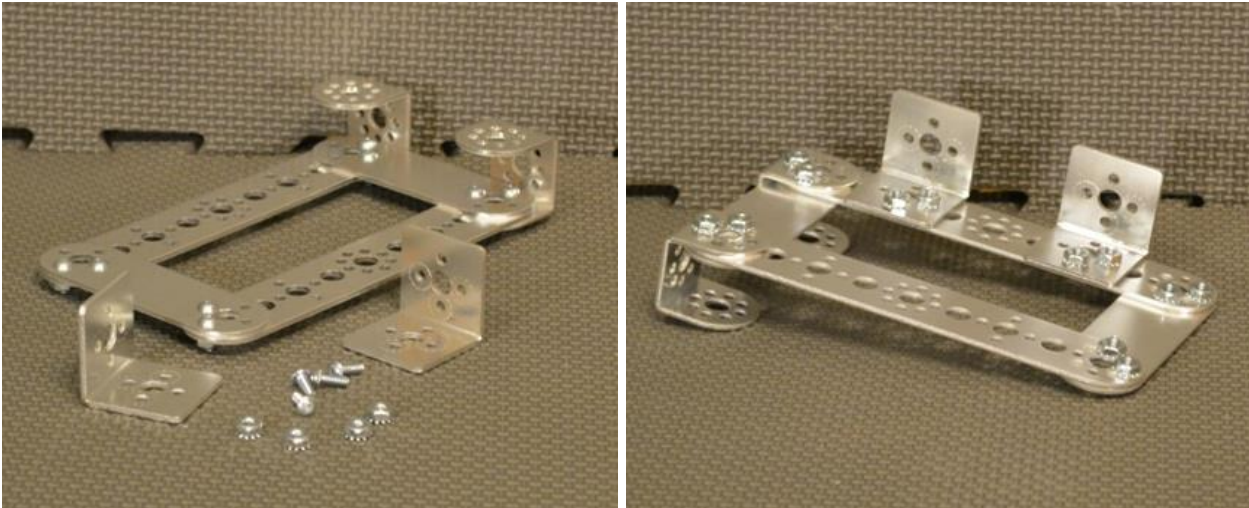
Step 1: 160mm flats (2), flat bracket (1), button head cap screws (4), keps nuts (4). The second picture shows the completed step.



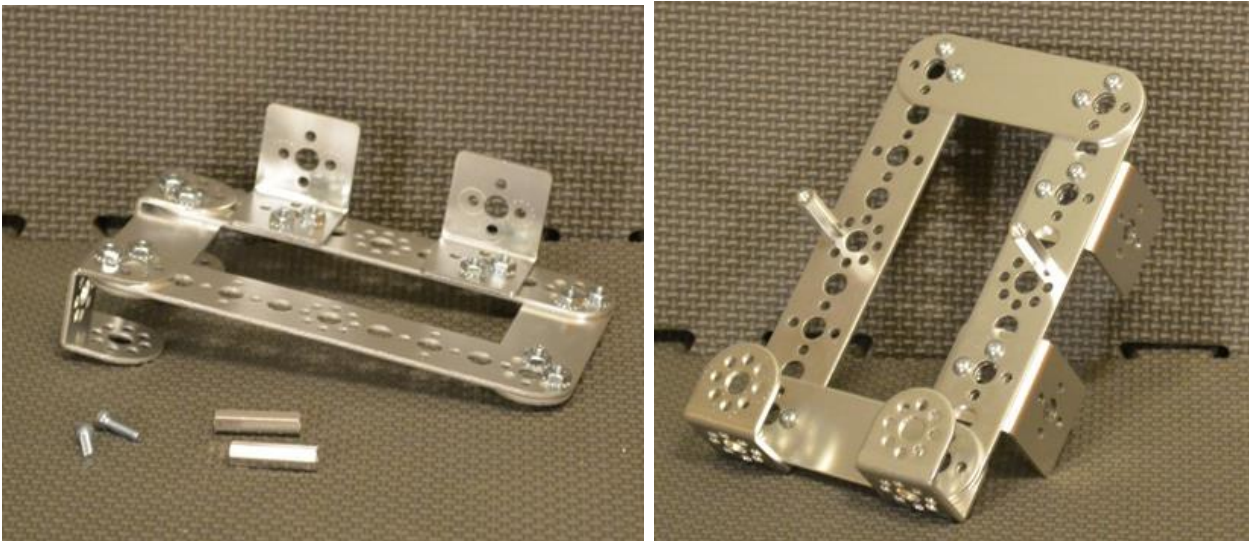
Step 2: assembly from step 1, flat bracket (1), inside C connector (2), button head cap screws (4), and keps nuts (4). The second picture shows the completed step. Note: the inside C connector is below the flat.



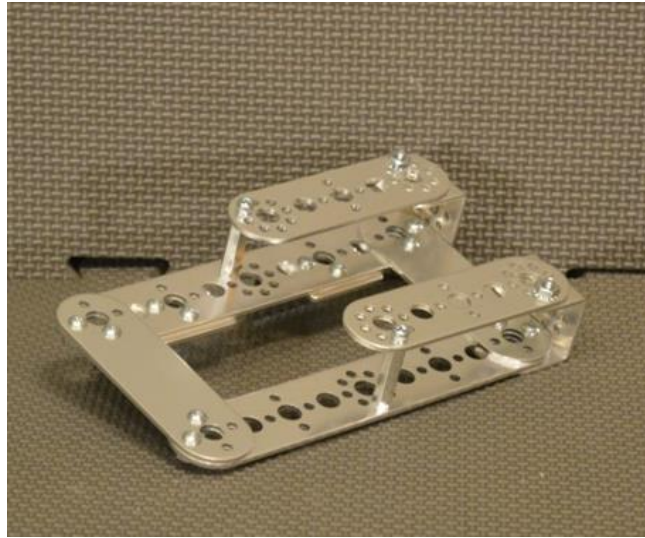
Step 3: assembly from step 2, inside corner bracket (2), button head cap screws (4), and keps nuts (4). The second picture shows the completed step.



Step 4: assembly from step 3, 1" stand-off posts (2), 1/2" socket head cap screws (4). The second picture shows the completed step. Note: the stand-off posts are not placed symmetrically.

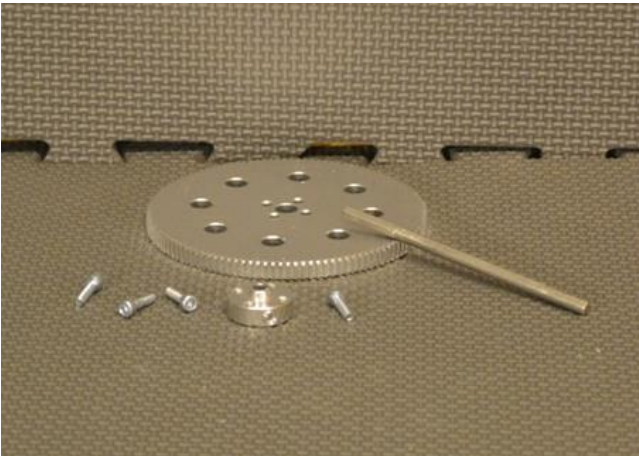


Step 5: assembly from step 4, 96mm flats (2), 5/16" socket head cap screws (2), button head cap screws (2), and keps nuts (2). The second picture shows the completed phone holder assembly.

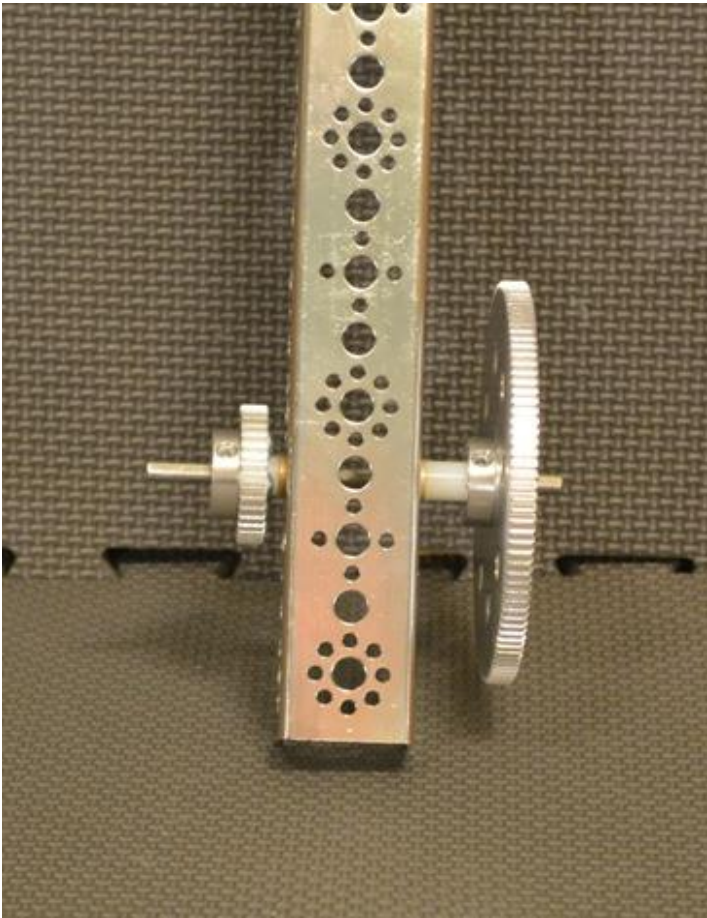
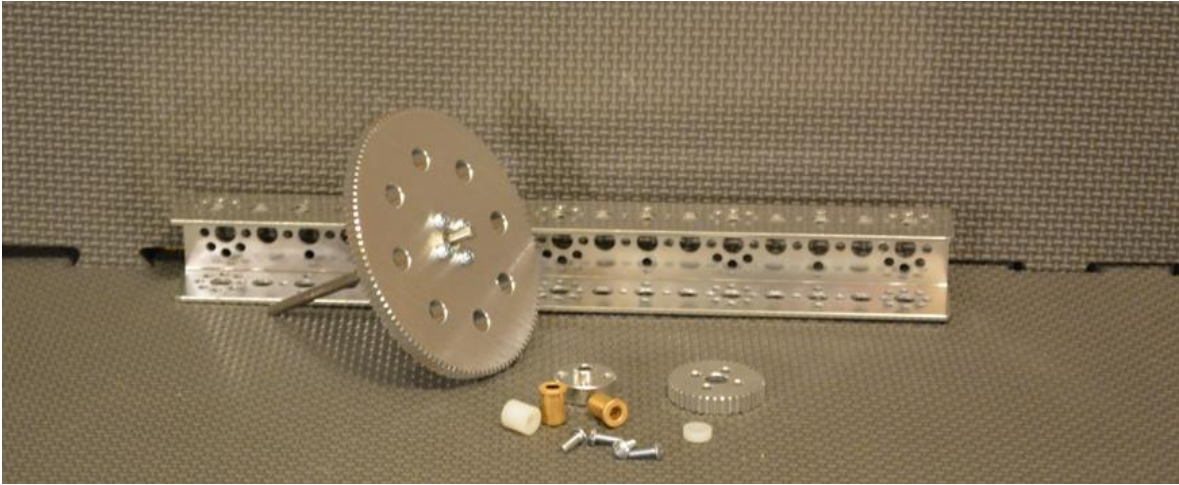


Tower Assembly

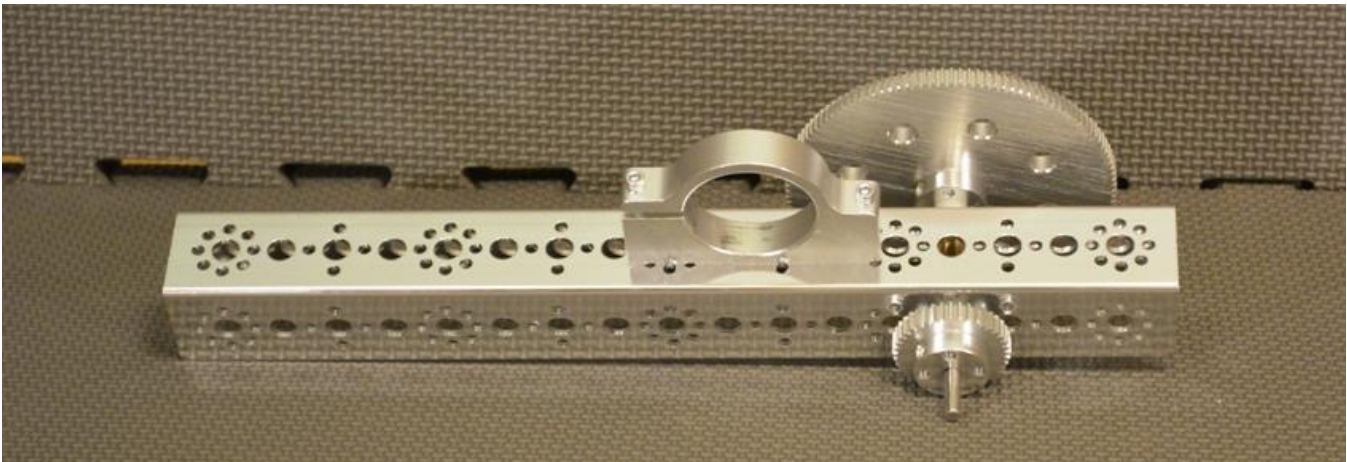
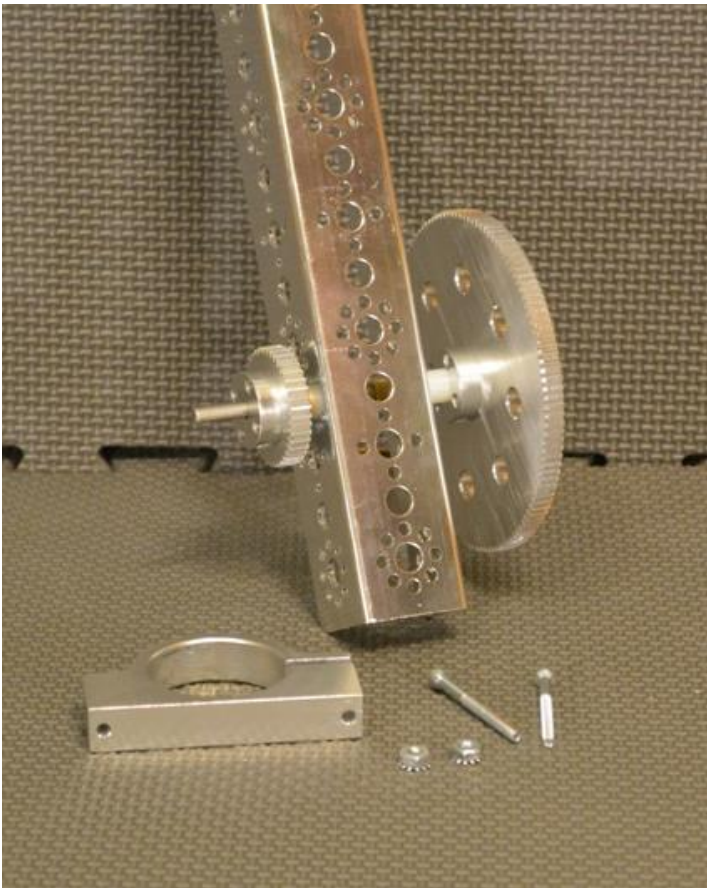
Step 1: 120-tooth gear (1), axle hub (1), 1/2" socket head cap screws (4), axle (1). The second picture shows the completed step.



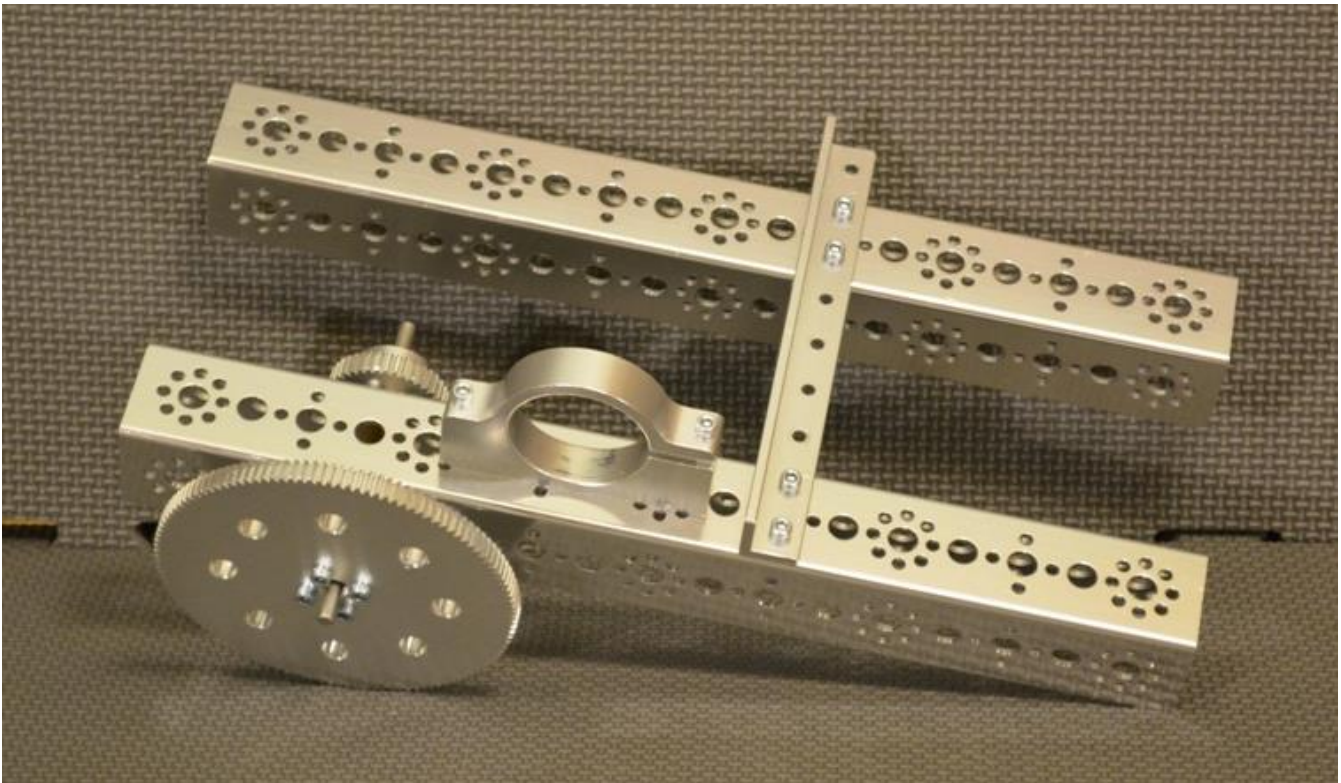
Step 2: assembly from step 1, 288mm channel, 3/8" axle spacer (1), bronze bushing (2), 1/8" ax axle el spacer (1), 40-tooth gear (1), axle hub (1), and button head cap screws (4). The second picture shows the completed step. Note the order on the axle. 120-tooth gear, axle hub, 3/8" axle spacer, bronze bushing, channel, bronze bushing, 1/8" spacer, 40-tooth gear, and axle hub.



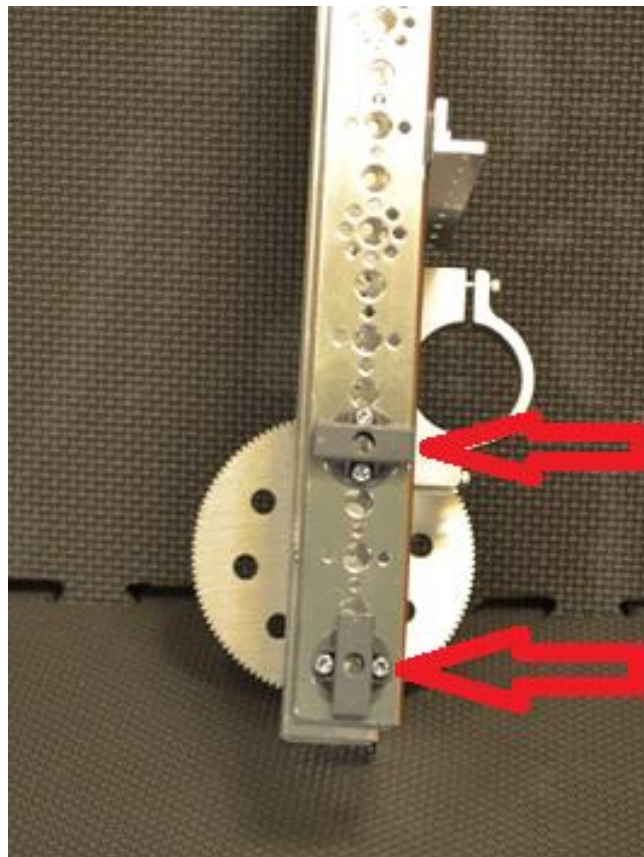
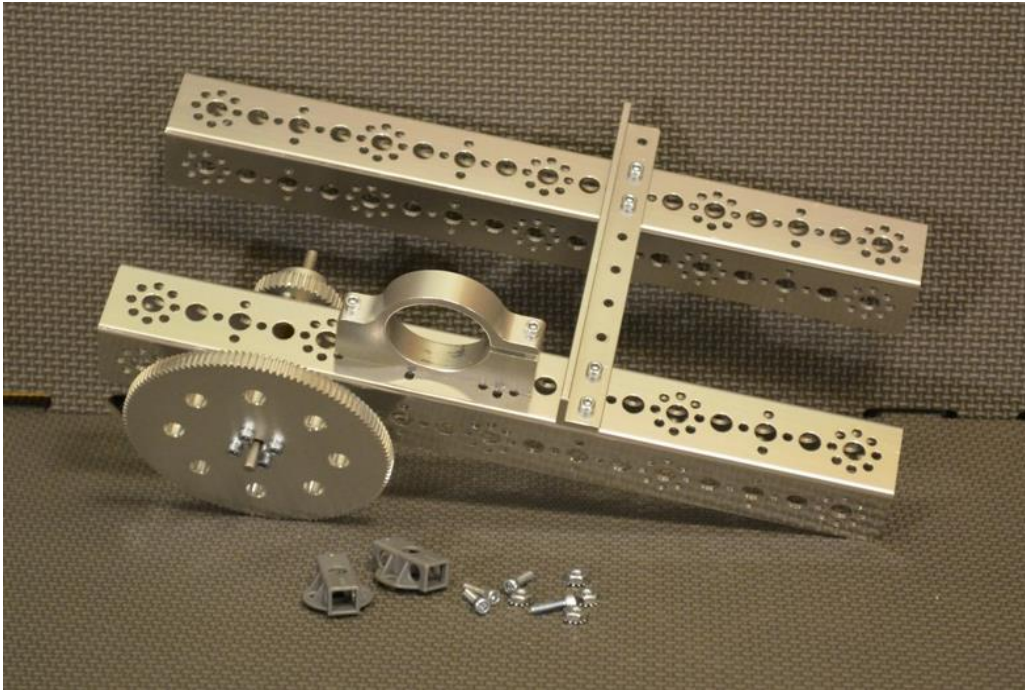
Step 3: assembly from step 2, motor mount (1) with included screws (2) and keps nuts (2). The second picture shows the completed step.



Step 4: assembly from step 3, 288mm channel (1), 144mm angle (1), 1/2" socket head cap screws (4), and keps nuts (4). The second picture shows the completed step.

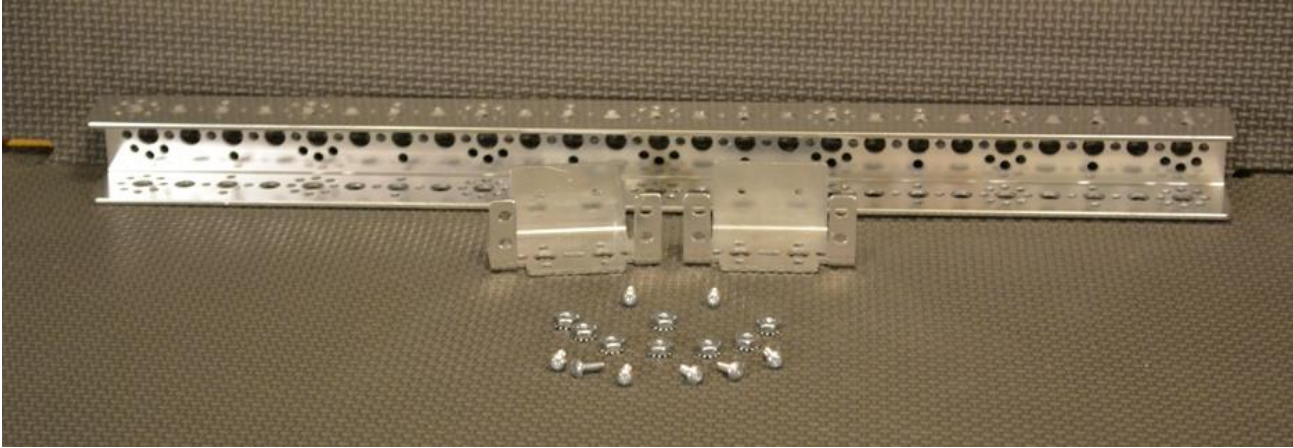


Step 5: assembly from step 4, hard point connectors (2), 1/2" socket head cap screws (4), and keps nuts (4). The second picture shows the completed tower assembly.

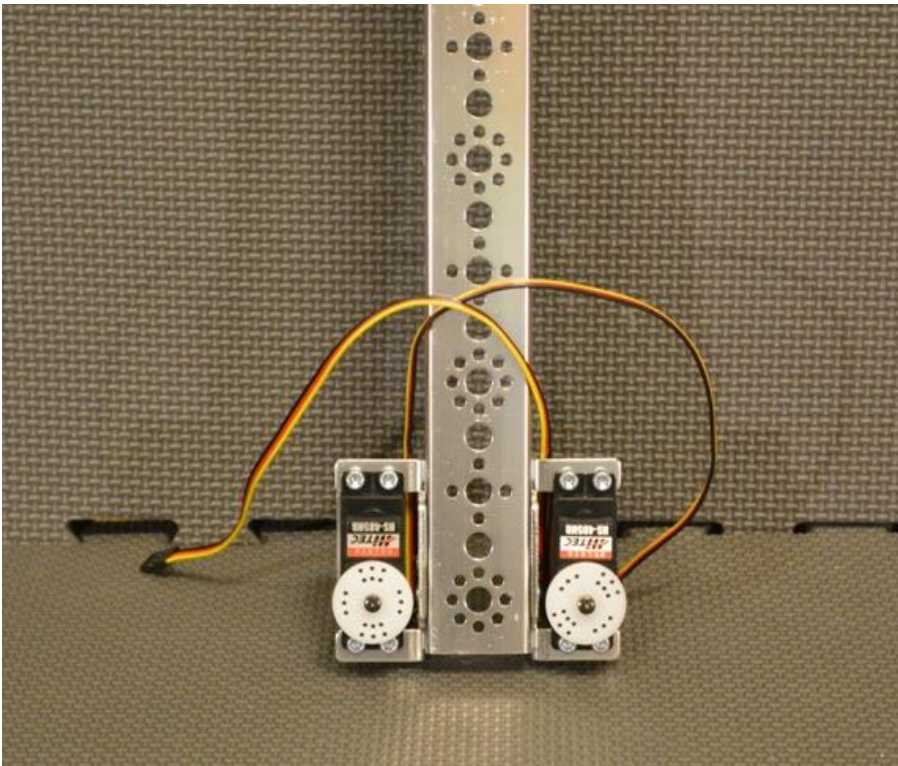
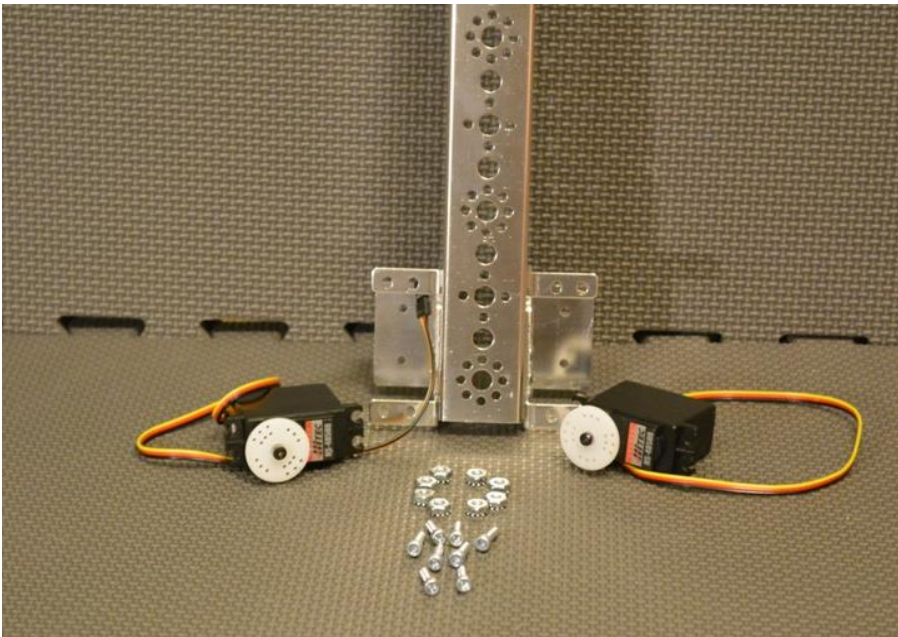


Arm Assembly

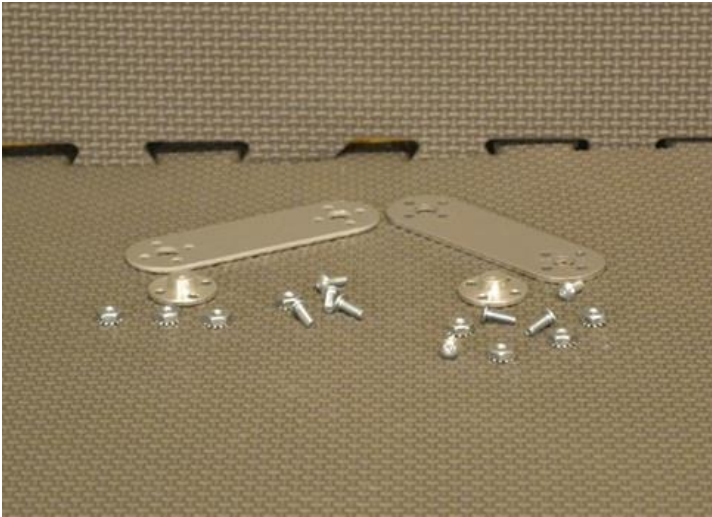
Step 1: 416mm channel (1), single standard-scale servo motor mounting bracket (2), button head cap screws (8), and keps nuts (8). The second picture shows the completed step.



Step 2: assembly from step 1, 180° standard-scale HS-485HB servo motor (2), 1/2" socket head cap screws (8), and keps nuts (8). The second picture shows the completed step.



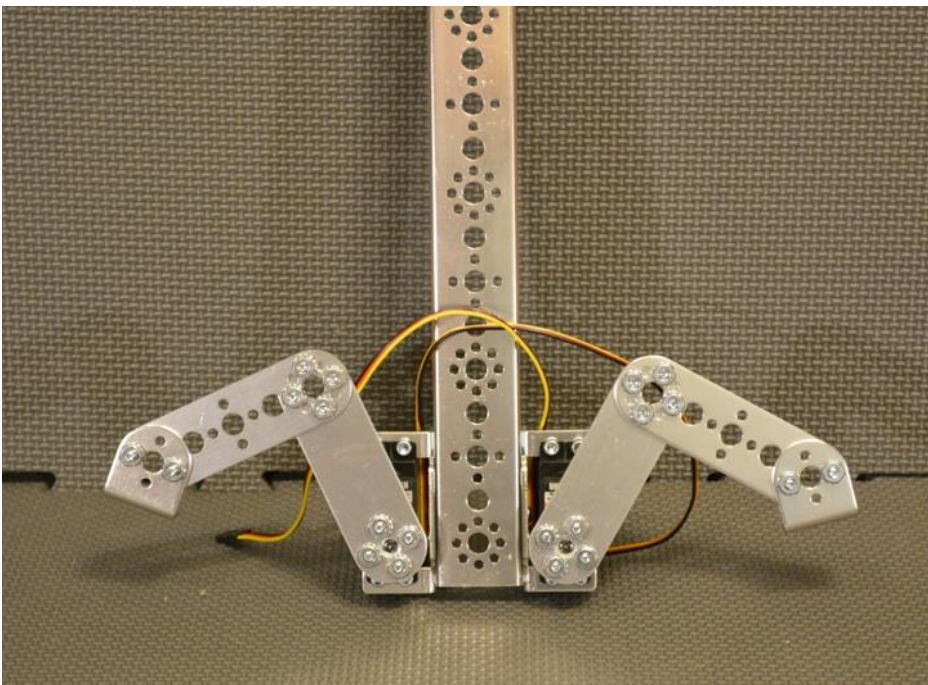
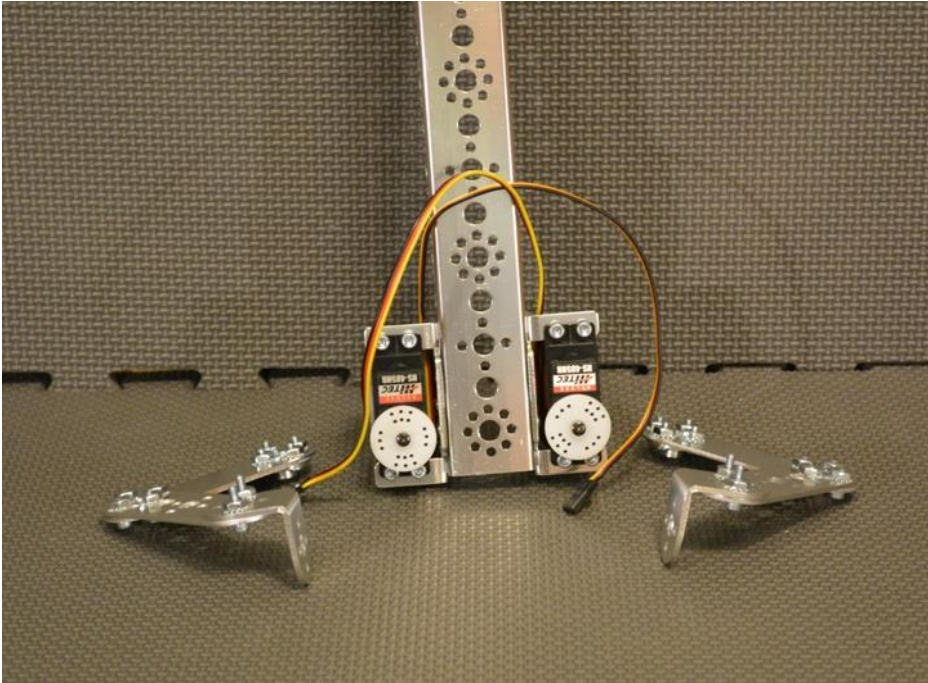
Step 3: servo horn - metal (2), flat bracket (2), button head cap screws (8), and keps nuts (8). The second picture shows the completed step.



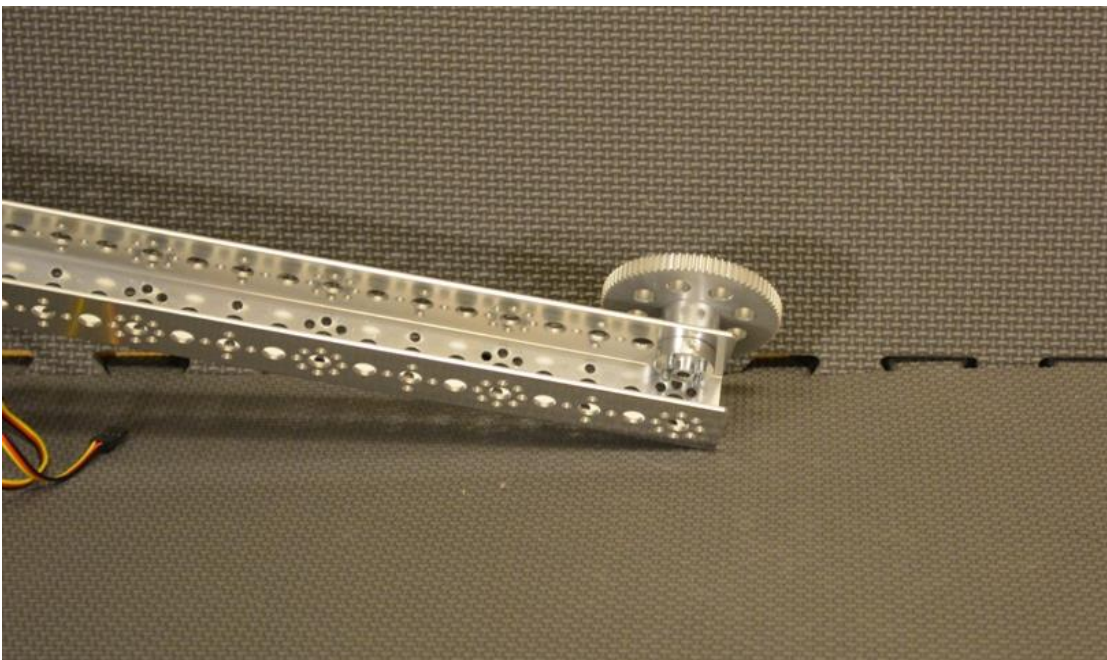
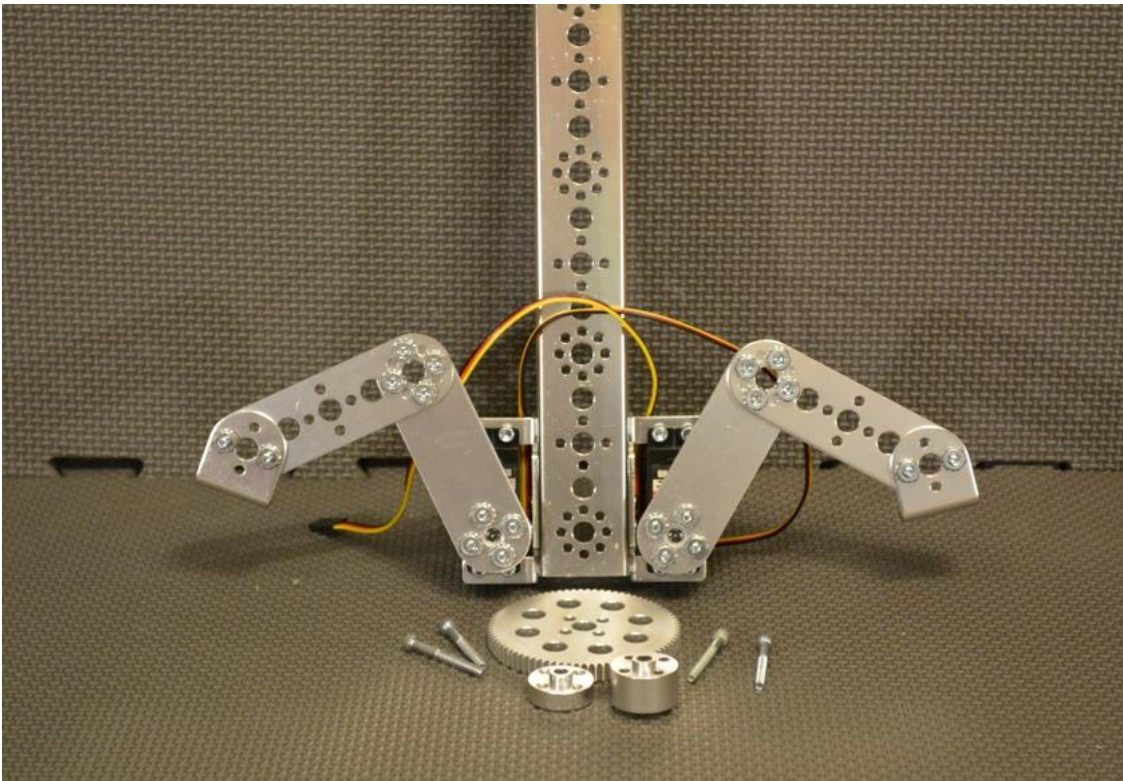
Step 4: assembly from step 3, 96mm flats (2), L bracket (2), 5/16" socket head cap screws (8), 1/2" socket head cap screws (4), and keps nuts (12). The second picture shows the completed step.



Step 5: assembly from step 2, assemblies from step 4. Remove the black screw from the center of the plastic servo horn. Use this screw to attach the grippers. To set the position of the grippers, rotate the servo to the fully open position and install the gripper so that it is opened just past vertical. This should get the servo in approximately the correct position. The second picture shows the completed step.



Step 6: assembly from step 5, 80-tooth gear (1), gear hub spacer (1) with included screws (4), and axle hub (1). The second picture shows the completed arm.



Drive Wheel Assembly

Make 2 copies

Step 1: 4" wheel (1), gear hub spacer (1) with included screws (4), axle hub (1), and axle (1). The order is wheel, gear hub spacer, gear then axle hub. The second picture shows the two completed assemblies.

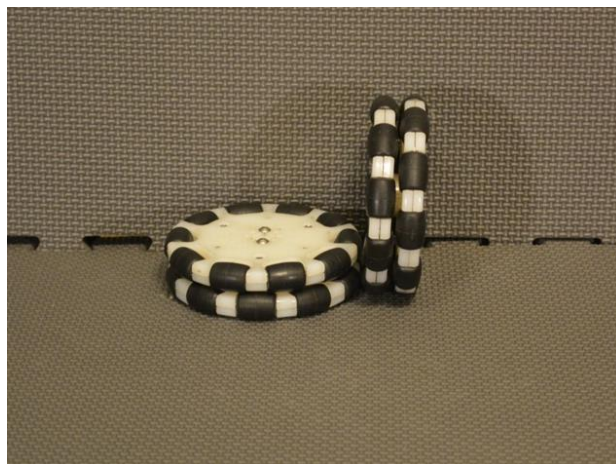


Omni Wheel Assembly

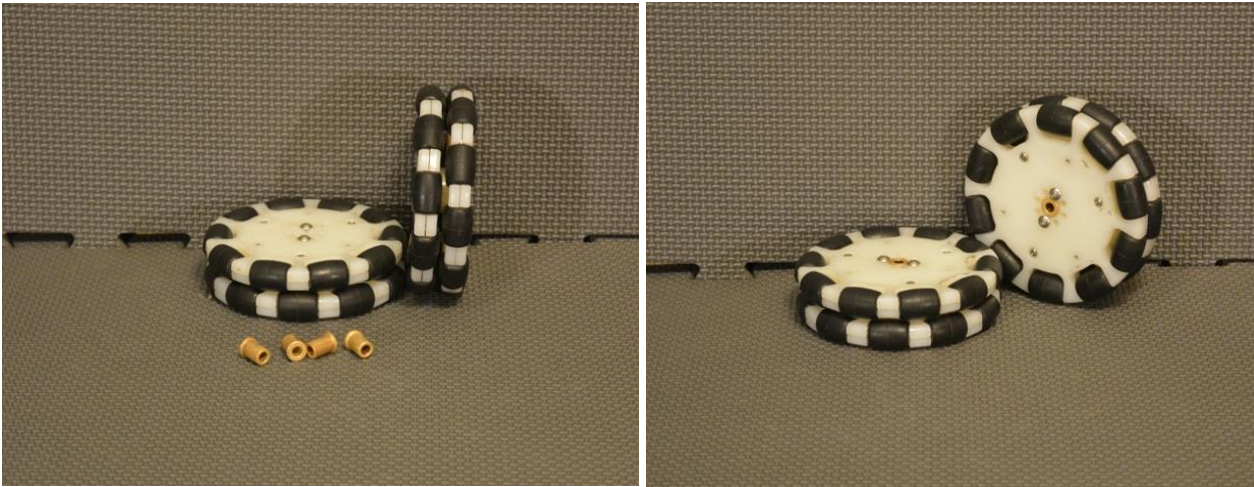
Make 2 copies

These are assembled as per the instructions that come in the omni wheel pack.

Step 1: 4" omni wheel half (2), joining ring (1) with included screws (4). The order is omni wheel, joining ring, omni wheel. The second picture shows the two completed assemblies.

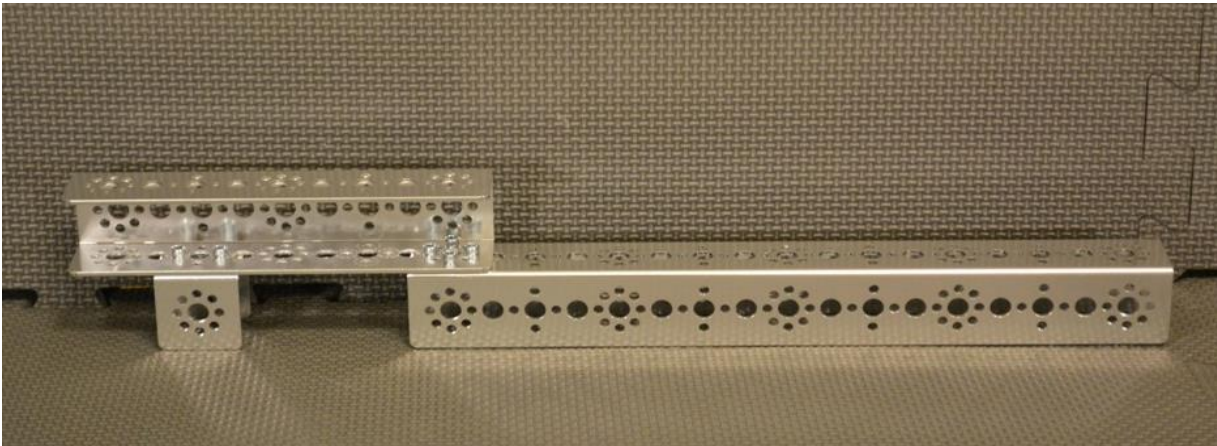
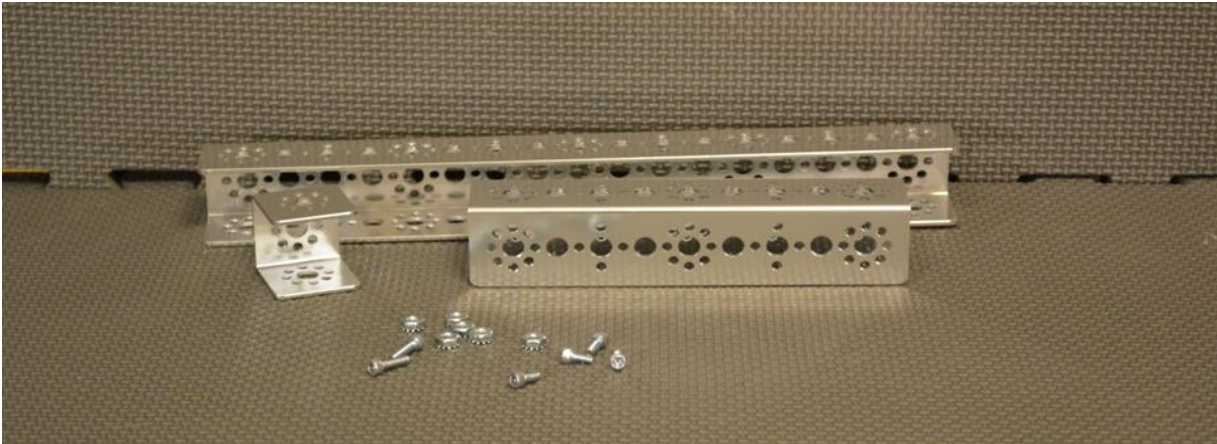


Step 2: assemblies from prior step and bronze bushings (4). The second picture shows the two completed assemblies.



Right Chassis Rail Assembly

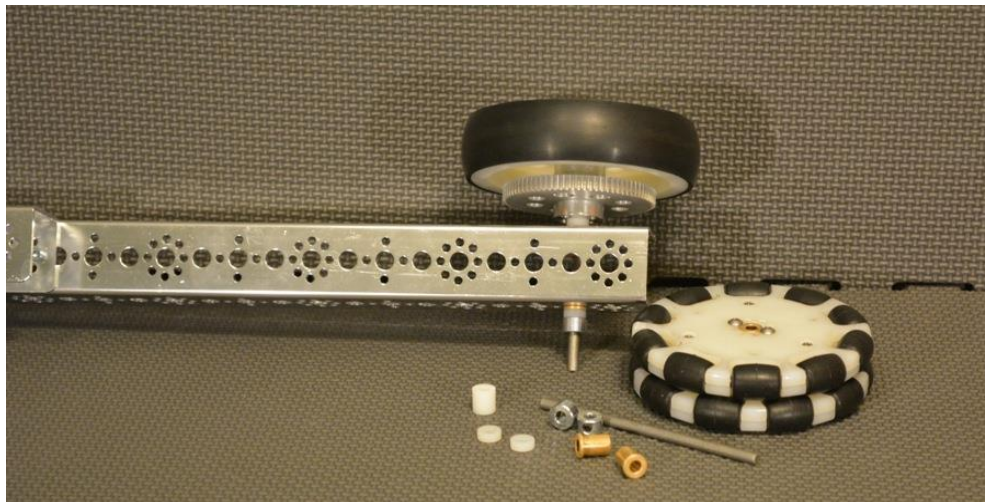
Step 1: 32mm channel (1), 160mm channel (1), 288mm channel (1), 5/16" socket head cap screws (4), 1/2" socket head cap screws (2), and keps nuts (6). The second picture shows the completed step.



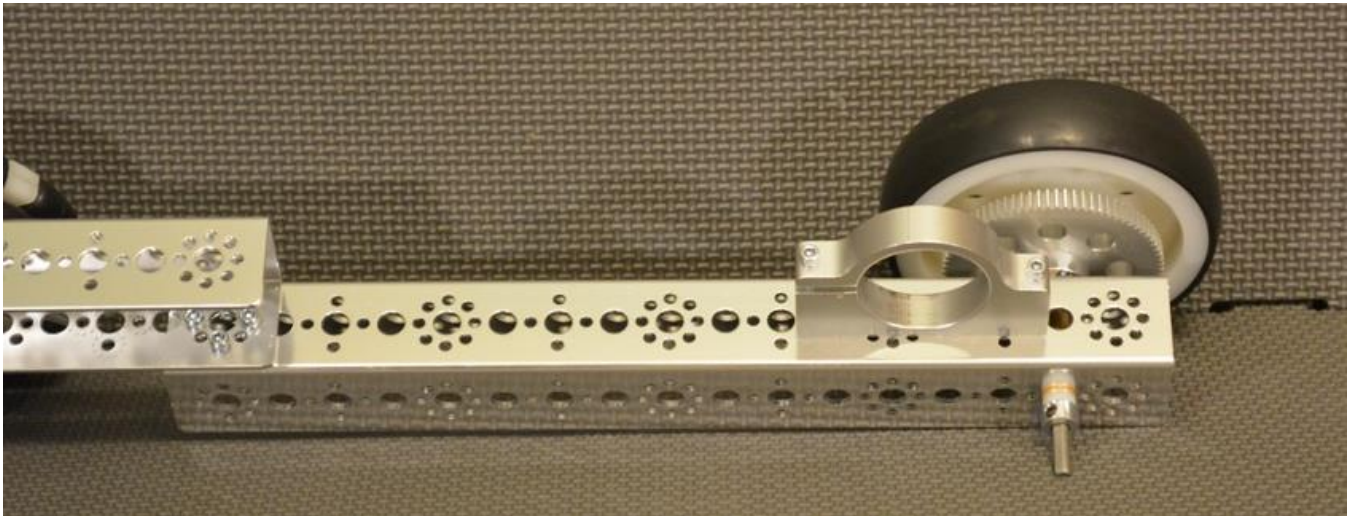
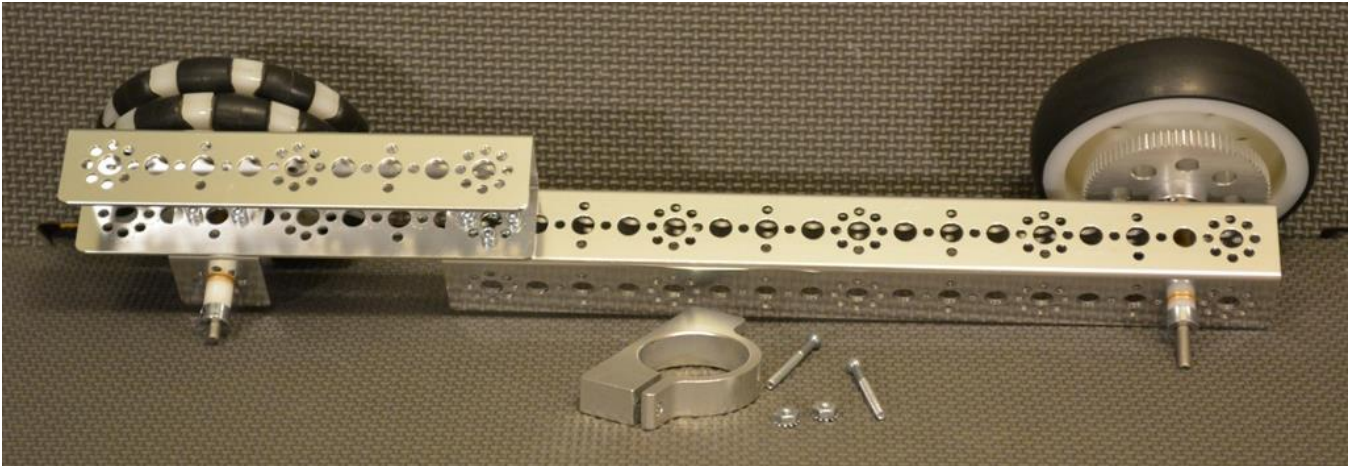
Step 2: assembly from step 1, drive wheel assembly (1), 1/8" axle spacer (2), axle collar (1), and bronze bushing (2). Note: assembly order is drive wheel assembly, 1/8" spacer, bronze bushing, channel, bronze bushing, 1/8" spacer, and axle collar. The second picture shows the completed step.



Step 3: assembly from step 2, omni wheel assembly (1), 1/8" axle spacer (2), 3/8" axle spacer, axle collar (2), bronze bushing (2), and axle (1). Note: assembly order is axle collar, 1/8" spacer, omni wheel assembly, 1/8" spacer, bronze bushing, channel, bronze bushing, 3/8" spacer, and axle collar. The second picture shows the completed step.



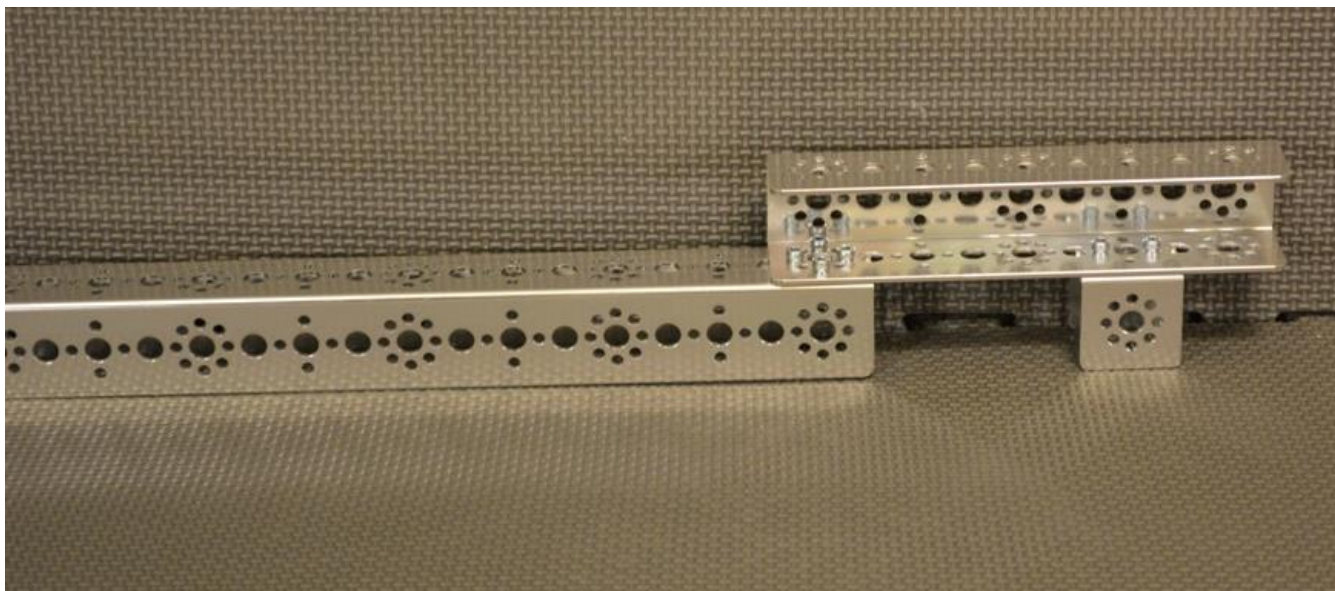
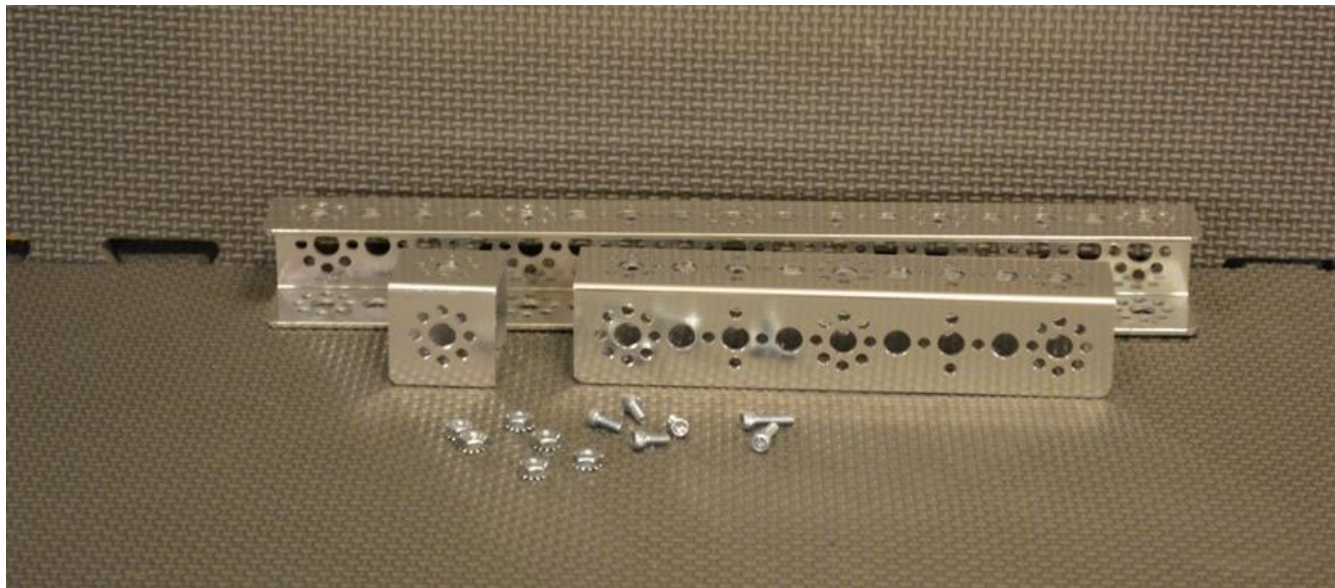
Step 4: assembly from step 3, motor mount (1) with included screws (2), and keps nuts (2). The second picture shows the completed right chassis rail.



Left Chassis Rail Assembly

This is built as a mirror image to the right chassis rail assembly.

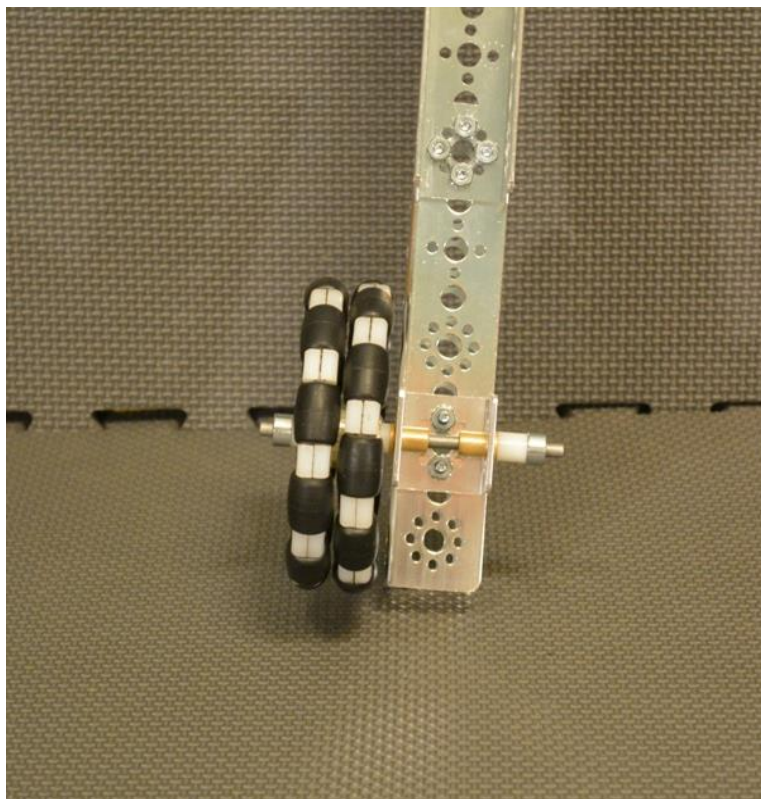
Step 1: 32mm channel (1), 160mm channel (1), 288mm channel (1), 5/16" socket head cap screws (4), 1/2" socket head cap screws (2), and keps nuts (6). The second picture shows the completed step.



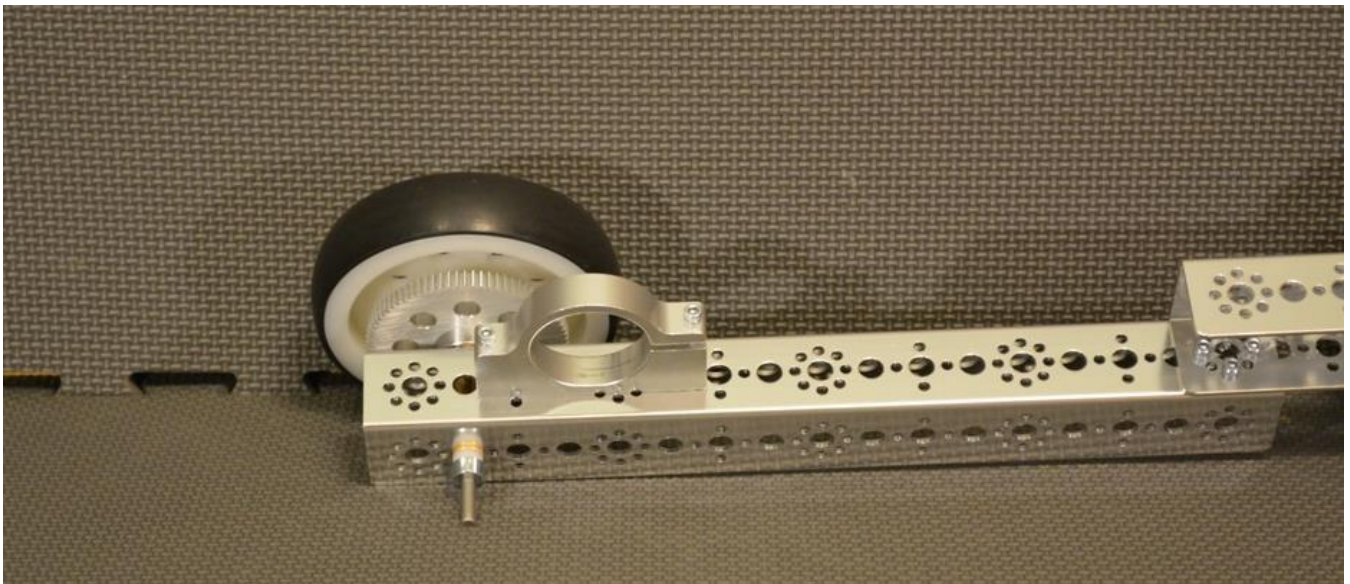
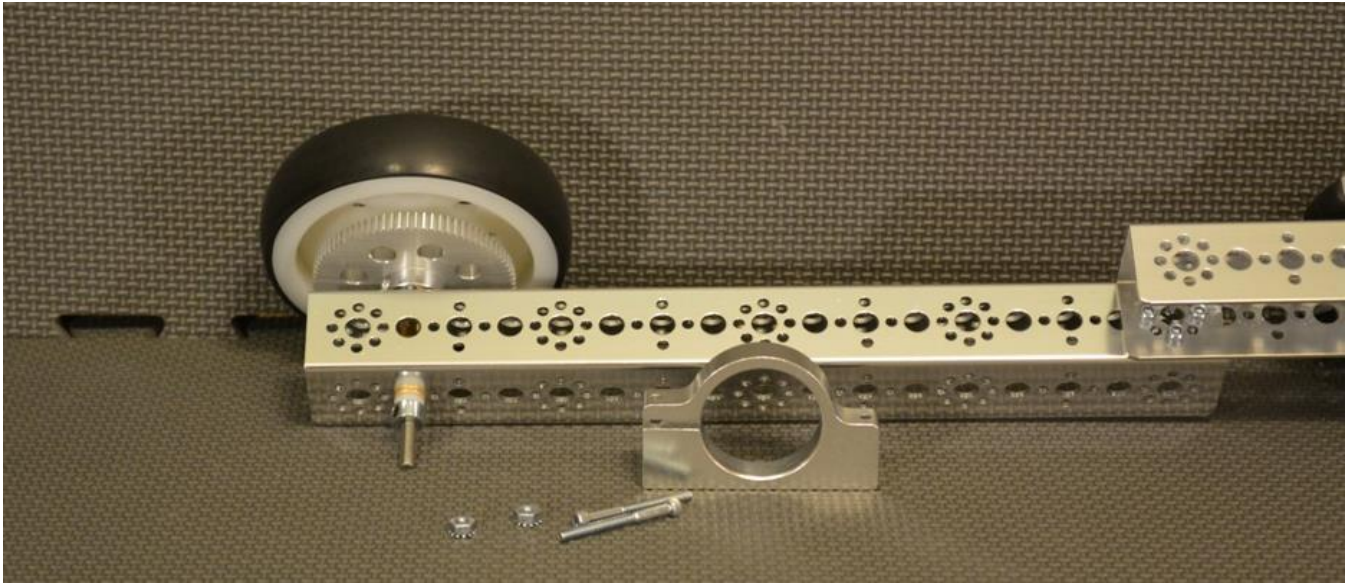
Step 2: assembly from step 1, drive wheel assembly (1), 1/8" axle spacer (2), axle collar (1), and bronze bushing (2). Note: assembly order is drive wheel assembly, 1/8" spacer, bronze bushing, channel, bronze bushing, 1/8" spacer, and axle collar. The second picture shows the completed step.



Step 3: assembly from step 2, omni wheel assembly (1), 1/8" axle spacer (2), 3/8" axle spacer, axle collar (2), bronze bushing (2), and axle (1). Note: assembly order is axle collar, 1/8" spacer, omni wheel assembly, 1/8" spacer, bronze bushing, channel, bronze bushing, 3/8" spacer, and axle collar. The second picture shows the completed step.

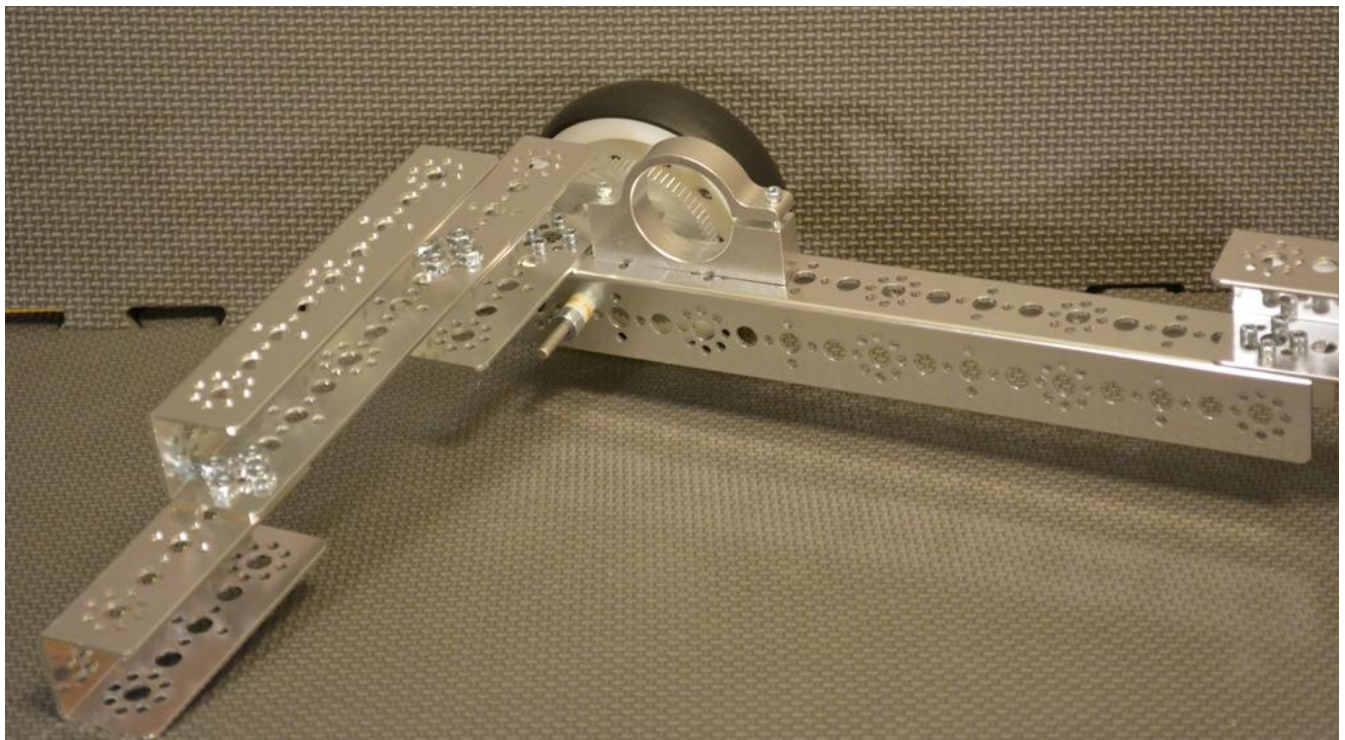
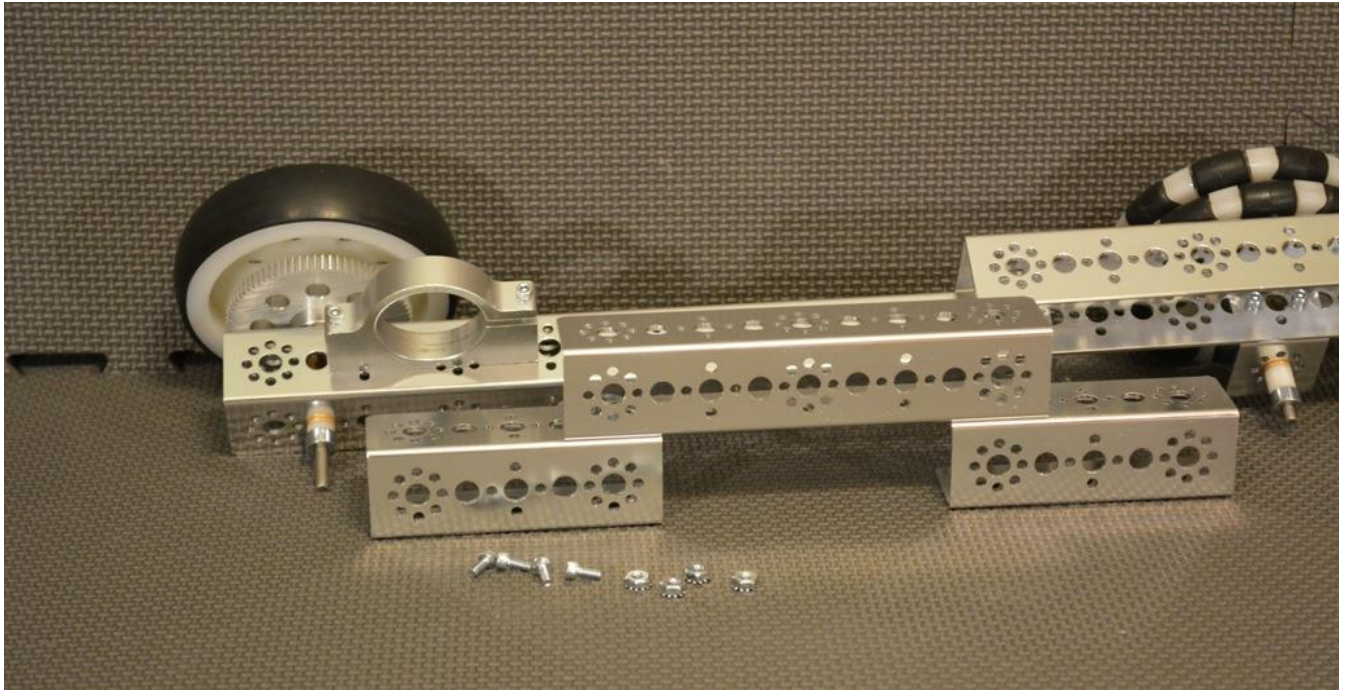


Step 4: assembly from step 3, motor mount (1) with included screws (2), and keps nuts (2). The second picture shows the completed left chassis rail.

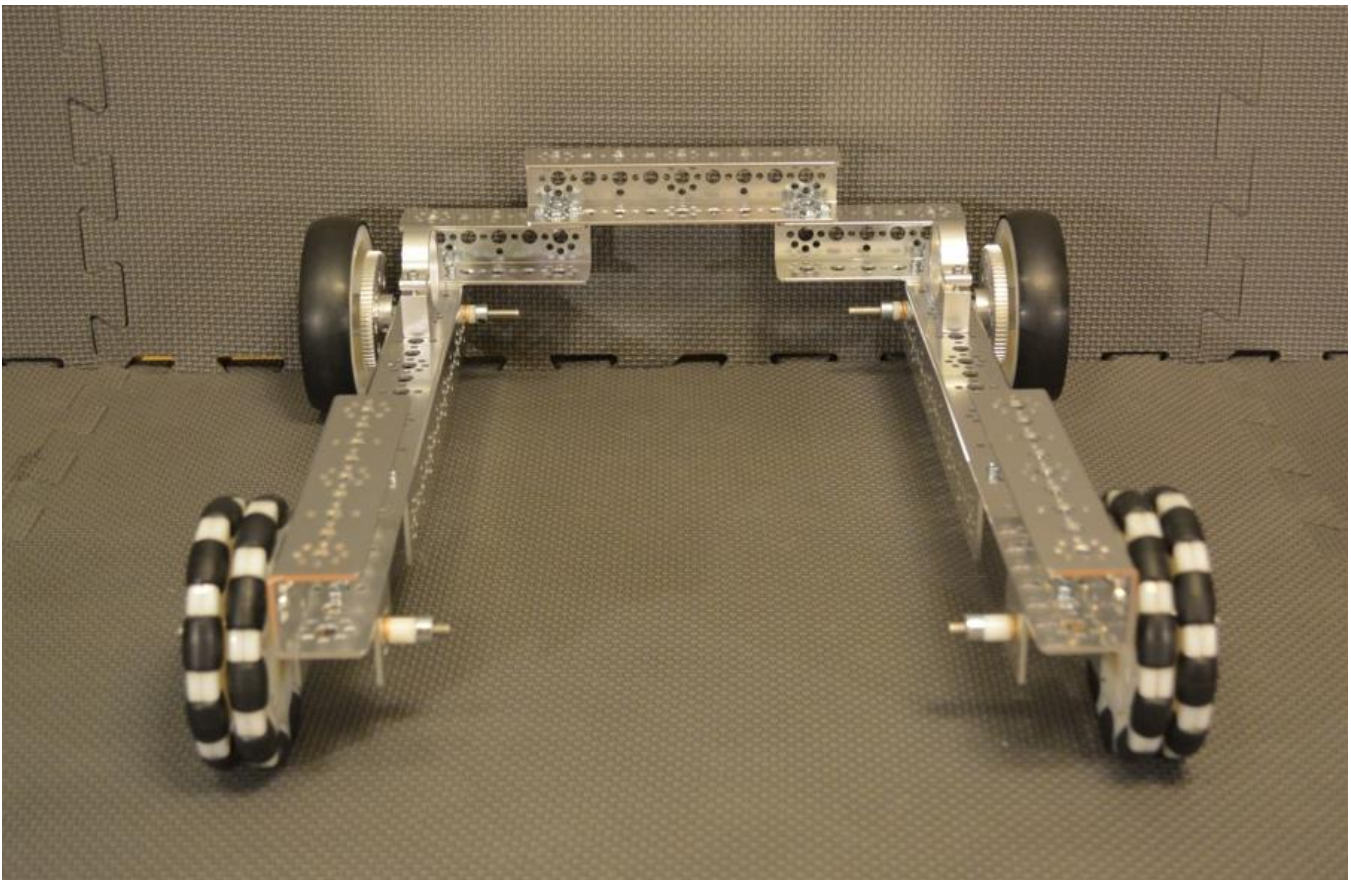
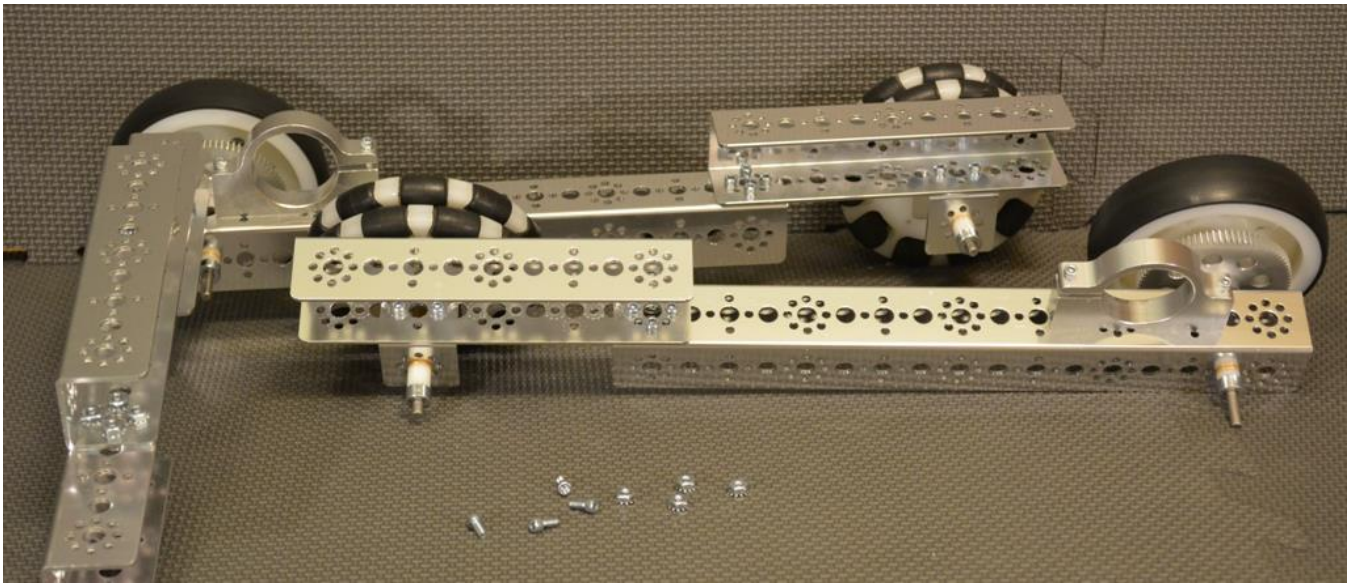


Robot Assembly

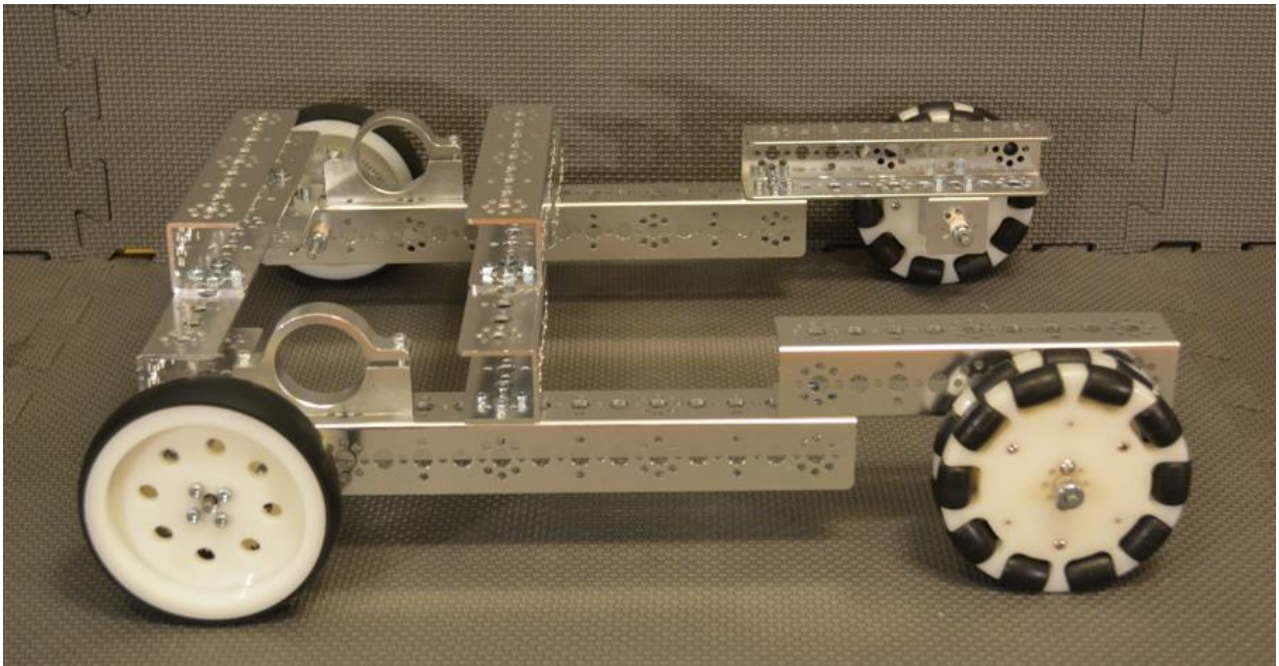
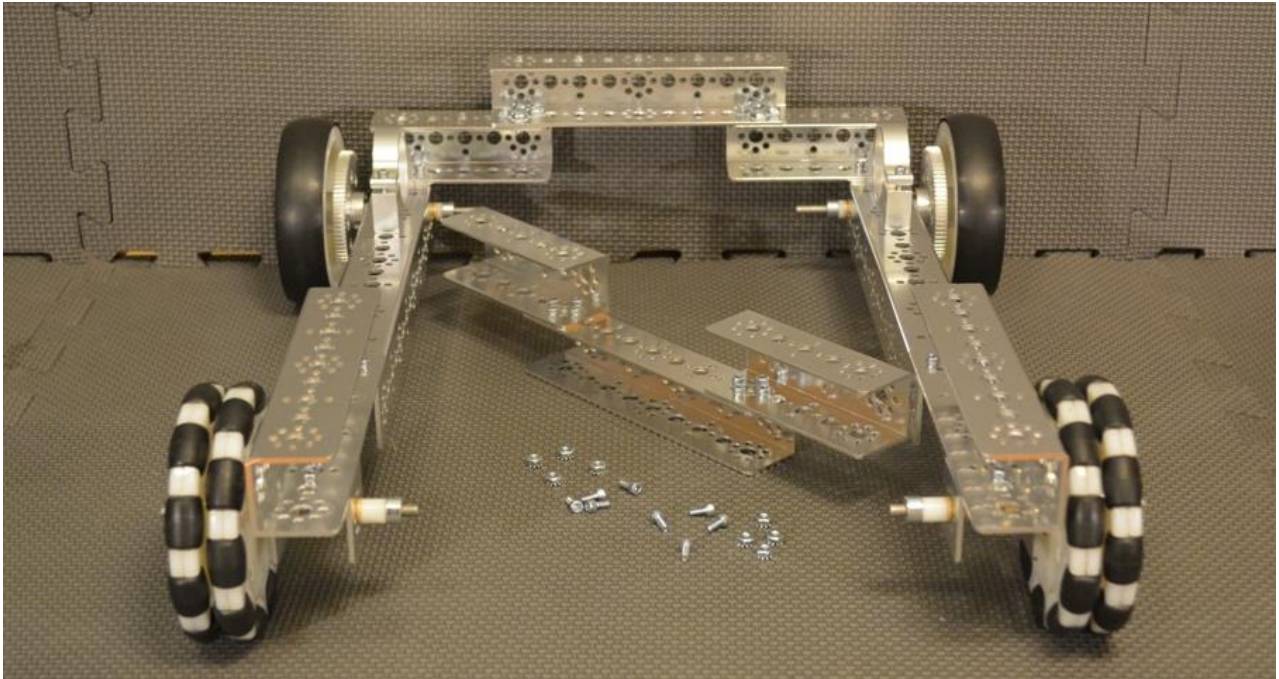
Step 1: left chassis rail assembly (1), cross bar assembly (1), 5/16" socket head cap screws (4), and keps nuts (4). The second picture shows the completed step.



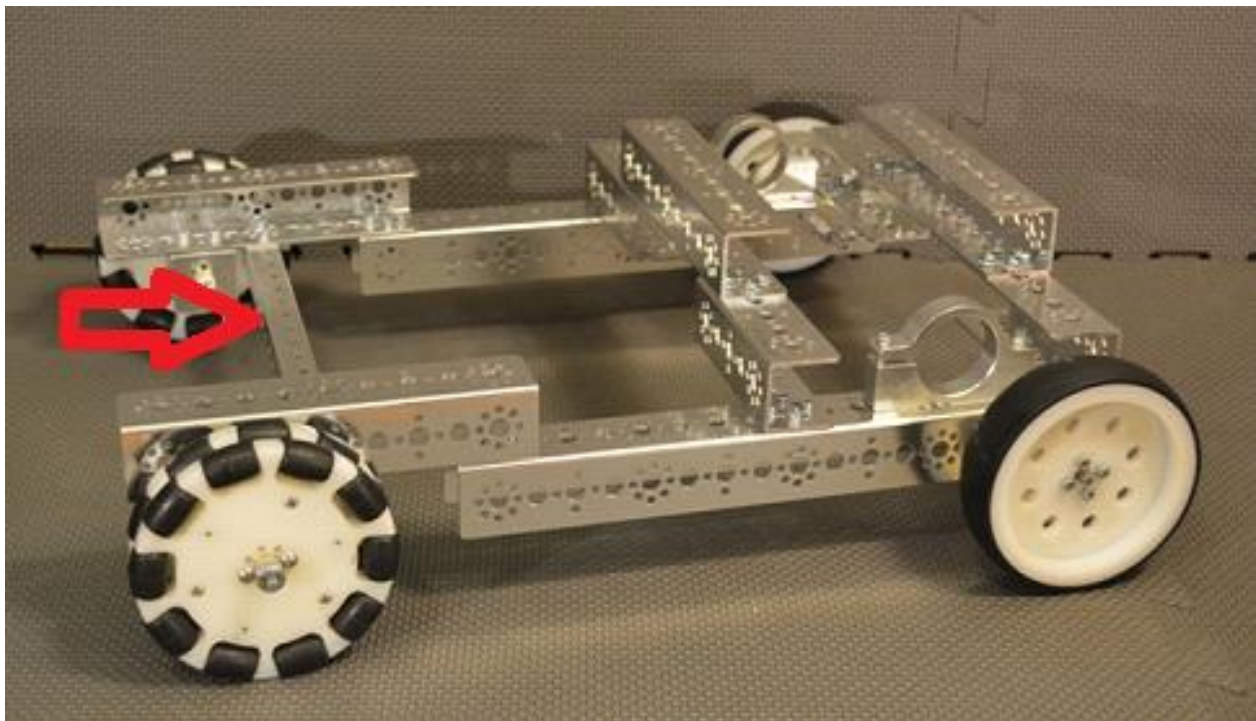
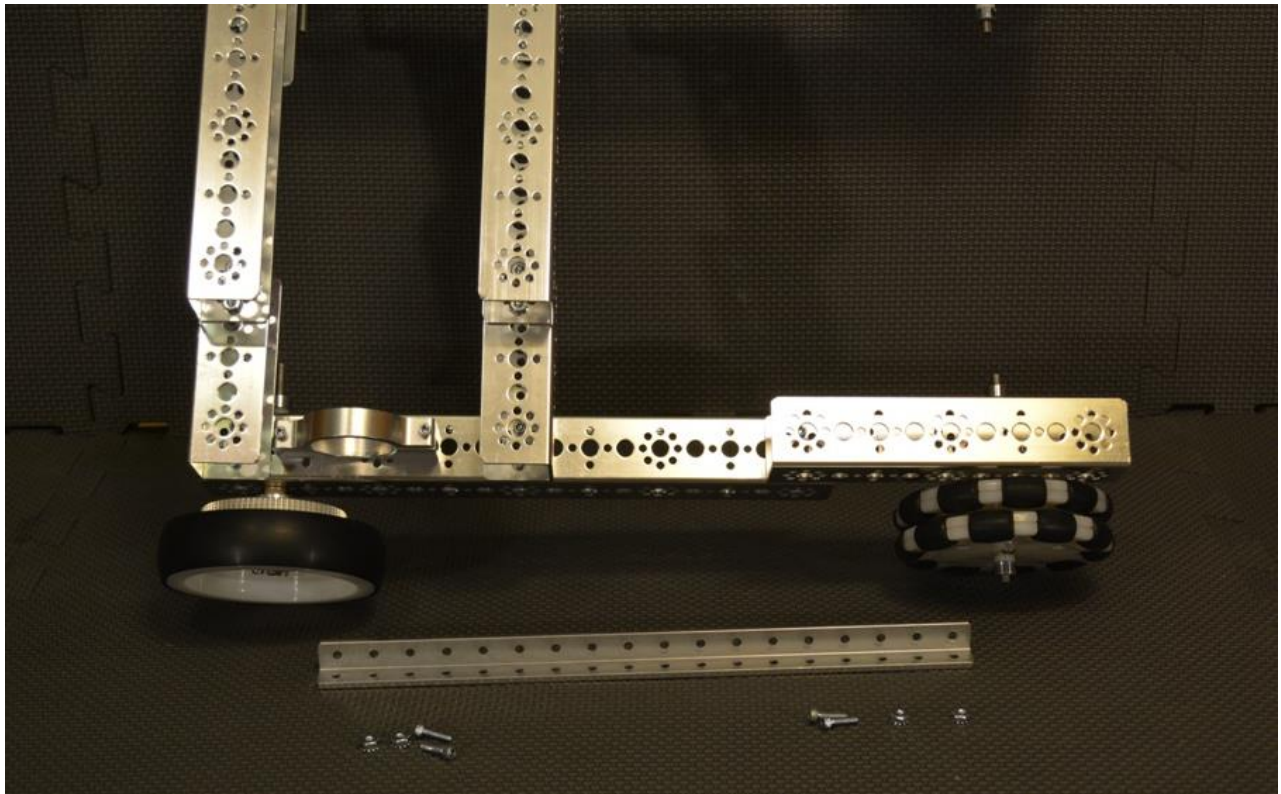
Step 2: assembly from step 1, right chassis rail assembly (1), 5/16" socket head cap screws (4), and keps nuts (4). The second picture shows the completed step.



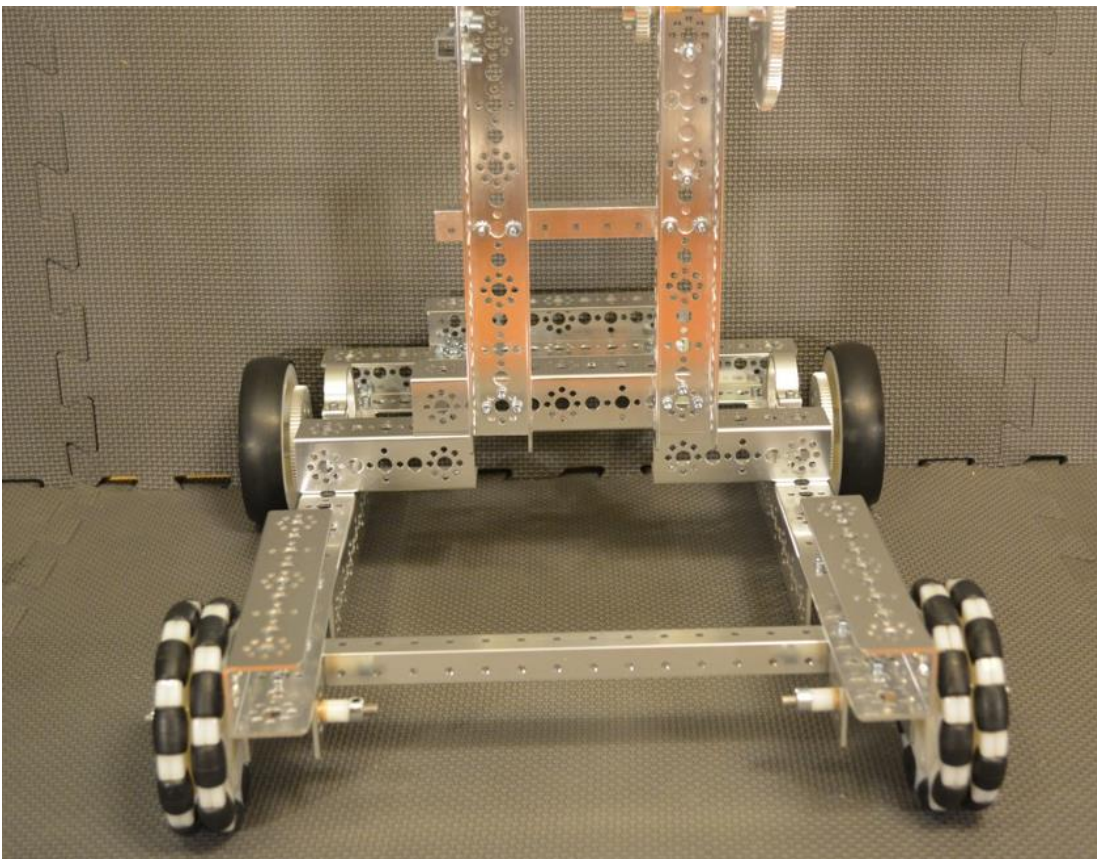
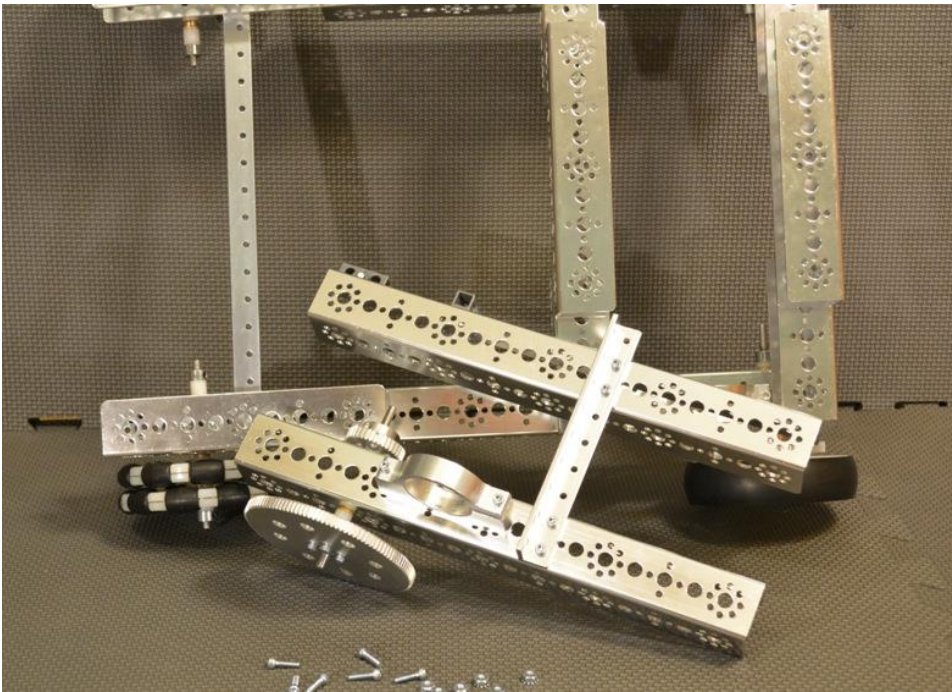
Step 3: assembly from step 2, cross bar assembly (1), 5/16" socket head cap screws (8), and keps nuts (8). The second picture shows the completed step.



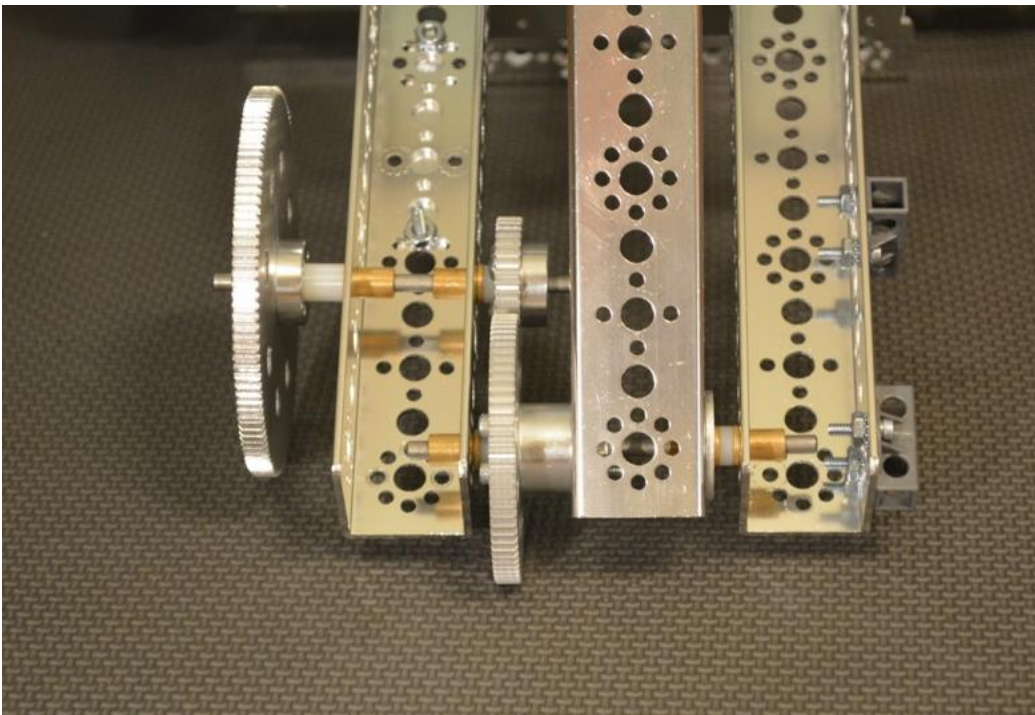
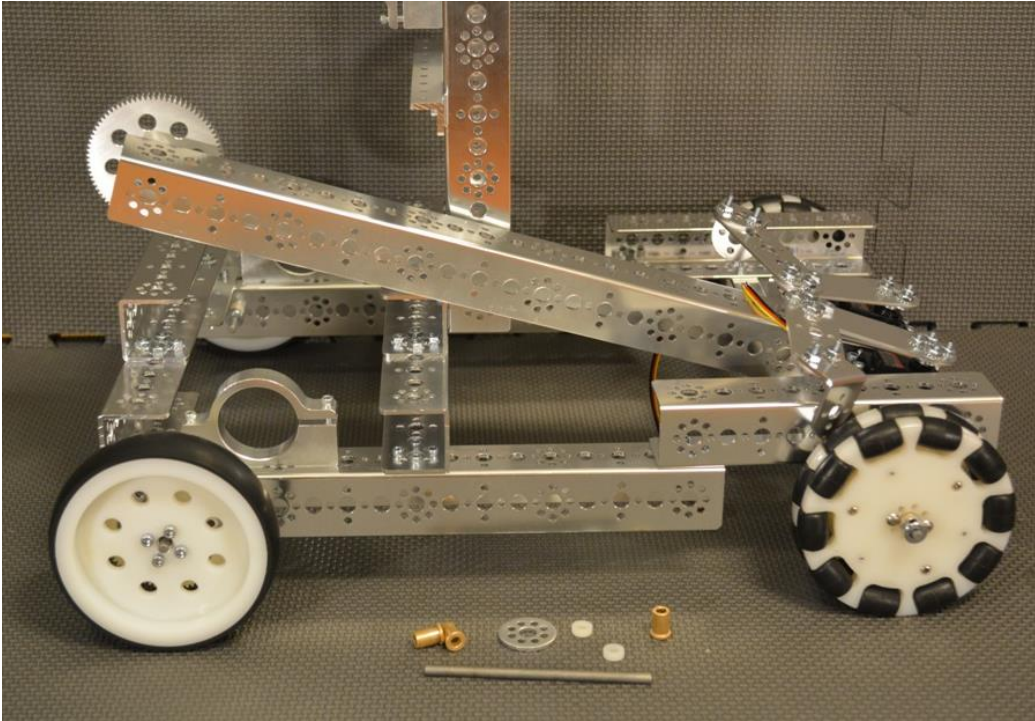
Step 4: assembly from step 3, 288mm angle (1), 1/2" socket head cap screws (4), and keps nuts (4). The second picture shows the completed step.



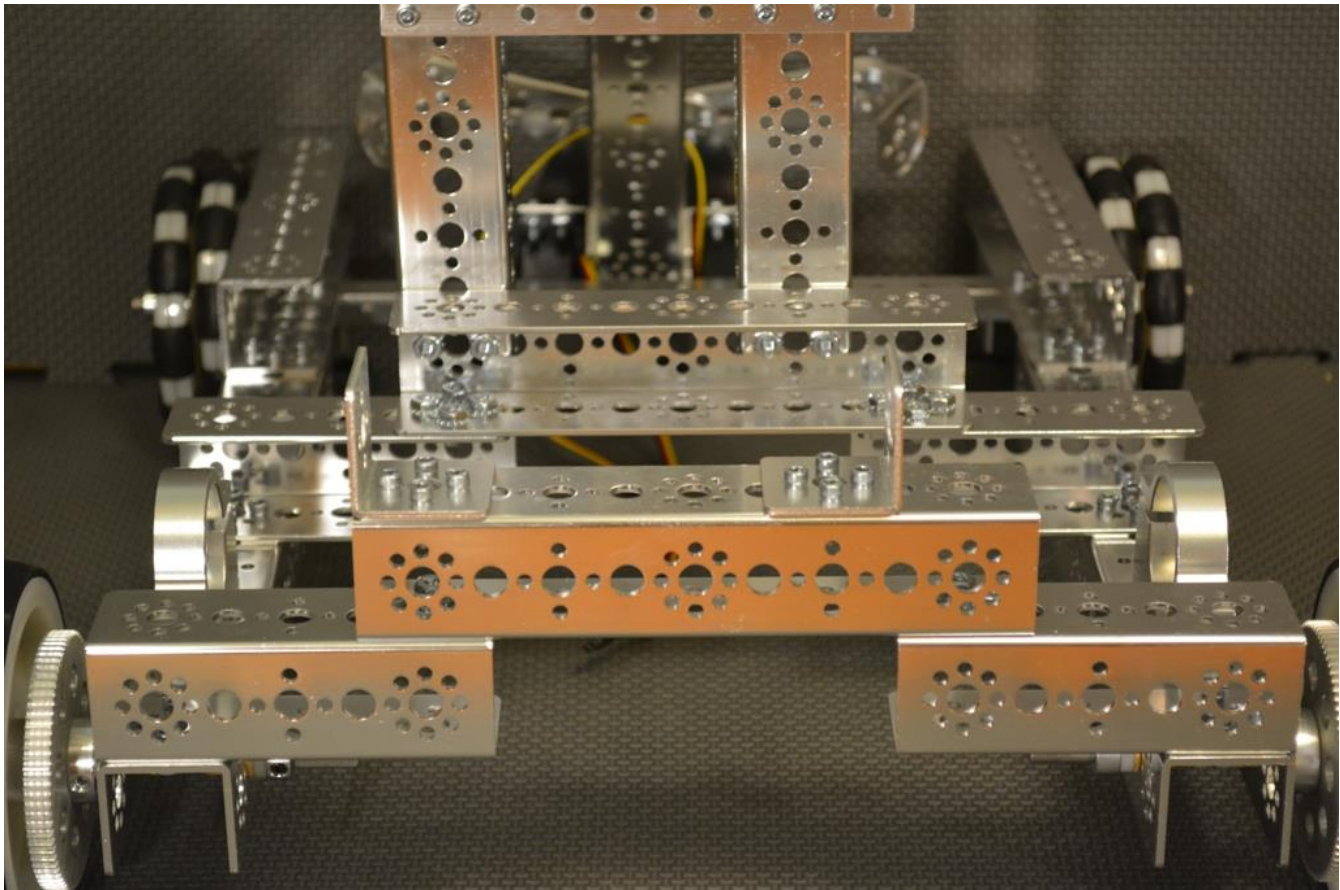
Step 5: assembly from step 4, tower assembly, 1/2" socket head cap screws (6), and keps nuts (6). The second picture shows the completed step.



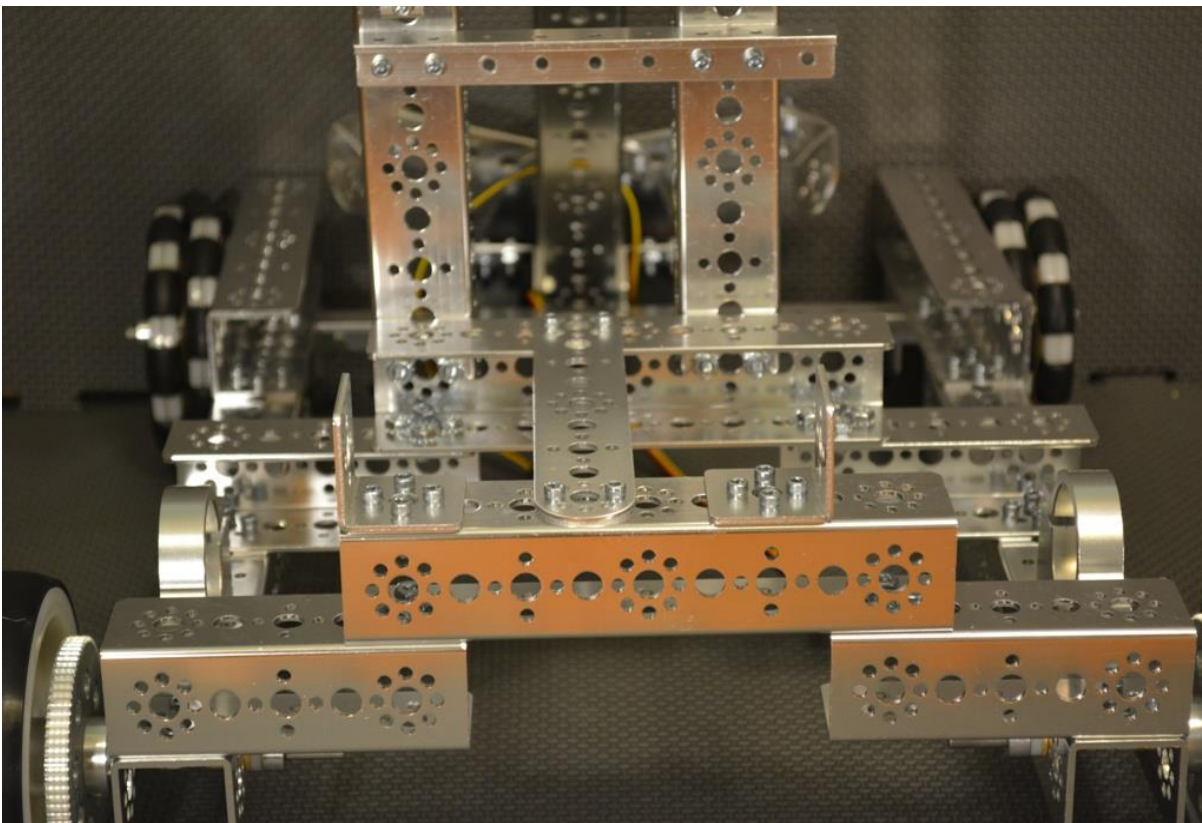
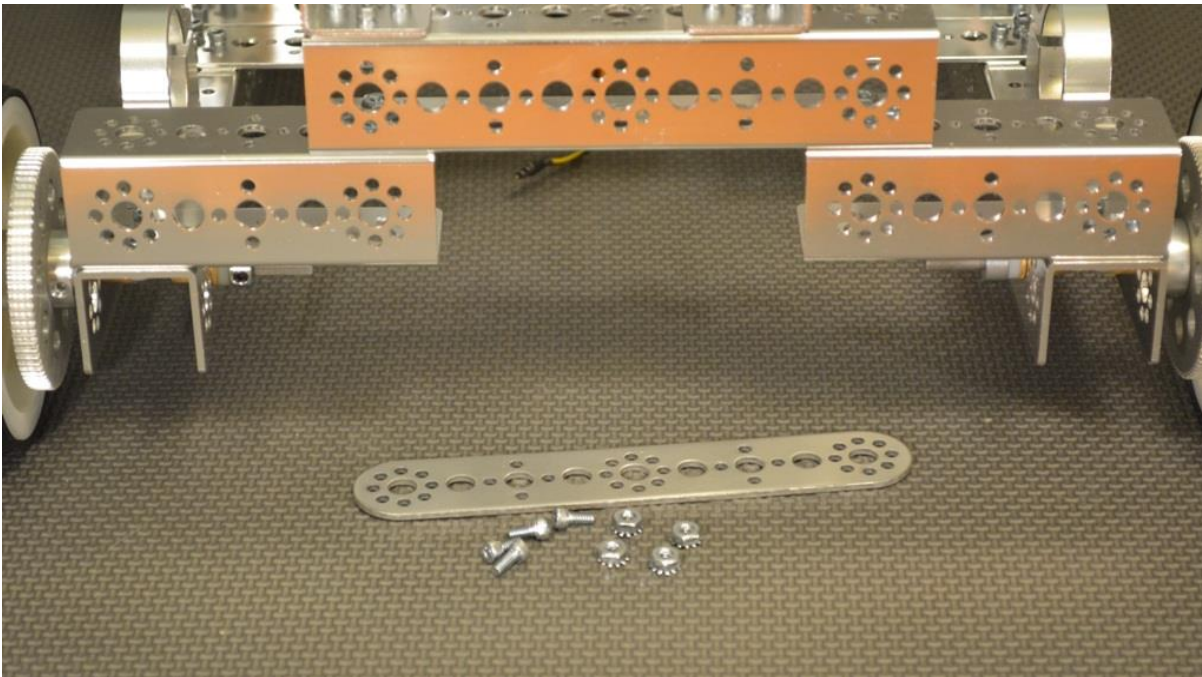
Step 6: assembly from step 5, arm assembly, bronze bushing (3), 1/8" axle spacer (2), axle (1), and flat spacer (1). Note: order of assembly is tower channel with hardpoint connector, bronze bushing, axle spacer, bronze bushing, flat spacer, arm, axle spacer, bronze bushing, tower channel with gears. The second picture shows the completed step.



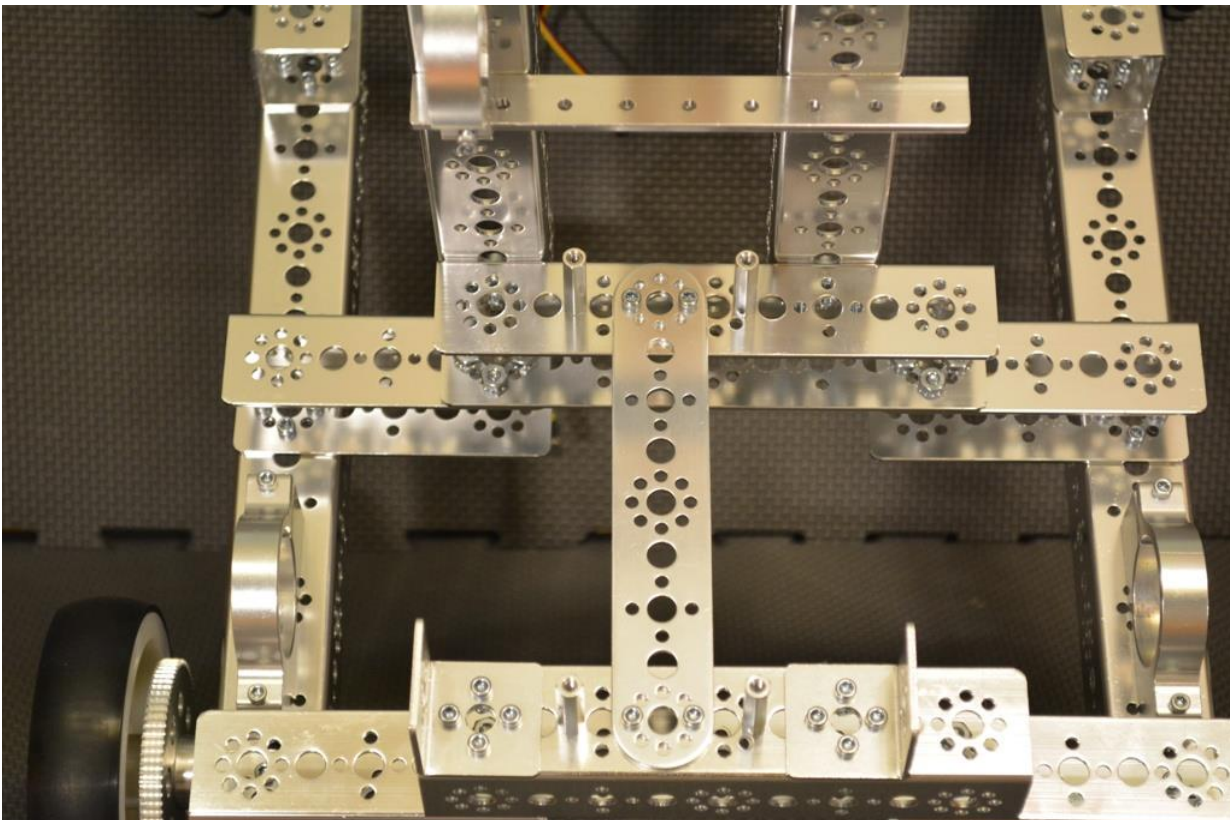
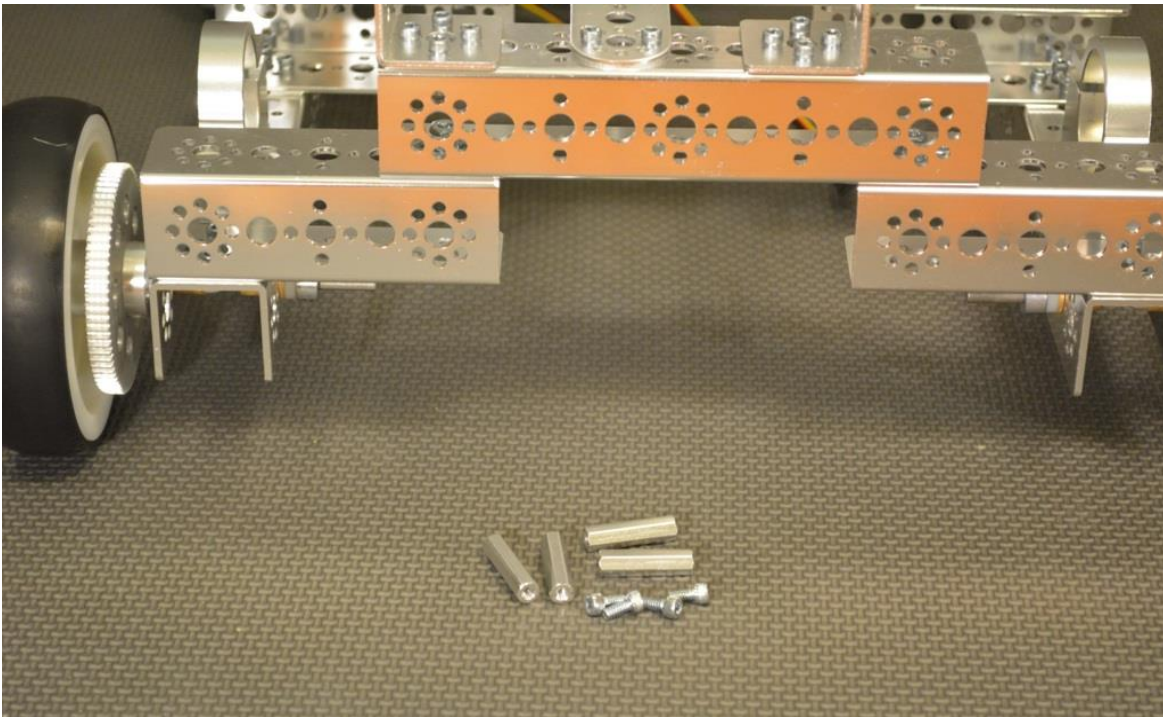
Step 7: assembly from step 6, inside corner bracket (2), 1/2" socket head cap screw (8), and keps nut (8). The second picture shows the completed step.



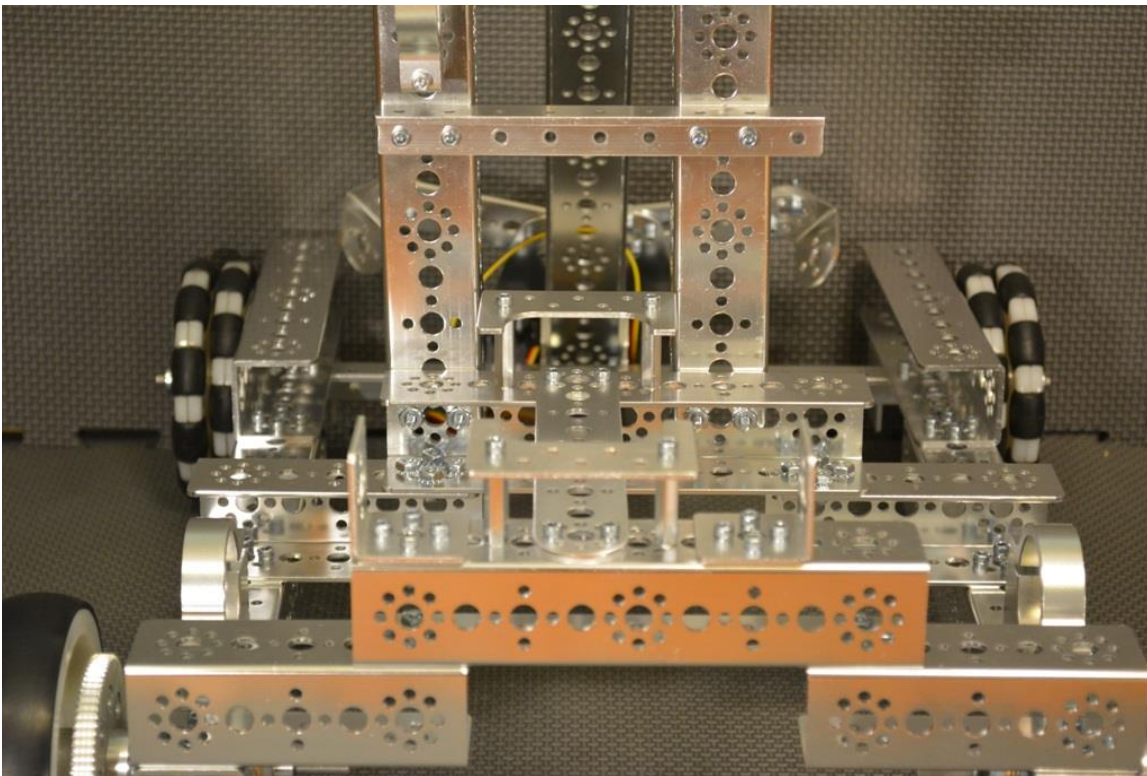
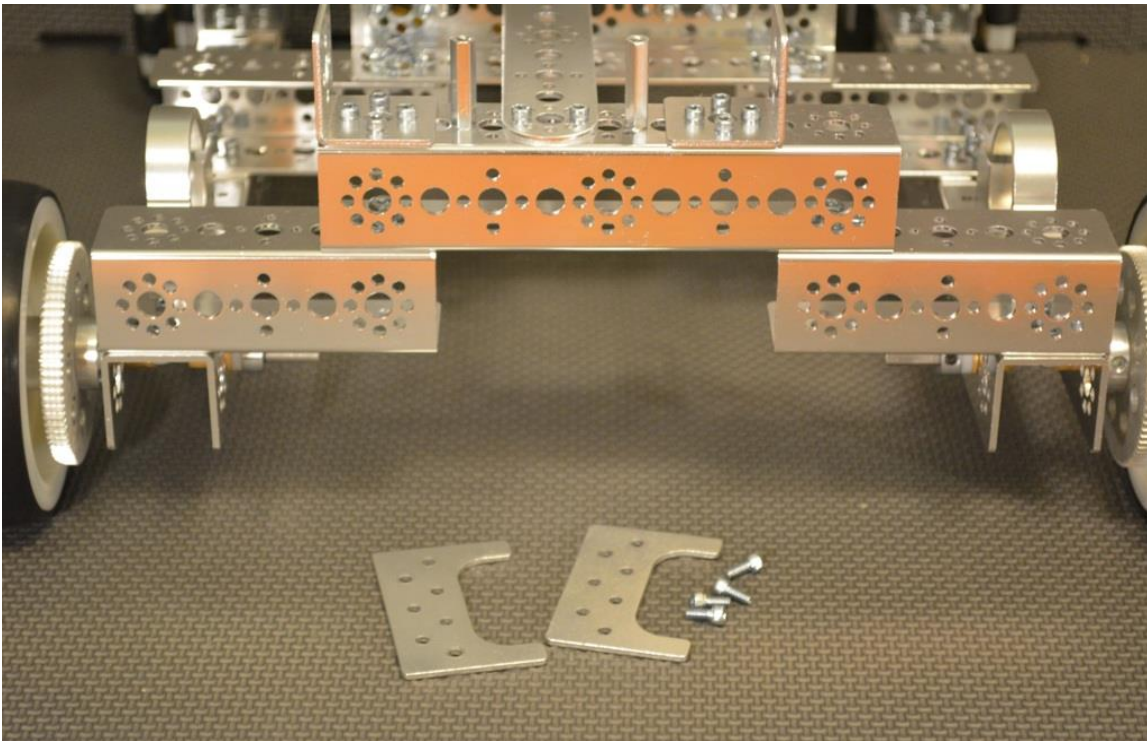
Step 8: assembly from step 7, 160mm flat (1), 5/16" socket head cap screw (4), and keps nut (4). The second picture shows the completed step.



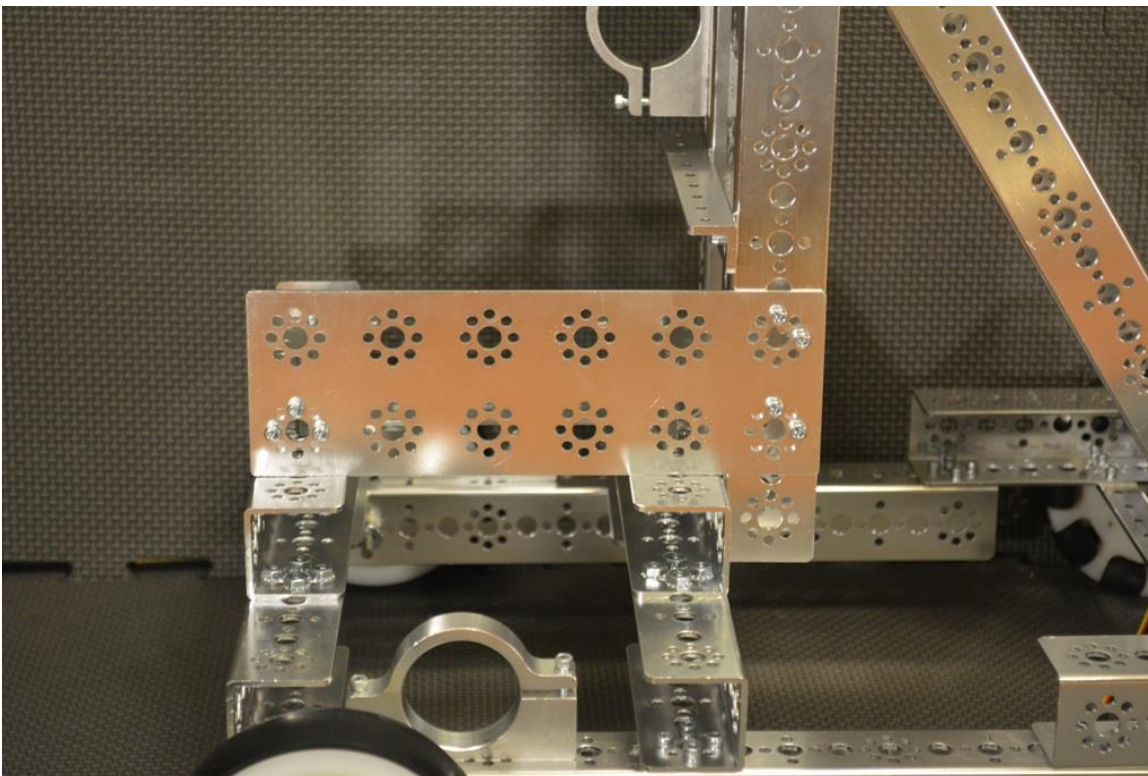
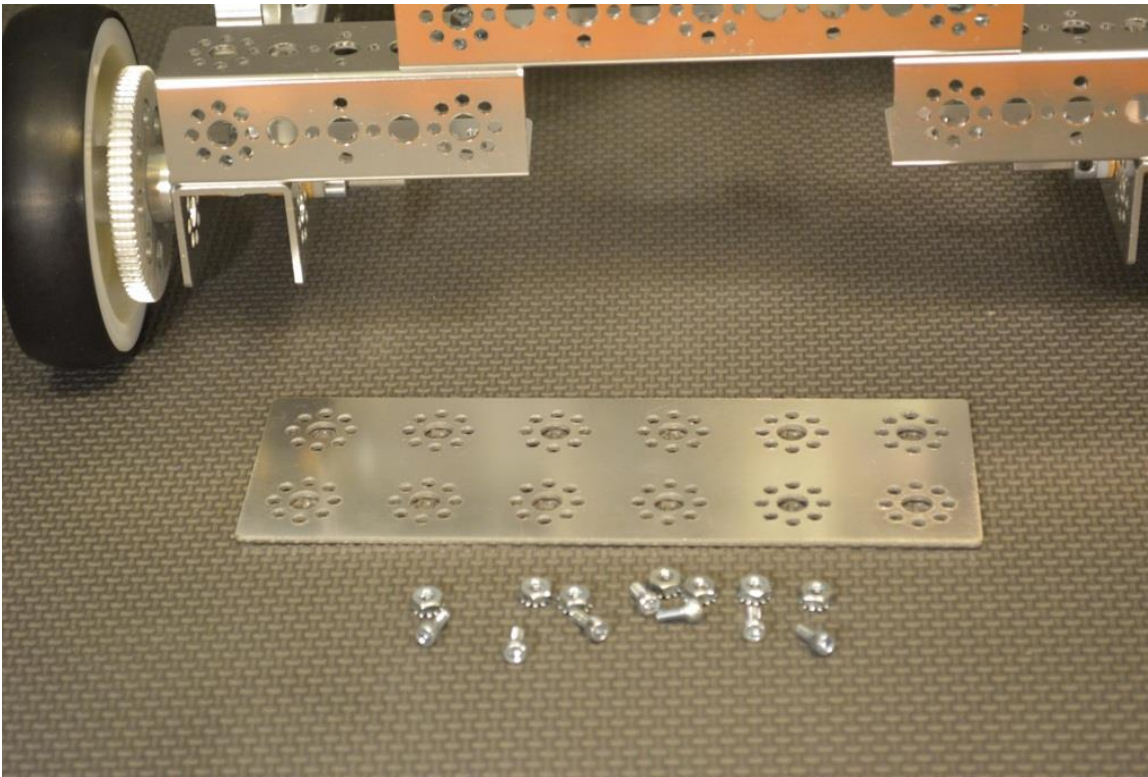
Step 9: assembly from step 8, 160mm flat (1), 1" stand-off posts (4), and 5/16" socket head cap screw (4). The second picture shows the completed step.



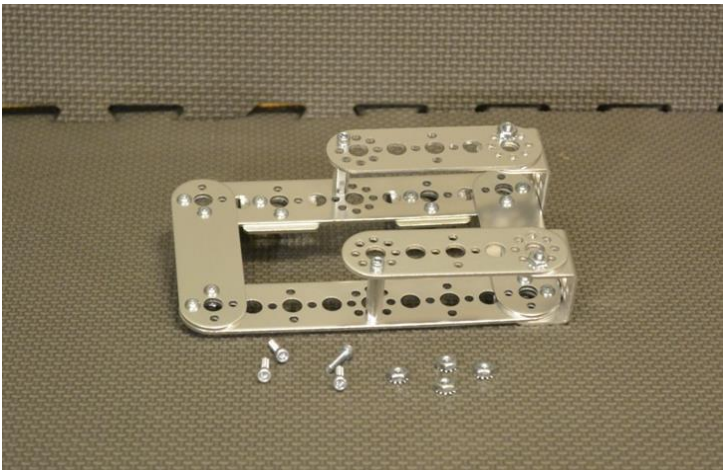
Step 9: assembly from step 8, battery clip (2), 5/16" socket head cap screw (4). The second picture shows the completed step.



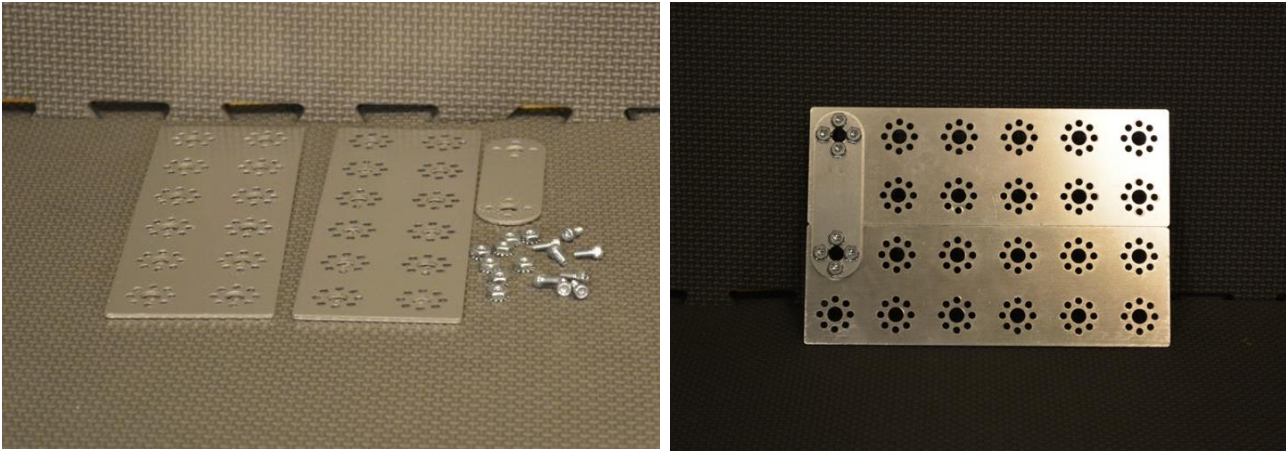
Step 10: assembly from step 9, flat building plate (1), 5/16" socket head cap screw (7), and keps nut (7). The second picture shows the completed step.



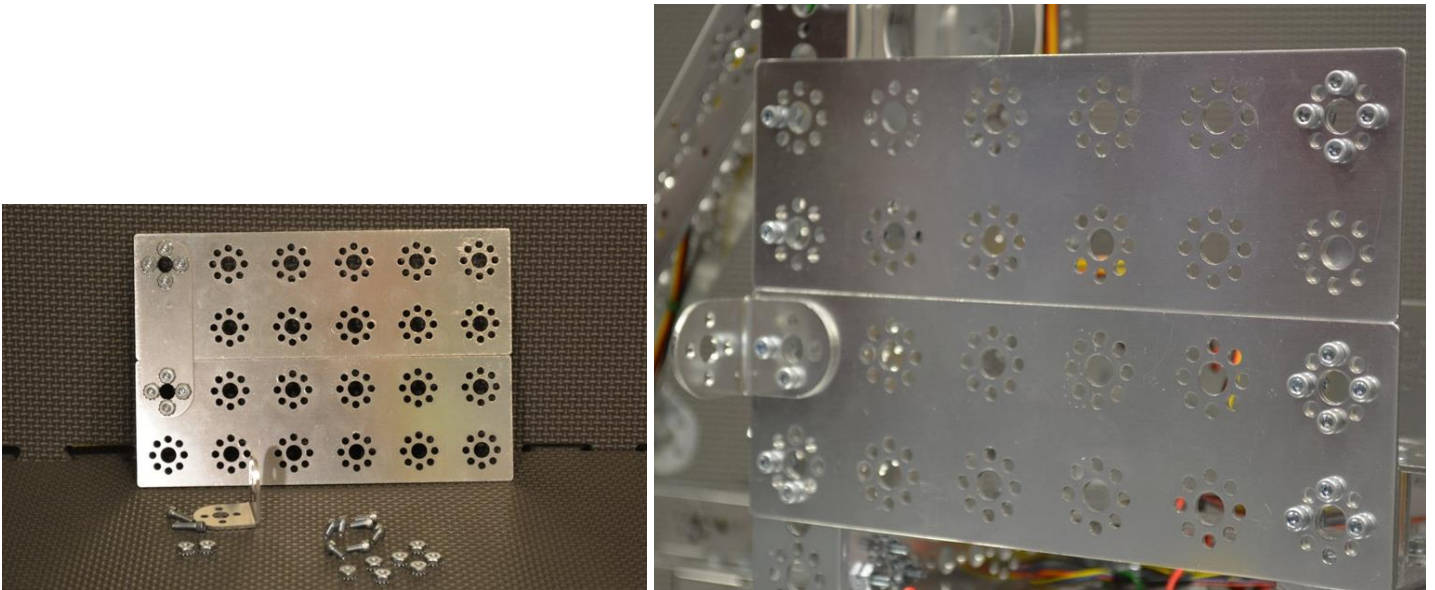
Step 11: assembly from step 10, phone mounting assembly (1), 1/2" socket head cap screw (4), and keps nut (4). The second picture shows the completed step.



Step 12: flat building plate (2), flat bracket (1), 5/16" socket head cap screw (8), and keps nut (8). The second picture shows the completed step.



Step 13: Robot from step 11, assembly from step 12, L bracket (1), 5/16" socket head cap screw (7), 1/2" socket head cap screw (2), and keps nut (9). The second picture shows the completed step.



Install Encoders on Motors

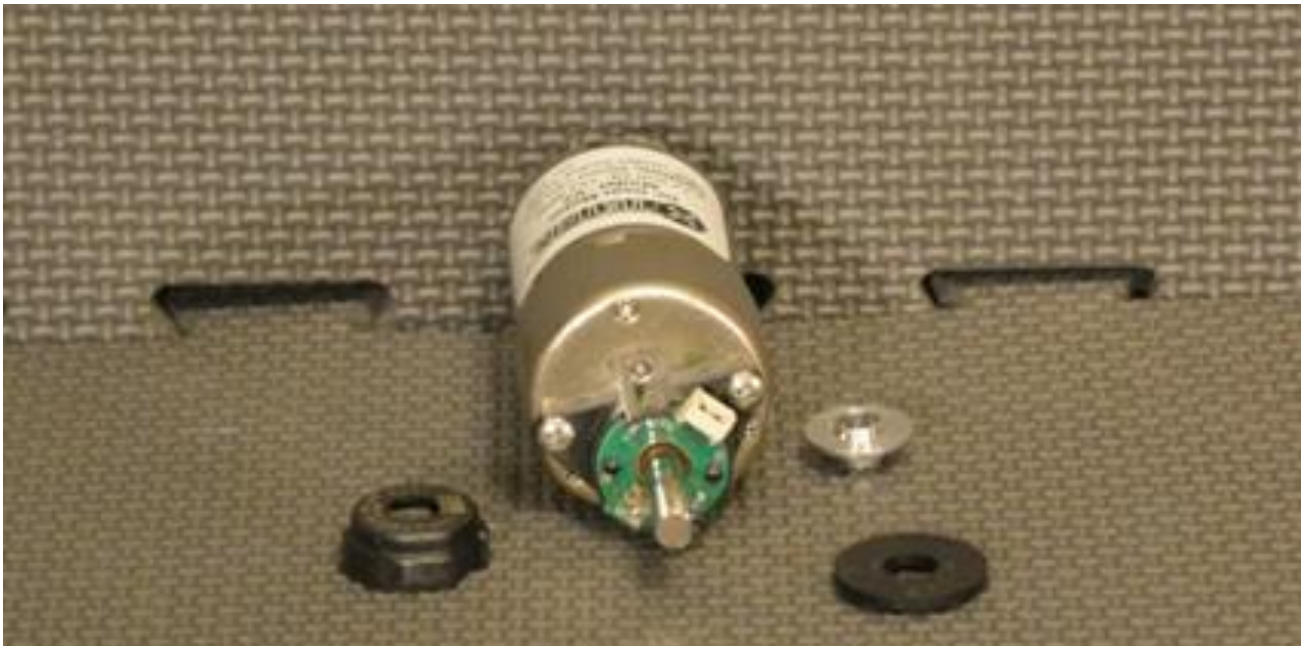
Make 2 copies

These are assembled as per the instructions that come in the encoder pack.

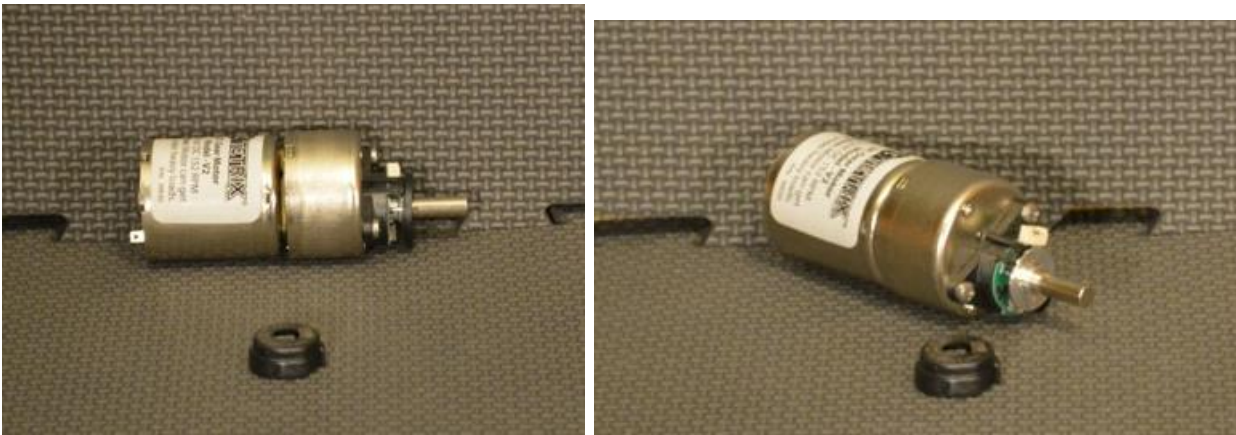
Step 1: encoder kit (1) and TETRIX DC motor (1). The second picture shows the separate encoder parts.



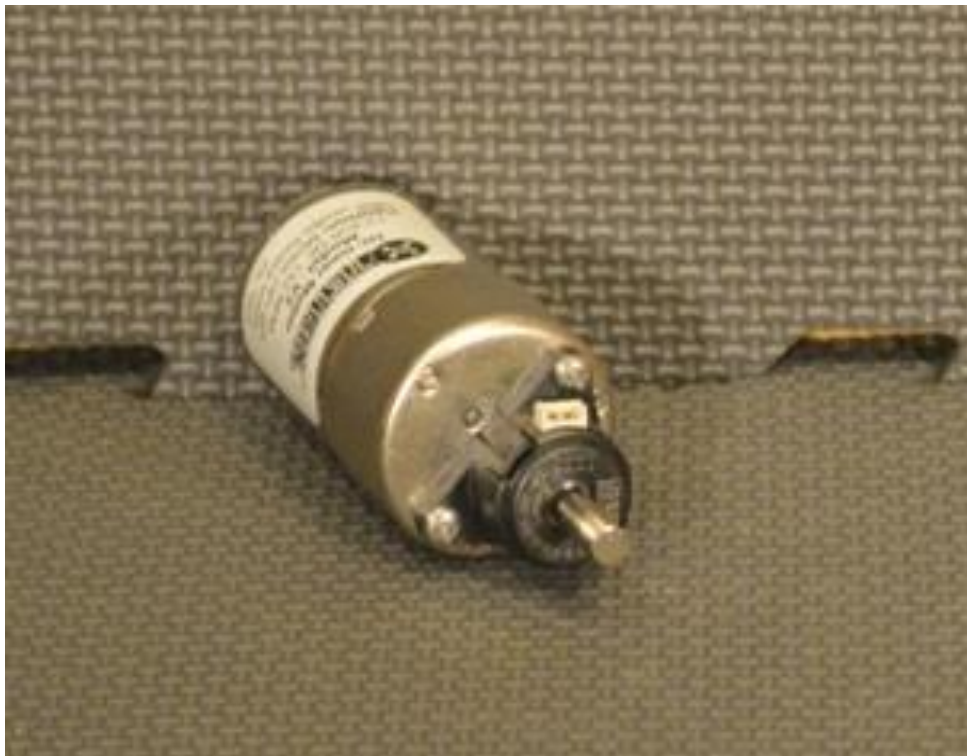
Step 2: install the encoder printed circuit board assembly to the motor with the provided screws.



Step 3: install the encoder disk on the motor shaft using the provided spacer tool. The first photo shows the spacer tool on the motor shaft setting the disk position. The second photo shows it without the spacer tool.



Step 4: snap the encoder cover onto the motor encoder.



Motor Hub and Gear Assembly

Make 3 copies

Step 1: 40-tooth gear (1), motor shaft hub (1), and button head cap screws (4). The second picture shows the completed assemblies.



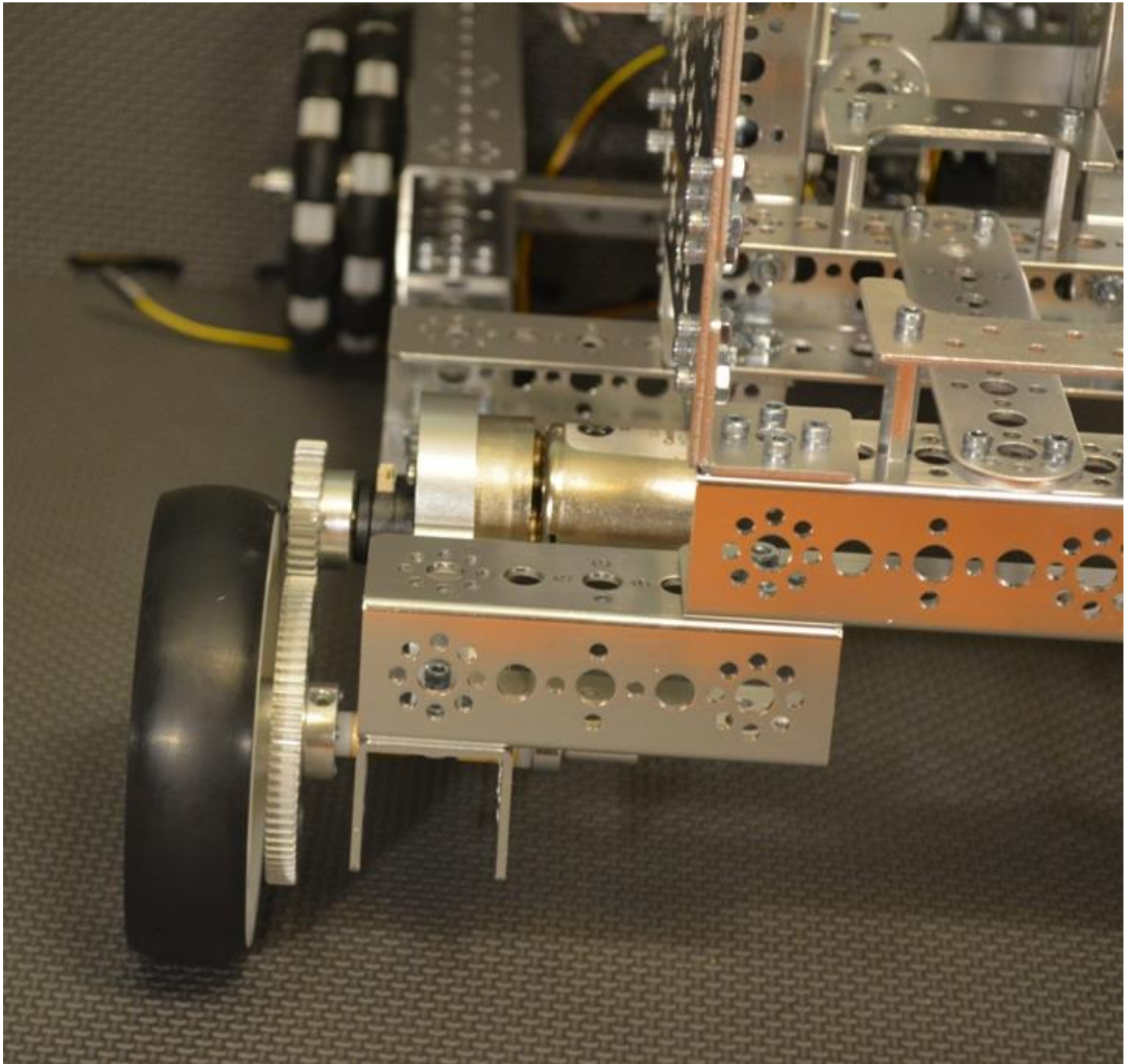
Drive Motor Assembly

Step 1: motor and encoder assembly (2) and motor hub and gear assembly (2). The second picture shows the completed assemblies. Note: install the hub as close as possible to the encoder without touching.

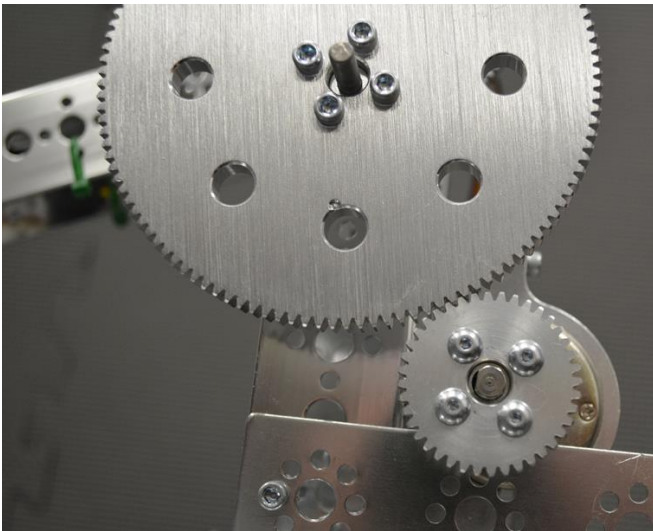
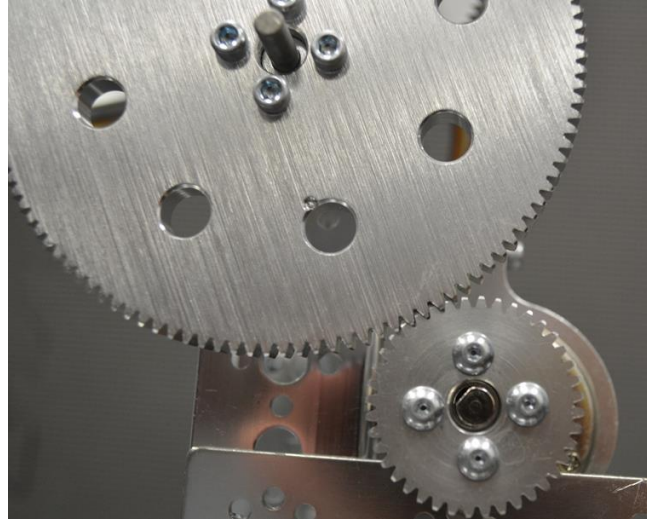
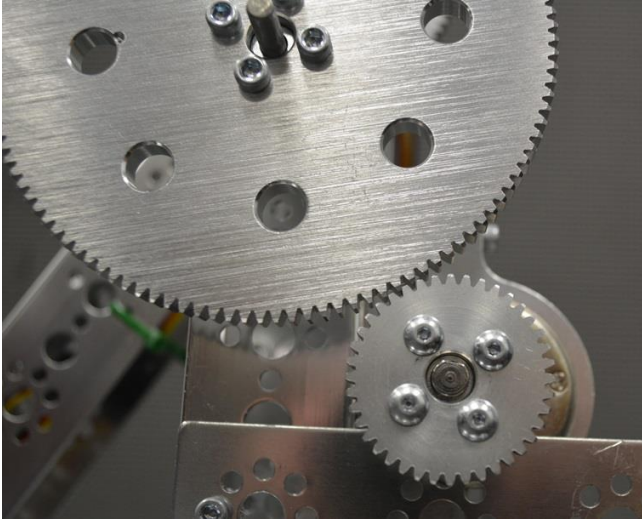


Install Drive Motors on Robot

Step 1: drive motor assembly (2) and Robot (1) The picture shows the completed assembly for the left chassis rail. The right chassis rail will be similar. Note: rotate the motor in the motor mount until the gears mesh well but not too tight. If they are too tight, they will bind, if they are too loose, they will slip.



The following pictures show examples of meshing the gears. The first is too loose, the second is too tight, the third is a good mesh. To test, rotate the mechanism by hand. If the gear teeth slip, then it is too loose, if the mechanism binds, then it is too tight



We will finish the final assembly after the Robot is programmed and we have determined which motor controller will be operating the drive motors and which will be operating the arm motor.

Programming the Robot

Android Studio is one of the Integrated Development Environments approved for FTC 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 8 computer.

Android Studio is built around Java, so the Java Development Kit (JDK) must be installed before Android Studio is installed. Version 7 has been tested with the FTC SDK, so this document will show all of the steps to download and install it onto a Windows 8 computer. Note that Java version 8 is available, but hasn't been tested with the FTC SDK.

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 modes 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, then the PushBot package. 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 Screen. This screen will be shown when the Windows key is pressed (shown below). Once on the start screen, typing the name of the application will cause a list to be shown of known applications. 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.

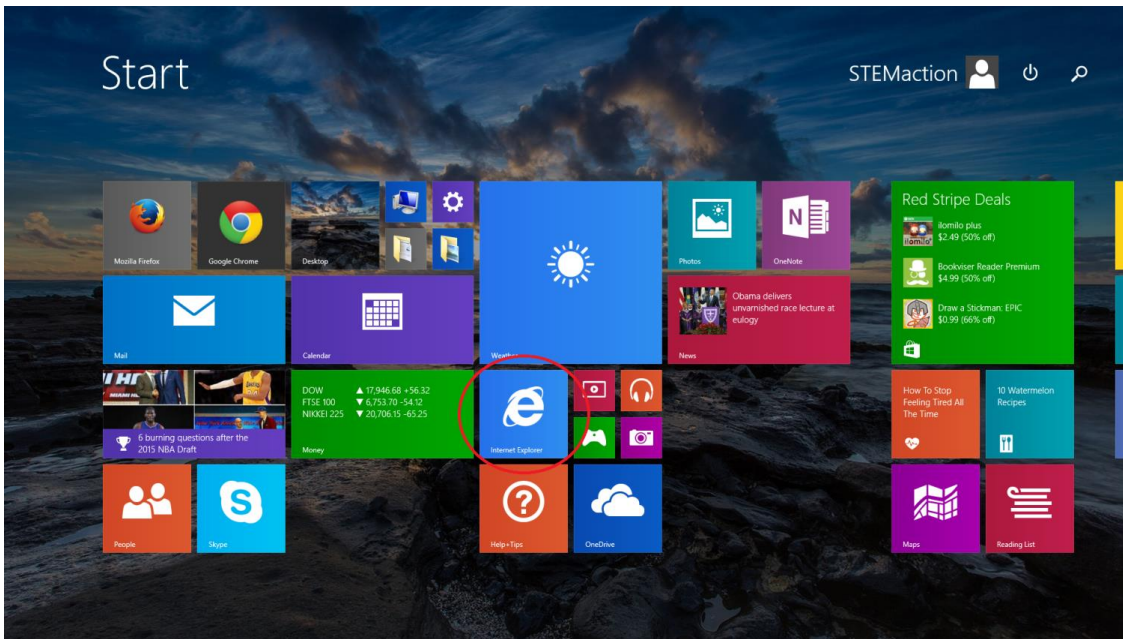


Note: Before beginning, charge the phones.

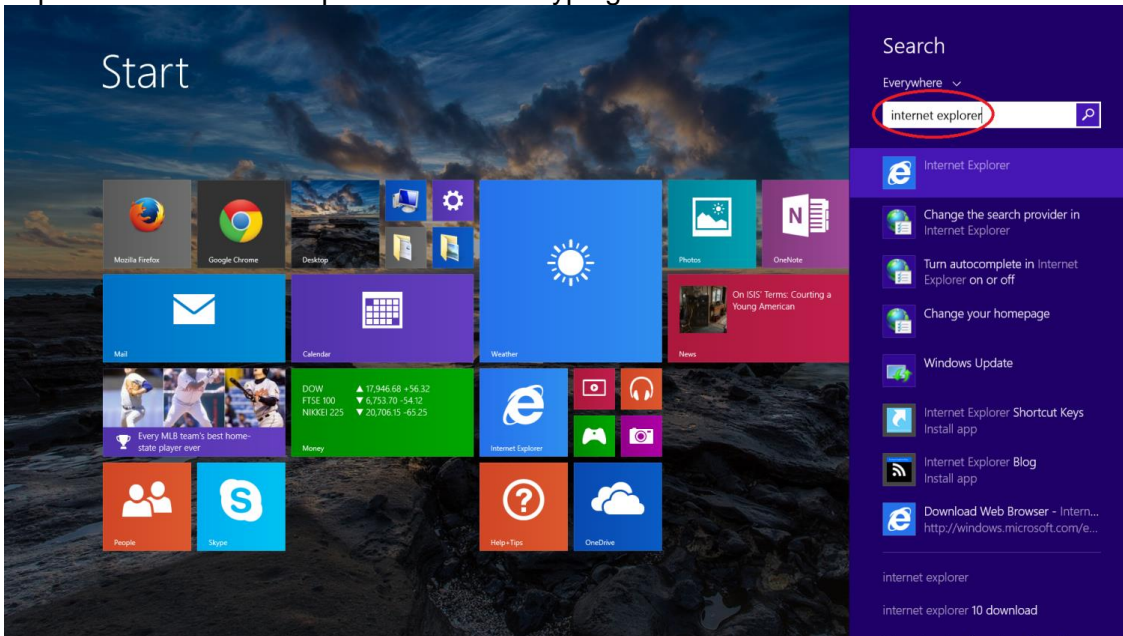
Opening Internet Explorer

Some of the steps in this section use Internet Explorer (IE). 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 IE 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 Internet 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 'Internet Explorer'. Remember to press enter after typing the name.



The Internet Explorer application is shown below.

New tab



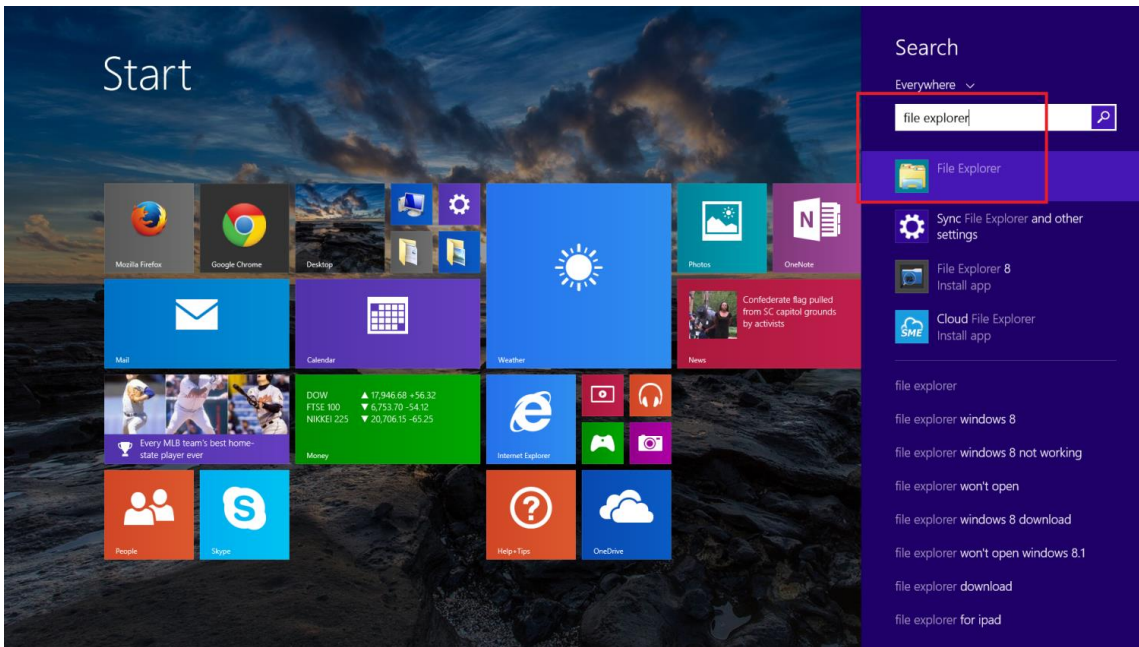
Opening File Explorer (aka Windows Explorer)

Some of the steps in this section use File Explorer (FE). The instructions for opening it 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 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.

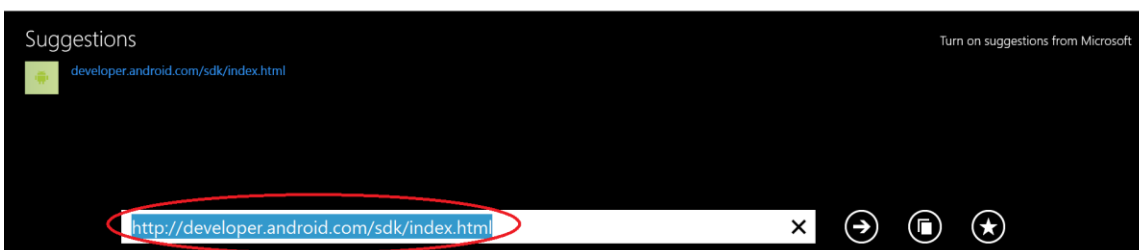


Determining whether Android Studio can be installed

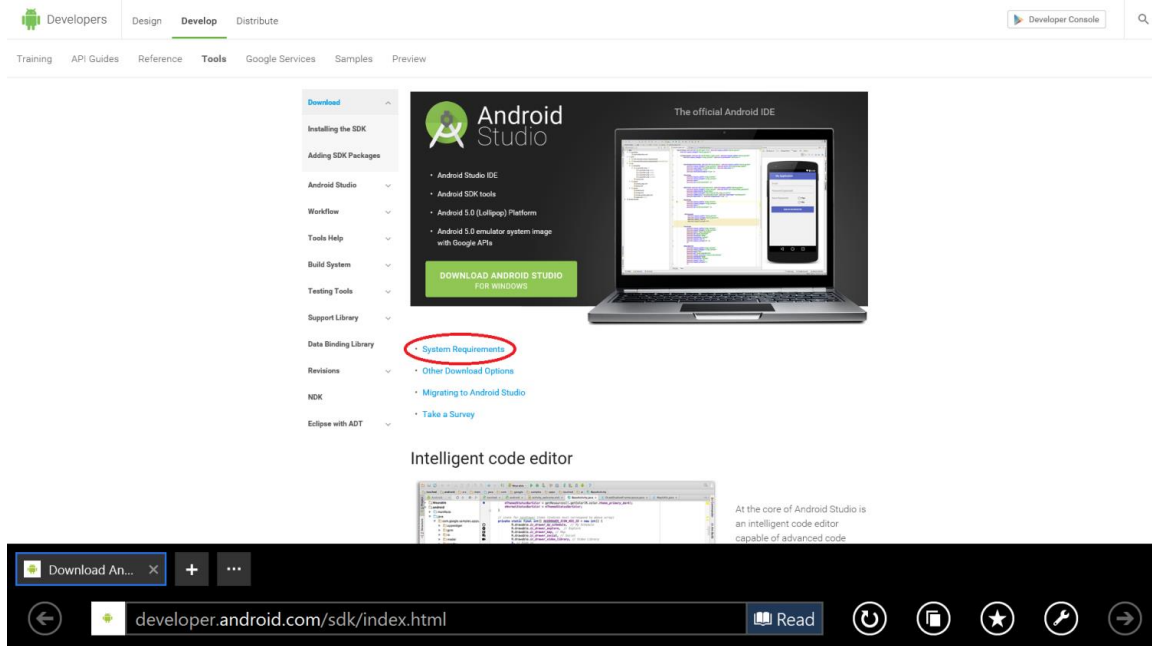
Android Studio publishes the qualities a computer must have to be able to run the IDE. Before installing the Android Studio IDE, Java, or the FTC SDK, check whether the machine can run the IDE. If the machine can't support the Android Studio IDE's minimum requirements, then there is no need to install 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 Internet Explorer and enter 'http://developer.android.com/sdk/index.html' into the Address field.

New tab



Step 2: Select 'System Requirements'.



Developers Design Develop Distribute

Training API Guides Reference Tools Google Services Samples Preview

Download

- Installing the SDK
- Adding SDK Packages
- Android Studio
- Workflow
- Tools Help
- Build System
- Testing Tools
- Support Library
- Data Binding Library
- Revisions
- NDK
- Eclipse with ADT

System Requirements

- Other Download Options
- Migrating to Android Studio
- Take a Survey

Intelligent code editor

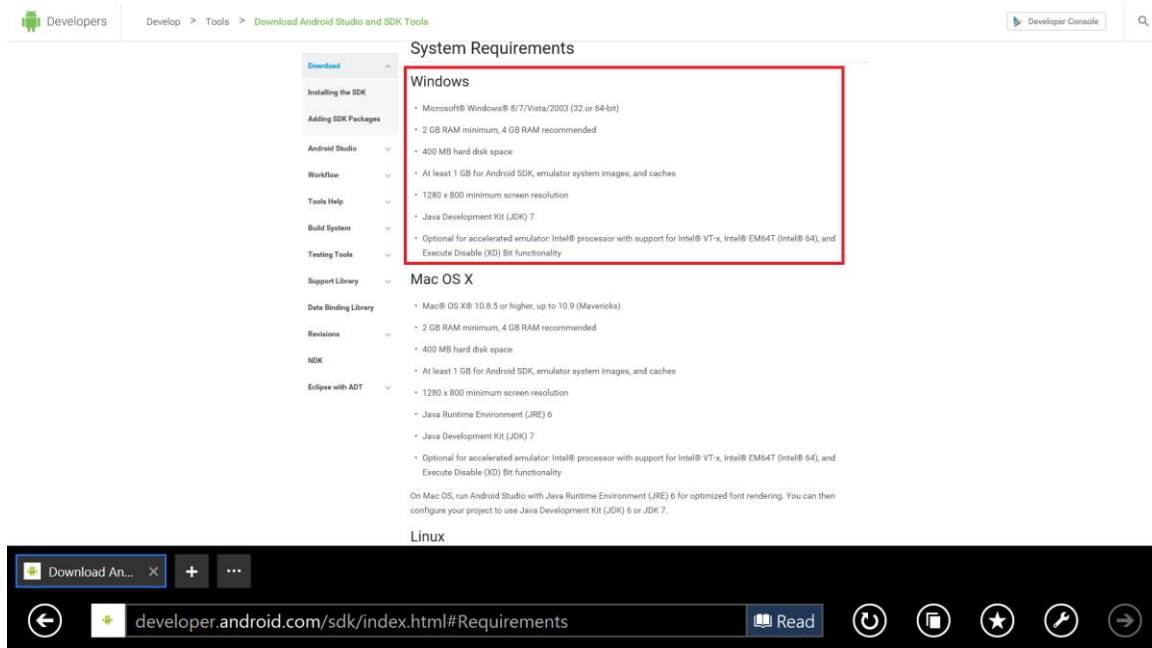
At the core of Android Studio is an intelligent code editor capable of advanced code

Download An... x + ...

developer.android.com/sdk/index.html

Read

Step 3: Examine the system requirements that match the installation operating system's type. The system requirements are farther down the page shown in the previous step; clicking the button will scroll the page downward to the System Requirements portion. The requirements for the Windows operating system are shown in the red box.



Developers Design Develop Distribute

Training API Guides Reference Tools Google Services Samples Preview

Download

- Installing the SDK
- Adding SDK Packages
- Android Studio
- Workflow
- Tools Help
- Build System
- Testing Tools
- Support Library
- Data Binding Library
- Revisions
- NDK
- Eclipse with ADT

System Requirements

Windows

- Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit)
- 2 GB RAM minimum, 4 GB RAM recommended
- 400 MB hard disk space
- At least 1 GB for Android SDK, emulator system images, and caches
- 1280 x 800 minimum screen resolution
- Java Development Kit (JDK) 7
- Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

Mac OS X

- Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks)
- 2 GB RAM minimum, 4 GB RAM recommended
- 400 MB hard disk space
- At least 1 GB for Android SDK, emulator system images, and caches
- 1280 x 800 minimum screen resolution
- Java Runtime Environment (JRE) 6
- Java Development Kit (JDK) 7
- Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

Linux

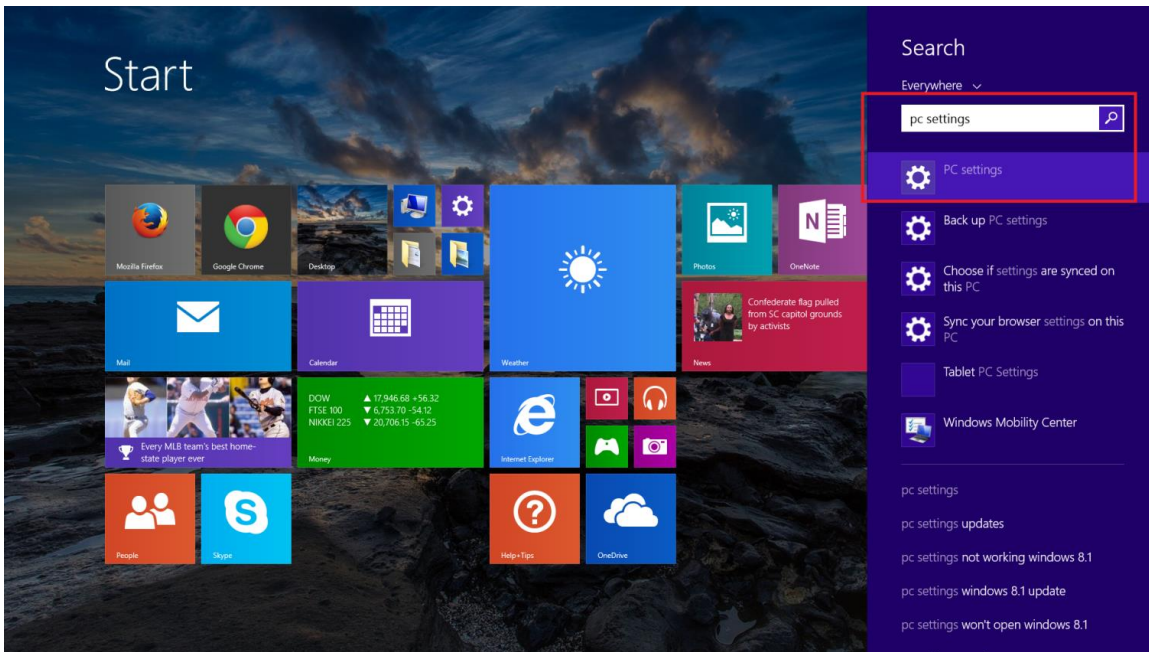
On Mac OS, run Android Studio with Java Runtime Environment (JRE) 6 for optimized font rendering. You can then configure your project to use Java Development Kit (JDK) 6 or JDK 7.

Download An... x + ...

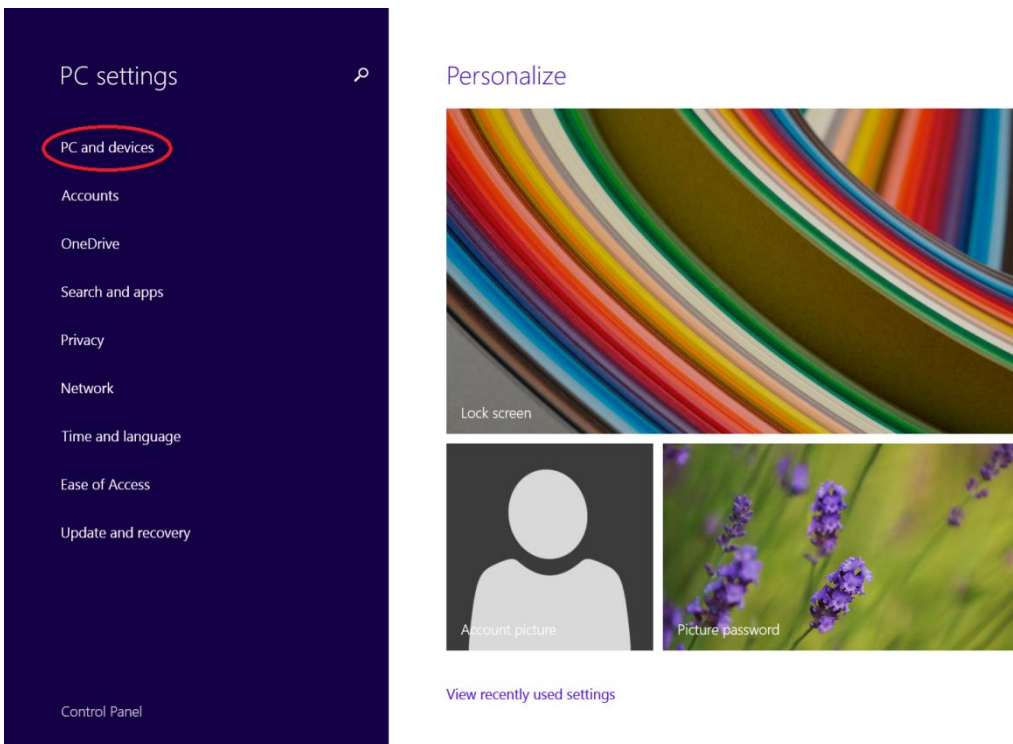
developer.android.com/sdk/index.html#Requirements

Read

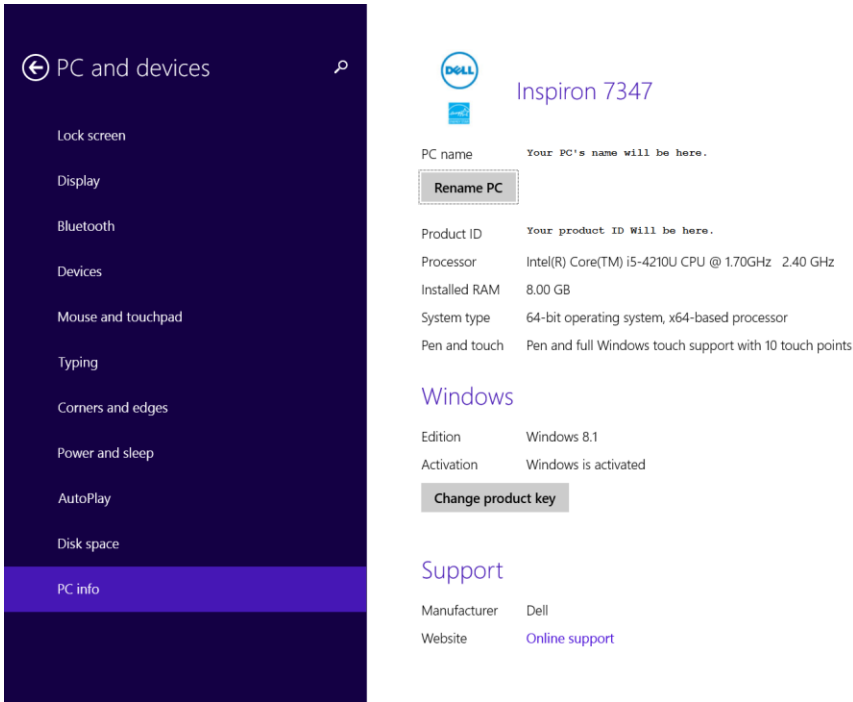
Step 4: Press the Windows key and type 'PC settings'. Remember to press enter after typing the name.



Step 5: Select PC and devices.



Step 6: Match the information from the following window with the requirements shown in step 3 – 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 8/7/Vista/2003 (32 or 64-bit). Since Windows 8.1 is higher than Windows 8, the requirement has been met.

Windows

Edition

Windows 8.1

Activation

Windows is activated

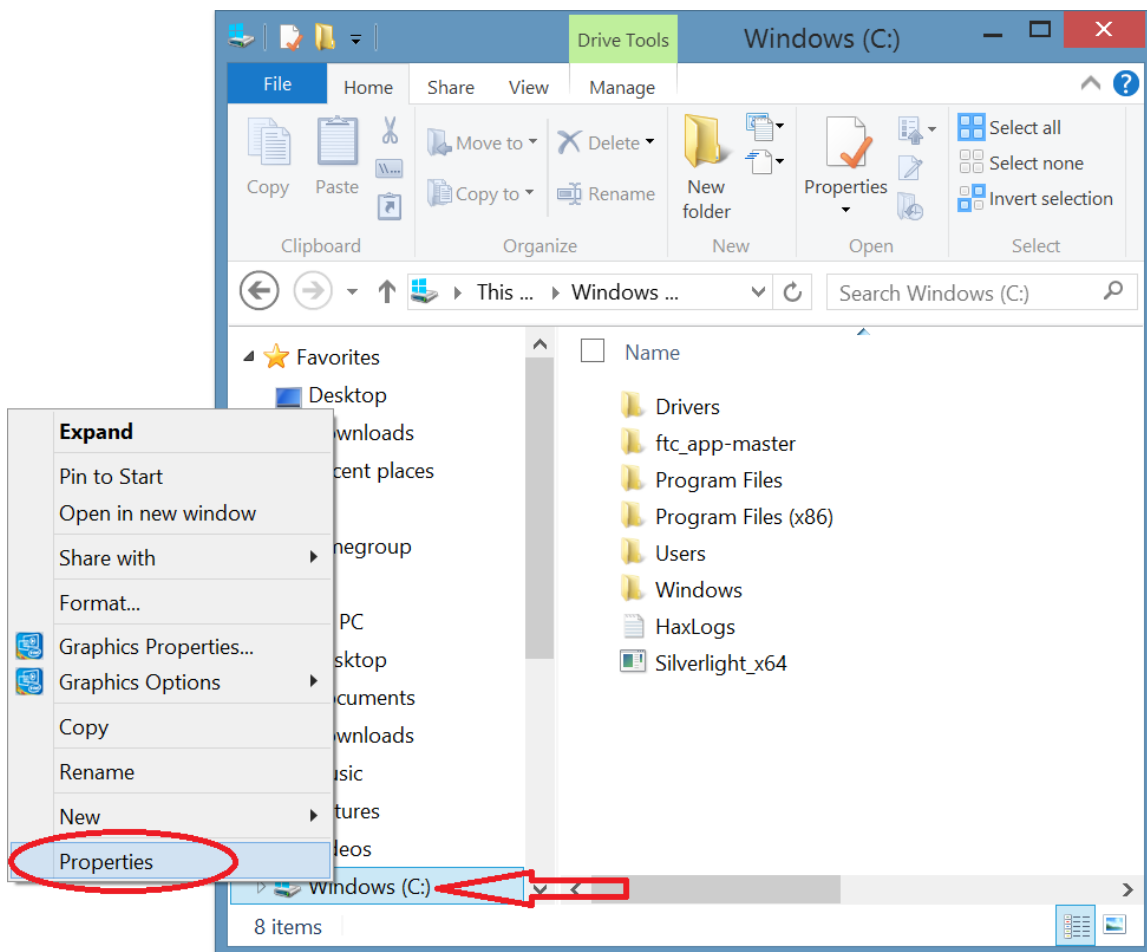
Change product key

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

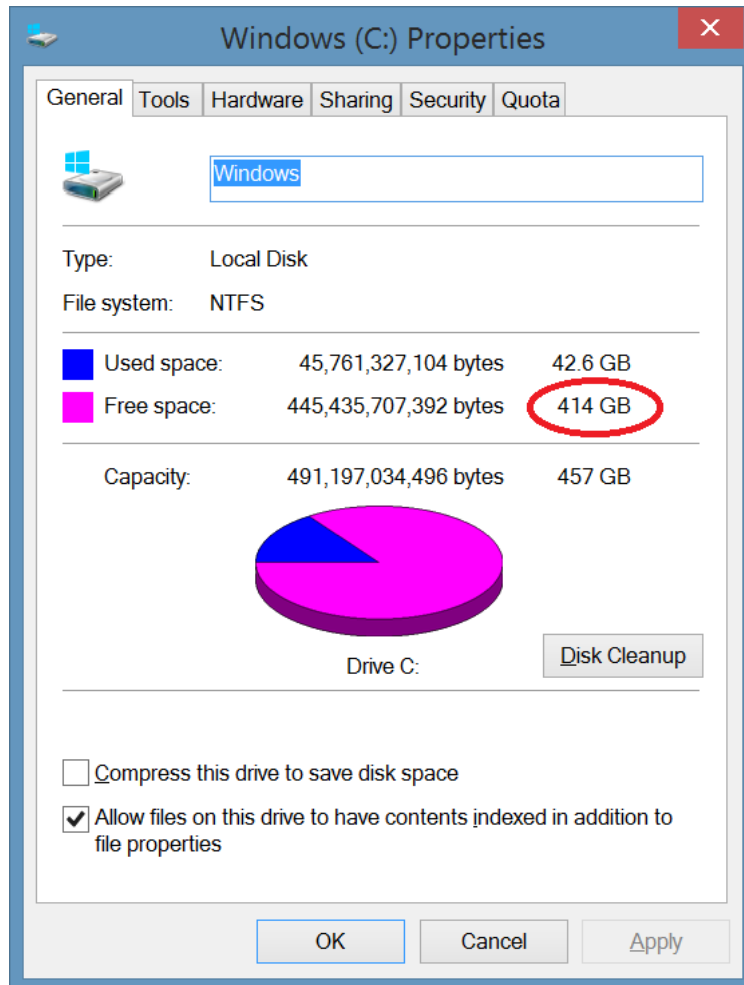
Processor	Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40 GHz
Installed RAM	8.00 GB
System type	64-bit operating system, x64-based processor
Pen and touch	Pen and full Windows touch support with 10 touch points

Windows

Step 7: To compare the hard disk space another Windows tool will be needed. Open File Explorer (known as Windows Explorer on older systems). Right click “Windows (C:)” and select Properties. If the computer has a touch screen, hold a stylus or finger on “Windows (C:)” until the context menu is shown.

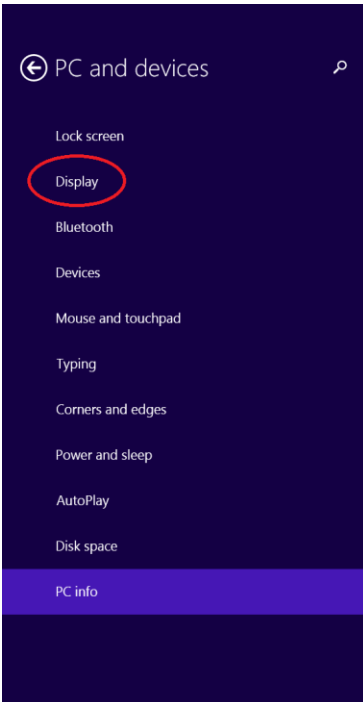


Requirement 3: 400MB hard disk space. Since 414 GB is greater than 400MB (1GB=1000MB), the requirement has been met.



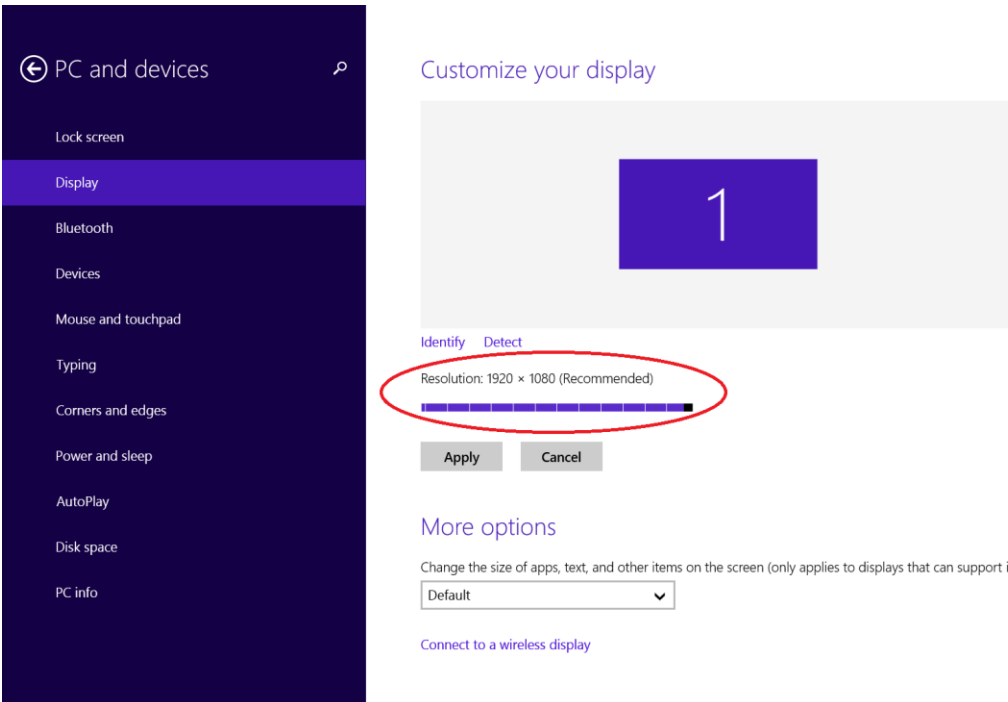
Requirement 4: At least 1 GB for Android SDK, emulator system images, and caches. Android Studio uses the Android SDK and thus requires hard disk space for the SDK. Since 414 GB (refer to the image above) is greater than 1GB, the requirement has been met. The total hard drive space required for the IDE installation mentioned in this document is 1GB + 400MB, which is 1.4GB. Since 414 GB is larger, both of the above requirements have been met.

Step 8: To compare the screen resolution, return to the PC Settings page (see steps 4 and 5) and select Display.



The screenshot shows the Windows Settings application. On the left, the 'PC and devices' sidebar is visible, with 'Display' highlighted in a red circle. The main content area shows the 'Inspiron 7347' system information page, including PC name, Product ID, Processor, Installed RAM, System type, and Pen and touch settings. Below this, the 'Windows' section shows the Edition (Windows 8.1) and Activation status (Windows is activated). The 'Support' section lists the Manufacturer (Dell) and Website (Online support).

Step 9: Note the scroll bar near the center of the window.



The screenshot shows the Windows Settings application with the 'Display' option selected in the left sidebar. The main content area shows the 'Customize your display' page. A red circle highlights the scroll bar near the center of the window. The page displays the resolution 'Resolution: 1920 x 1080 (Recommended)' and a slider for adjusting the resolution. Below the slider are 'Apply' and 'Cancel' buttons. The 'More options' section is visible below, showing a dropdown menu set to 'Default' and a link to 'Connect to a wireless display'.

Step 10: Drag the cursor left and right until the resolution is greater than or equal to the minimum resolution. As the cursor is being dragged, the 1280 x 800 minimum screen resolution might be shown.

Requirement 5: 1280 x 800 minimum screen resolution. Since the supported width (1920) is greater than (or equal to) the minimum required width (1280) and the supported height (1080) is greater than (or equal to) the minimum required height (800), the requirement has been met.

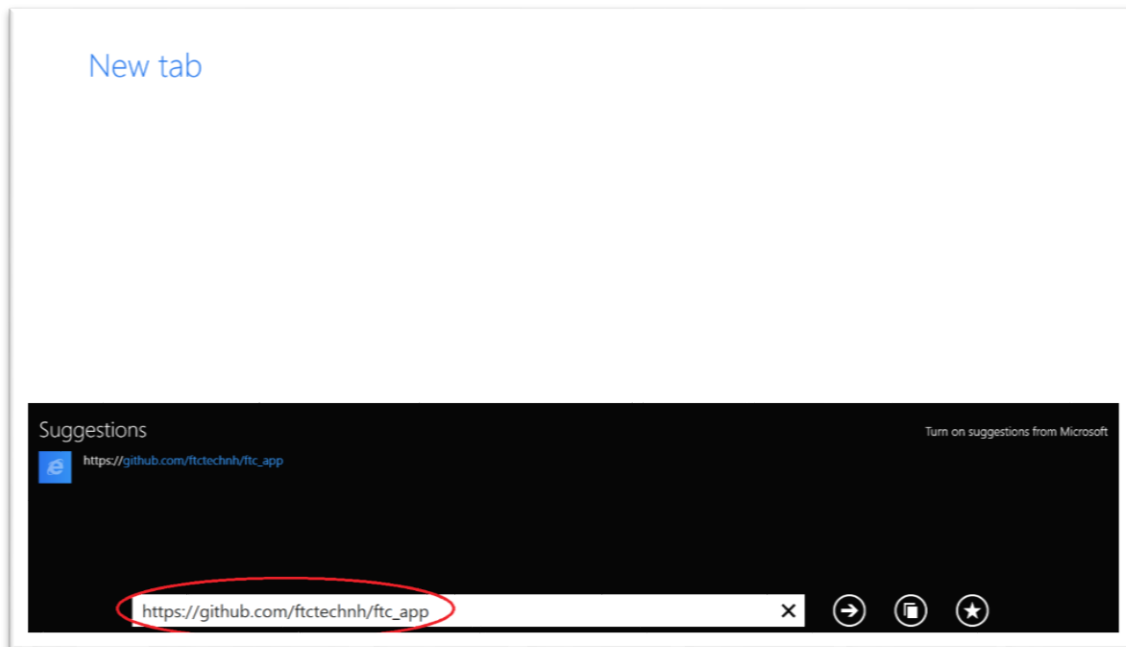
Requirement 6: Java Development Kit (JDK) 7. Since this document describes the process to install JDK 7, this requirement will be met.

Optional Requirement 7: Since the last requirement 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 Internet Explorer and enter 'https://github.com/ftctechnh/ftc_app' into the Address field.



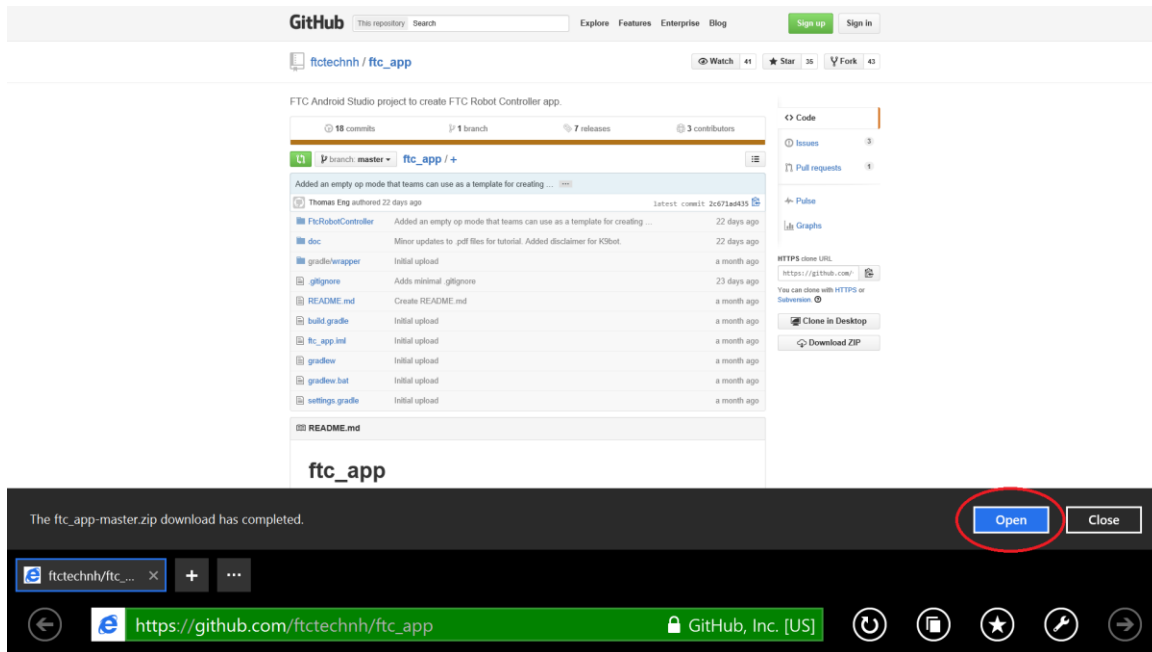
Step 2: After entering the URL from the above step, the following page is shown. Select the “Download ZIP” option.

The screenshot shows the GitHub repository page for `ftctechnh/ftc_app`. The repository is an Android Studio project to create an FTC Robot Controller app. The commit history is visible, showing the latest commit by Thomas Eng. The right-hand sidebar contains options for cloning the repository, including a red circle around the **Download ZIP** button.

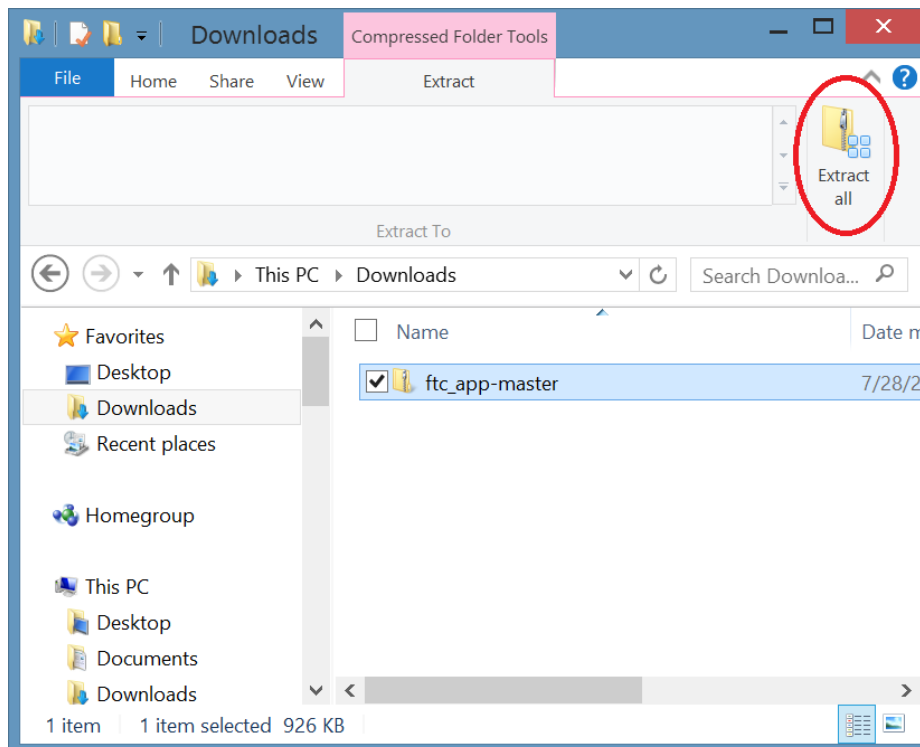
Step 3: After selecting the Download ZIP button, the browser displays a banner to open the file using a decompression utility or to save the file to a location on the machine. Select Save.

The screenshot shows the browser displaying a download dialog box for the file `ftc_app-master.zip` from `codeload.github.com`. The dialog box asks, "Do you want to open or save `ftc_app-master.zip` from `codeload.github.com`?" and provides three buttons: **Open**, **Save** (circled in red), and **Cancel**. The browser's address bar shows the URL `https://github.com/ftctechnh/ftc_app`.

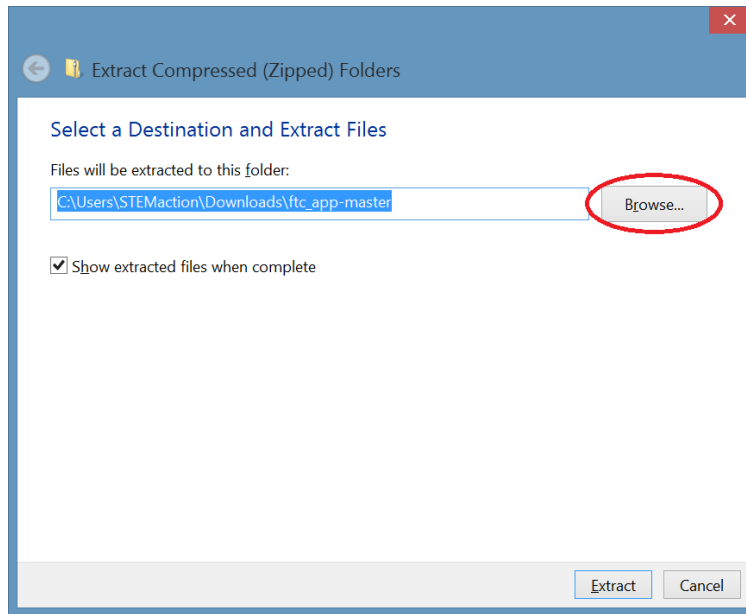
Step 4: After downloading the file, select Open.



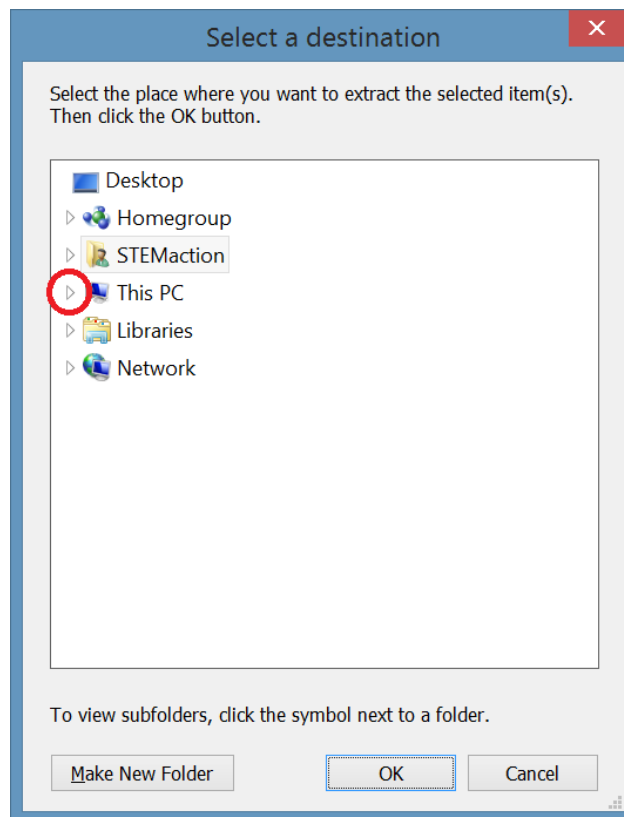
Step 5: After performing the previous step, File Explorer should open. The FTC SDK is a compressed file with a '.zip' extension. The full name of the file is ftc_app-master.zip or ftc_app.zip; the name will vary depending on how the zip file is obtained. If Windows is configured to hide file extensions (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 all'.



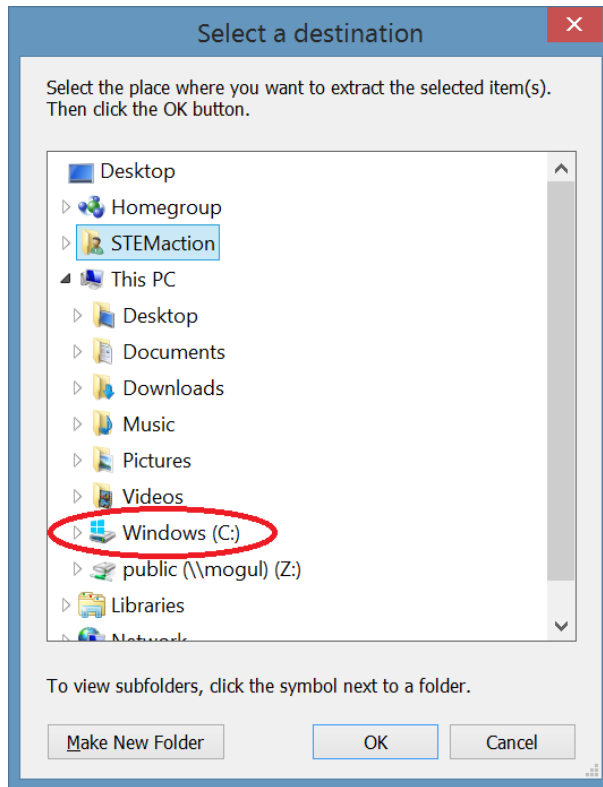
Step 6: Select 'Browse'.



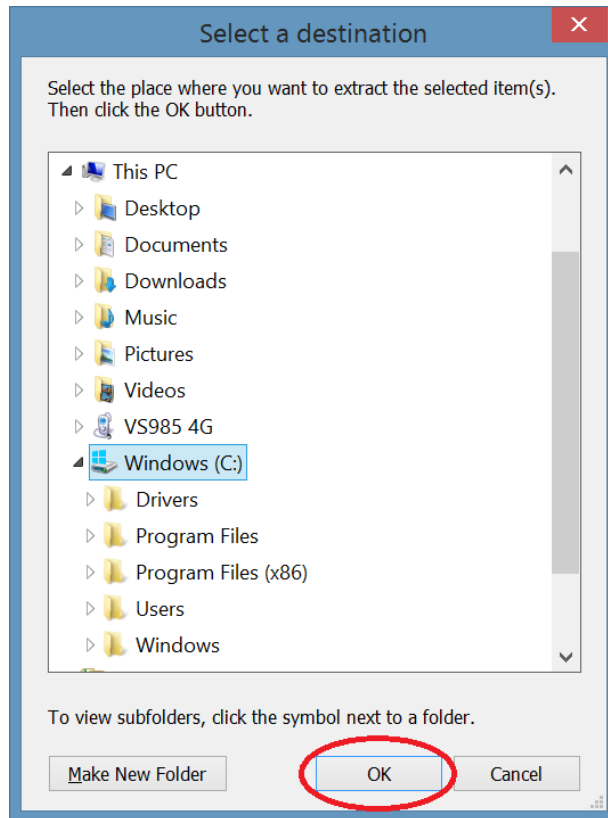
Step 7: Select a directory on your machine's hard drive for the location of the copied file. For this document the "Windows (C:)" folder will eventually be chosen. It will take several steps to find it. Select the small arrow next to 'This PC'.



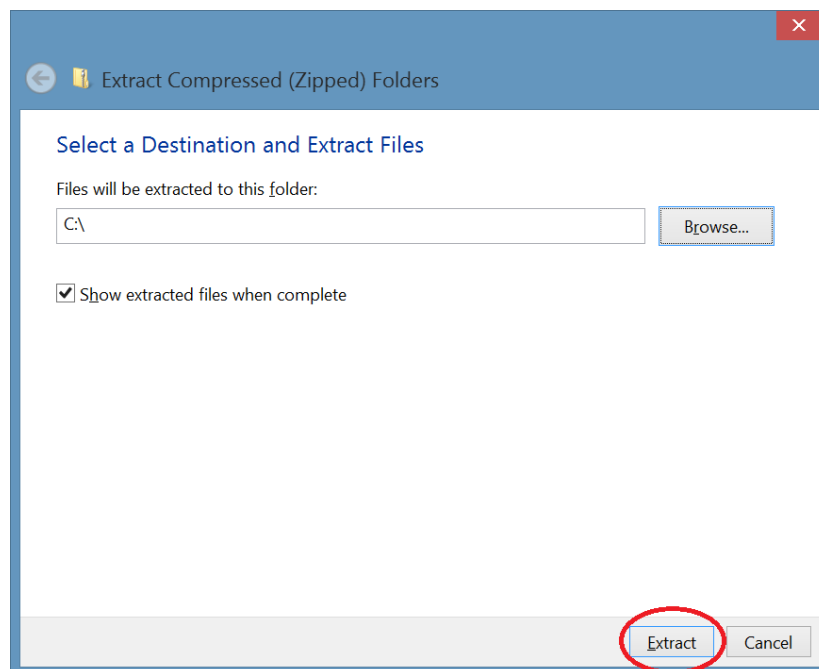
Step 8: Select 'Windows (C:)'.
Step 9: Select 'C:\Program Files (x86)\Common-IO\Software\PushBot\'.
Step 10: Click 'Extract All'.



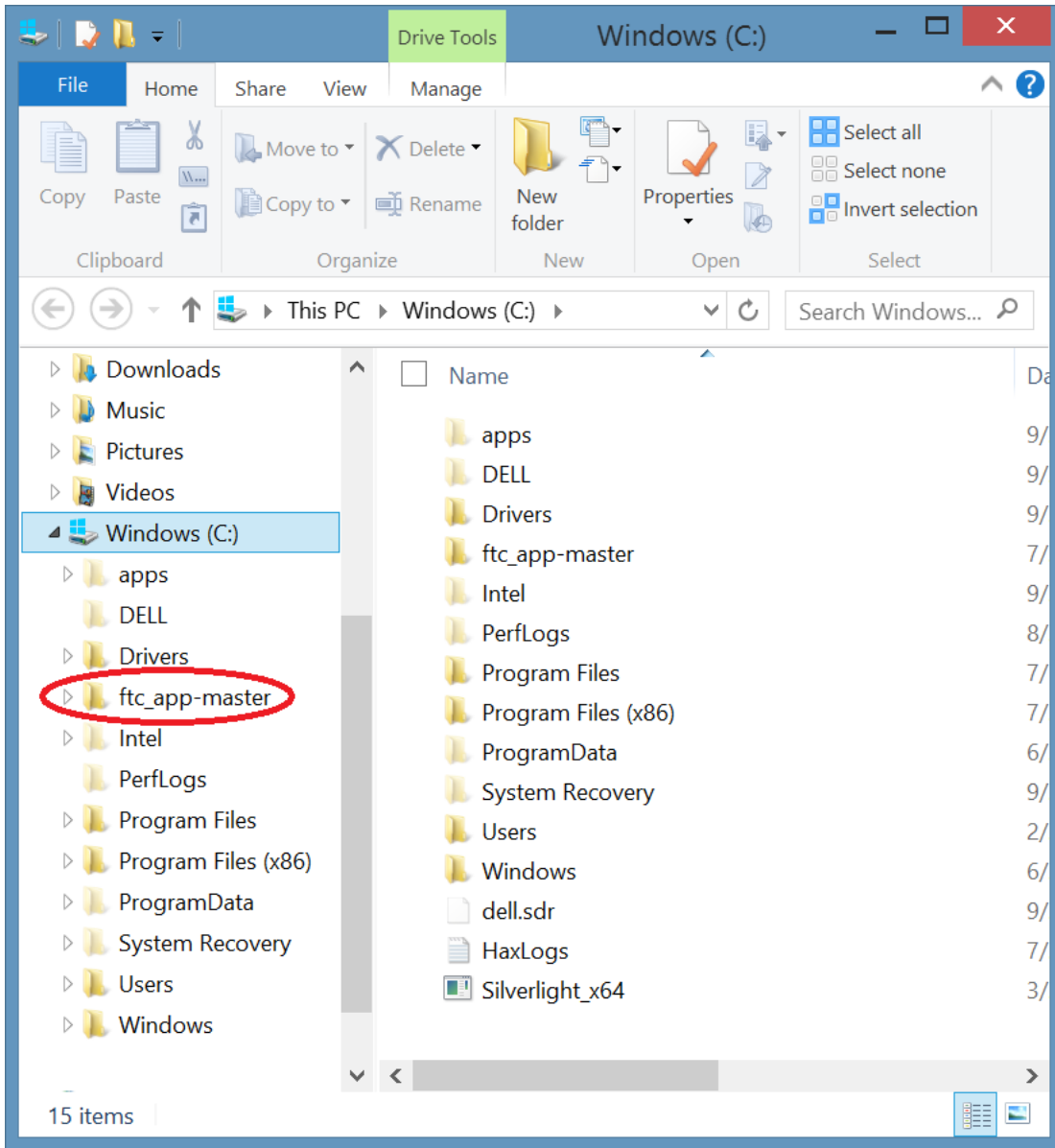
Step 9: Select OK and the dialog will close and the Extract Compressed (Zipped) Folders dialog will become active (shown in the next step).



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/jre7-downloads-1880261.html> Accept the License Agreement.

Java SE Development Kit 7u79

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux x86	130.4 MB	jdk-7u79-linux-i586.rpm
Linux x86	147.6 MB	jdk-7u79-linux-i586.tar.gz
Linux x64	131.69 MB	jdk-7u79-linux-x64.rpm
Linux x64	146.4 MB	jdk-7u79-linux-x64.tar.gz
Mac OS X x64	196.89 MB	jdk-7u79-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.79 MB	jdk-7u79-solaris-i586.tar.Z
Solaris x86	96.66 MB	jdk-7u79-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.67 MB	jdk-7u79-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u79-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140 MB	jdk-7u79-solaris-sparc.tar.Z
Solaris SPARC	99.4 MB	jdk-7u79-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24 MB	jdk-7u79-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.4 MB	jdk-7u79-solaris-sparcv9.tar.gz
Windows x86	138.31 MB	jdk-7u79-windows-i586.exe
Windows x64	140.06 MB	jdk-7u79-windows-x64.exe

[Back to top](#)

Java SE Runtime Environment 7u79

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux x86	31.62 MB	jre-7u79-linux-i586.rpm
Linux x86	46.28 MB	jre-7u79-linux-i586.tar.gz
Linux x64	32.14 MB	jre-7u79-linux-x64.rpm
Linux x64	44.92 MB	jre-7u79-linux-x64.tar.gz
Mac OS X x64	49.65 MB	jre-7u79-macosx-x64.dmg

Step 2: Select the download that matches your computer. Refer to section (Determining whether Android Studio can be installed, step 6 to determine the system type. For this example, it is `jdk-7u79-windows-x64.exe`.

7u79 JDK MOS Checksum
7u80 JDK MOS Checksum

What is the difference between a Java CPU (7u79) and PSU (7u80) release?
Java SE Critical Patch Updates (CPU) contain fixes to security vulnerabilities and critical bug fixes. Oracle strongly recommends that all Java SE users upgrade to the latest CPU releases as they are made available. Most users should choose this release.
Java SE Patch Set Updates (PSU) contain all of the security fixes in the CPU released up to that version, as well as additional non-critical fixes. Java PSU releases should only be used if you are being impacted by one of the additional bugs fixed in that version.

Visit Java CPU and PSU Releases Explained for details.

Looking for JDK on ARM?
JDK 7 for ARM downloads have moved to the JDK 7 for ARM download page.

Java SE Development Kit 7u79

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux i586	130.4 MB	jdk-7u79-linux-i586.rpm
Linux i586	147.6 MB	jdk-7u79-linux-i586.tar.gz
Linux i64	131.69 MB	jdk-7u79-linux-i64.rpm
Linux i64	146.4 MB	jdk-7u79-linux-i64.tar.gz
Mac OS X i64	196.89 MB	jdk-7u79-macosx-i64.dmg
Solaris x86 (SVR4 package)	140.79 MB	jdk-7u79-solaris-i586.tar.Z
Solaris x86	96.66 MB	jdk-7u79-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.67 MB	jdk-7u79-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u79-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140 MB	jdk-7u79-solaris-sparc.tar.Z
Solaris SPARC	99.4 MB	jdk-7u79-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24 MB	jdk-7u79-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.4 MB	jdk-7u79-solaris-sparcv9.tar.gz
Windows i586	138.31 MB	jdk-7u79-windows-i586.exe
Windows x64	140.06 MB	jdk-7u79-windows-x64.exe

Java SE Development Kit 7u79 Demos and Samples Downloads

You must accept the Oracle BSD License to download this software.

Accept License Agreement Decline License Agreement

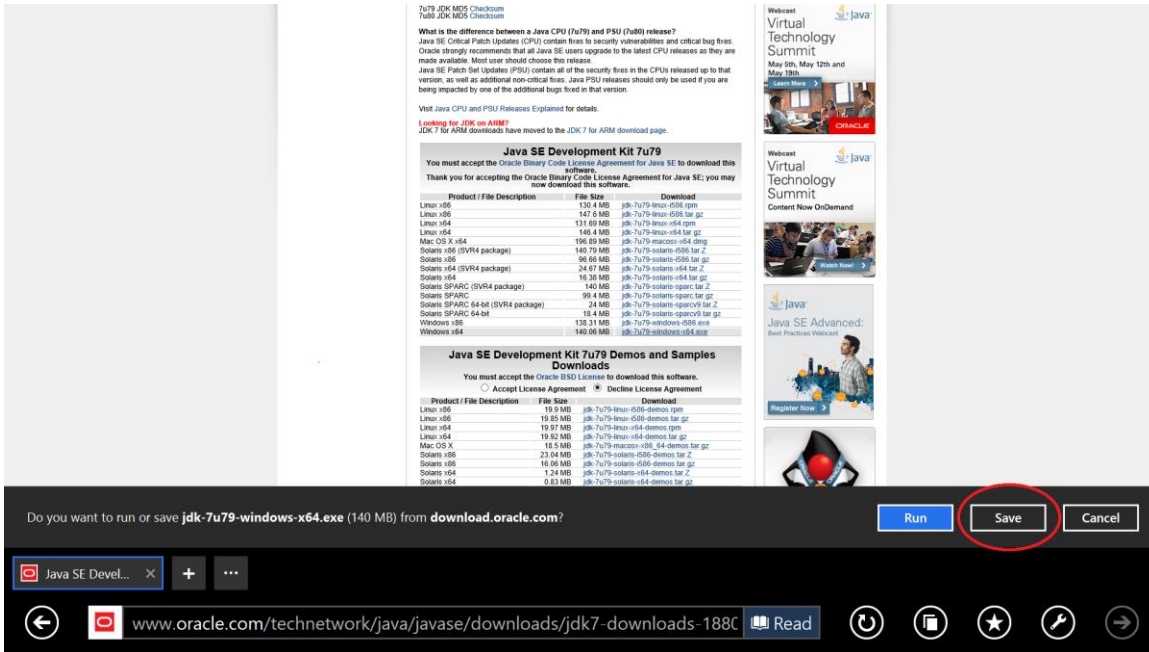
Product / File Description	File Size	Download
Linux i586	19.9 MB	jdk-7u79-linux-i586-demos.rpm
Linux i586	19.85 MB	jdk-7u79-linux-i586-demos.tar.gz
Linux i64	19.97 MB	jdk-7u79-linux-i64-demos.rpm
Linux i64	19.92 MB	jdk-7u79-linux-i64-demos.tar.gz
Mac OS X	19.5 MB	jdk-7u79-macosx-i64_demos.tar.gz
Solaris i586	23.04 MB	jdk-7u79-solaris-i586-demos.tar.Z
Solaris i586	16.86 MB	jdk-7u79-solaris-i586-demos.tar.gz
Solaris i64	1.24 MB	jdk-7u79-solaris-i64-demos.tar.Z
Solaris i64	0.23 MB	jdk-7u79-solaris-i64-demos.tar.gz
Solaris SPARC	23.34 MB	jdk-7u79-solaris-sparc-demos.tar.Z
Solaris SPARC	16.11 MB	jdk-7u79-solaris-sparc-demos.tar.gz
Solaris SPARC 64-bit	1.35 MB	jdk-7u79-solaris-sparcv9-demos.tar.Z
Solaris SPARC 64-bit	0.89 MB	jdk-7u79-solaris-sparcv9-demos.tar.gz
Windows i586	20.1 MB	jdk-7u79-windows-i586-demos.zip
Windows x64	20.6 MB	jdk-7u79-windows-x64-demos.zip

http://download.oracle.com/otn-pub/java/jdk/7u79-b15/jdk-7u79-windows-x64.exe

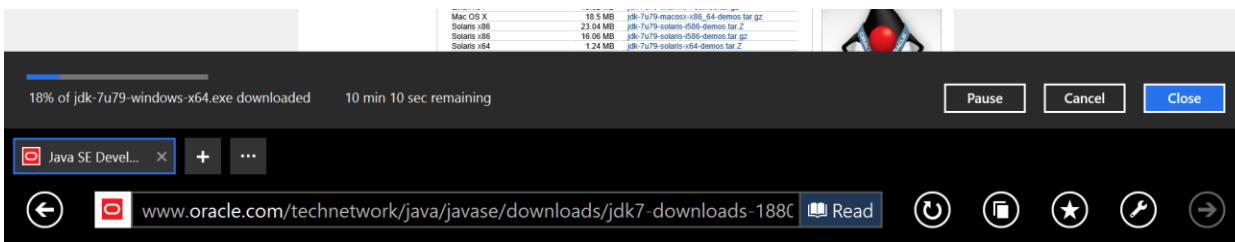
Java SE Devel... x + ...

www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880261.html

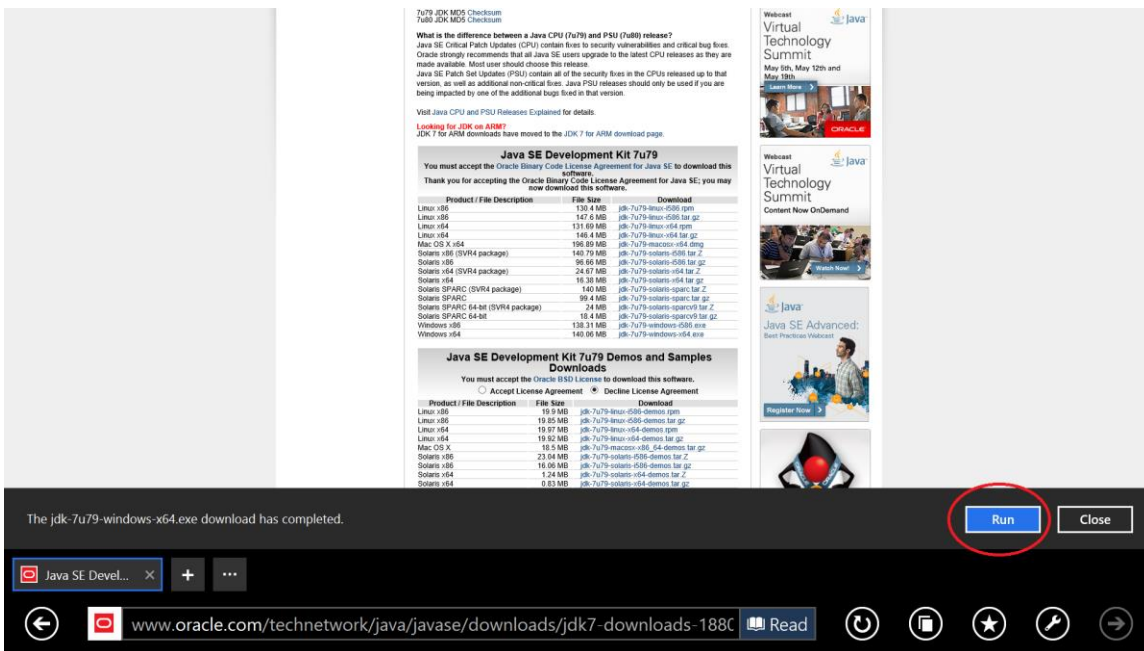
Step 3: Internet Explorer should display a downloading banner. Select Save (arrow).



After selecting save, the download will begin. Internet Explorer will display the estimated amount of time remaining.



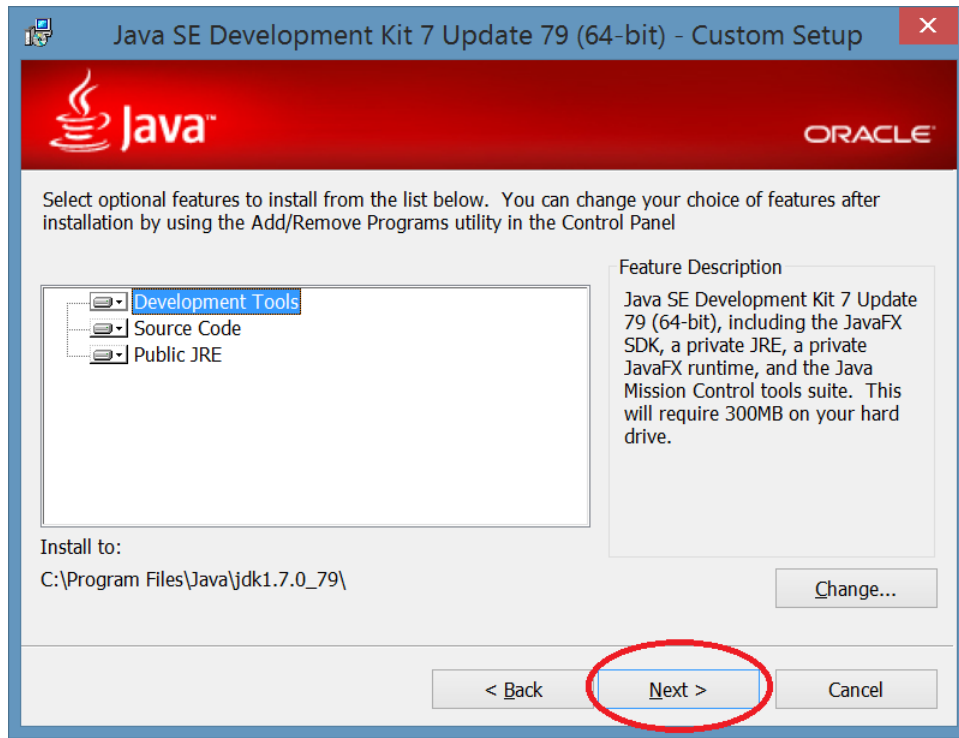
Step 4: After the download is complete, select Run. If a window is displayed that asks “Do you want to allow the following program to make changes to this computer?”, select Yes.



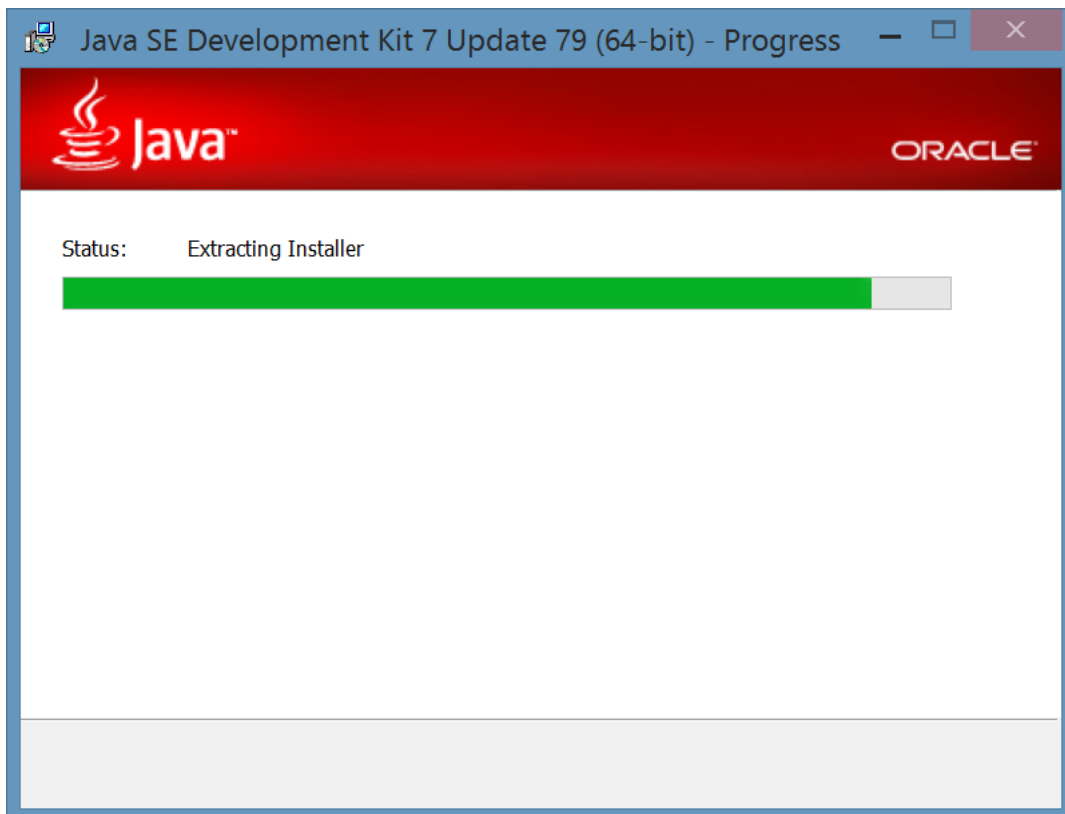
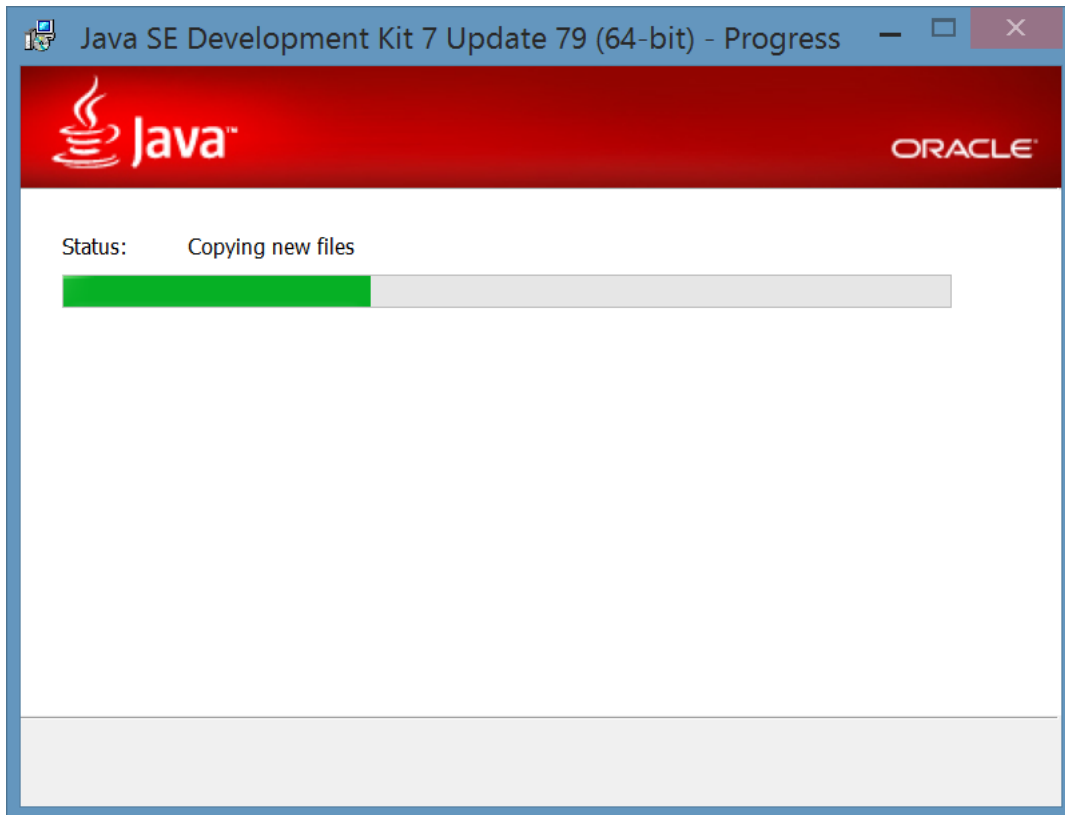
Step 5: The installation of Java will begin by displaying the following window. Select Next.



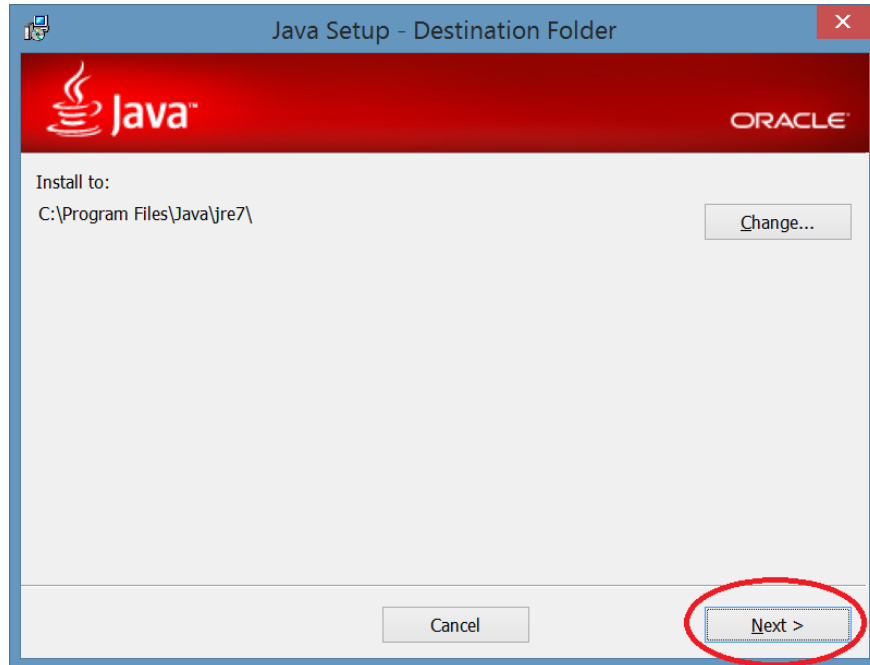
Step 6: The following dialog can be used to select which parts are installed and where the files will be located. Choose Next to accept the default installation.



While the files are being installed the installer will display the following windows.



Step 7: The installation continues by selecting where the files will be located. Choose Next to accept the default installation.



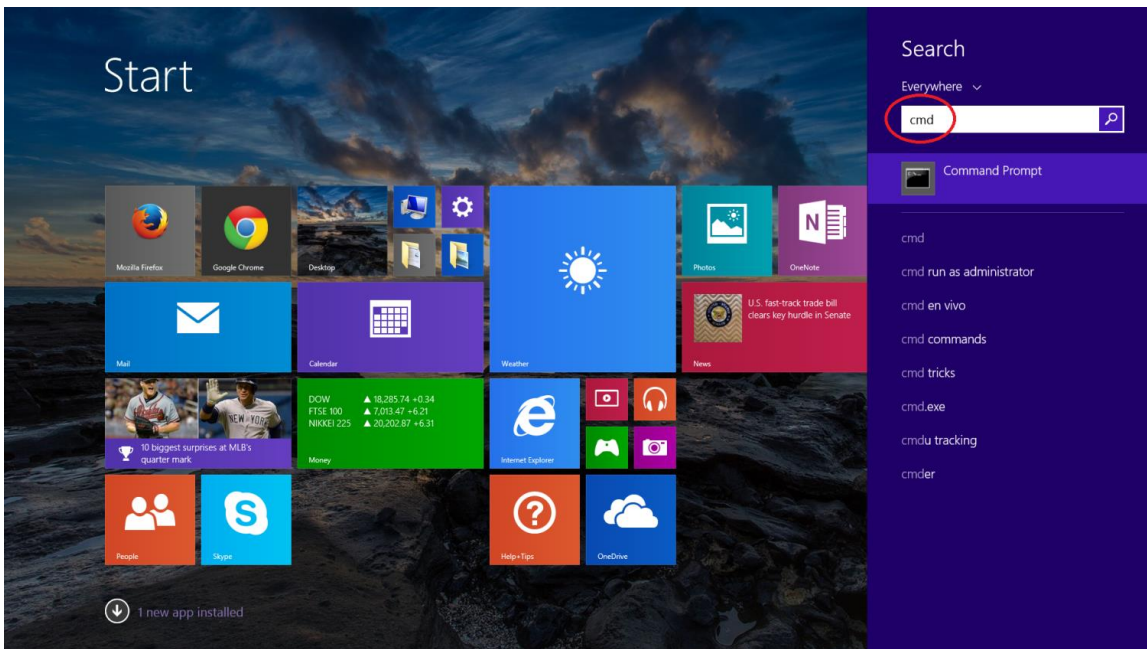
While the files are being installed the following dialog will be displayed.



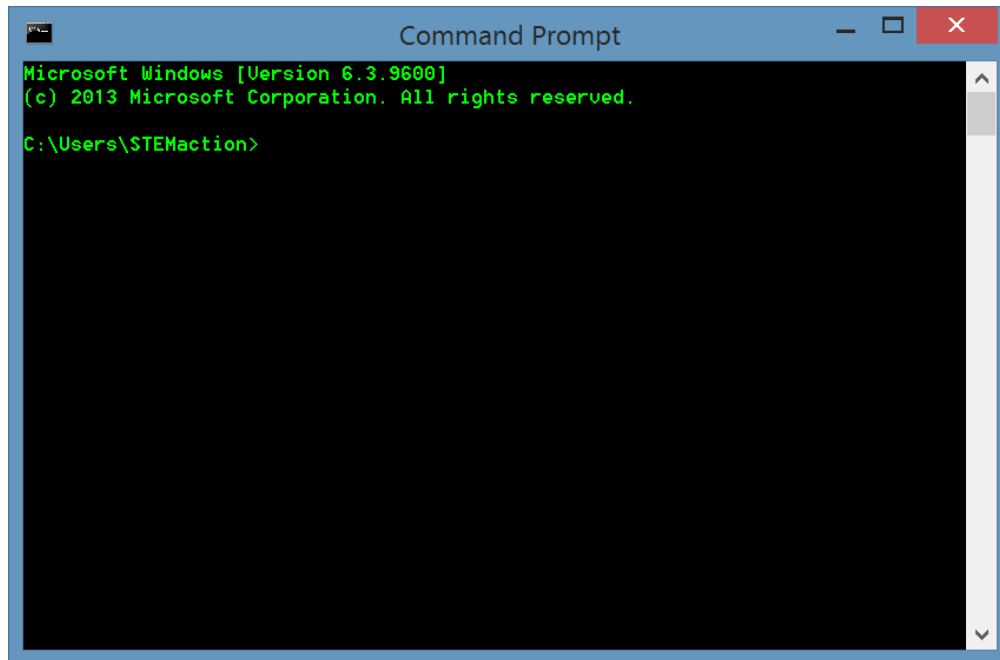
Step 8: After the installation is complete, choose Close to exit the installer.



Step 9: 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.



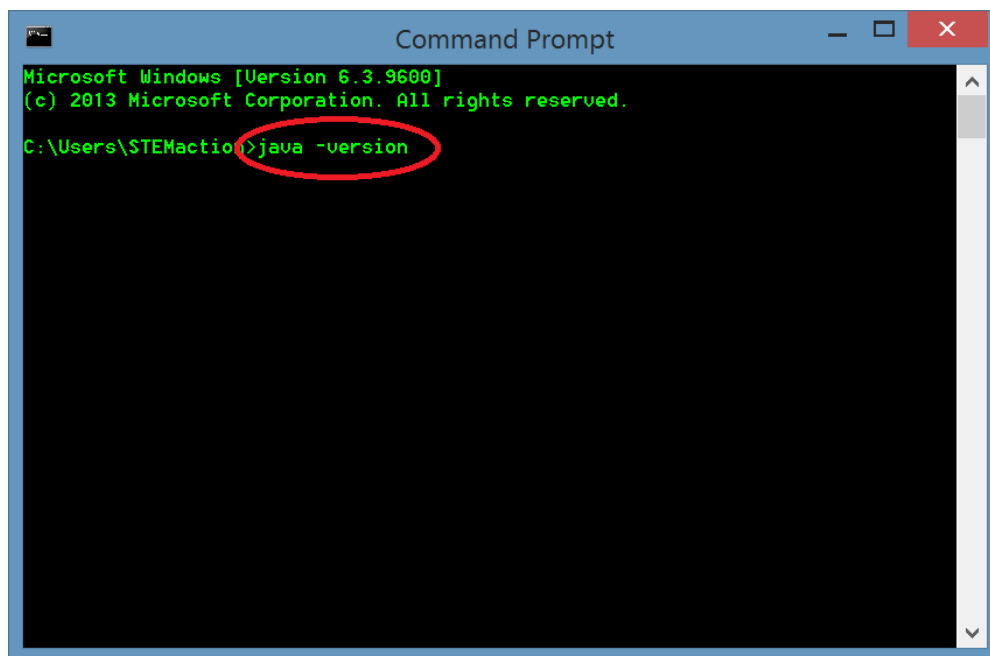
The following window will be displayed.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.


C:\Users\STEMaction>
```

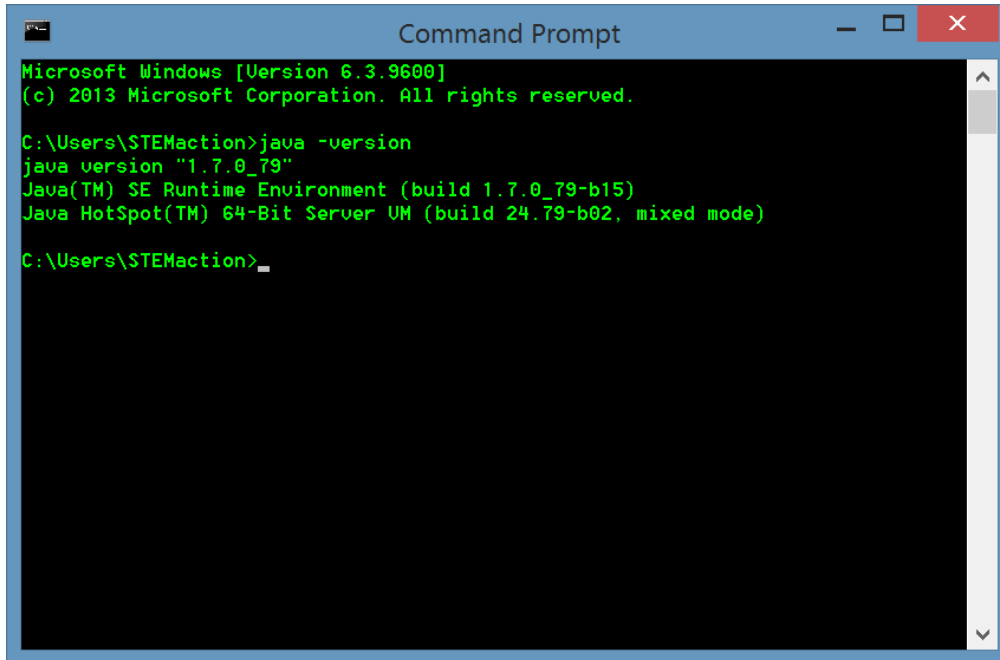
Step 10: Type “java -version” and press return/enter



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\STEMaction> java -version
```


Step 11: Verify that the version is "1.7.XXX". The close button () can be used to close the window.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\STEMaction>java -version
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)

C:\Users\STEMaction>
```

Installing the Android Studio Integrated Development Environment (IDE)

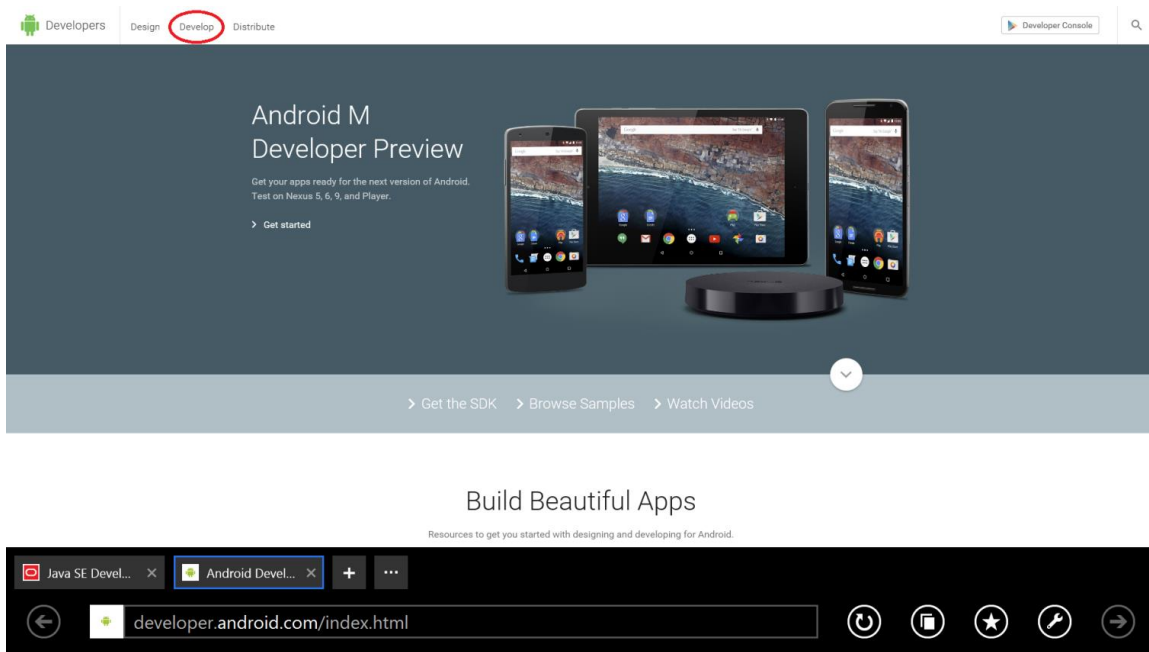
The FTC SDK and Java Development kit have previously been installed. The next task is to install the Android Studio IDE.

Step 1: The IDE is available for Windows, Mac OS X and Linux and can be downloaded from the <http://developer.android.com> website. Open Internet Explorer and enter the URL as shown below.

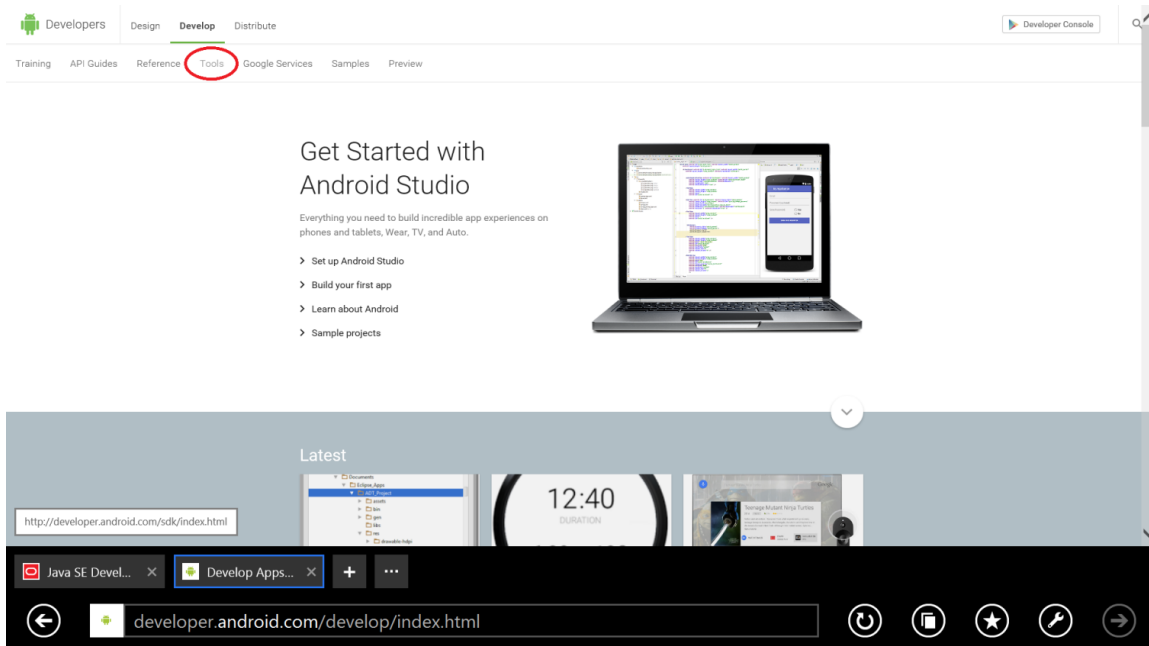
New tab



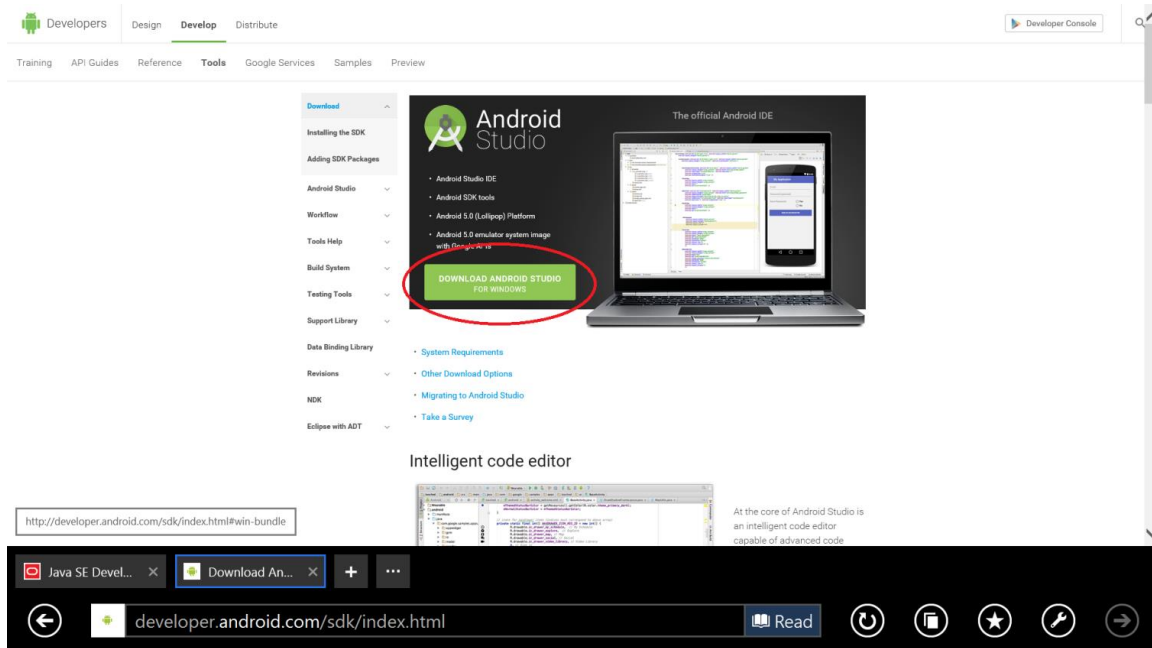
Step 2: Select “Develop”.



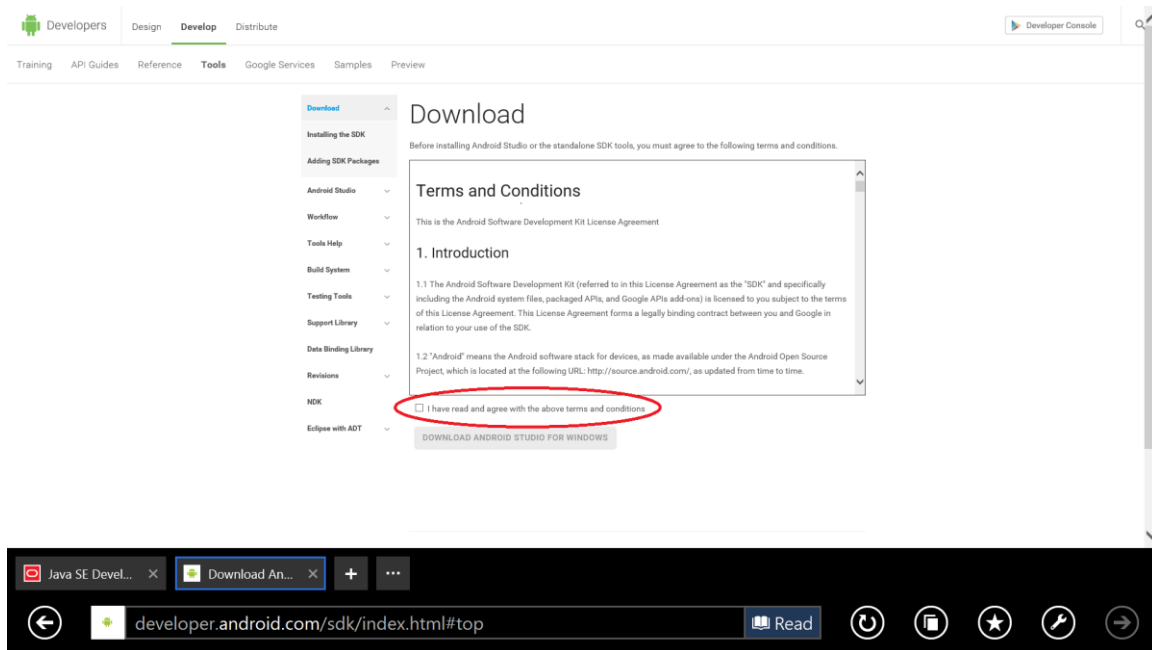
Step 3: Select Tools on the submenu.



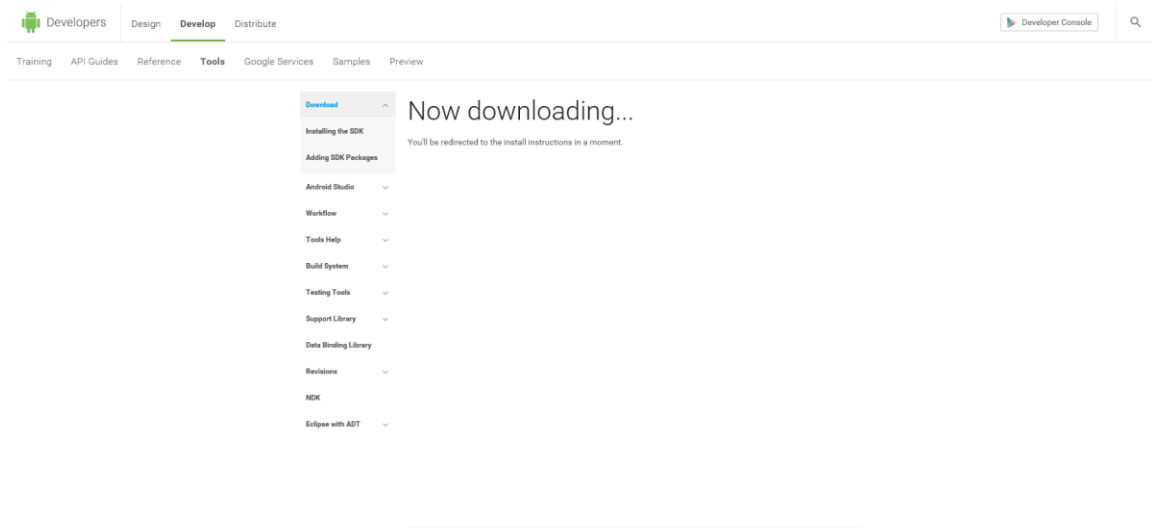
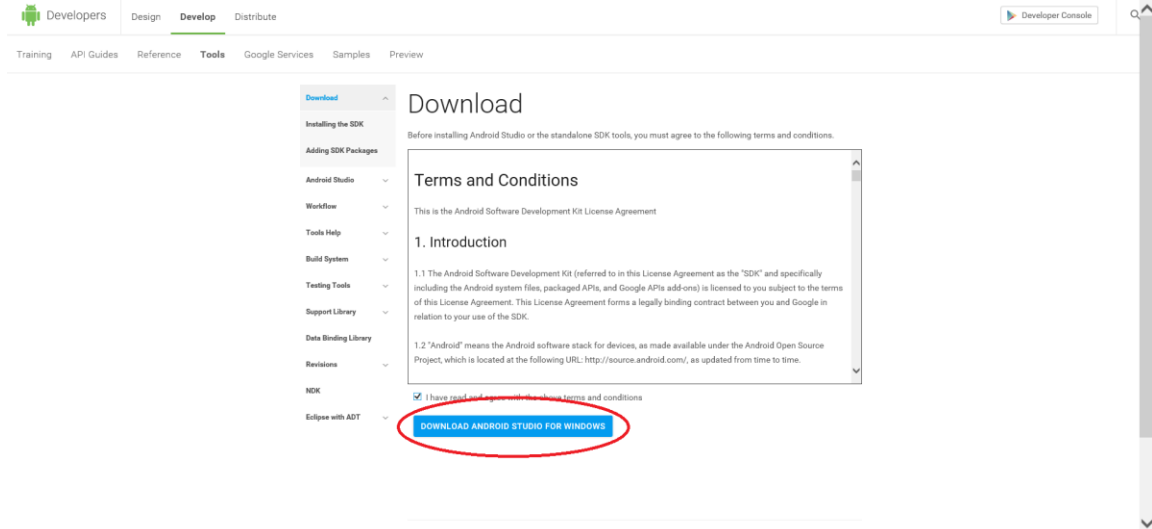
Step 4: Select the green “Download Android Studio for Windows” button.



Step 5: Accept the terms and conditions from the next page to begin the download.



Step 6: After checking the terms and conditions, the download button will be selectable. Select the Download Android Studio for Windows button. Another page might be displayed before the download starts.



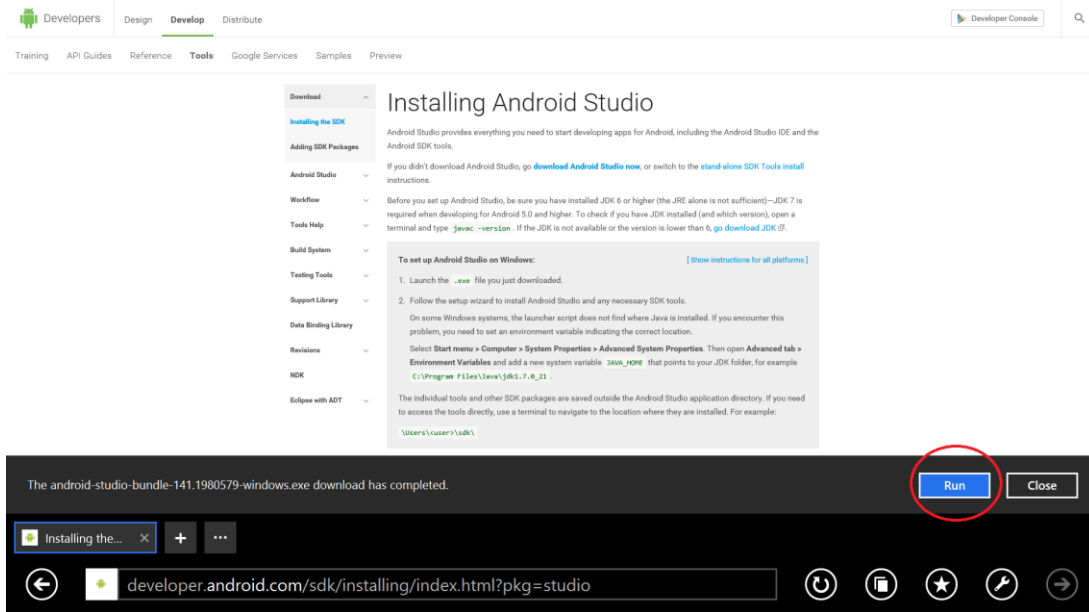
Step 7: Internet Explorer should display a downloading banner. Select Save. This might take a while (as shown in the image below the following image).

The image consists of two screenshots of the Android Studio installation page on the developer.android.com website, overlaid with a Windows taskbar showing the download progress.

Top Screenshot: The browser window displays the "Installing Android Studio" page. The "Run" button in the download banner is circled in red. The page content includes instructions for installing Android Studio on Windows, such as launching the .exe file and following the setup wizard.

Bottom Screenshot: The browser window displays the same page, but the download progress bar shows "47% of android-studio-bundle-141.1980579-windows.exe downloaded" with "3 min 21 sec remaining". The "Close" button in the download banner is visible.

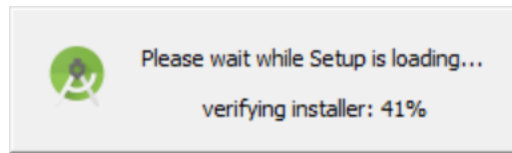
Step 8: After the download is complete, select Run. As the installer is loading, the image below the following image may be shown.



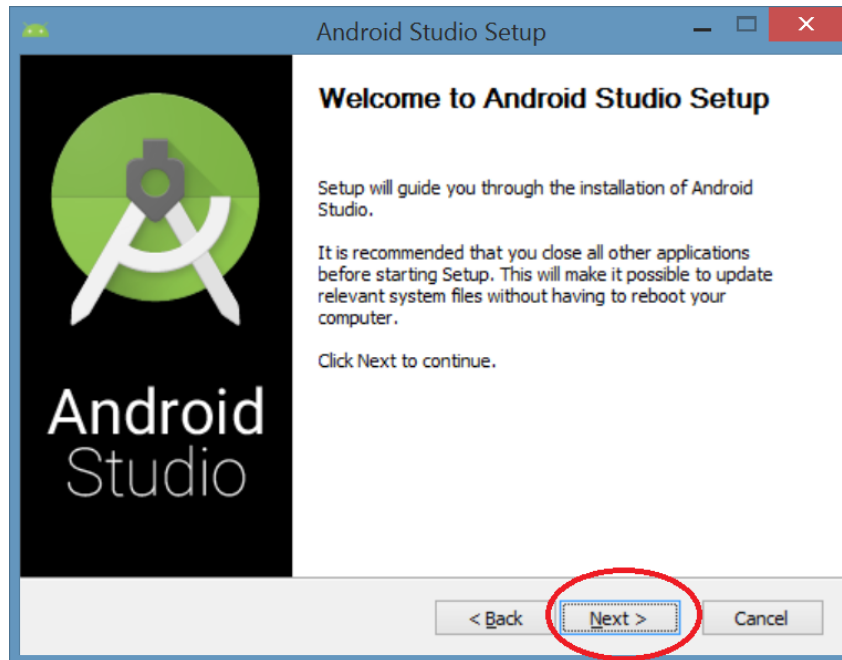
The image shows a screenshot of the Android Studio installation page on the developer.android.com website. The page title is "Installing Android Studio". The content includes instructions for downloading the SDK, adding SDK packages, and setting up Android Studio on Windows. A red circle highlights the "Run" button in the Windows download notification bar.

The notification bar text reads: "The android-studio-bundle-141.1980579-windows.exe download has completed." The "Run" button is circled in red.

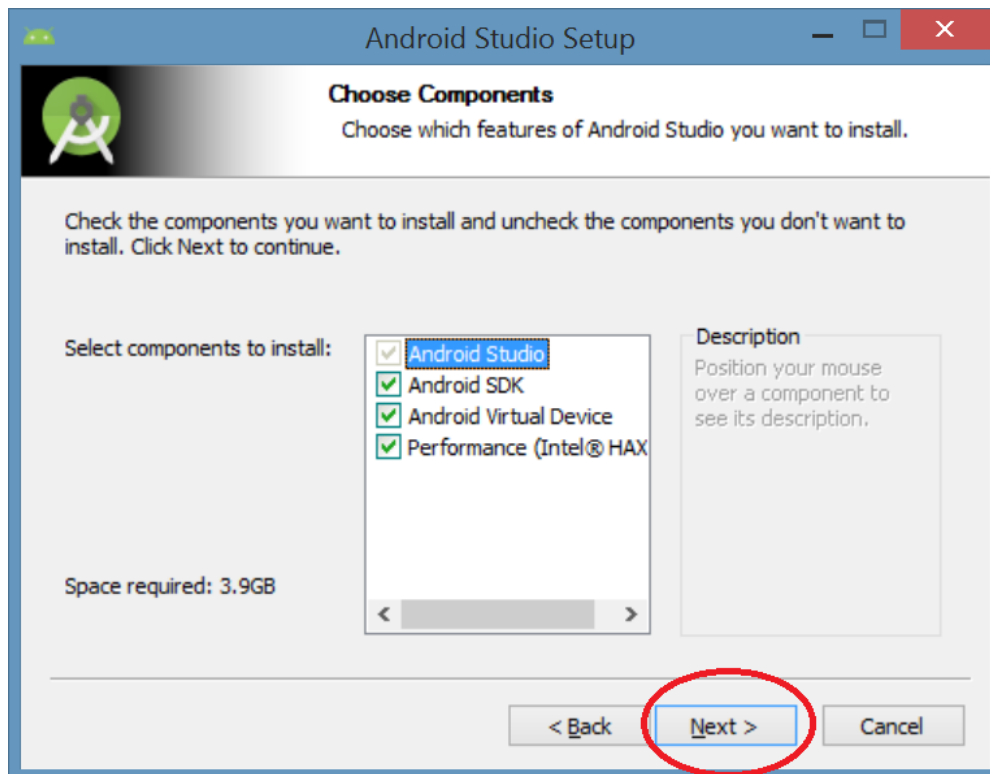
The browser address bar shows: `developer.android.com/sdk/installing/index.html?pkg=studio`



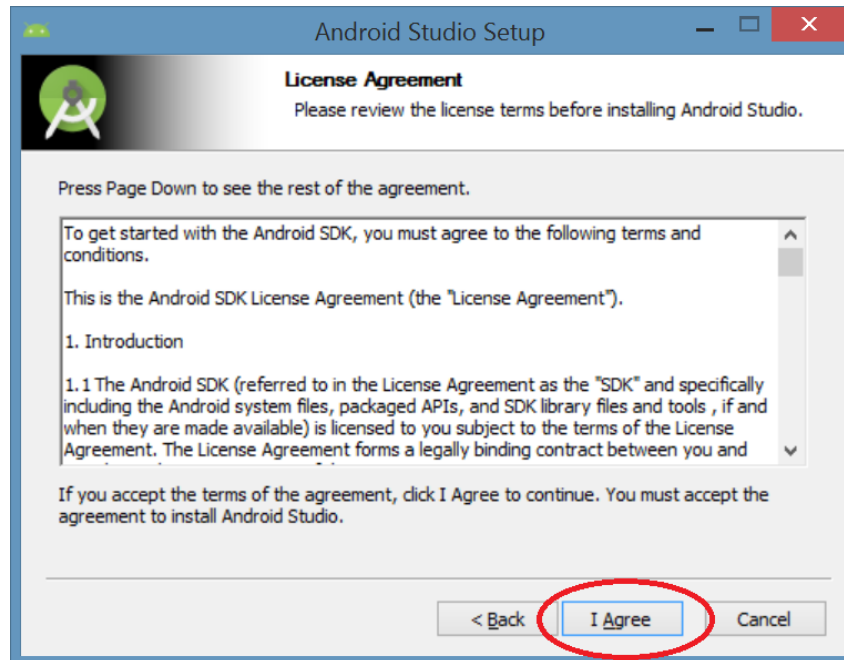
Step 9: To begin the installation process select Next. If a window is displayed that asks “Do you want to allow the following program to make changes to this computer?”, select Yes.



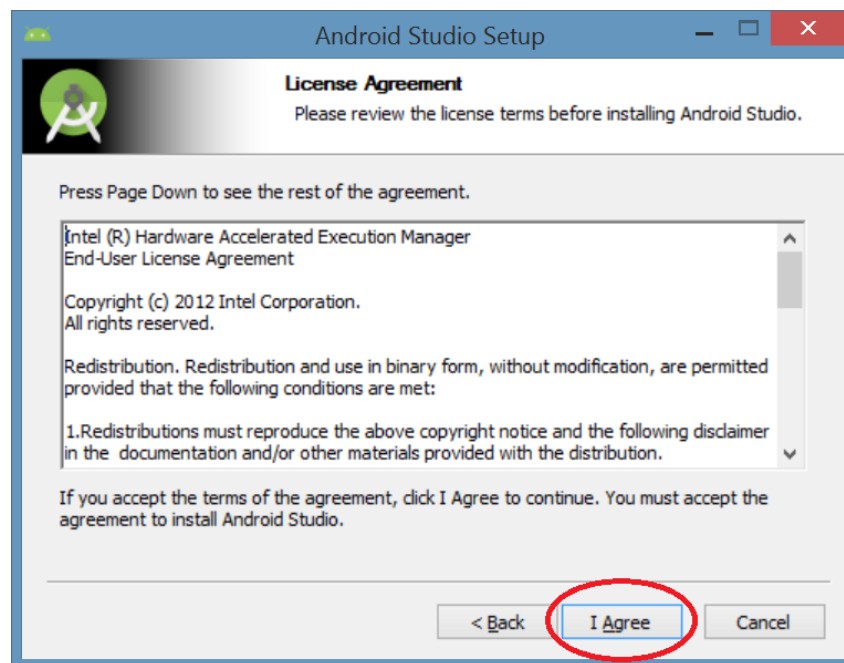
Step 10: The following dialog can be used to select which parts are installed and where the files will be located. Choose Next to accept the default installation.



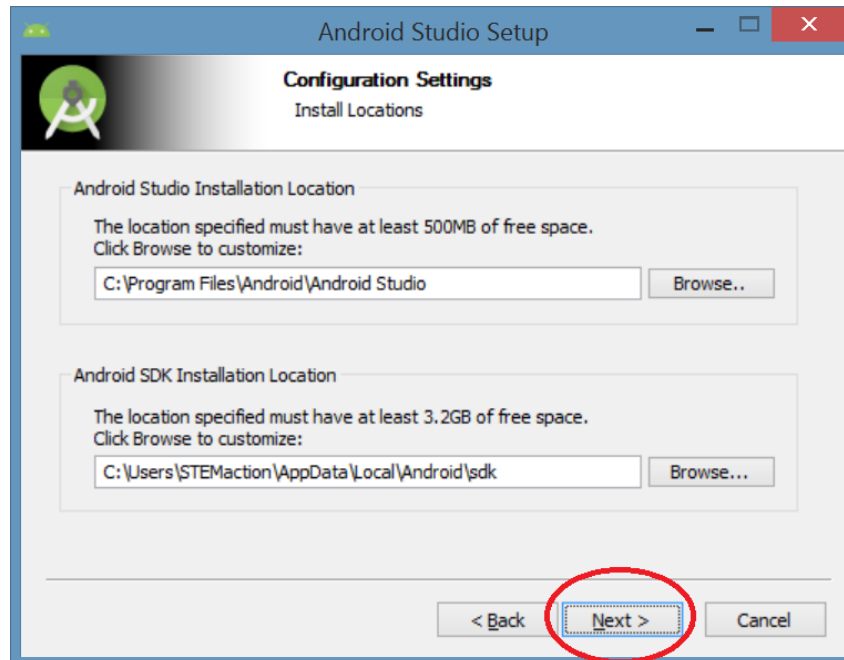
Step 11: Accept the License Agreement for the **Android SDK** by selecting 'I Agree'.



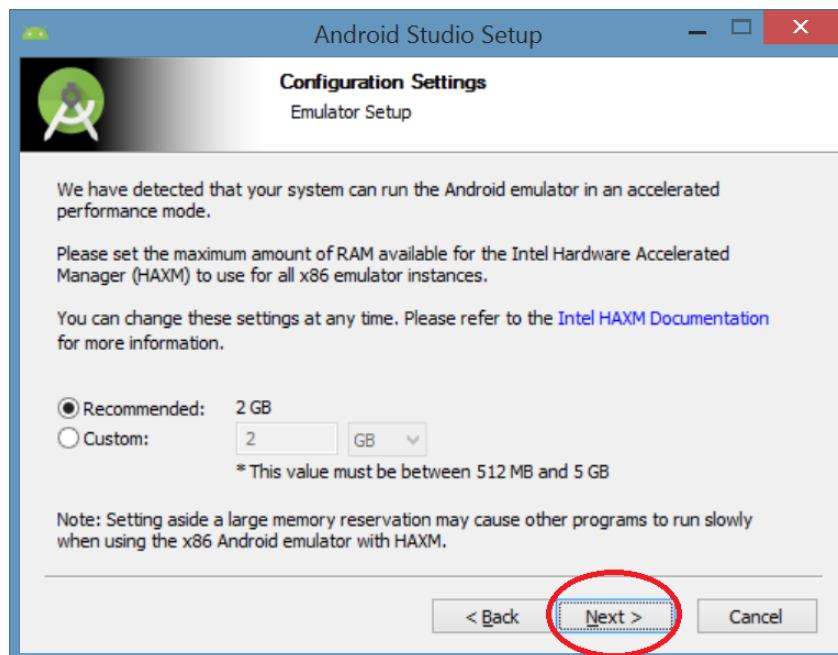
Step 12: Accept the License Agreement for the **Android Studio IDE** by selecting 'I Agree'.



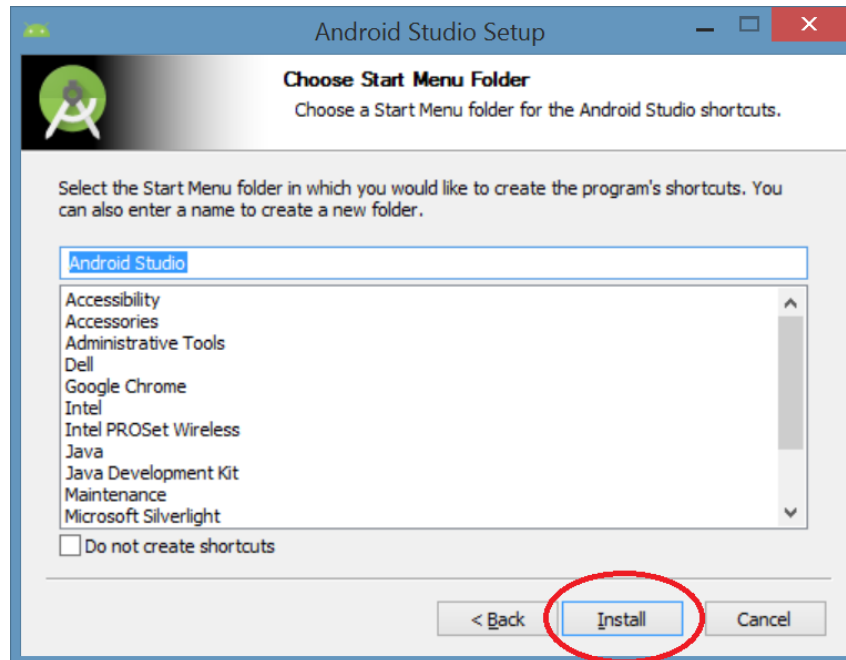
Step 13: Choose Next to accept the default configuration settings.



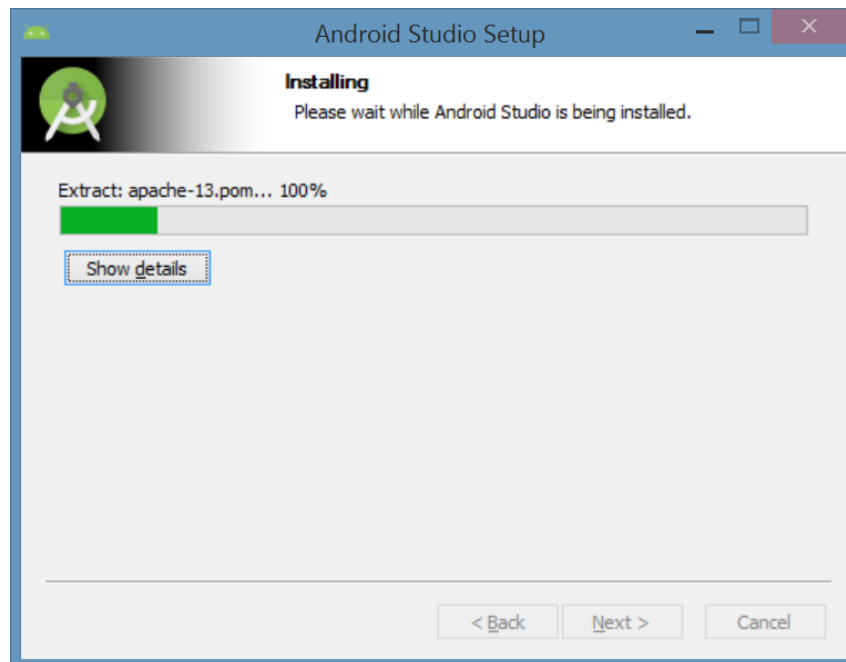
Step 14: Choose 'Next' to accept the default emulator settings.

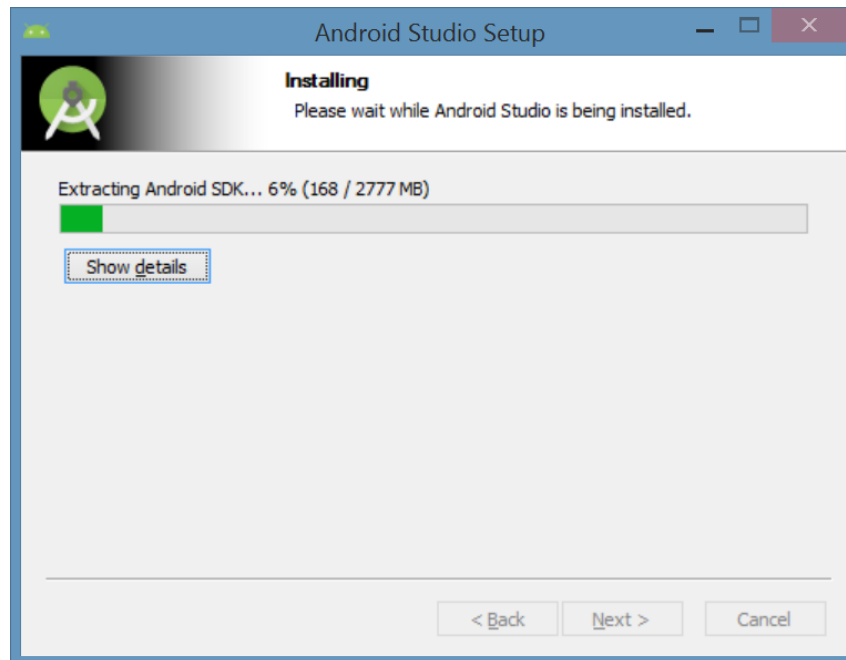


Step 15: Choose 'Install' to accept the default start menu folder.

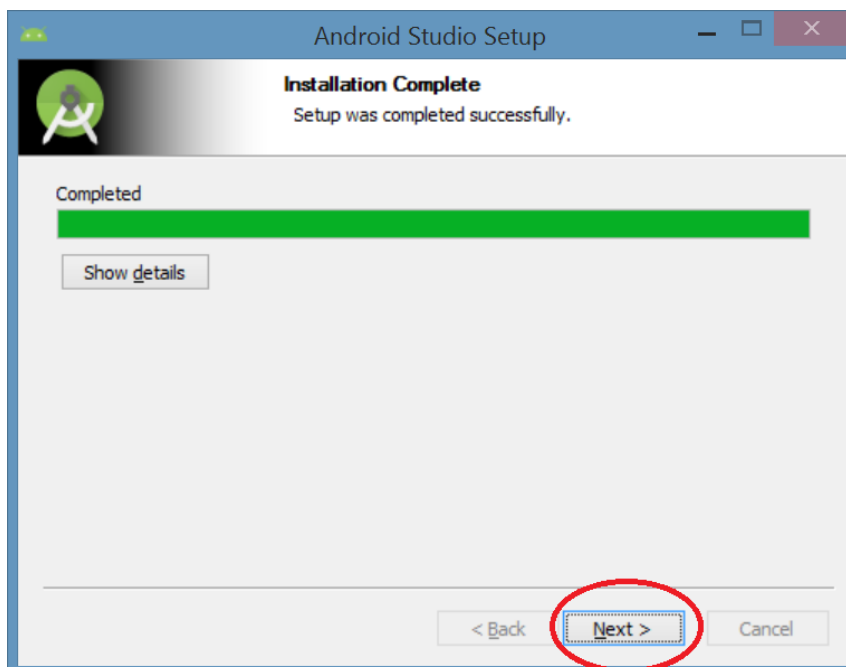


While the install continues, the following pages may be displayed.

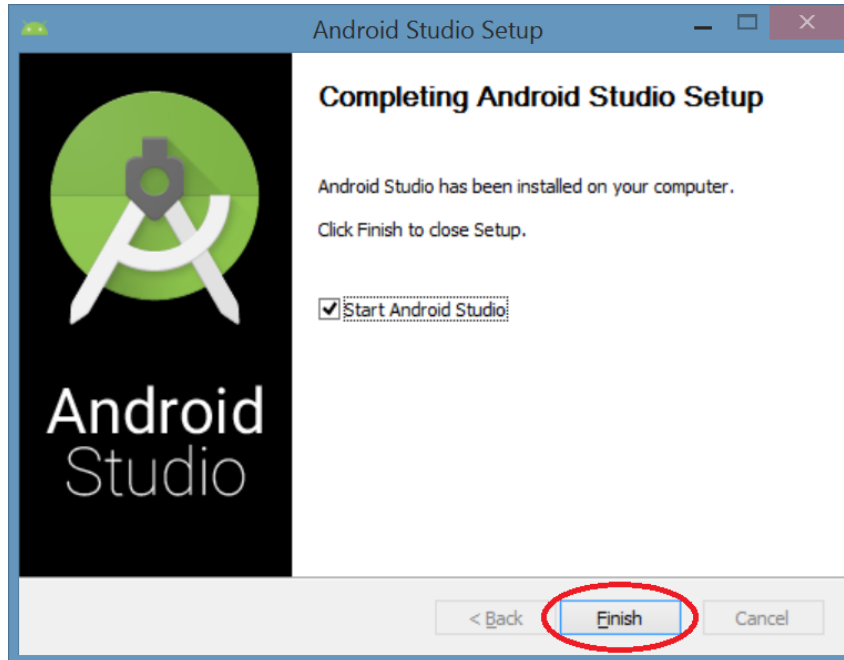




Step 16: When the installation is complete, select 'Next'.



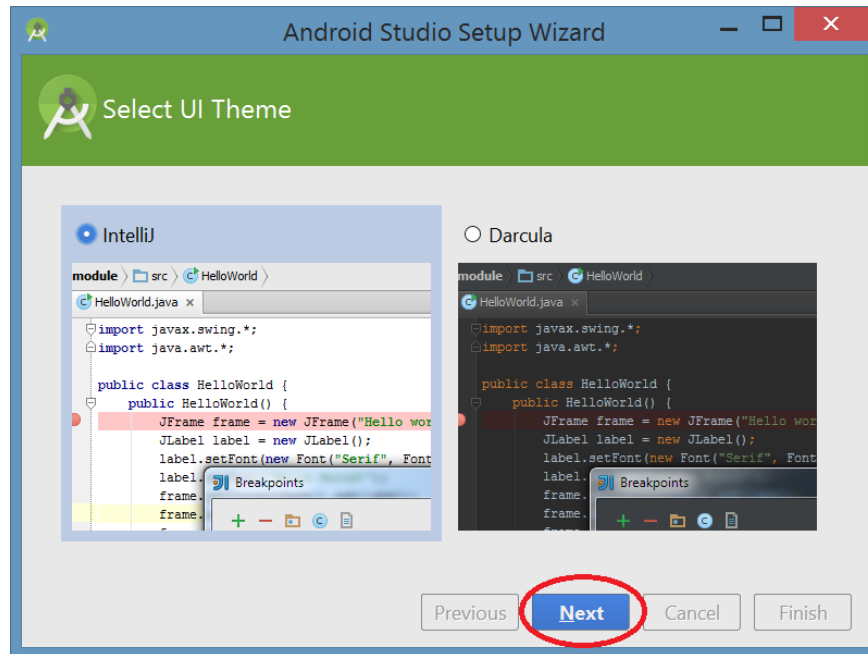
Step 17: Open Android Studio by selecting 'Finish'.



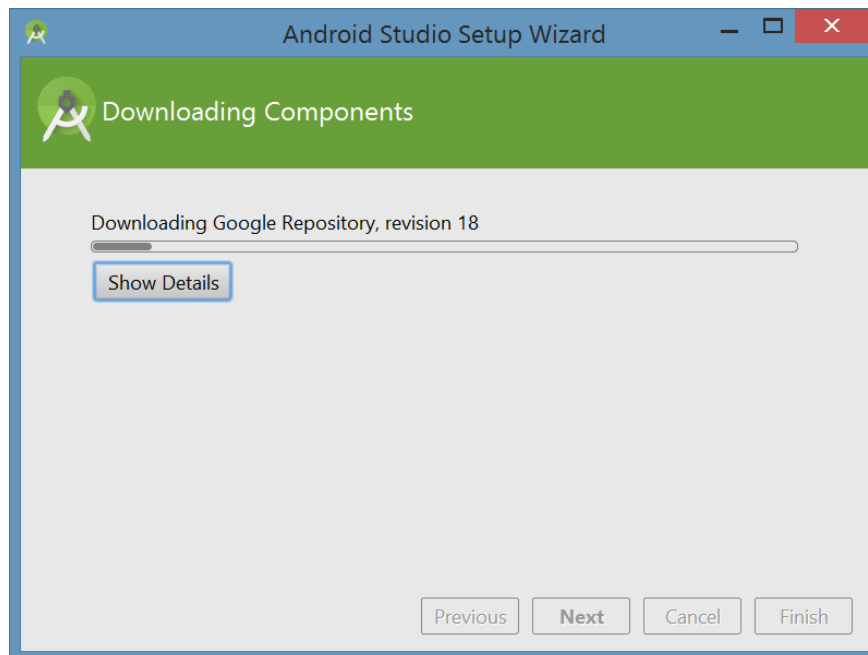
The following screen will be displayed while Android Studio is opening.

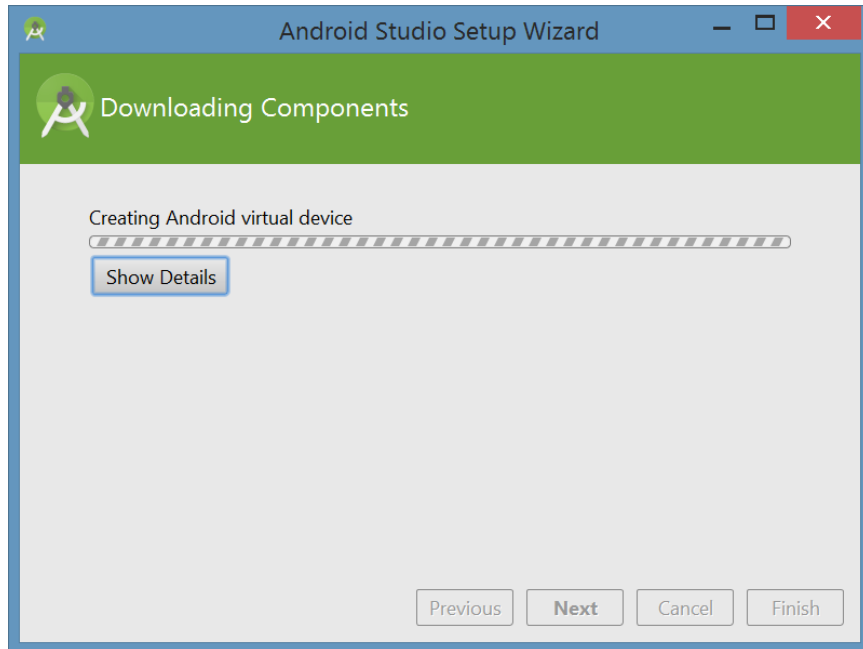


Step 18: Choose 'Next' to select the default UI Theme.

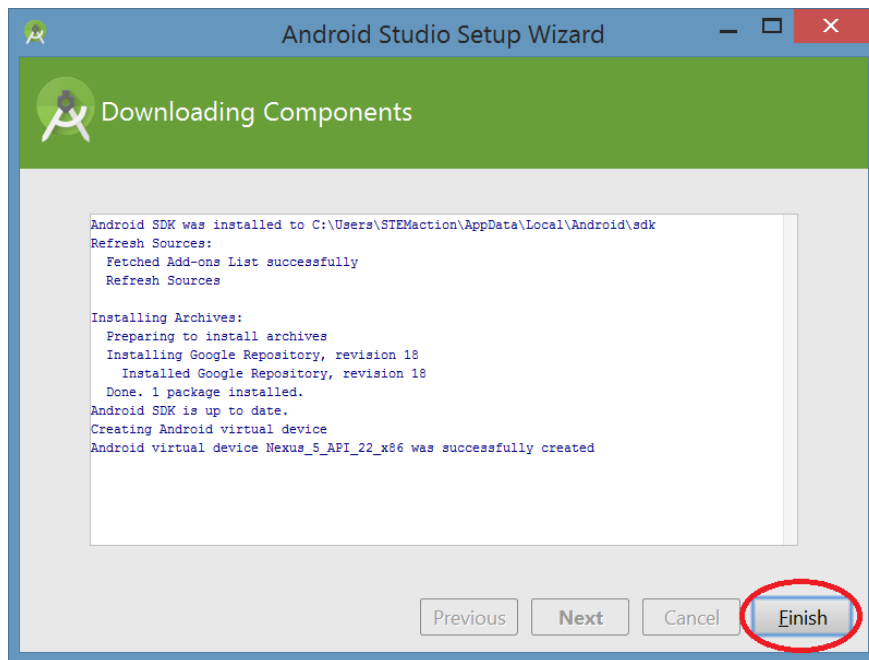


The following windows will be displayed while the setup wizard downloads necessary components.

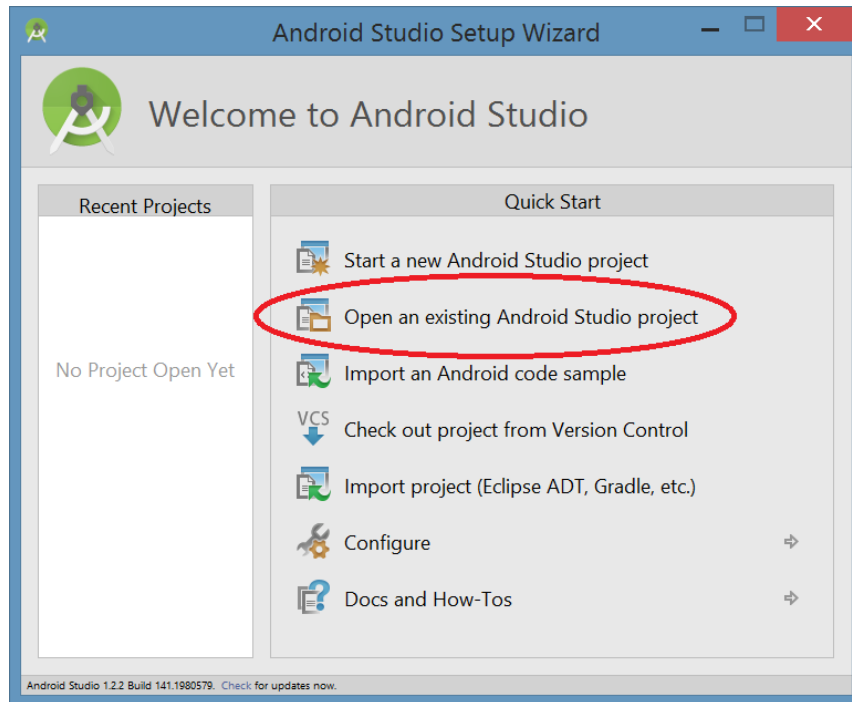




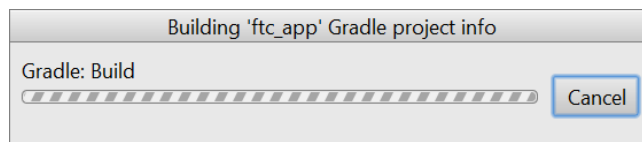
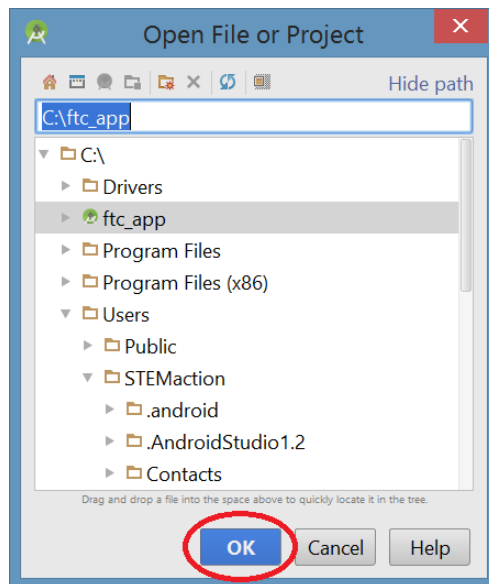
Step 19: Select 'Finish' and the Android Studio Welcome page will be shown.



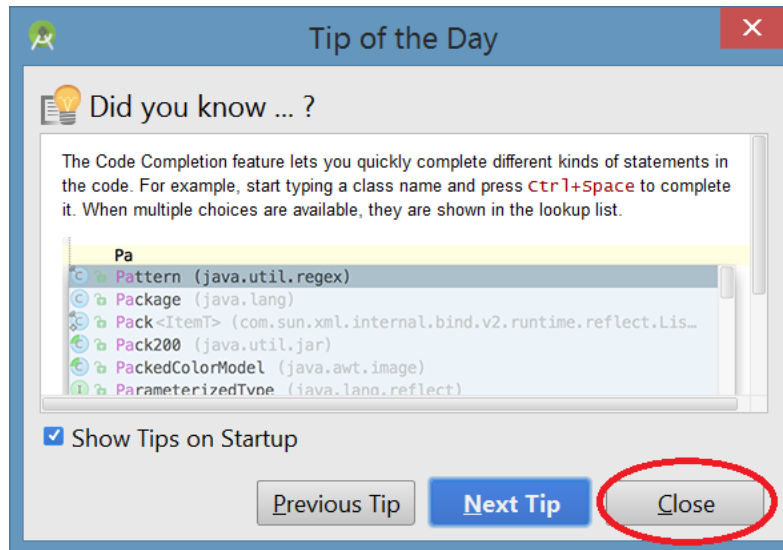
Step 20: Open the FTC SDK by selecting 'Open an existing Android Studio project'.



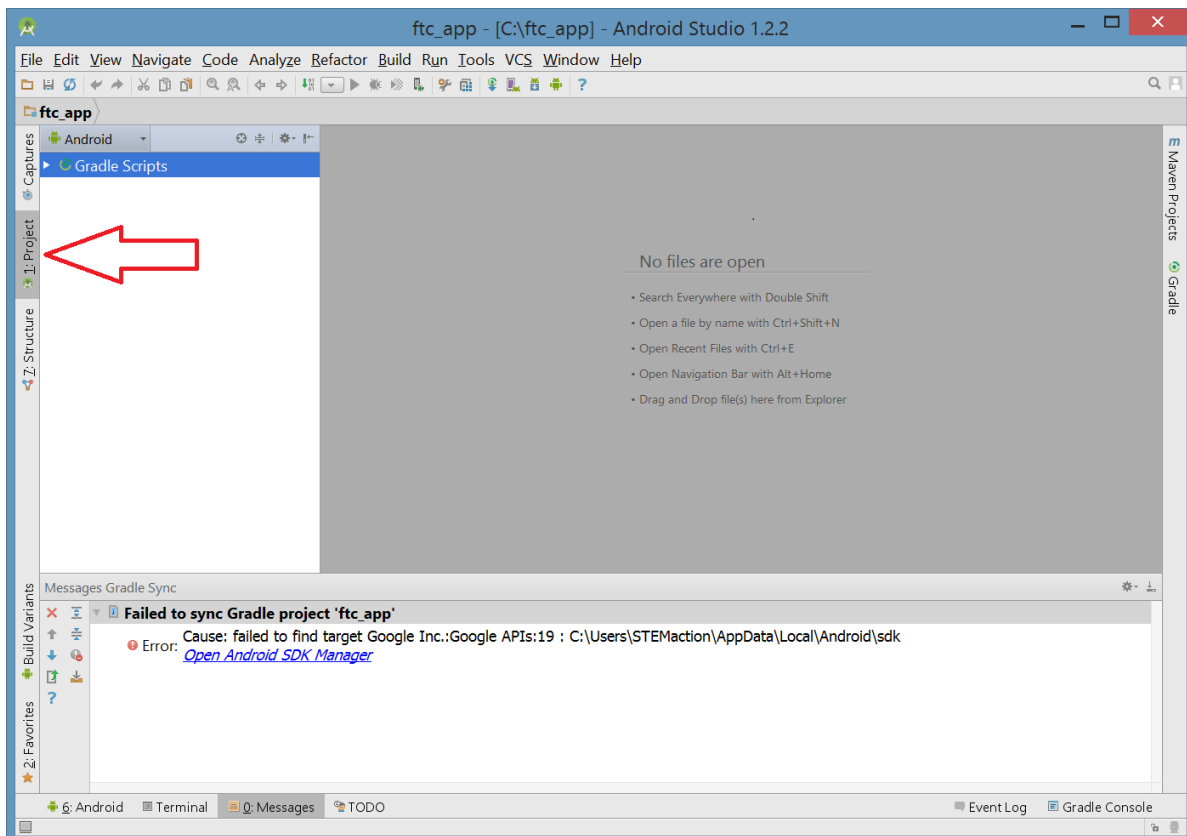
Step 21: It might take a while for the dialog to completely load. If 'ftc_app-master' doesn't show in the list after a long while, close the "c:" folder by clicking the triangle to the left of 'C:\' and reopen the folder by clicking the triangle again. Single click the 'ftc_app-master' folder. Select OK. Android Studio will attempt to build the project and several windows will be displayed before the IDE opens.



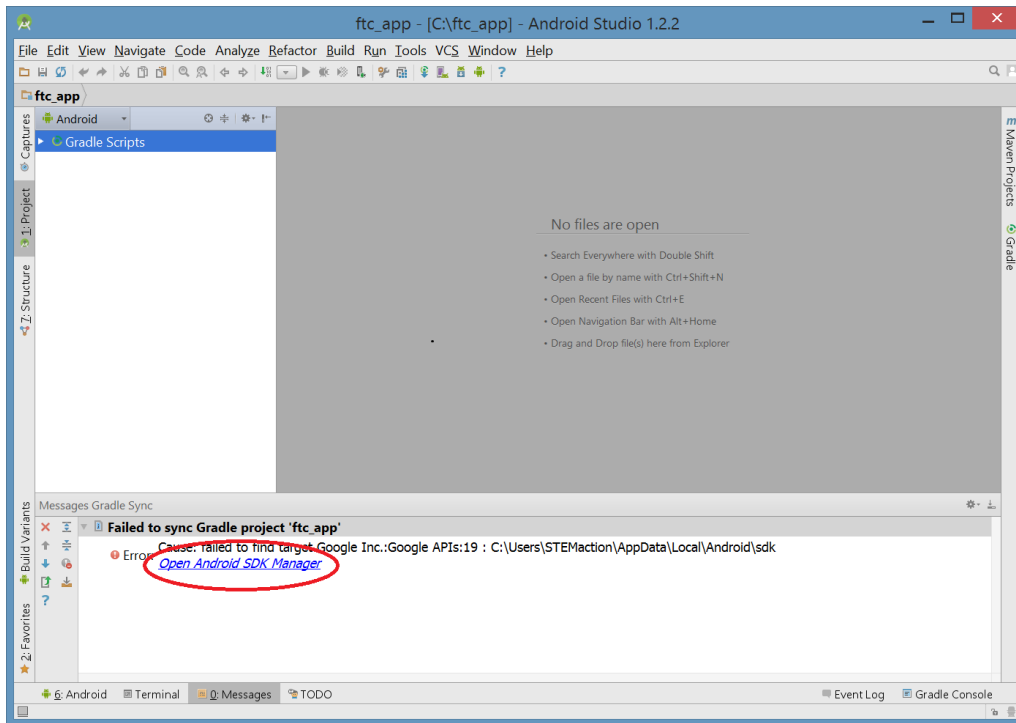
Step 22: Close the Tip of the Day window.



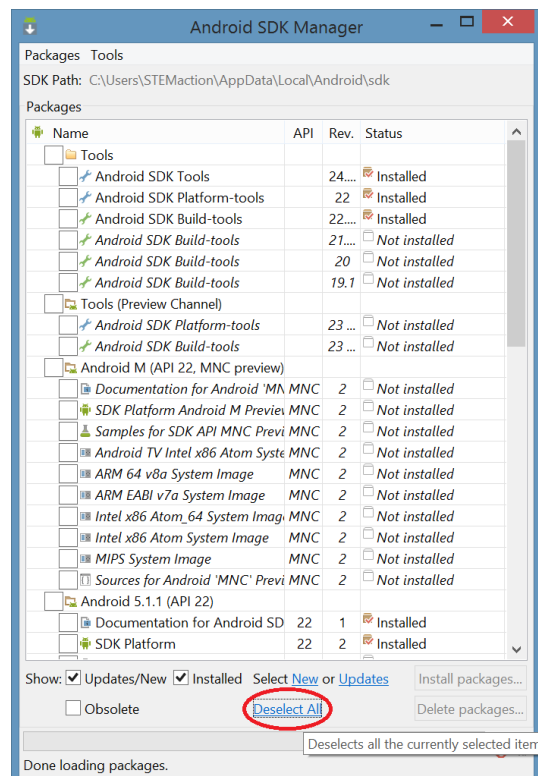
Step 23: If the ftc_app-master project isn't shown, click on the triangle next to Android (near the top left portion) and select Project from the drop down list. The image below is sometimes how the project opens (i.e. without the SDK showing).



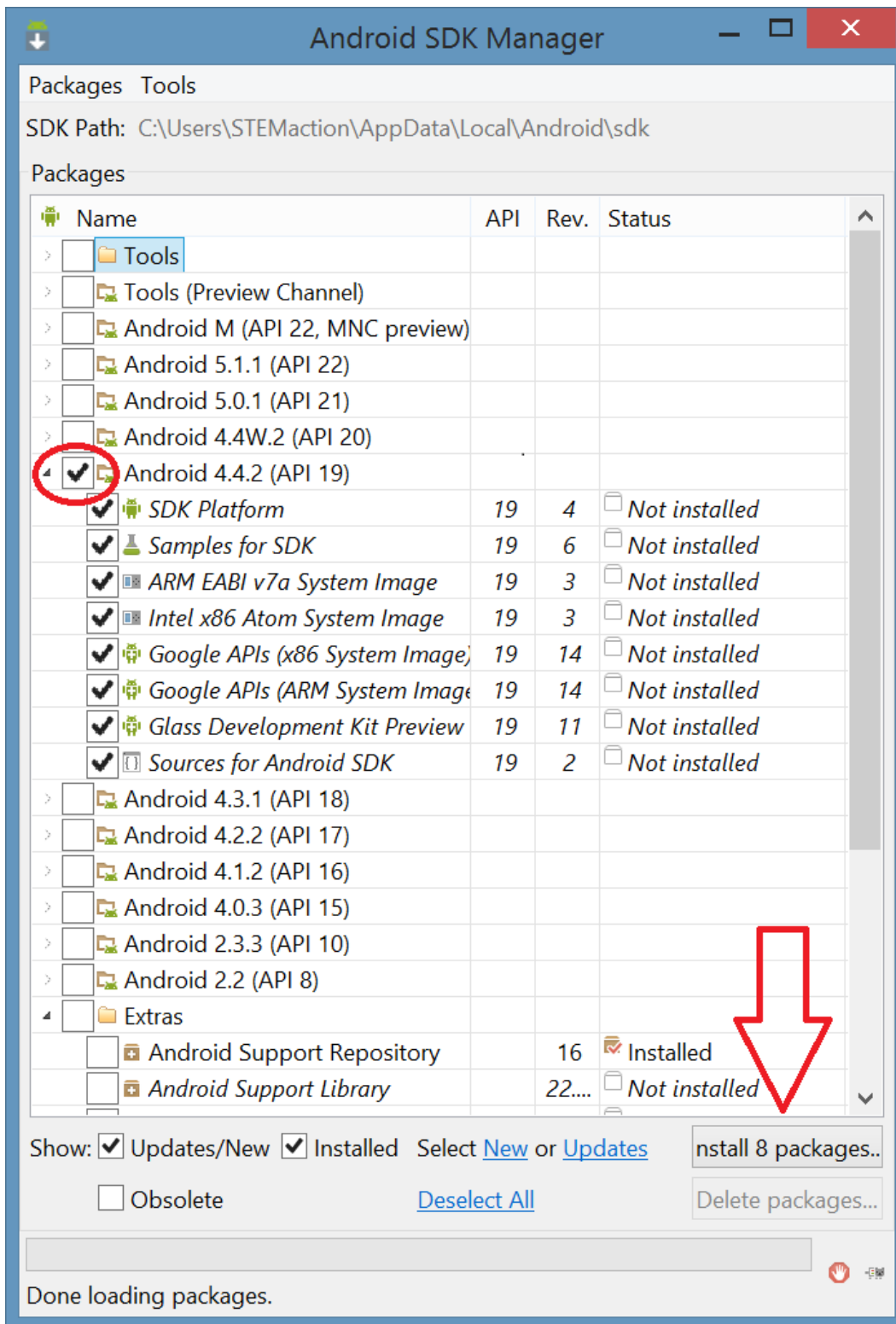
Step 24: If an error message is displayed at the bottom of the window, then install the Android SDK Manager by selecting the link near the bottom of the page. If there is no message that indicates “failed to find Build Tools...”, then the installation is complete and proceed to the next section “Installing the Push Bot Op modes”.



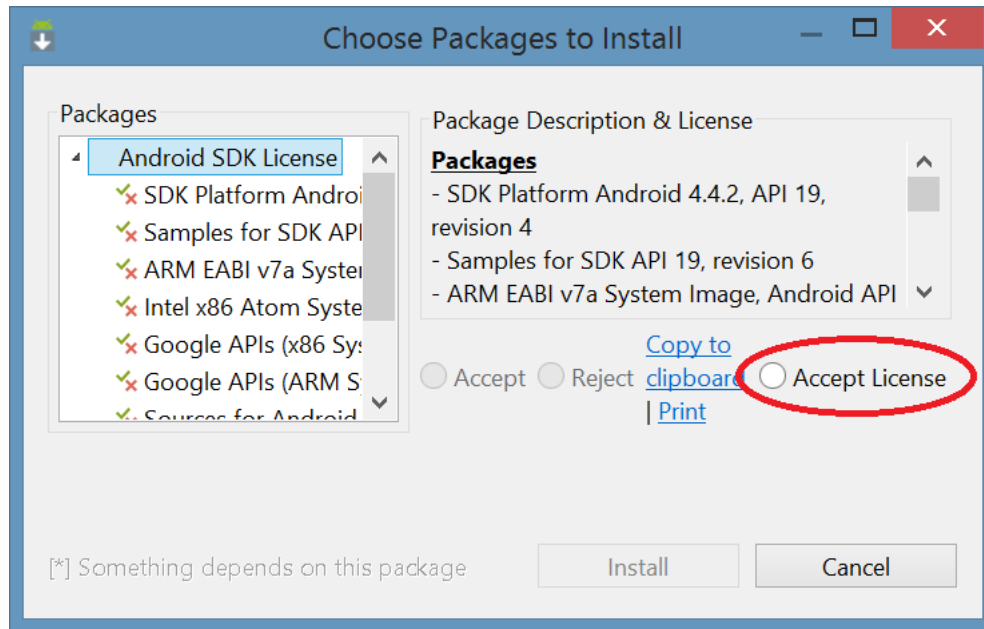
Step 25: Deselect all components using the ‘Deselect All’ button near the bottom of the window.



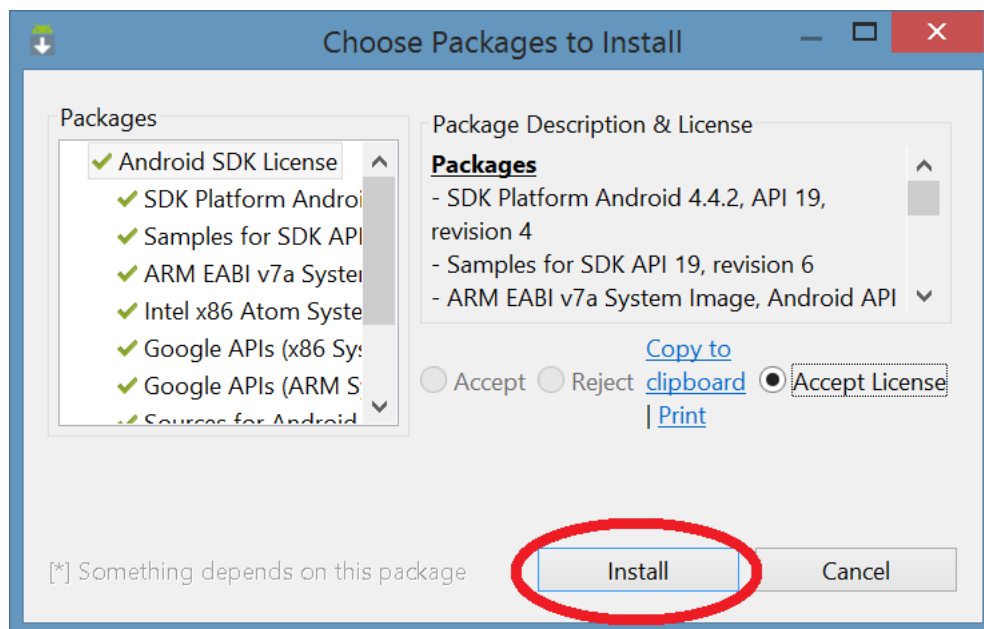
Step 26: Select the box (red circle) next to Android 4.4.2 (API 19). All components beneath that element will become checked. Install the components (red arrow).



Step 27: Accept the License Agreements.



Step 28: 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).



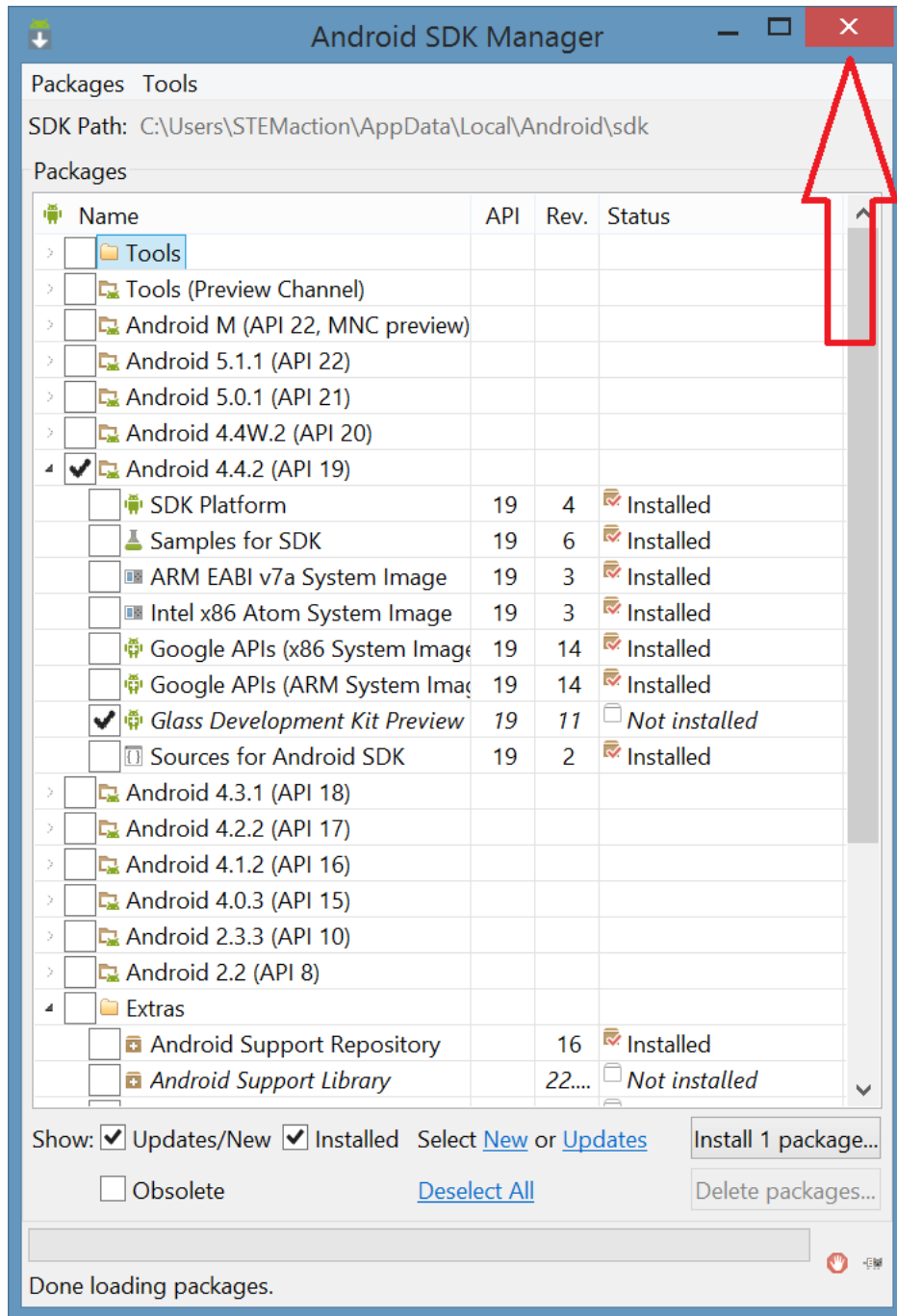
The screenshot shows the Android SDK Manager interface. At the top, the title bar reads "Android SDK Manager". Below it, there are tabs for "Packages" and "Tools". The "SDK Path" is set to "C:\Users\STEMaction\AppData\Local\Android\sdk".

The main area displays a list of packages with columns for Name, API, Rev., and Status. The following table represents the data shown in the screenshot:

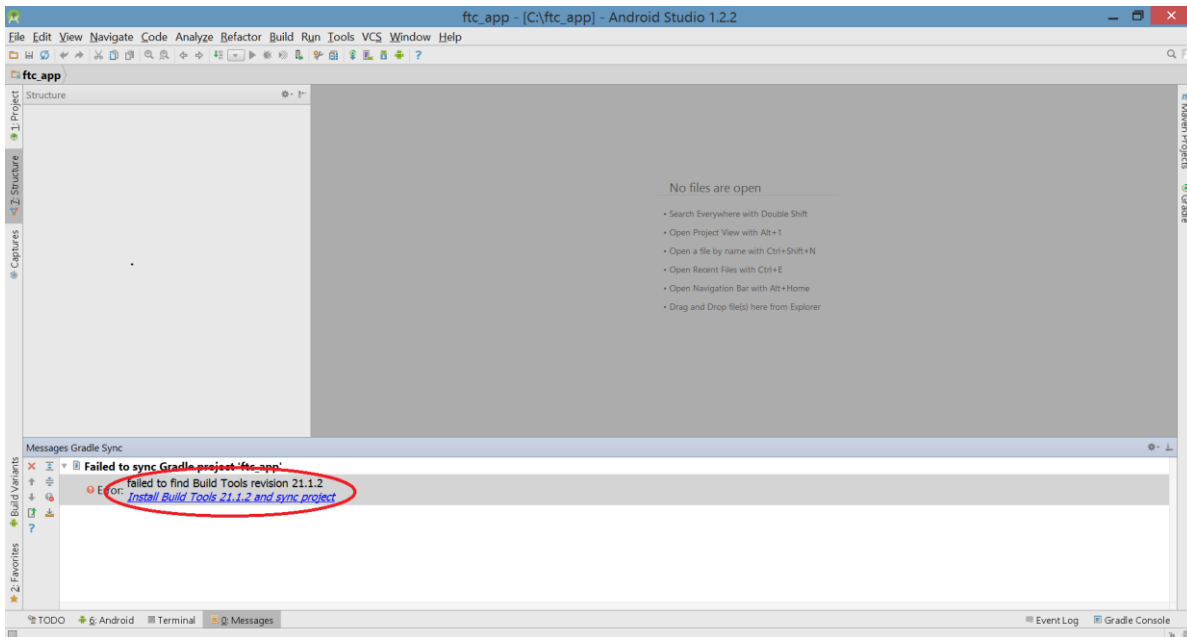
Name	API	Rev.	Status
<input type="checkbox"/> Tools			
<input type="checkbox"/> Tools (Preview Channel)			
<input type="checkbox"/> Android M (API 22, MNC preview)			
<input type="checkbox"/> Android 5.1.1 (API 22)			
<input type="checkbox"/> Android 5.0.1 (API 21)			
<input type="checkbox"/> Android 4.4W.2 (API 20)			
<input checked="" type="checkbox"/> Android 4.4.2 (API 19)			
<input checked="" type="checkbox"/> SDK Platform	19	4	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Samples for SDK	19	6	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> ARM EABI v7a System Image	19	3	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Intel x86 Atom System Image	19	3	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Google APIs (x86 System Image)	19	14	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Google APIs (ARM System Image)	19	14	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Glass Development Kit Preview	19	11	<input type="checkbox"/> Not installed
<input checked="" type="checkbox"/> Sources for Android SDK	19	2	<input type="checkbox"/> Not installed
<input type="checkbox"/> Android 4.3.1 (API 18)			
<input type="checkbox"/> Android 4.2.2 (API 17)			
<input type="checkbox"/> Android 4.1.2 (API 16)			
<input type="checkbox"/> Android 4.0.3 (API 15)			
<input type="checkbox"/> Android 2.3.3 (API 10)			
<input type="checkbox"/> Android 2.2 (API 8)			
<input type="checkbox"/> Extras			
<input type="checkbox"/> Android Support Repository		16	<input checked="" type="checkbox"/> Installed
<input type="checkbox"/> Android Support Library		22...	<input type="checkbox"/> Not installed

At the bottom of the window, there are controls for showing updates and installed packages, and buttons for installing and deleting packages. A progress bar at the very bottom shows the download progress for the "SDK Platform" package, with the text: "Downloading SDK Platform Android 4.4.2, API 19, revision 4 (12%, 2482 KiB/".

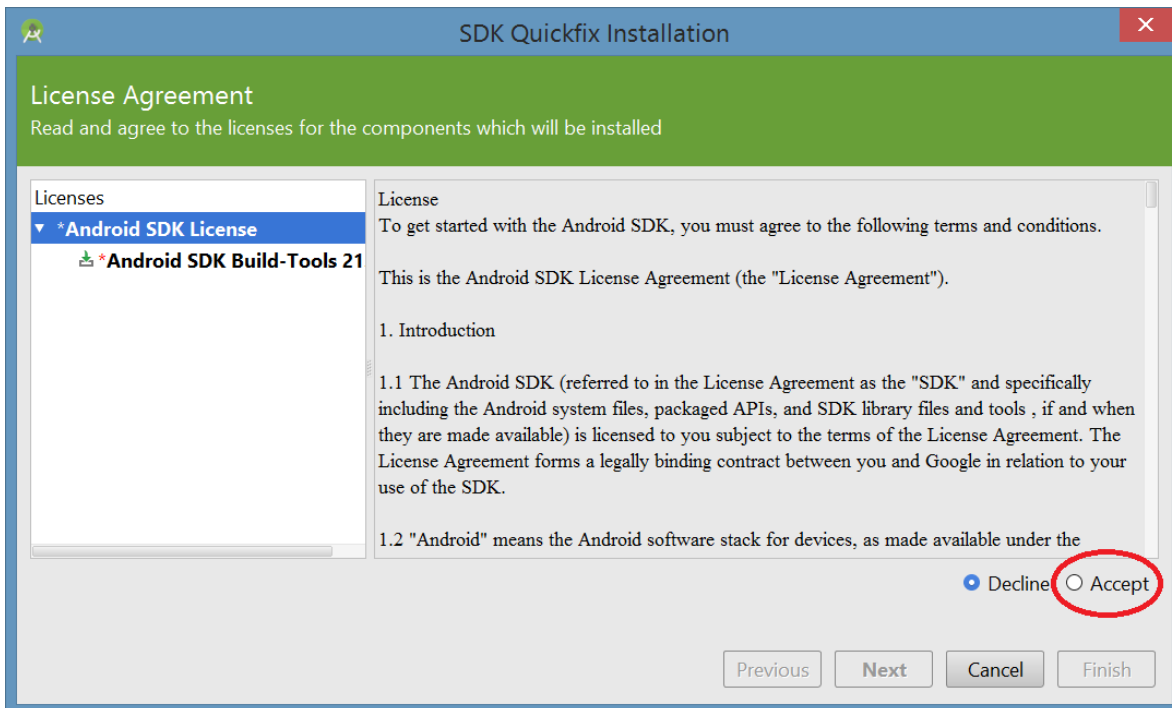
Step 29: After the download has completed, close the dialog.



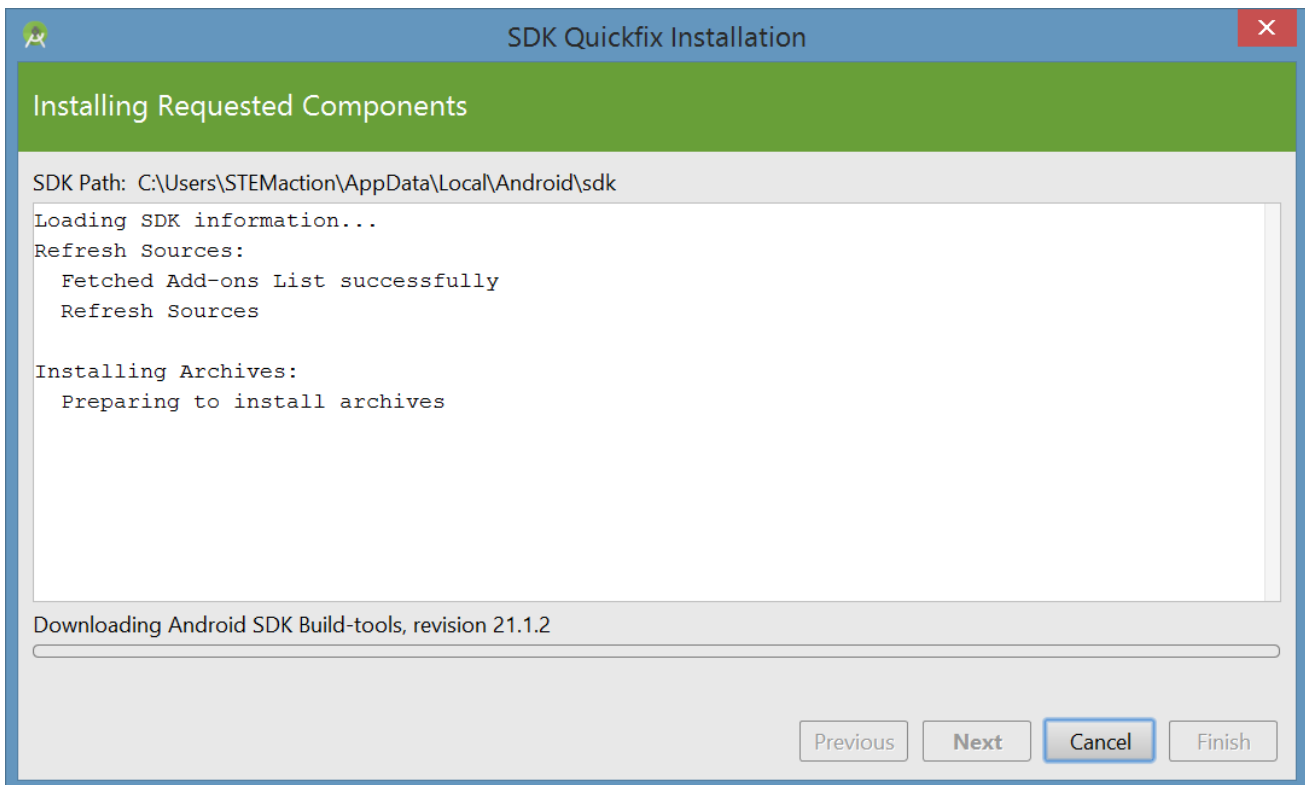
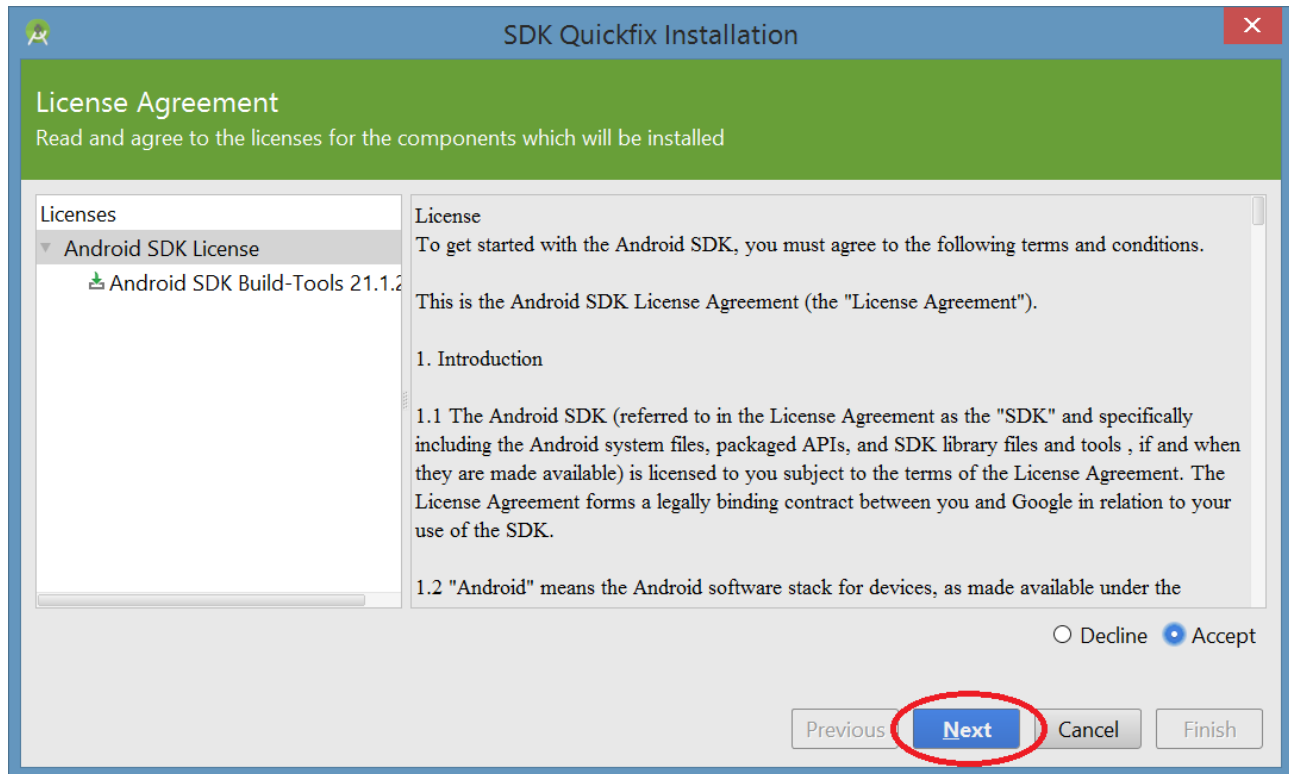
Step 30: If Android Studio doesn't automatically attempt a rebuild, close Android Studio and restart it. If a different error message than the previous error message, select the blue link again to install Build Tools 21.



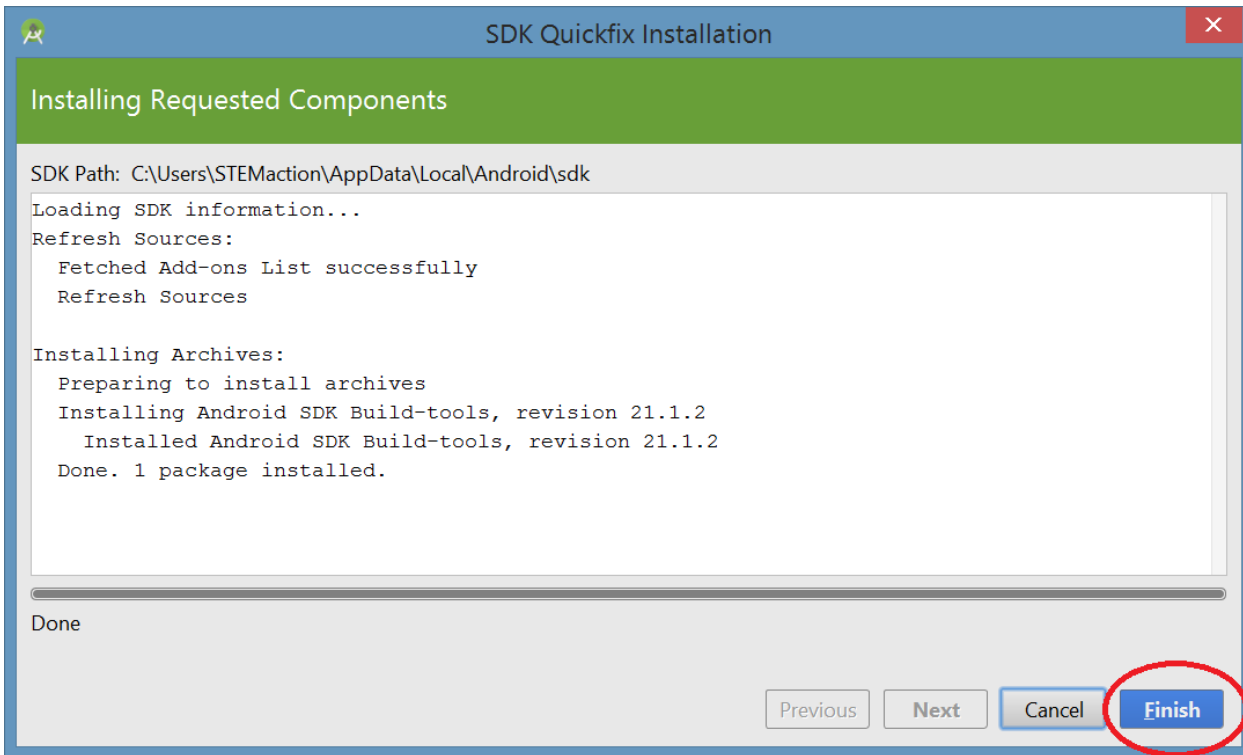
Step 31: Accept the quick fix license.



Step 32: Click 'Next' to install the components. The progress of the installation can be seen in the next dialog (shown below the following image).



Step 33: Return to Android Studio by selecting 'Finish'.

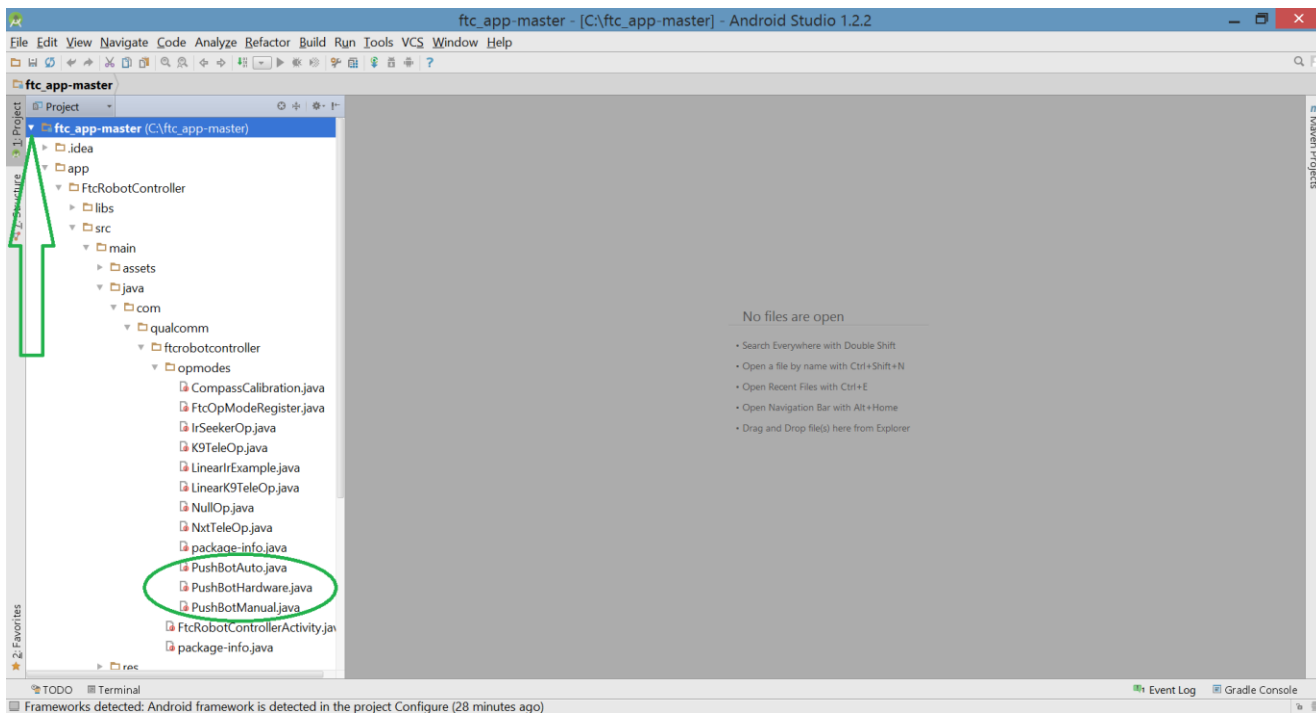


Step 34: Installation should be complete at this point. If an error message continues to be displayed, refer to the Android Studio website or USFIRST forum for more information.

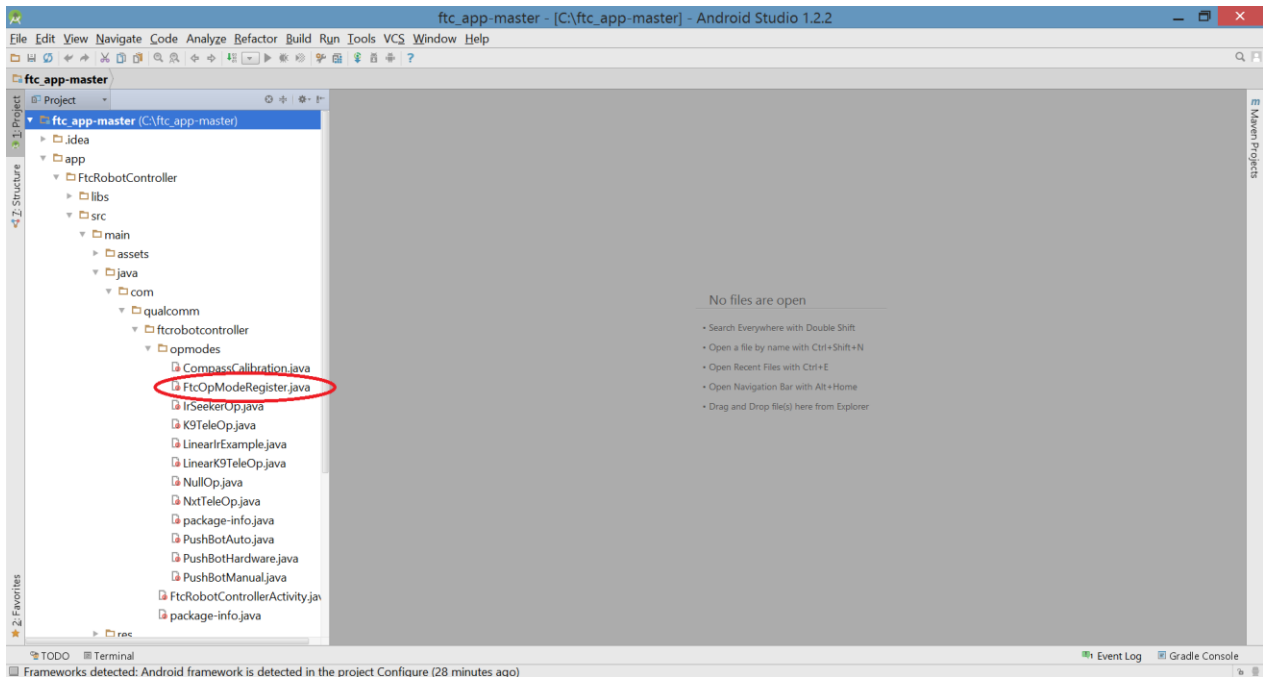
Registering the PushBot Op modes

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.

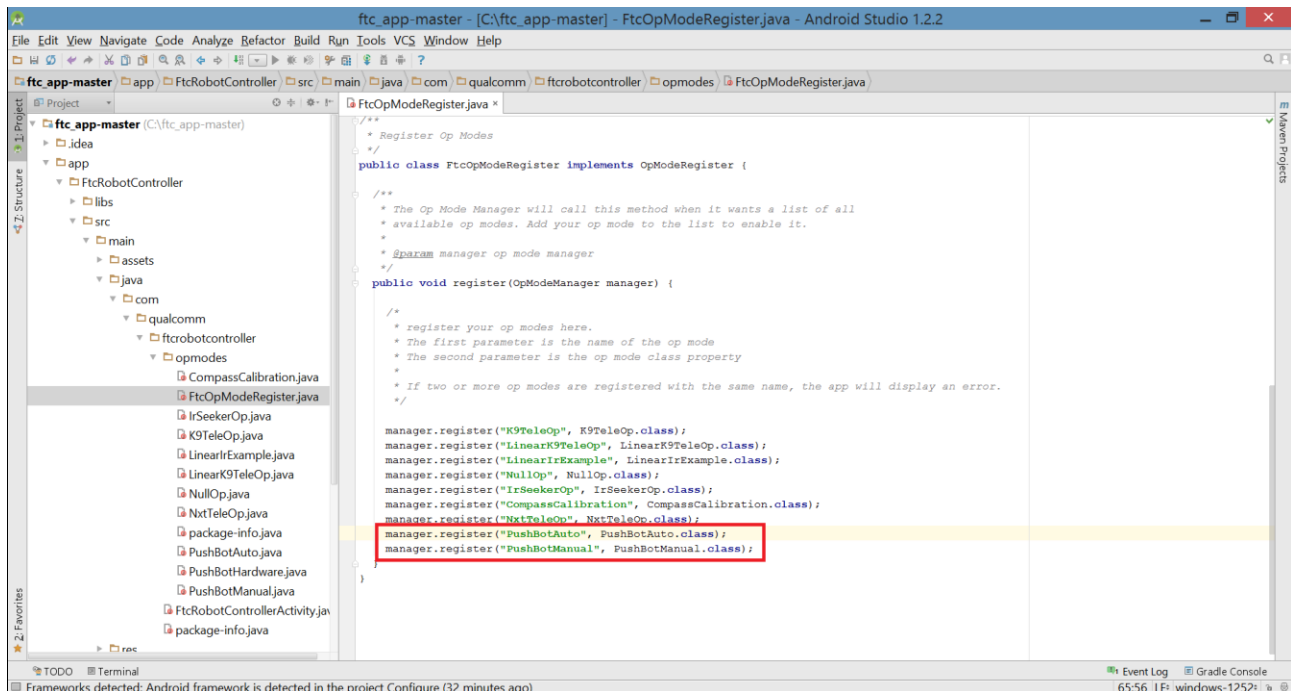
Step 1: If Android Studio isn't still open, open it now. Open the FtcRobotController tree using the small arrow to its left. Continue opening the tree until the Op modes item is shown (i.e. open FtcRobotController, then Java, then com.qualcomm.ftcrobotcontroller, then Op modes). Verify that the PushBot Op modes are in the tree. Note that there are three entries in the tree that begin with 'PushBot': PushBotAuto, PushBotHardware and PushBotManual (green circle).



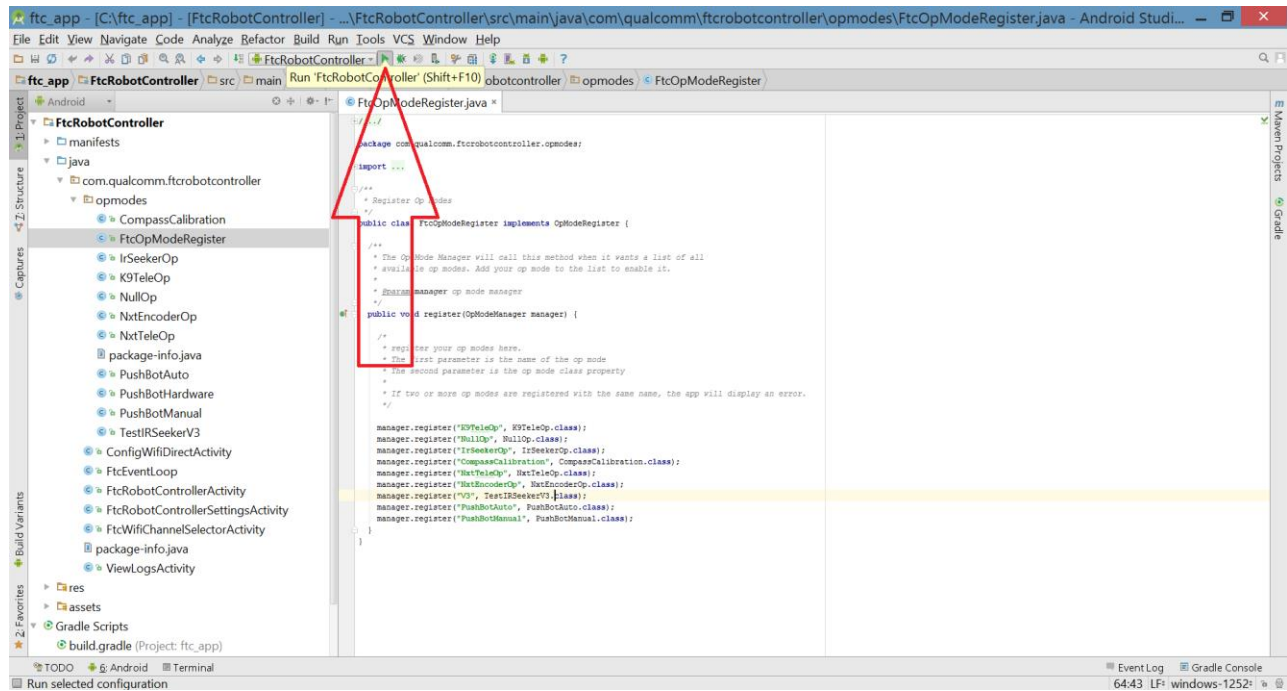
Step 2: Open the FtcOpModeRegister file (red oval) by double clicking it.



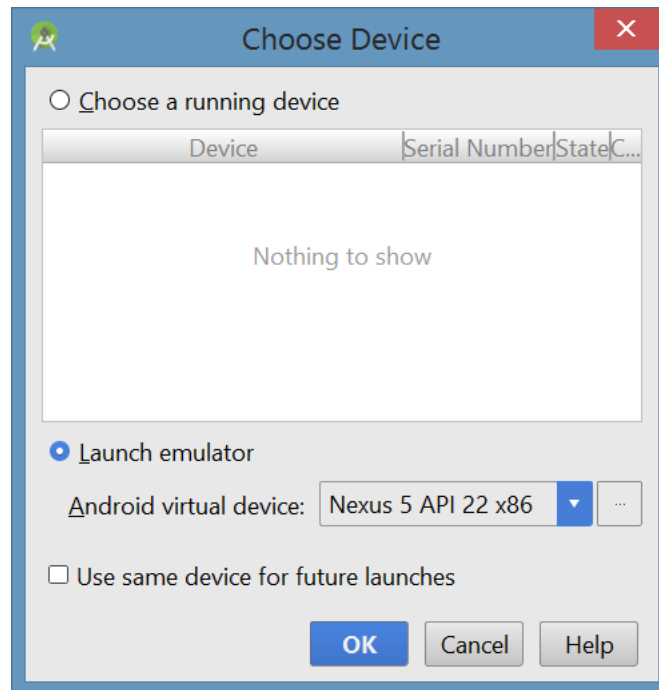
Step 3: Scroll to the bottom of the file until the 'manager.register' lines are visible; they will have green (strings) and blue (class names) text. Using the existing entries as a guide, type “**manager.register (“PushBotAuto”, PushBotAuto.class);**” and “**manager.register (“PushBotManual”, PushBotManual.class);**”. Do **NOT** create a line for the PushBotHardware class. The new lines are shown in the red rectangle.



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



Preparing the Robot Controller and Driver Station cell phones

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 activate 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.

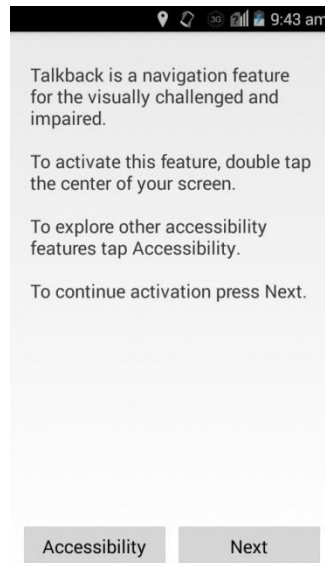


Step 10: Turn phones on. The circle is where the power switch is located. The phones will begin booting. The sequence of images shows the booting sequence.

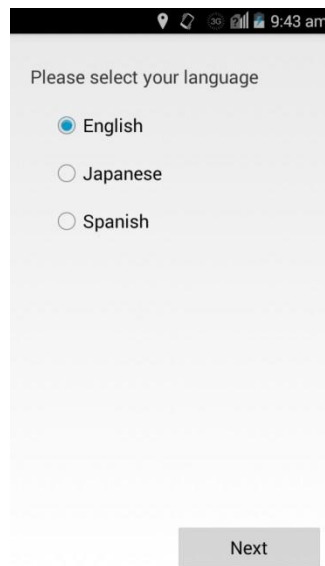




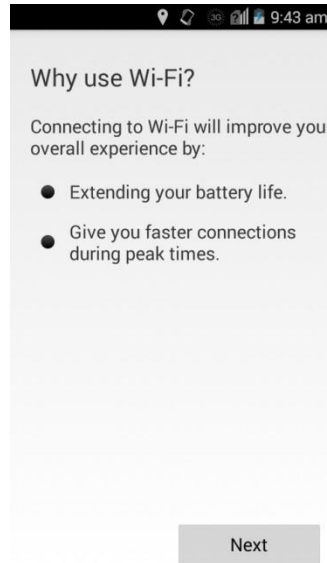
Step 11: The above images show instructions for both phones. This and all following steps may only show one image, but apply the step to both phones. Choose 'Next' to accept the default talkback setting.



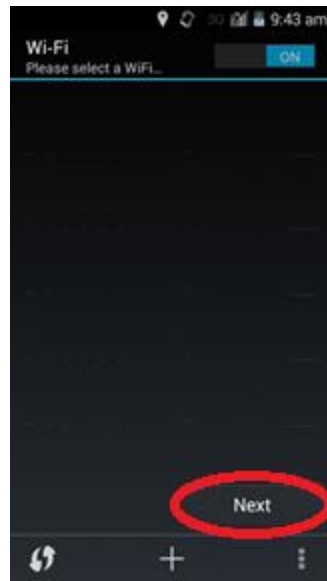
Step 12: Select 'English'.



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



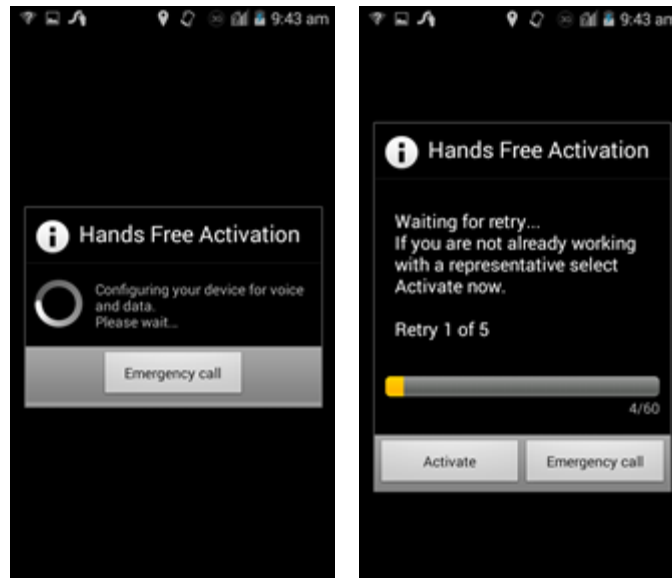
Step 14: Do not connect to Wi-Fi at this time. Select 'Next'.



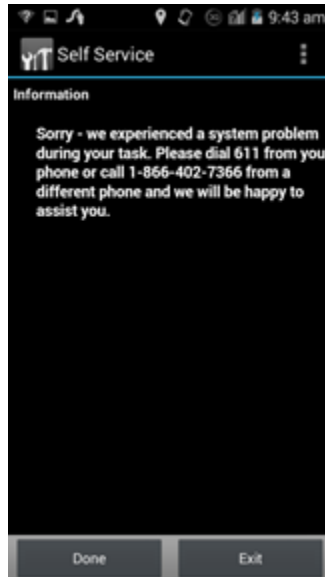
Step 15: When the UICC Error screen is shown, select 'OK'.



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.



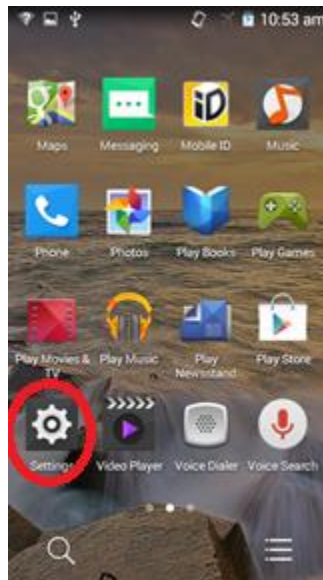
Step 17: The activation should fail. Select 'Done'.



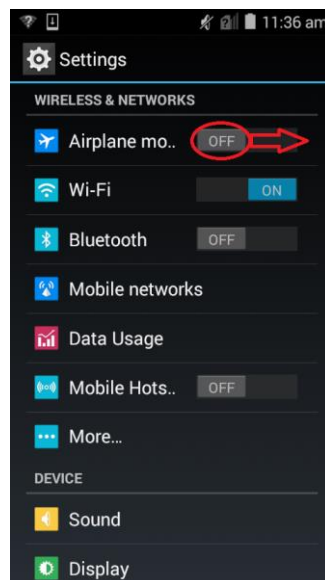
Step 18: Android will return to the home screen. Open the application activity by selecting its icon (shown in the red circle).



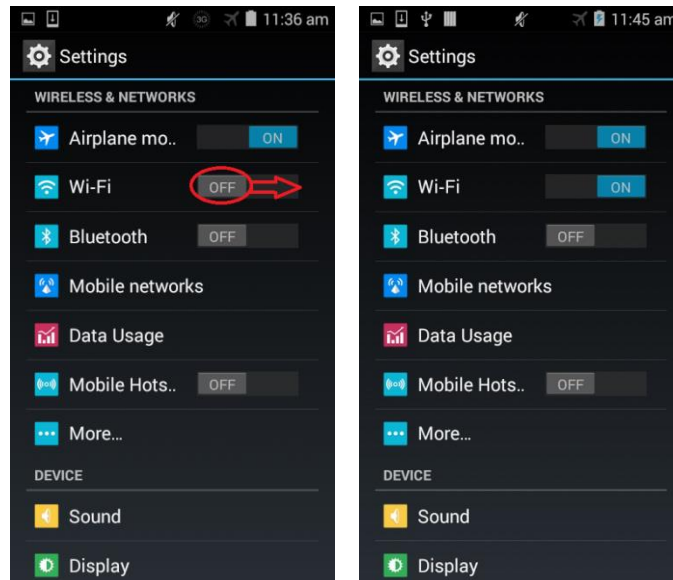
Step 19: Open the settings activity by selecting its icon (shown in the red circle). It might be necessary to swipe the screen left or right.



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.



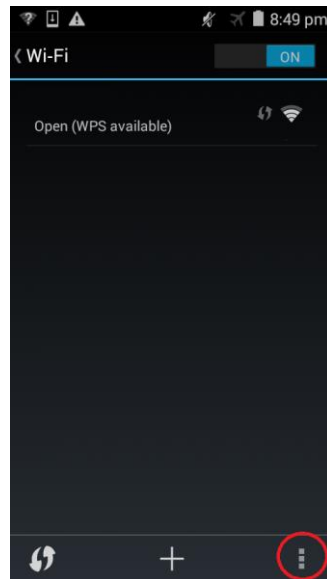
Naming the Cell Phone

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".

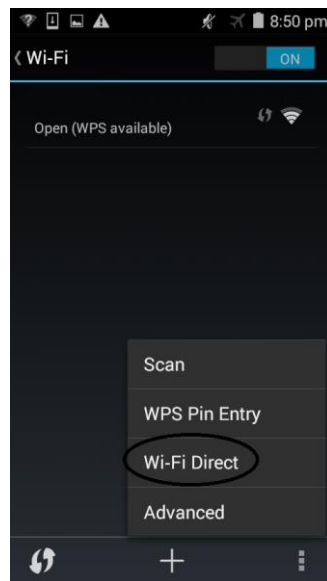
Step 1: (Continuing from the previous section – the settings page is displayed) Select Wi-Fi.



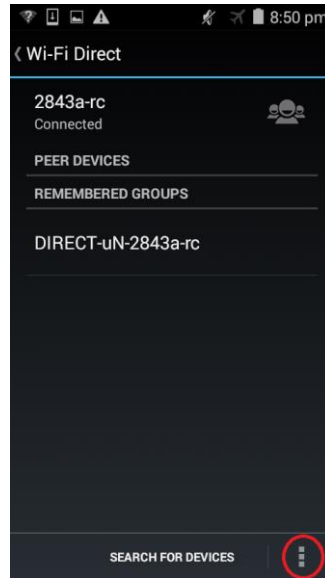
Step 2: Select the overflow settings (three dots on the bottom right).



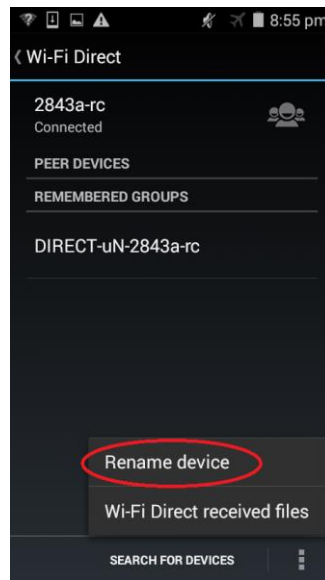
Step 3: Select Wi-Fi Direct



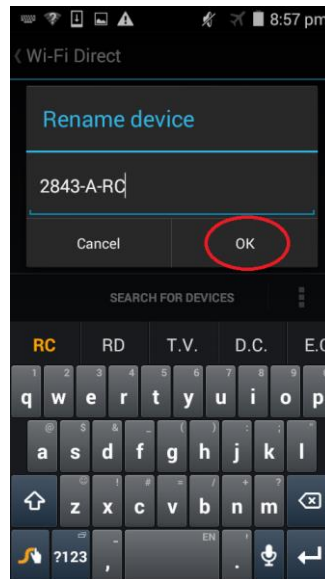
Step 4: If “RENAME DEVICE” is not shown at the bottom of the screen, then select the overflow settings and then select “Rename device”. In the image below, the phone had already been named. New phones will have a generic name.



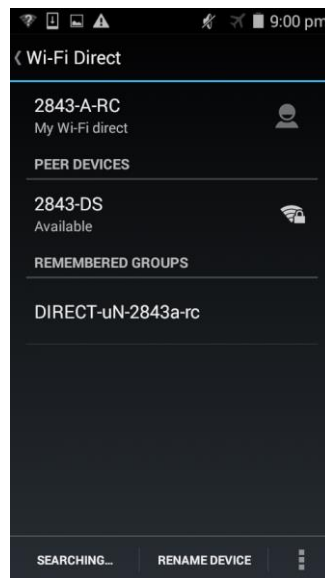
Step 5: Enter the new name according to the rules in Game Manual Part I.



Step 6: Select OK.



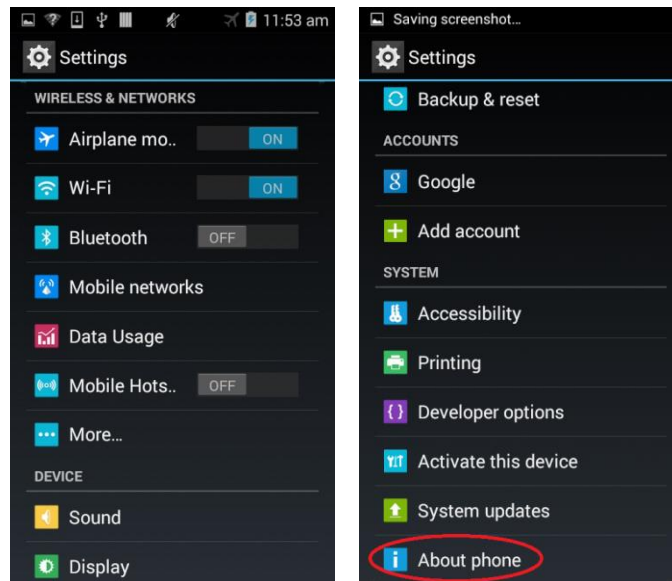
Step 7: From the page shown below, return to the settings activity by pressing the back button. It may take several presses to return to the settings activity.



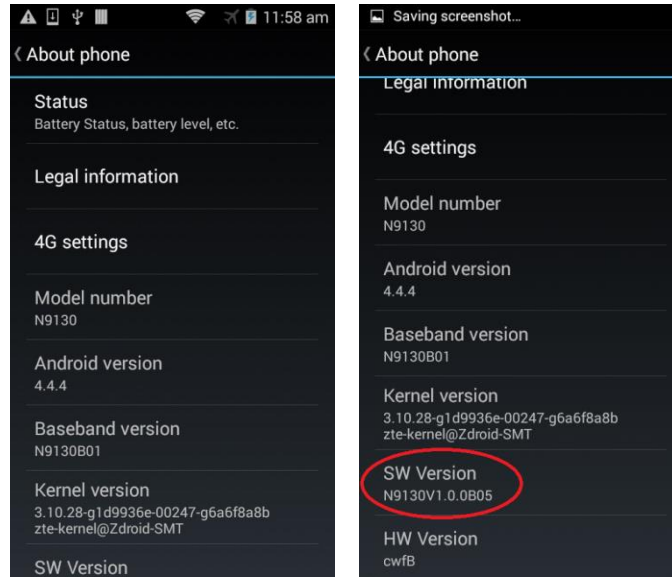
Enabling Debugging Mode on the Robot Controller Cell Phone

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 can't deploy the Robot Controller app until debugging is enabled on the phone.

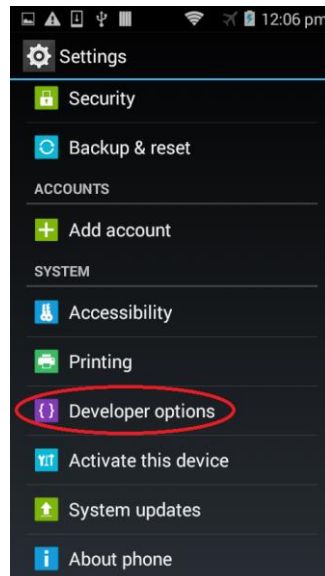
Step 1: From the previous section, the Settings activity should be shown (left image). Scroll downward until 'About phone.' is visible (right image) – it should be at the very bottom. Select 'About phone'.



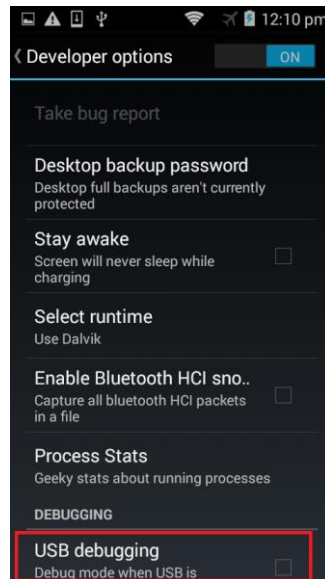
Step 2: The About phone activity (left screen) allows a developer to enable debugging. Scroll downward until 'SW Version' is shown (right image) - on some Android devices the text says 'Build number'. Tap 'SW Version' 7 times. A toast (a message that is displayed for a short amount of time) should display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



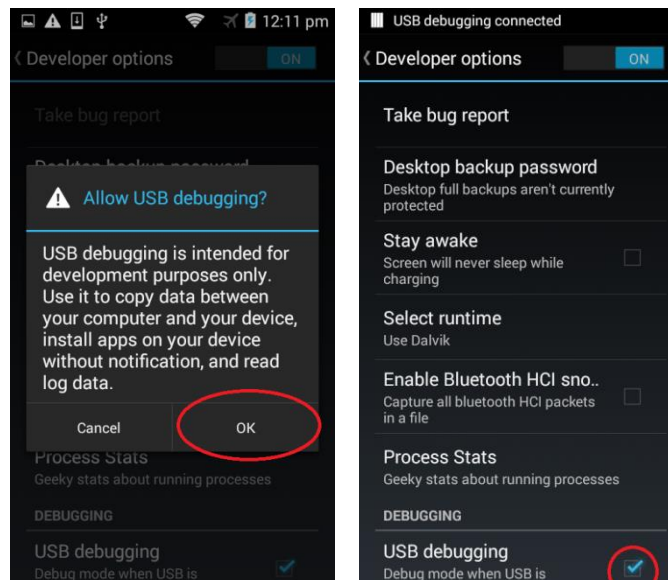
Step 3: If Android hasn't already switched to the Settings activity, then do so now. Scroll to the bottom of the page and select 'Developer options'. The developer options button is invisible until debugging is enabled (see previous step).



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 has a check in it (right image).

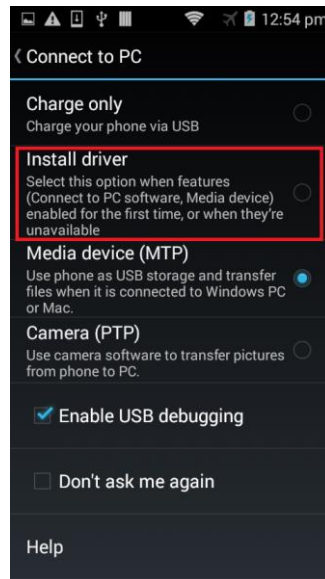


Installing the USB Device Driver

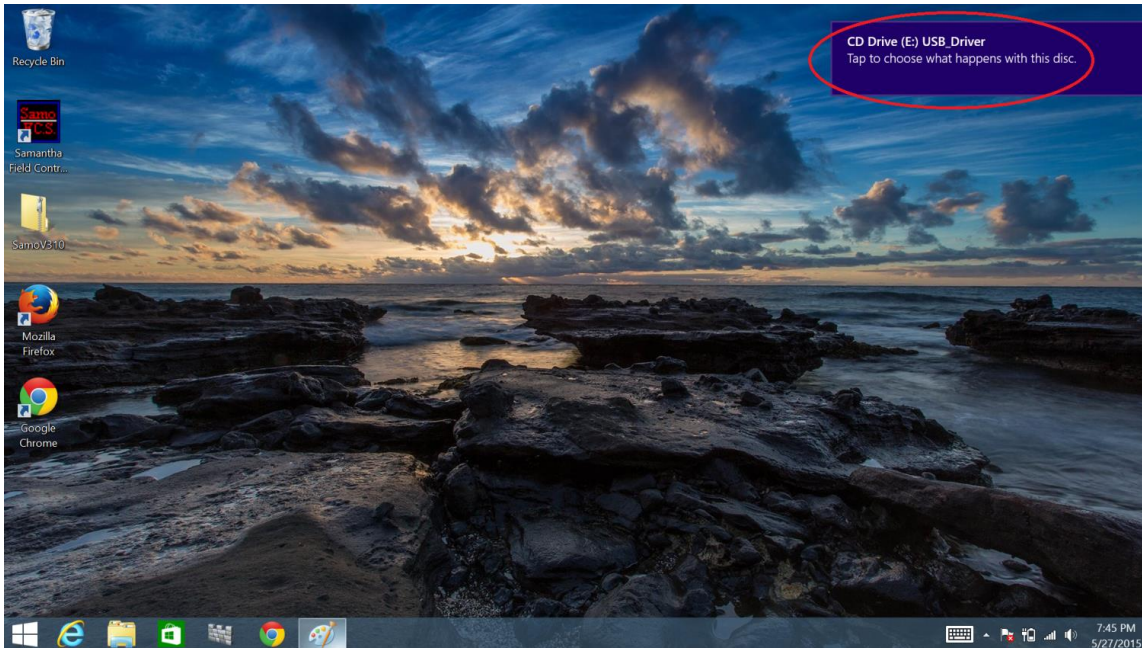
This section applies only to the Robot Controller cell phone. The correct USB driver will need to be installed onto the Windows computer hosting Android Studio. Without the correct driver, Android Studio can't connect to the cell phone when it is connected via a USB cable. Also the computer will treat 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.

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.

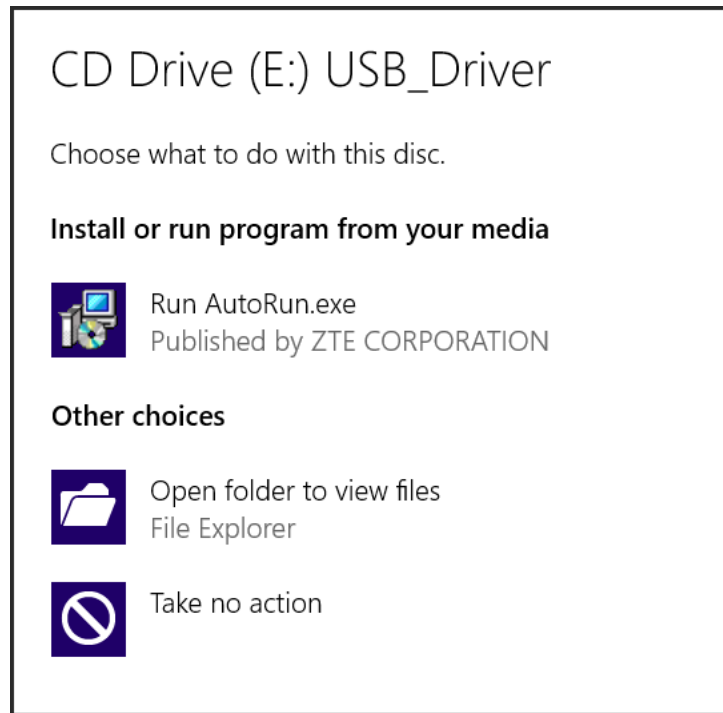
Step 2: Select 'Install driver' on phone.



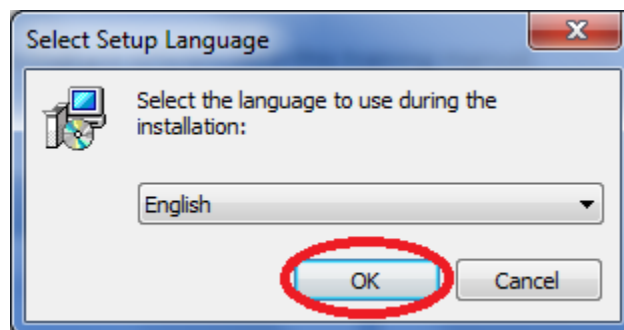
Step 3: When 'Install driver' is selected on the phone the computer will detect the connection. Confirm the computer's prompt.



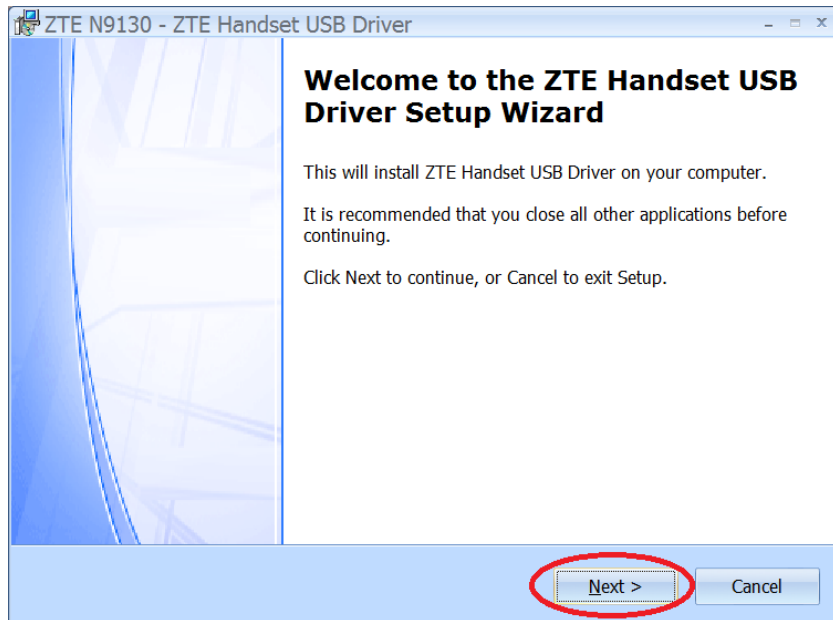
Step 4: After confirming the prompt, another prompt will be displayed. Select 'Run AuroRun.exe'. If the screen darkens and a dialog asks "Do you want to allow the following program to make changes to this computer?", select 'Yes'.



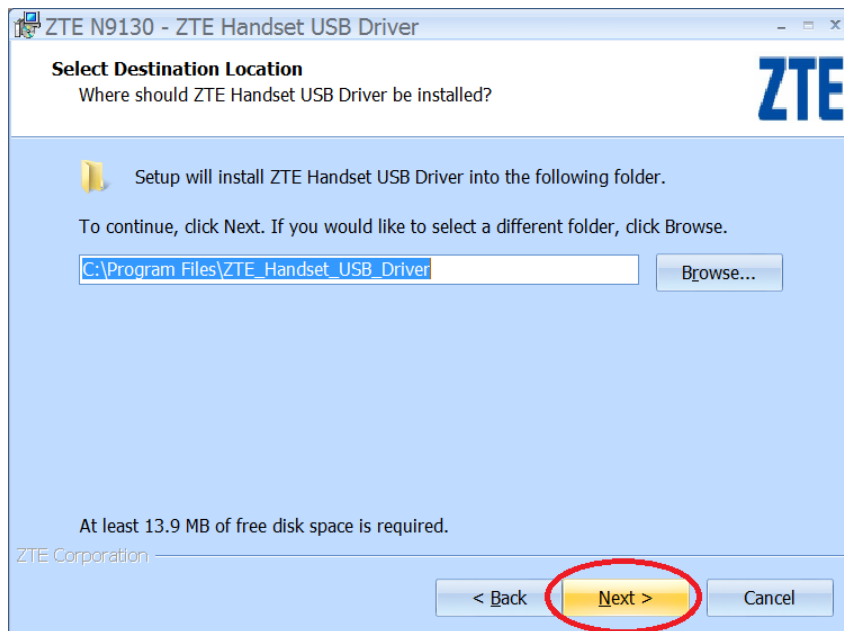
Step 5: The Select Setup Language dialog should be displayed. Select 'OK'.



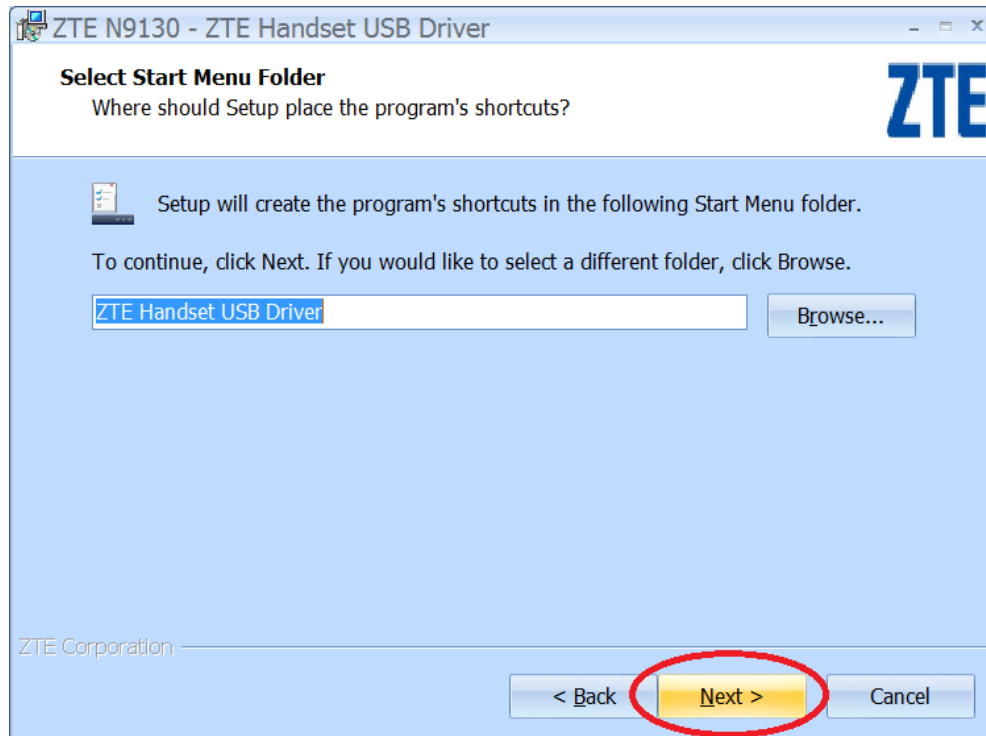
Step 6: Select 'Next' to continue with the installation.



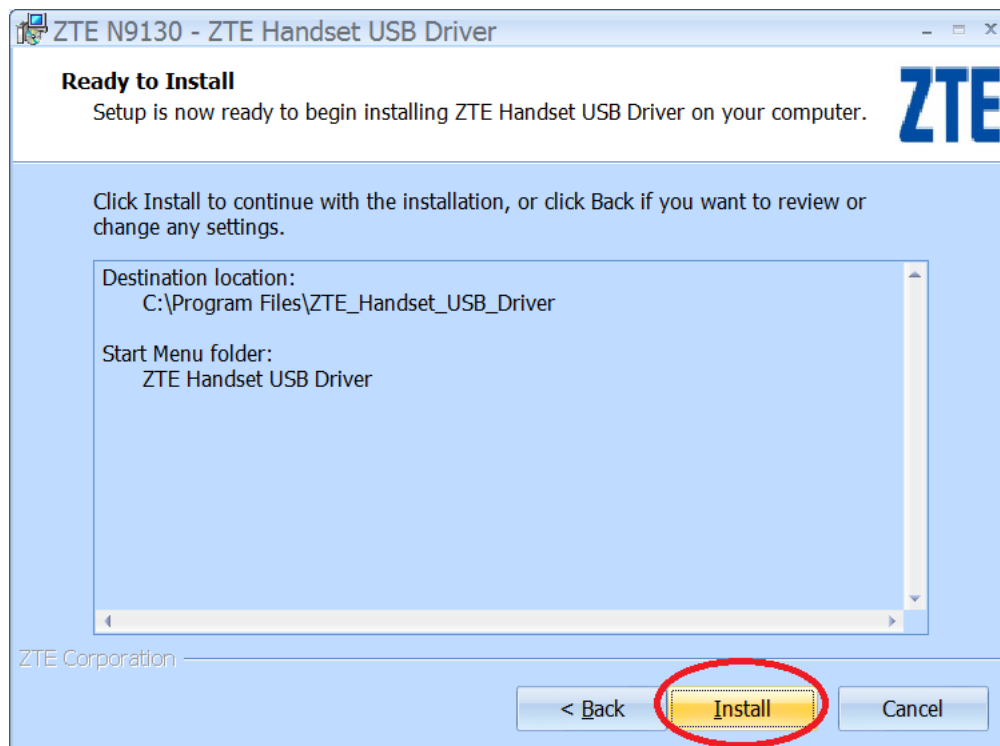
Step 7: Select 'Next' to accept the default location for the driver.

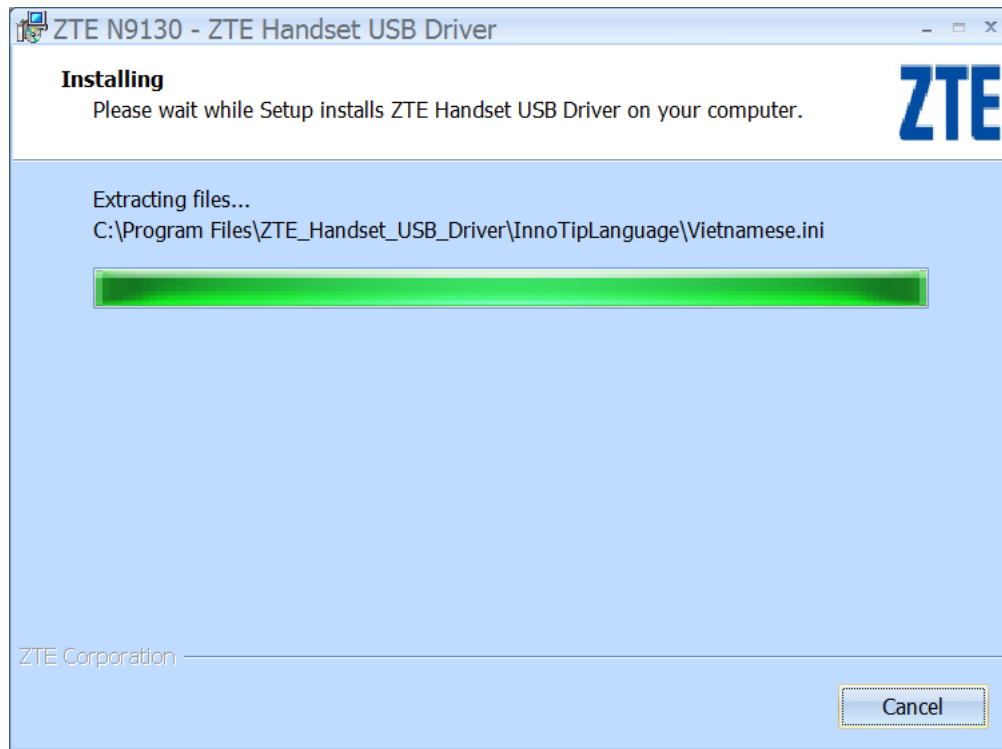


Step 8: Select 'Next' to accept the default location of the driver shortcuts.

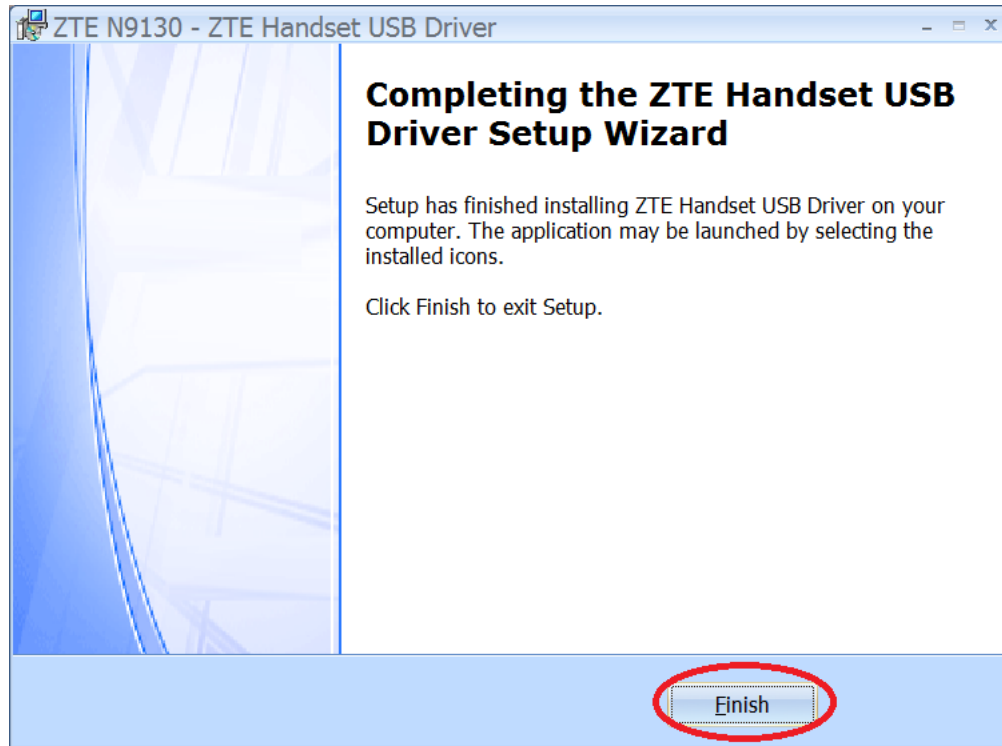


Step 9: Select 'Install'. The image below the following shows status as the driver is being installed.

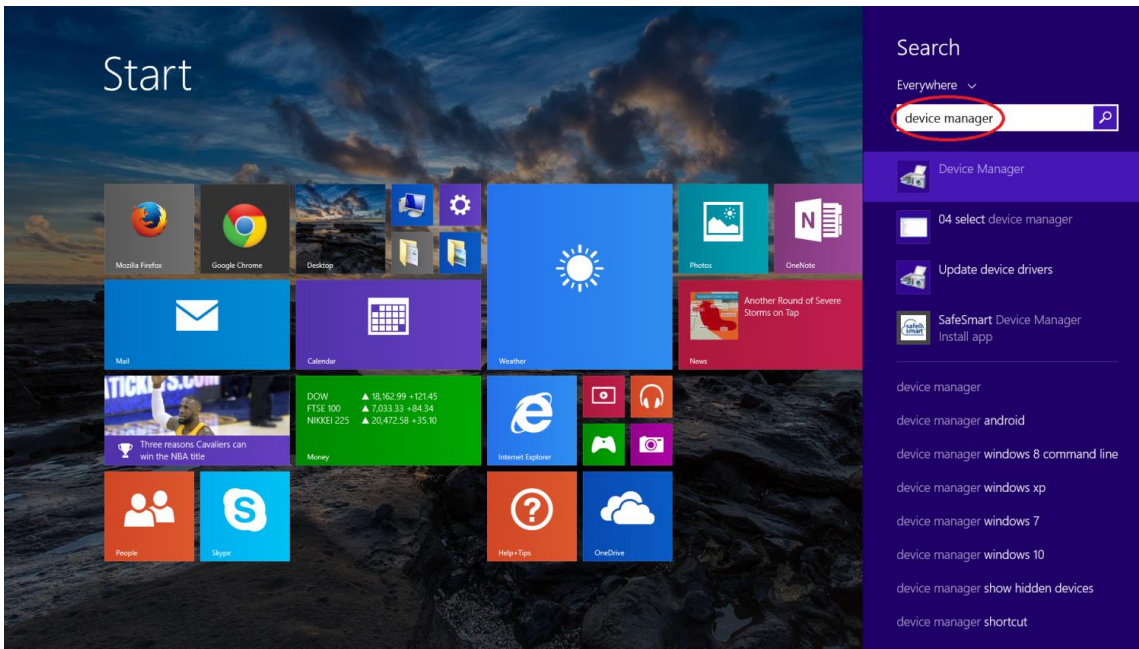




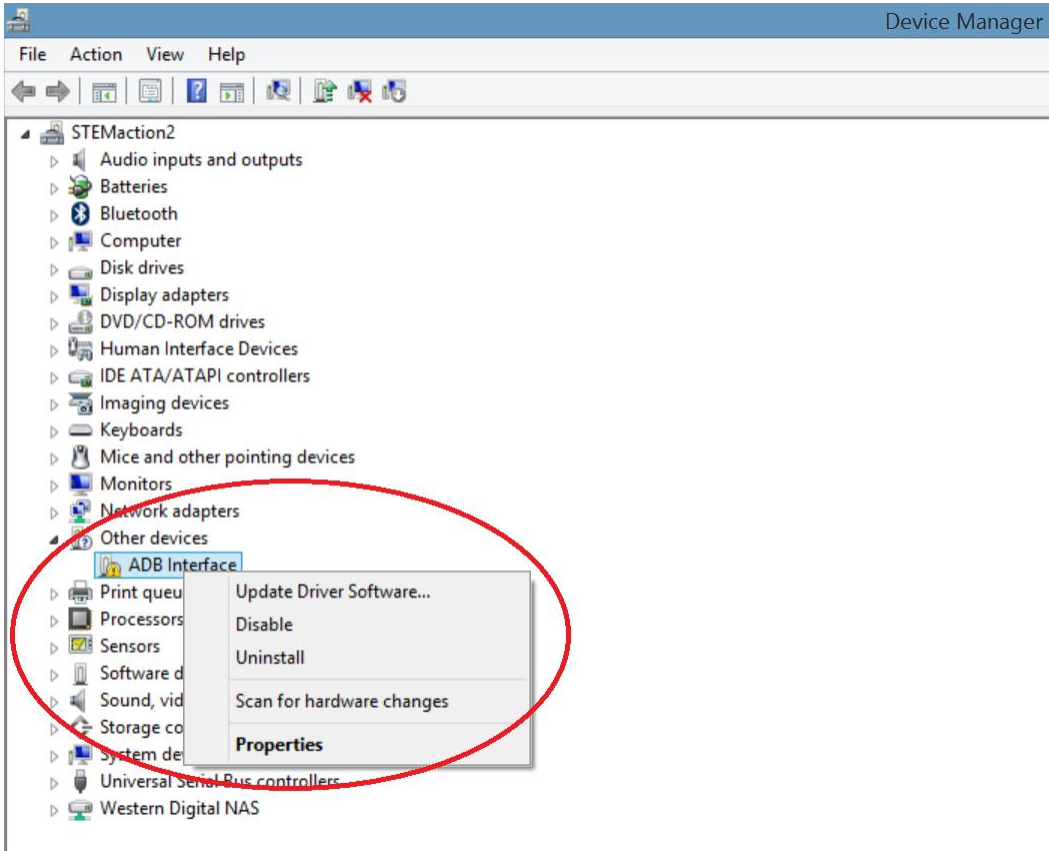
Step 10: Select 'Finish'.



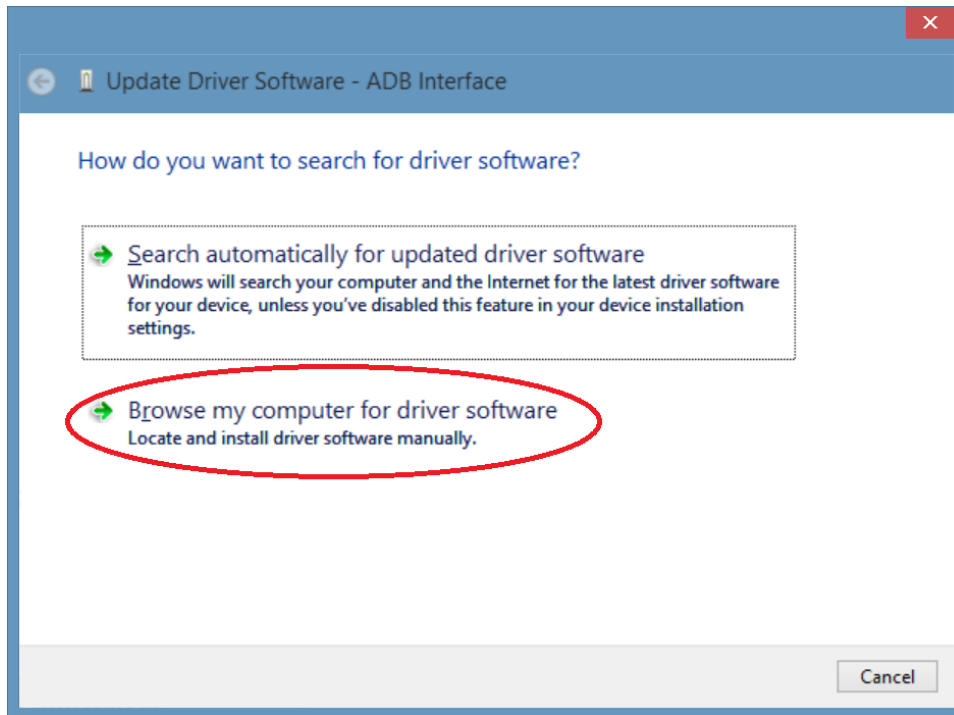
Step 11: Press the Window key and type 'device manager' and press return (similar to opening Internet Explorer).



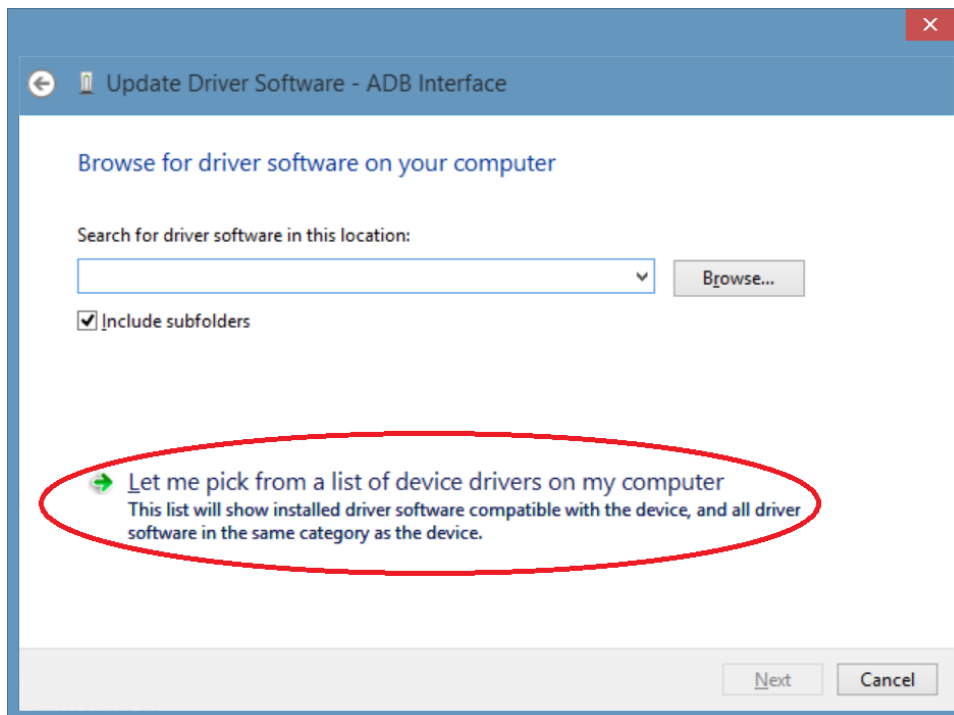
Step 12: Right click on ADB Interface and select Update Driver Software.



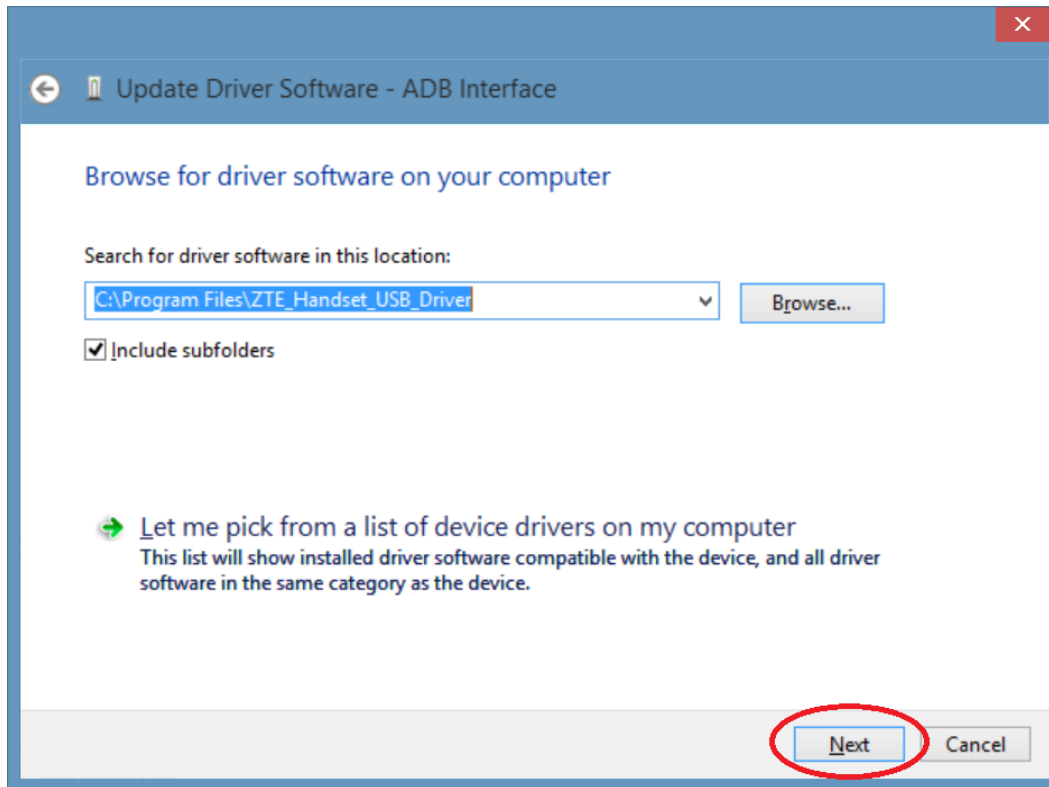
Step 13: Select 'Browse my computer for driver software'.



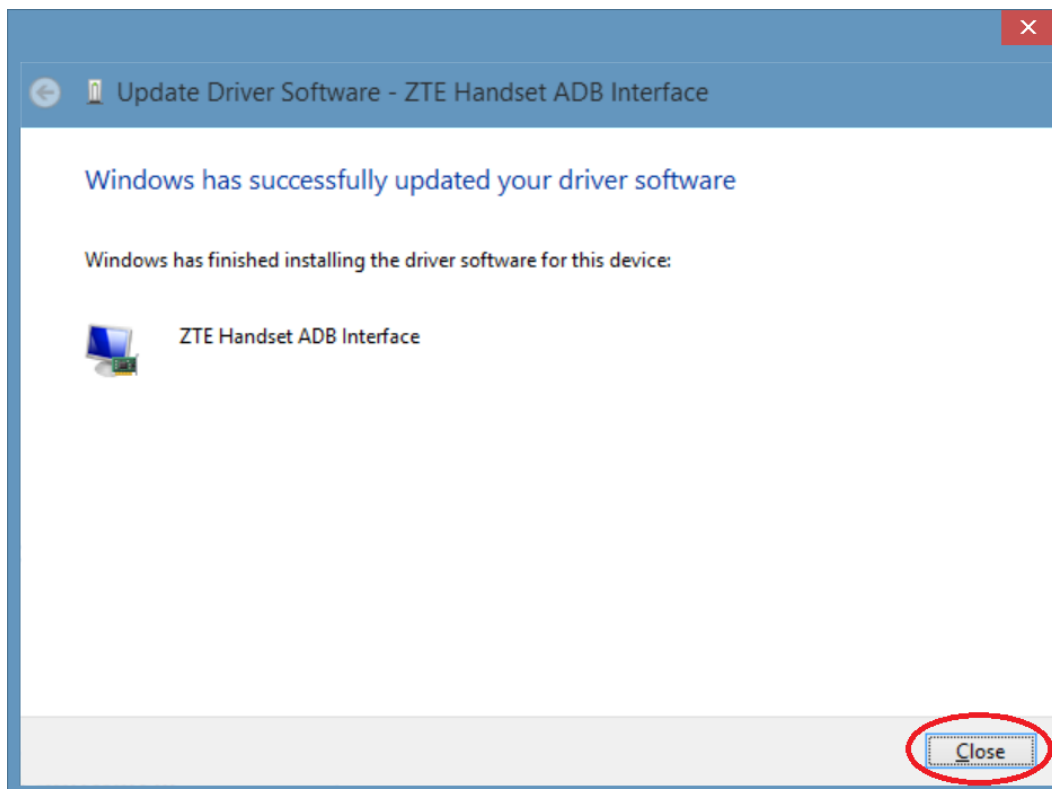
Step 14: Select 'Let me pick from a list of device drivers on my computer'.



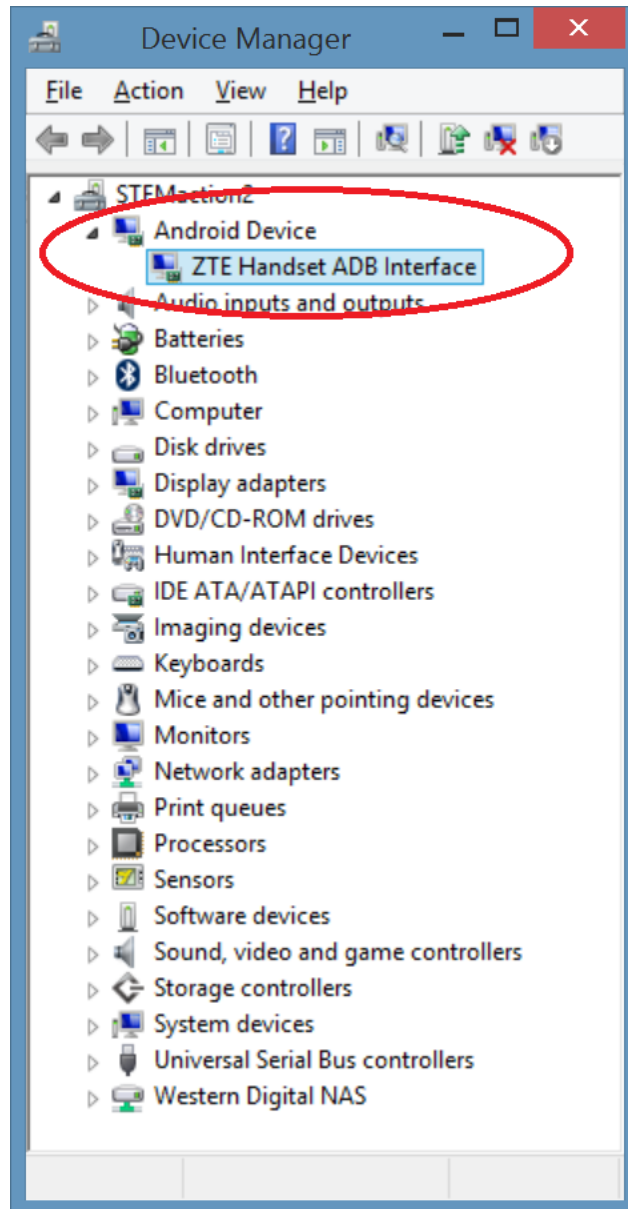
Step 15: Select 'Next'.



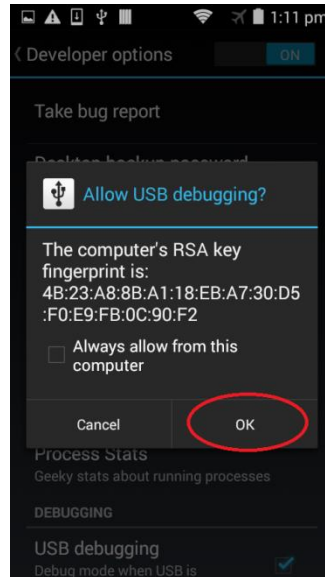
Step 16: Select 'Close'.



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'.

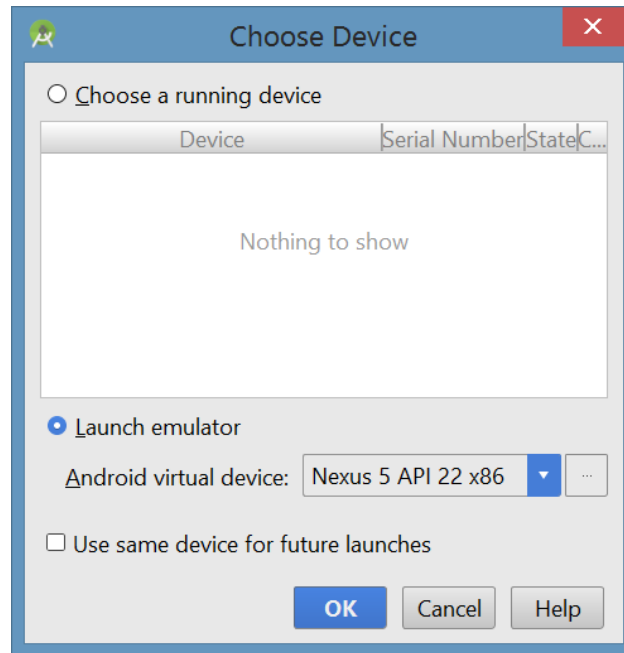


Deploying the Robot Controller

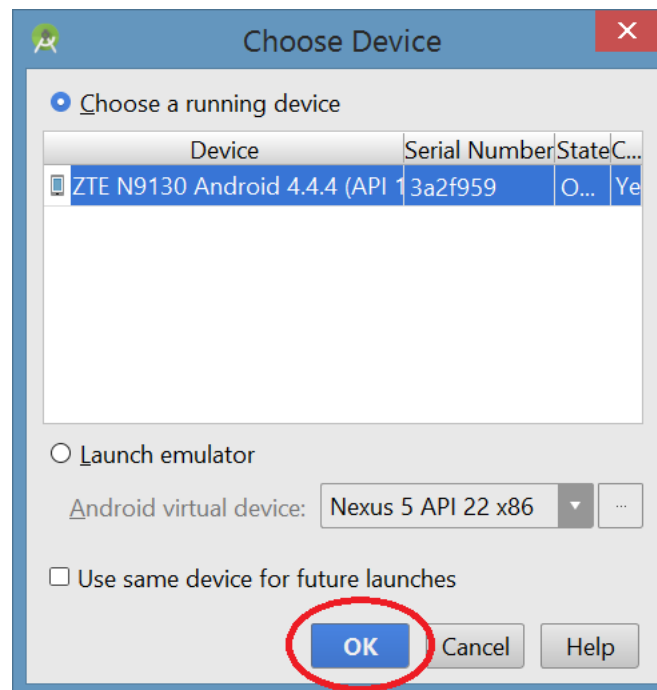
For this section choose one phone to be the Driver Station and the other to be the Robot Controller. The Robot Controller will host the Robot Controller app, which implements the PushBot Op modes. Android Studio can't deploy the Robot Controller app until debugging is enabled on the phone, so that will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

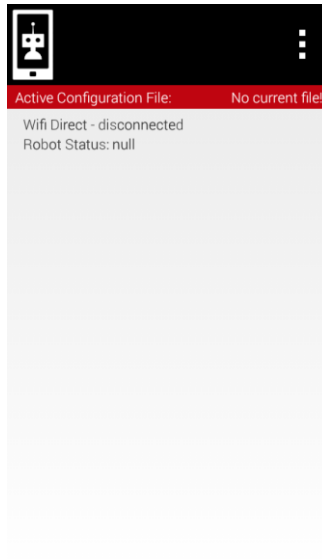
Step 1: From a previous section, Android Studio may still be open and displaying the Choose Device dialog. Close the Choose Device dialog if it is still open and the device list is empty (shown below). If Android Studio isn't open, then start it. Build the project (small green triangular button –shown 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.



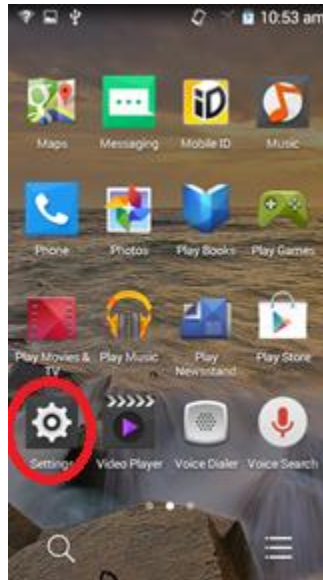
Deploying the Driver Station

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).



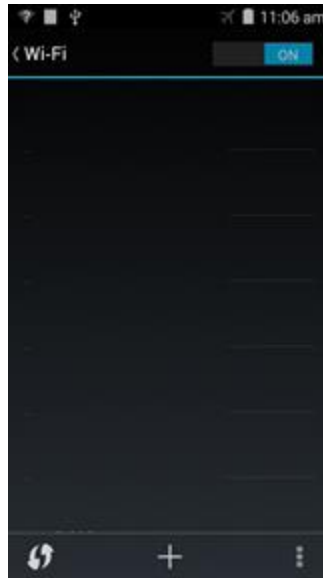
Step 2: Open the settings activity by selecting its icon (shown in the red circle). It might be necessary to swipe the screen left or right until the icon comes into view.



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

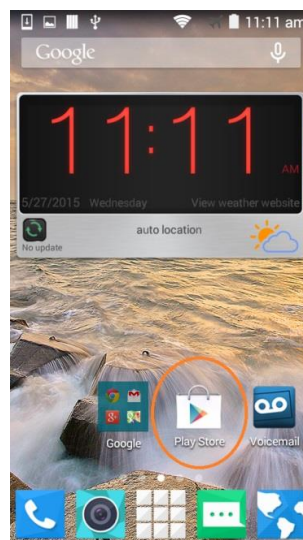


Step 4: Connect to a Wi-Fi network that is able to reach the Internet and to which the login information is known. The image below shows no networks. If networks are available, then they will be shown in the middle of the window. Select a Wi-Fi network and enter any required login information.

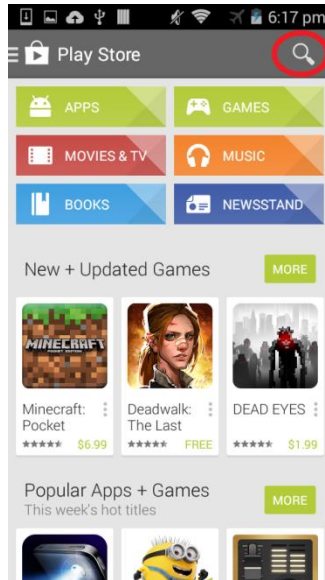


Step 5: Return to the home screen by pressing the home button (the center icon on the bottom of the phone's screen).

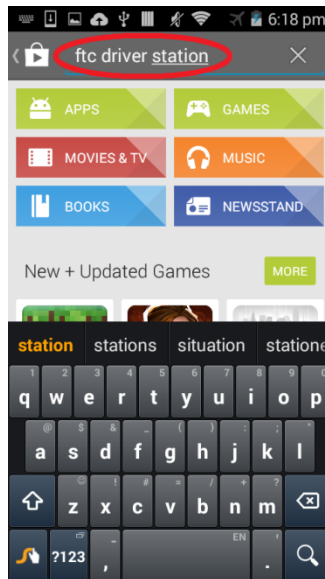
Step 6: Launch the Google Play Store app (orange circle). The Play Store app might prompt you to either login to an existing Google account or create a new one. Follow the onscreen instructions to either create a new (free) account or login to your existing account. The Driver Station app is free and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



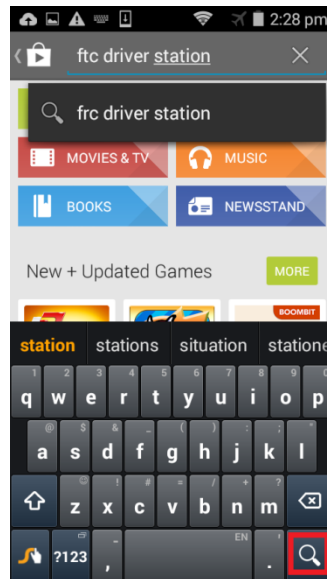
Step 7: Click on the search icon.



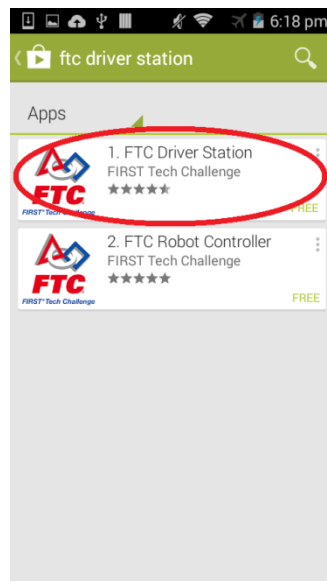
Step 8: Enter the phrase "FTC Driver Station".



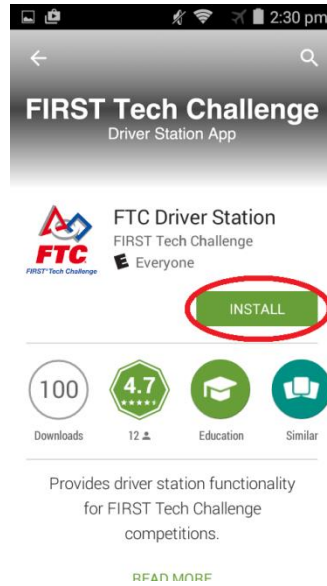
Step 9: And press the magnifying glass in the bottom right corner (on the keyboard).



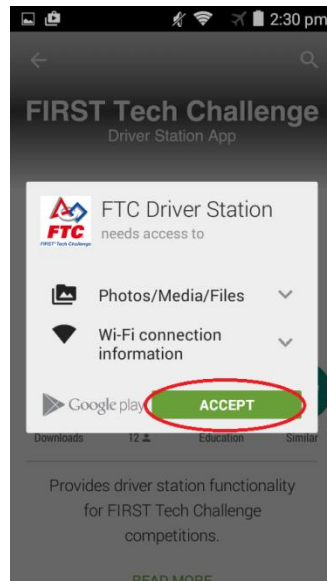
Step 10: Select the FTC Driver Station block.



Step 11: Select 'INSTALL'.



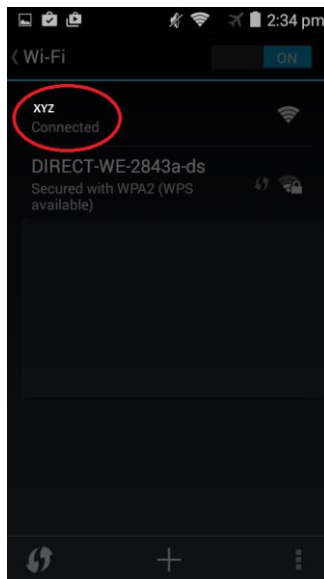
Step 12: Select 'Accept'.



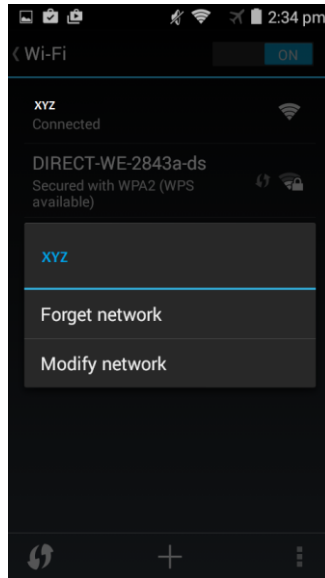
Step 13: Return to the home screen. Observe the FTC Driver Station icon is displayed there. If it isn't shown on the home screen then search for it in the app drawer.



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.



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.



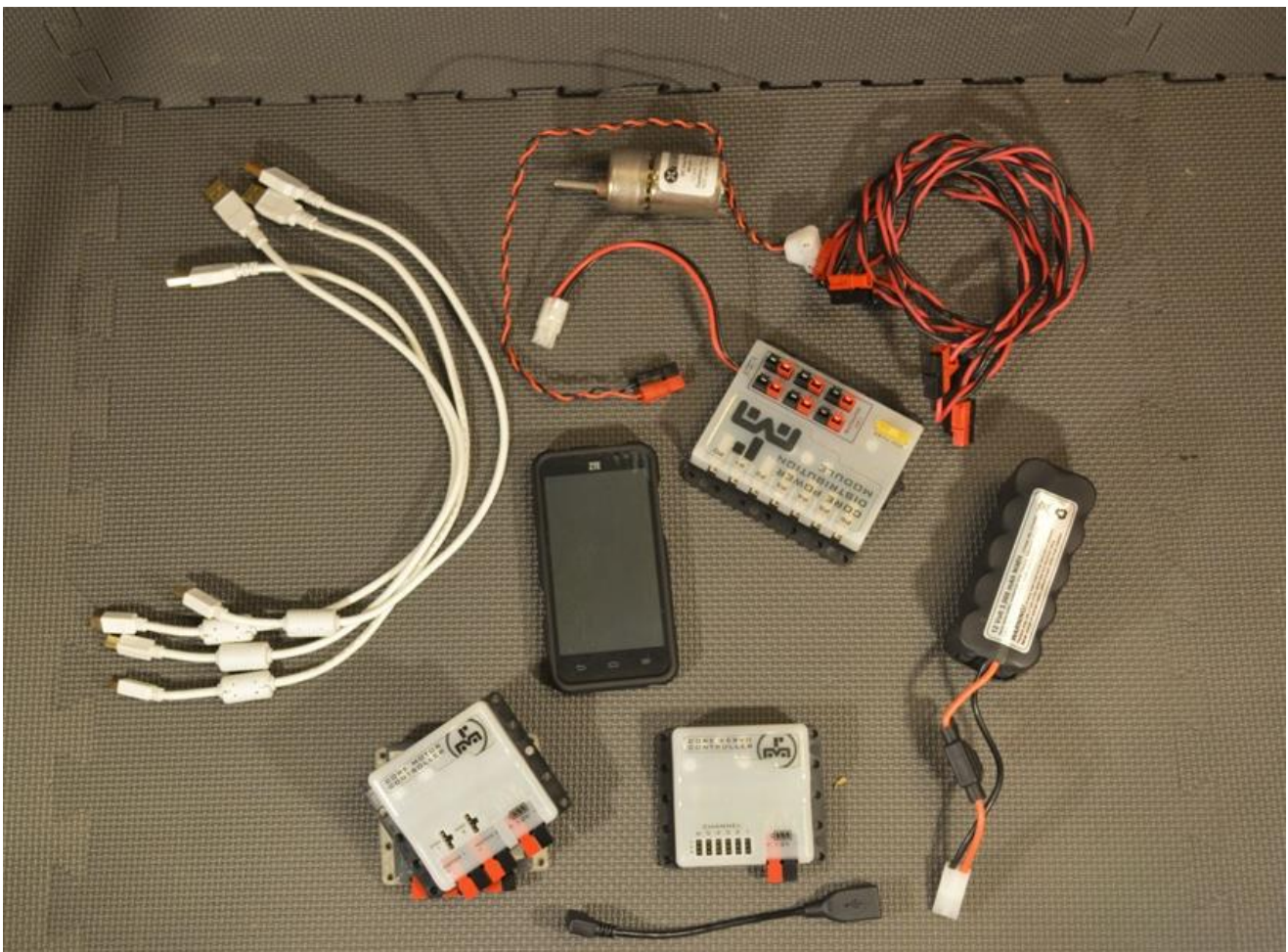
Integrating the Op modes and Electronics

From previous steps, the Driver Station and Robot Controller apps have been installed on the cell phones. The cell phones are powered on and the apps have been started. When the Robot Controller app configures the USB devices, the devices are assigned randomly – the order depends on which control 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.

Building the electronics test bench

This section will outline the testing of the electronics to determine which motor controller will be used to control the drive wheel motors and which will control the arm motor. This step happens prior to attaching the electronics to the Robot. Here is a picture showing the electronics that will be mounted to the Robot. Note: parts may vary slightly in appearance to what is shown.

The instructions in this section apply to the Robot Controller cell phone only.



Step 1: ZTE phone (1), USB On The Go (OTG) cable (1), USB A male to mini-B USB cable (1).



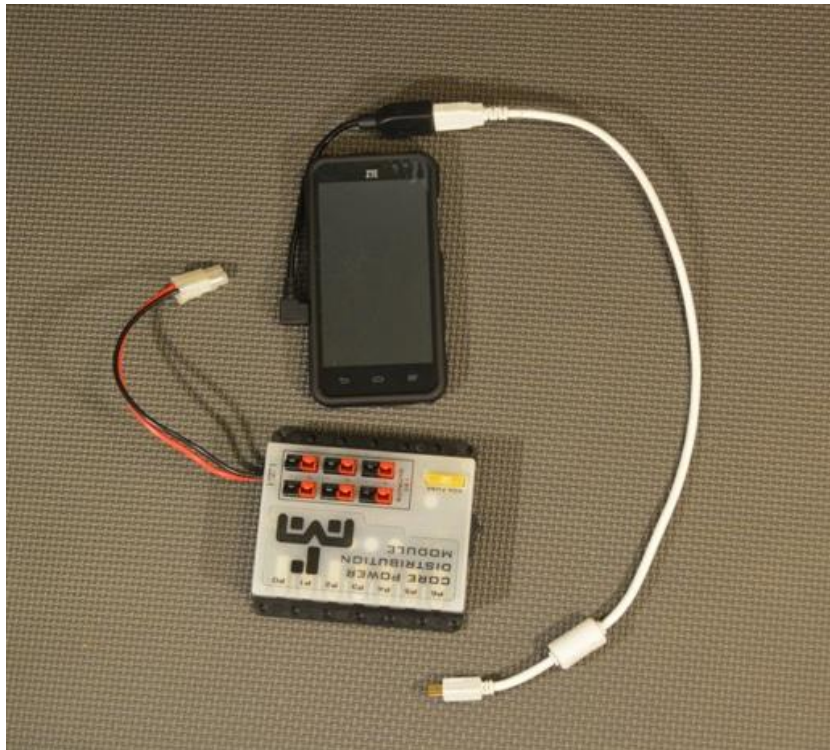
Step 2: Connect the USB OTG cable to the USB port on the phone



Step 3: Connect the USB A male to mini-B USB cable to the USB OTG cable.



Step 4: core power distribution module.



Step 5: Connect the mini-B connector to the core power distribution module.



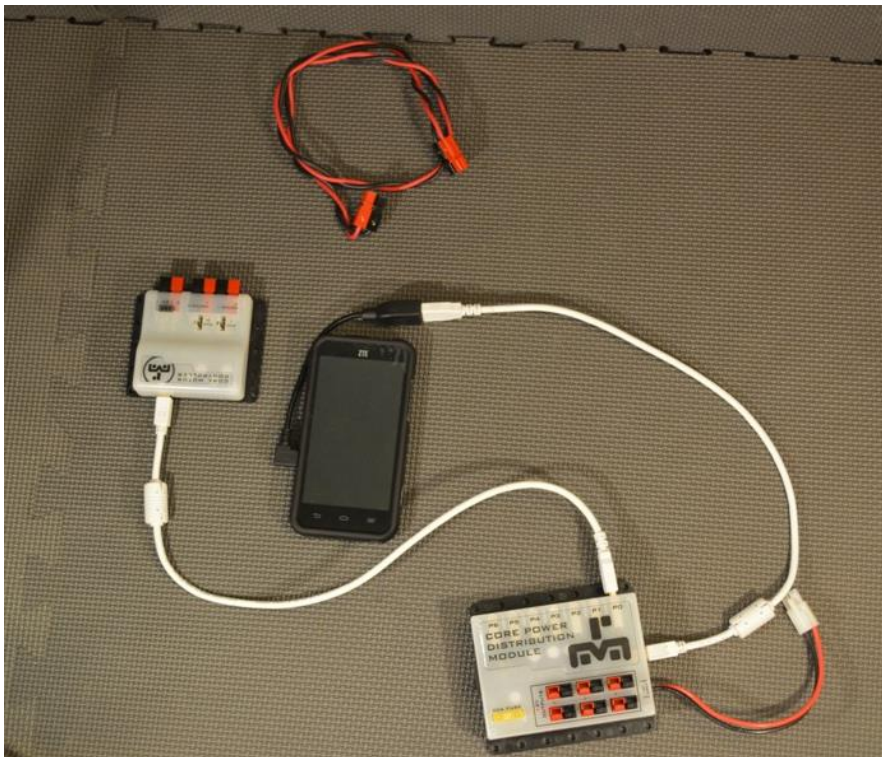
Step 6: core motor controller (1), USB A male to mini-B USB cable (1), DC power cable (1).



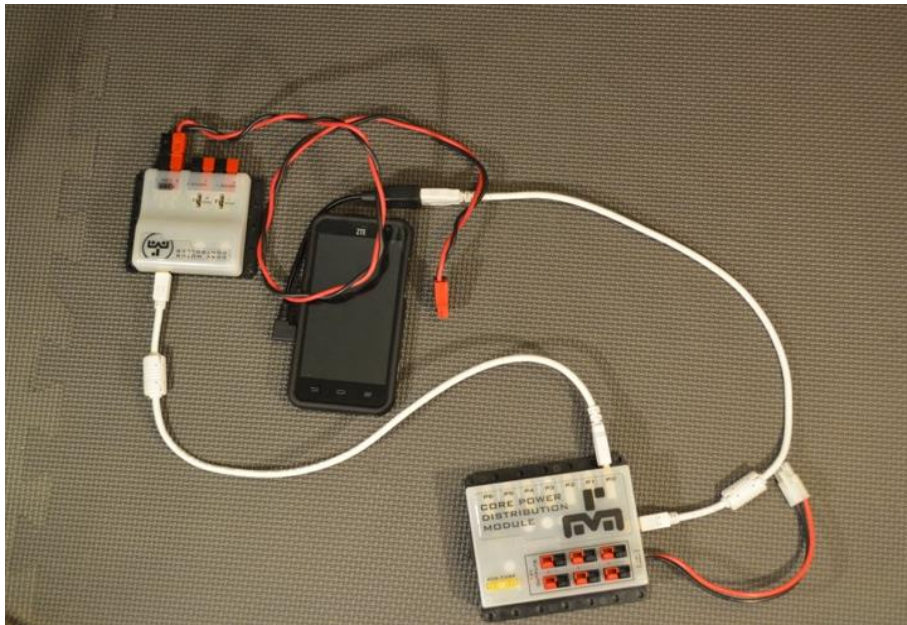
Step 7: Connect USB mini-B connector to the core motor controller.



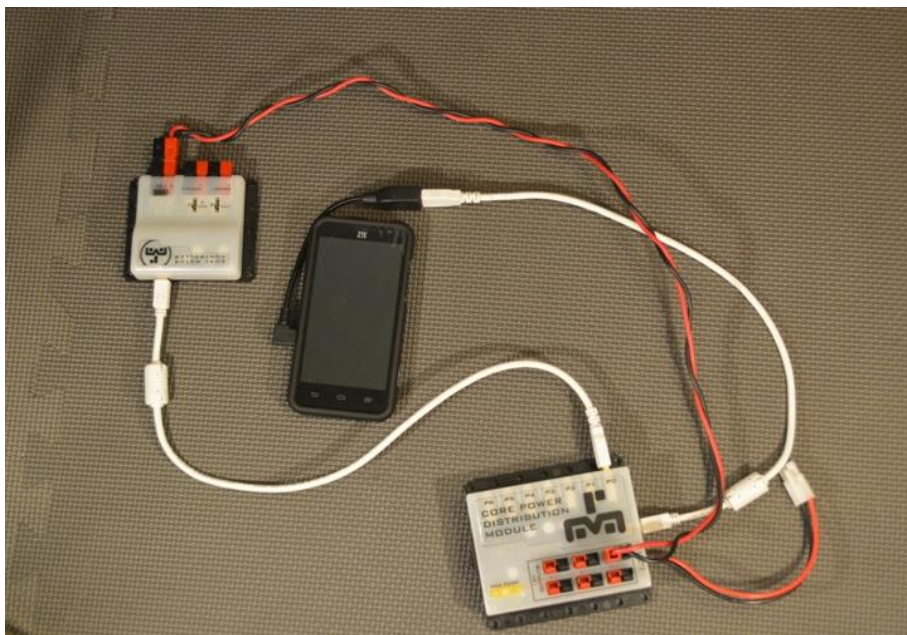
Step 8: Connect USB A connector to any of the remaining USB ports on the core power distribution module.



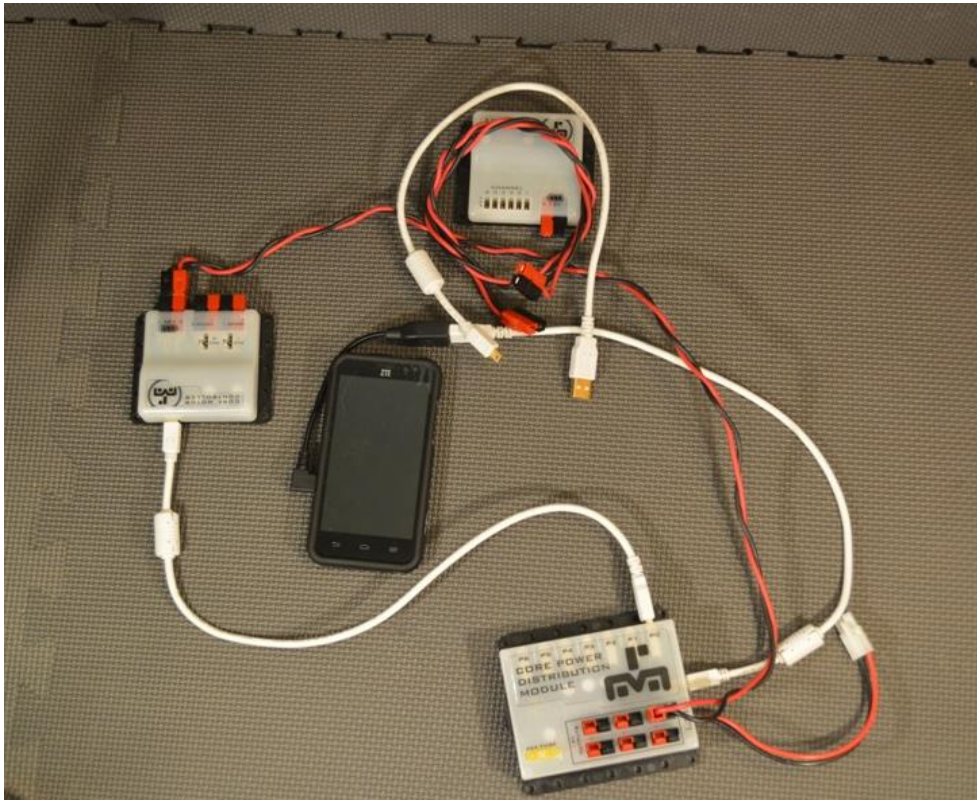
Step 9: Connect DC power wire to the core motor controller power input connector.



Step 10: Connect other end of the DC power wire to any of the power output connectors on the core power distribution module.



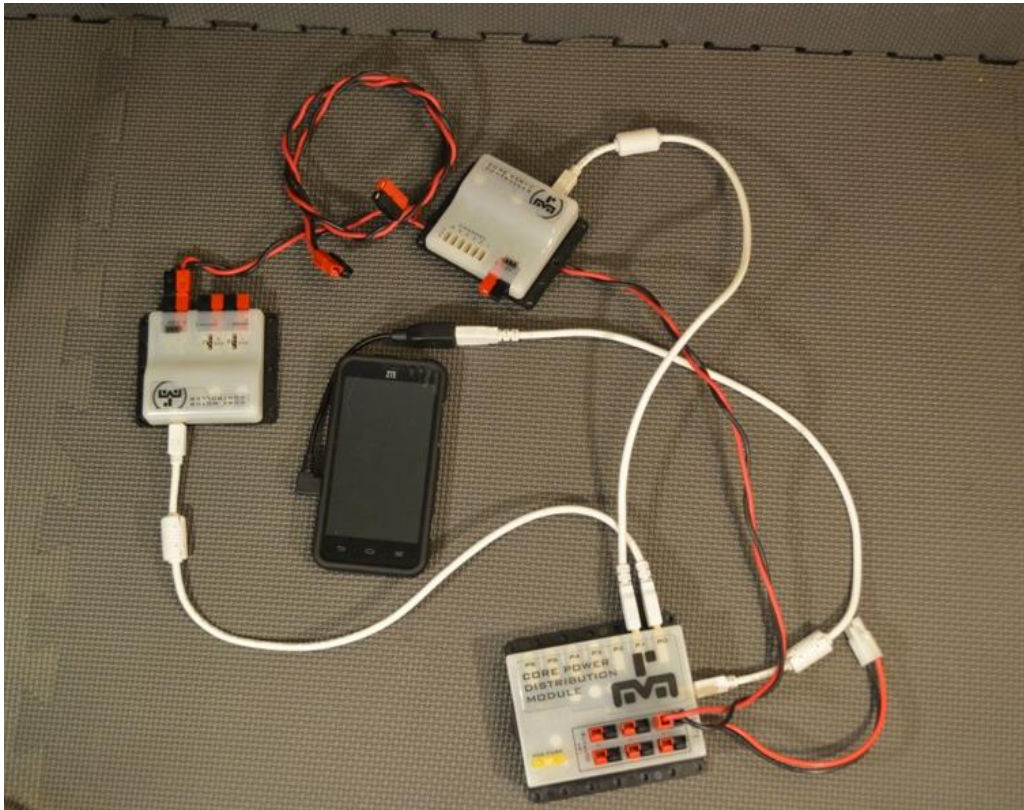
Step 11 core servo controller (1), USB A male to mini-B USB cable (1), DC power cable (1).



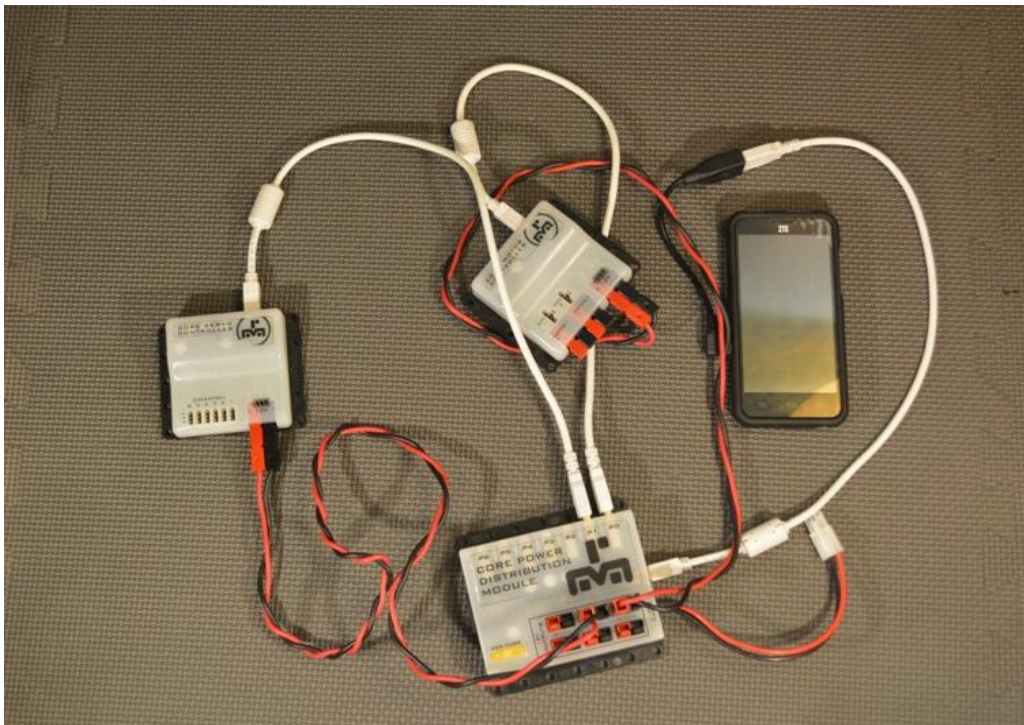
Step 12: Connect USB mini-B connector to the core servo controller.



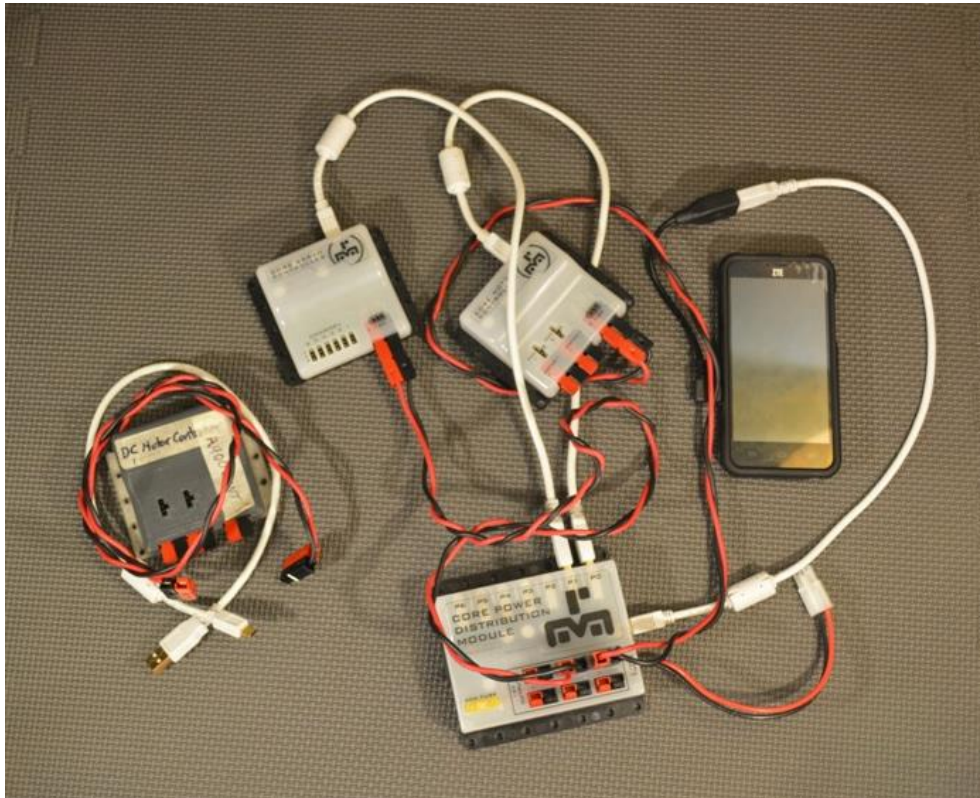
Step 13: Connect USB A connector to any of the remaining USB ports on the core power distribution module.



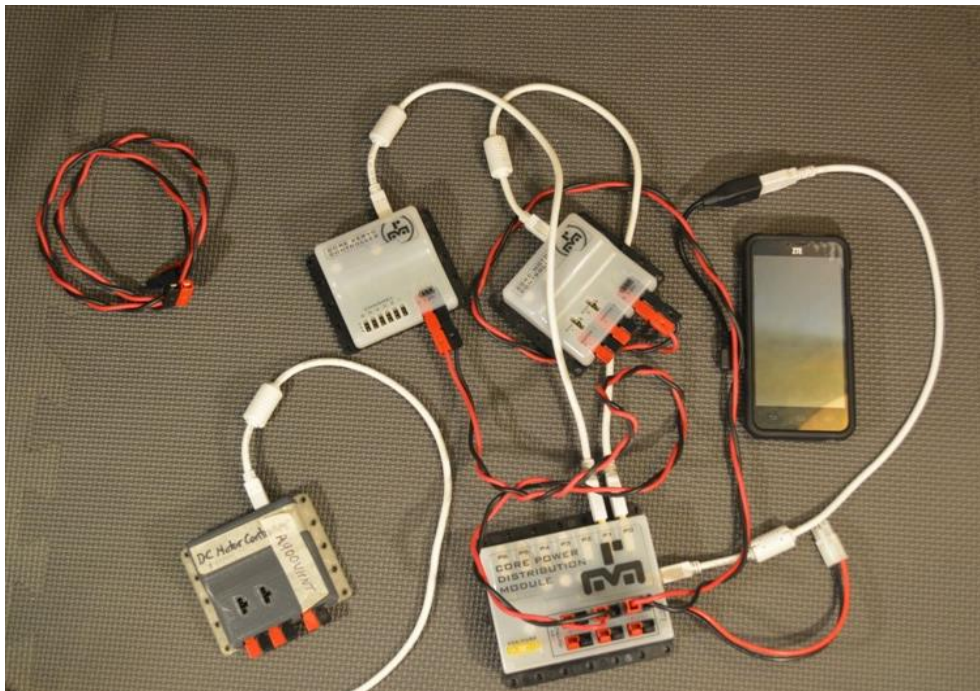
Step 14: Connect one end of the DC power wire to the core servo controller power input connector and connect the other end to any of the power output connectors on the core power distribution module.



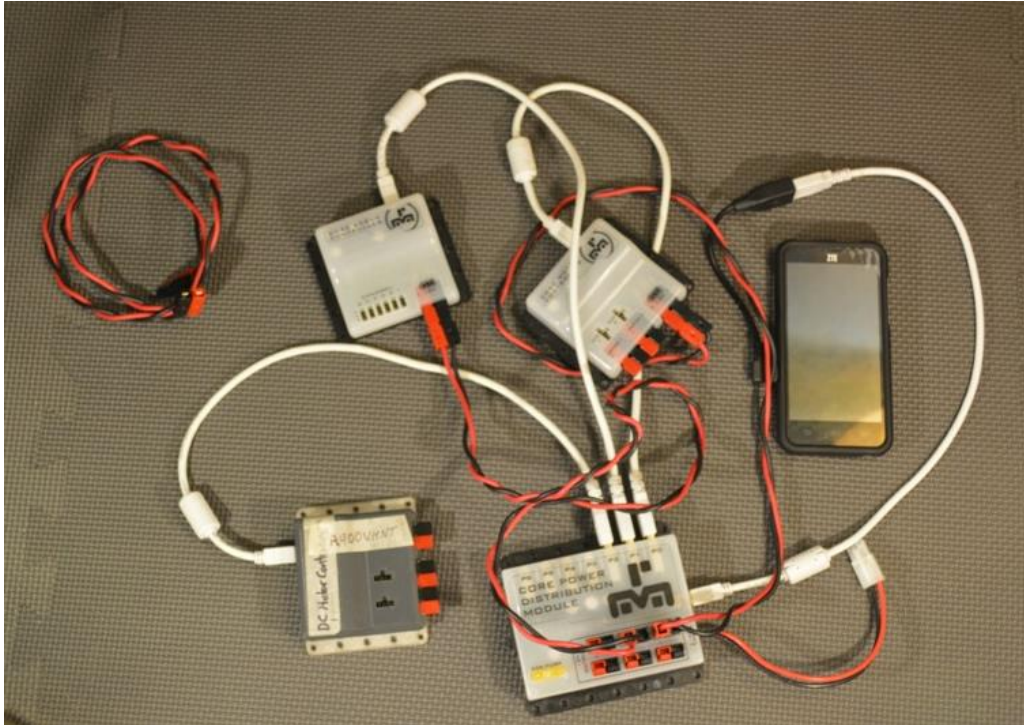
Step 15: core motor controller (1), USB A male to mini-B USB cable (1), DC power cable (1).



Step 16: Connect USB mini-B connector to the core motor controller.



Step 17: Connect USB A connector to any of the remaining USB ports on the core power distribution module.



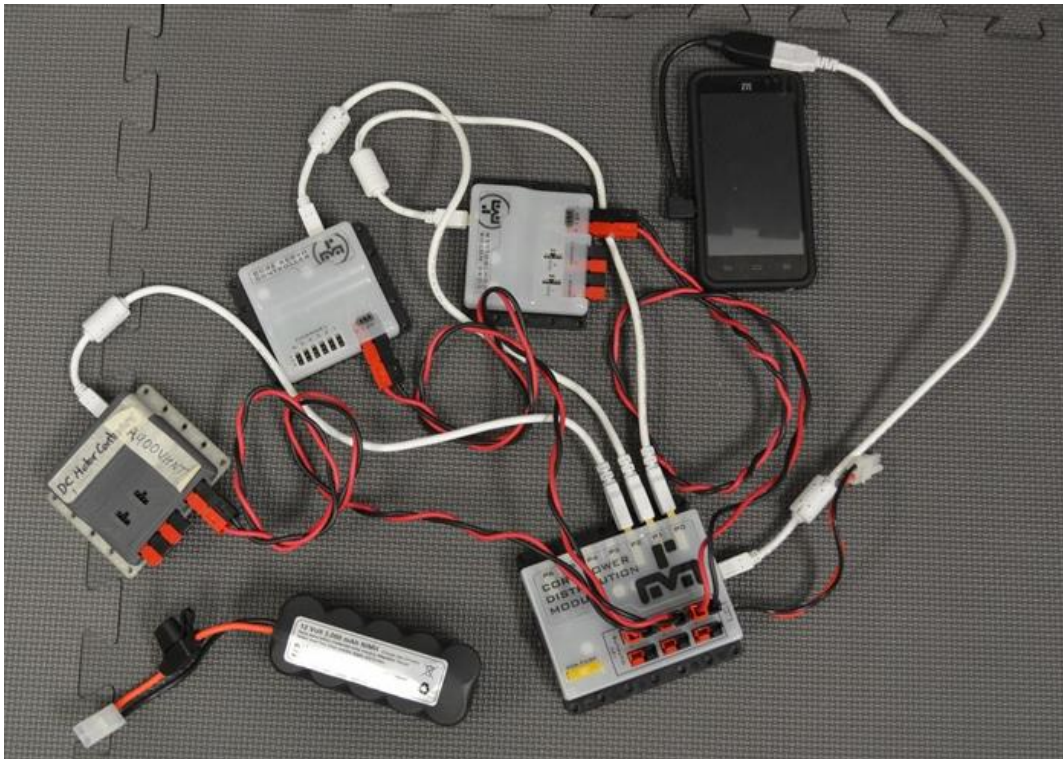
Step 18: Connect DC power wire to the core motor controller power input connector.



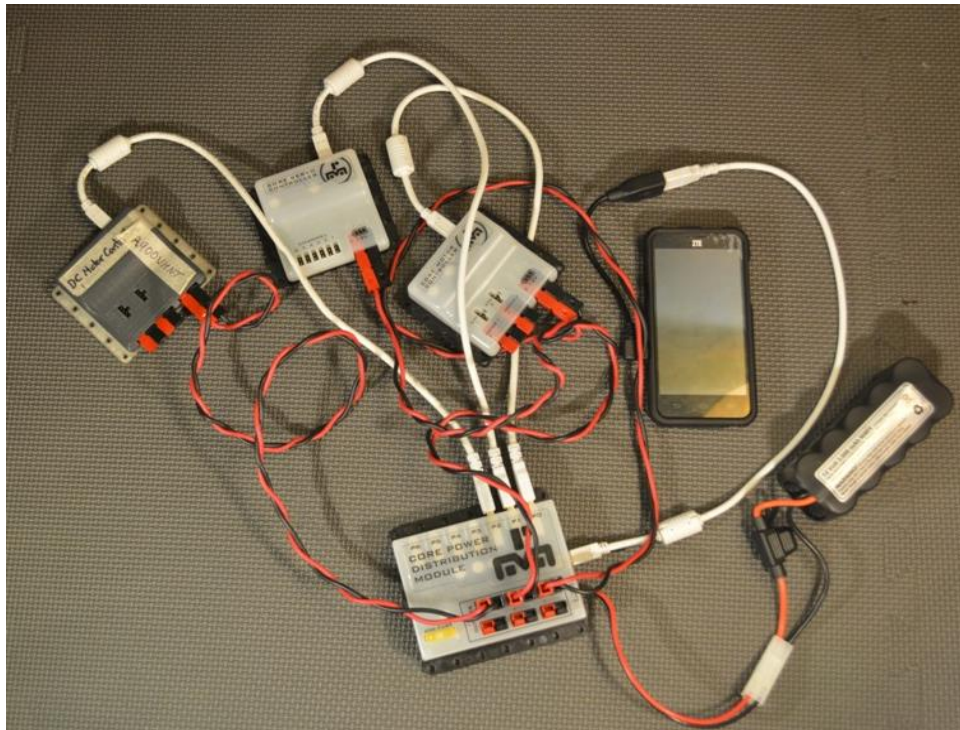
Step 19: Connect other end of the DC power wire to any of the power output connectors on the core power distribution module.



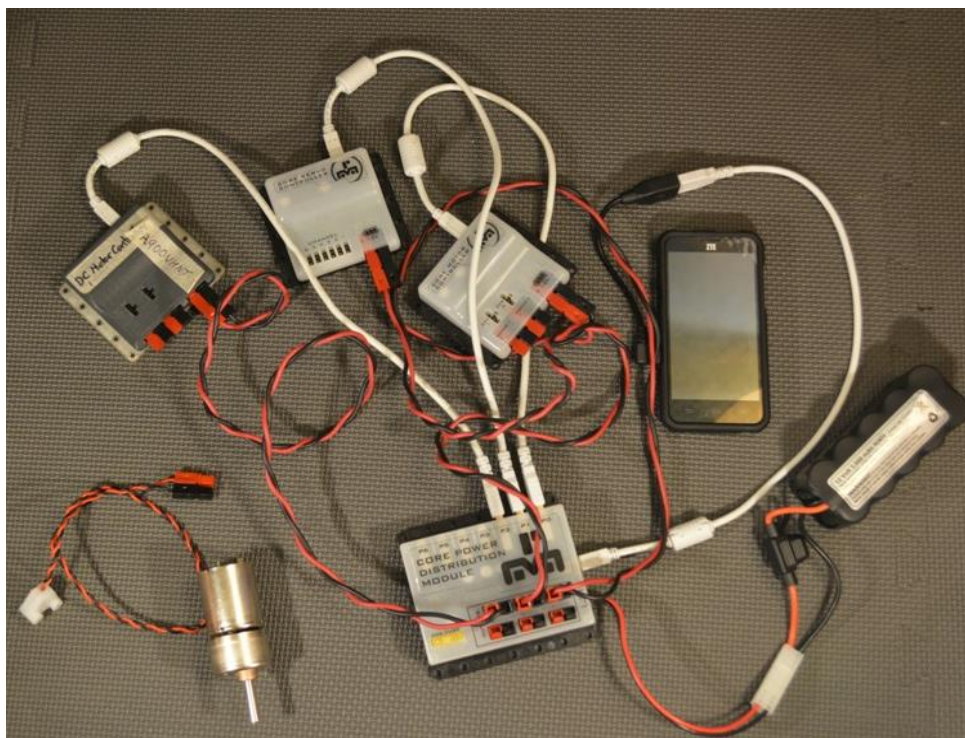
Step 20: TETRIX 12-volt battery (1).



Step 20: Connect TETRIX 12-volt battery to power input connector on core power distribution module.



Step 21: TETRIX DC gear motor (1), motor power cable (1).



Step 22: Connect motor power cable to DC gear motor.



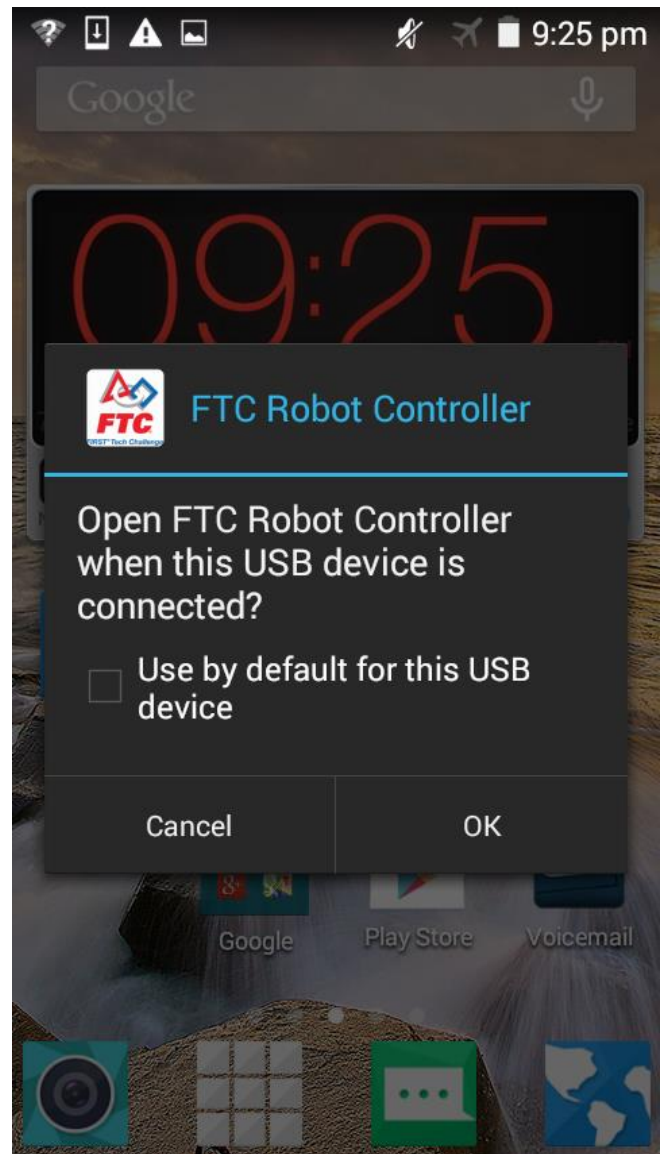
Configuring the USB devices

The Robot Controller app scans the devices connected to the Core Power Distribution Module. It builds a list of connected devices. To make the devices usable by an op-mode, they must be named. There is no identifying information provided via the interface, so a manual determination is required.

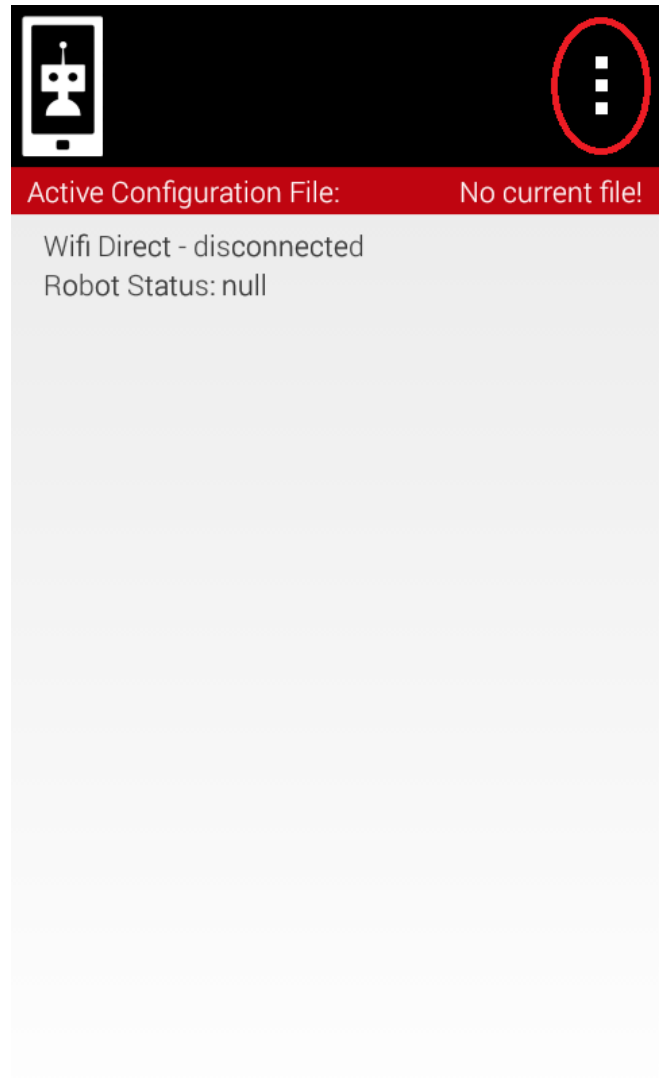
Step 1: Turn the core power distribution module on. The switch is on the side away from the power cable.



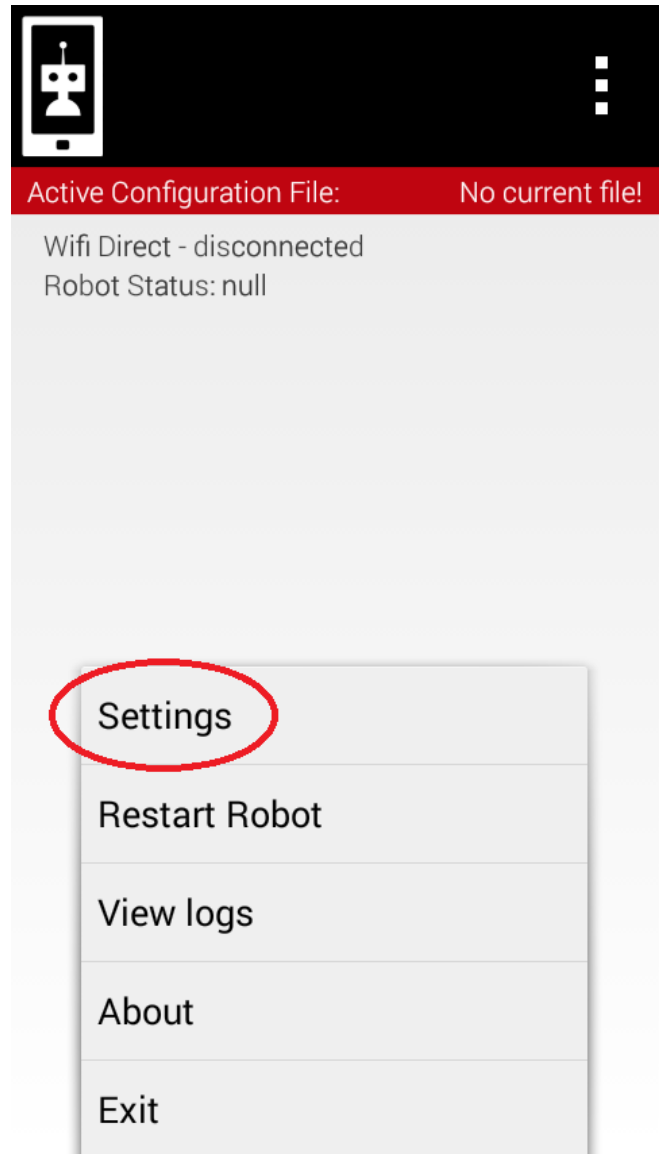
Step 2: When the core power distribution module is powered on, the Robot Controller app should auto-start (the home screen is replaced by the FTC Robot Controller activity) and may prompt the user to “remember” whether it can auto-start when the core power distribution module is detected in the future.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



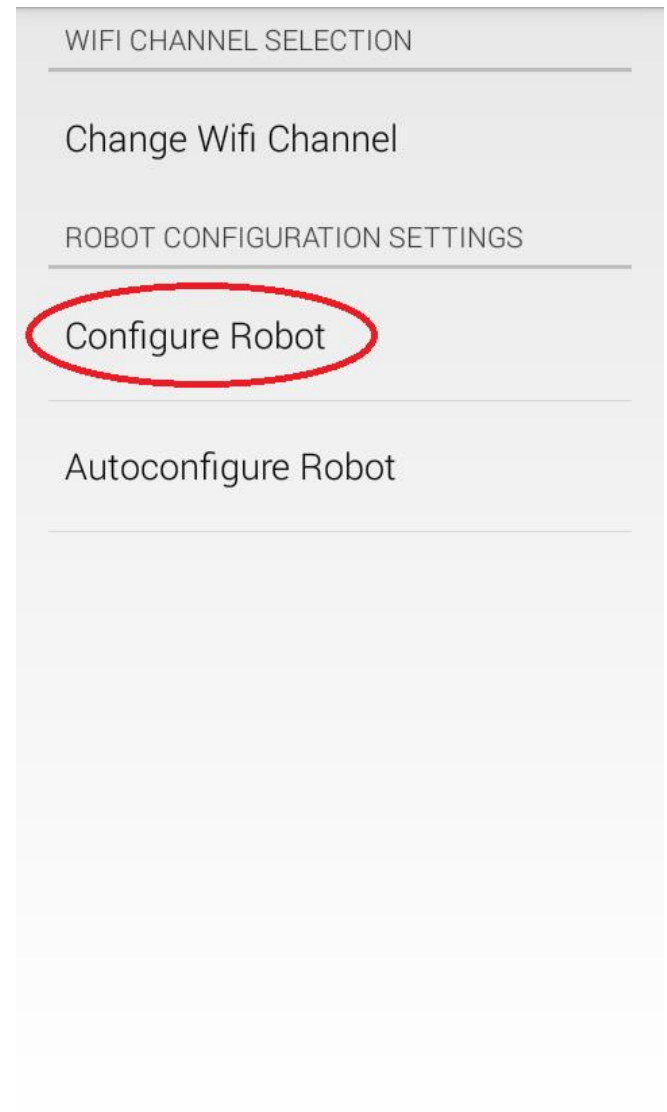
Step 3: Touch the overflow settings icon (three dots in the top right corner) to open a menu.



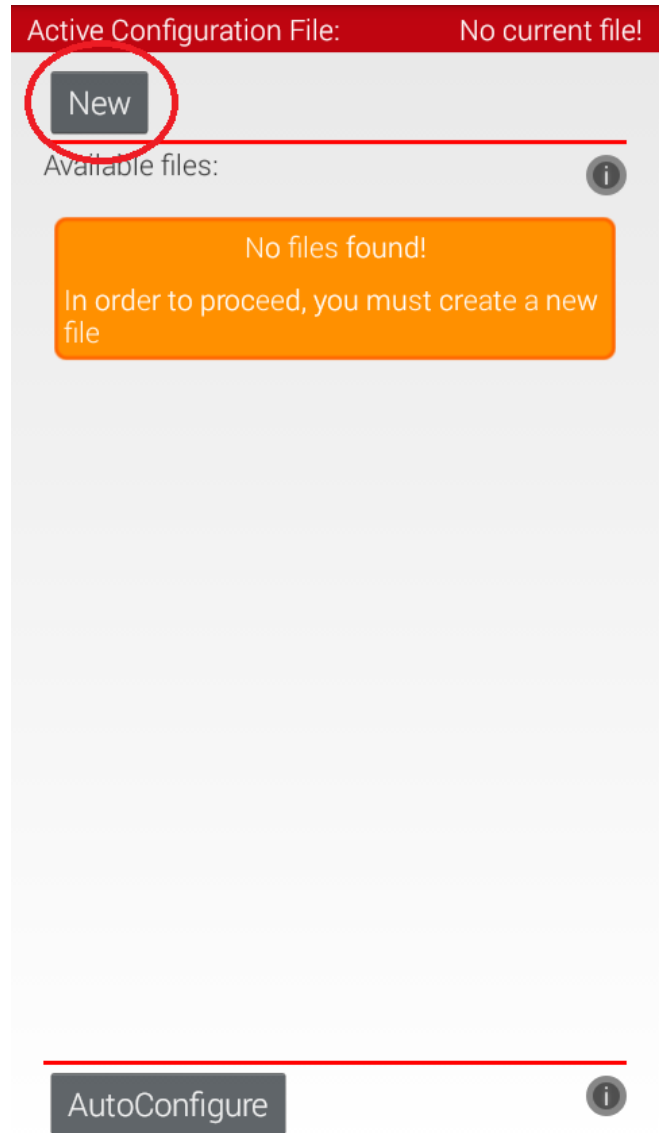
Step 4: Select settings.



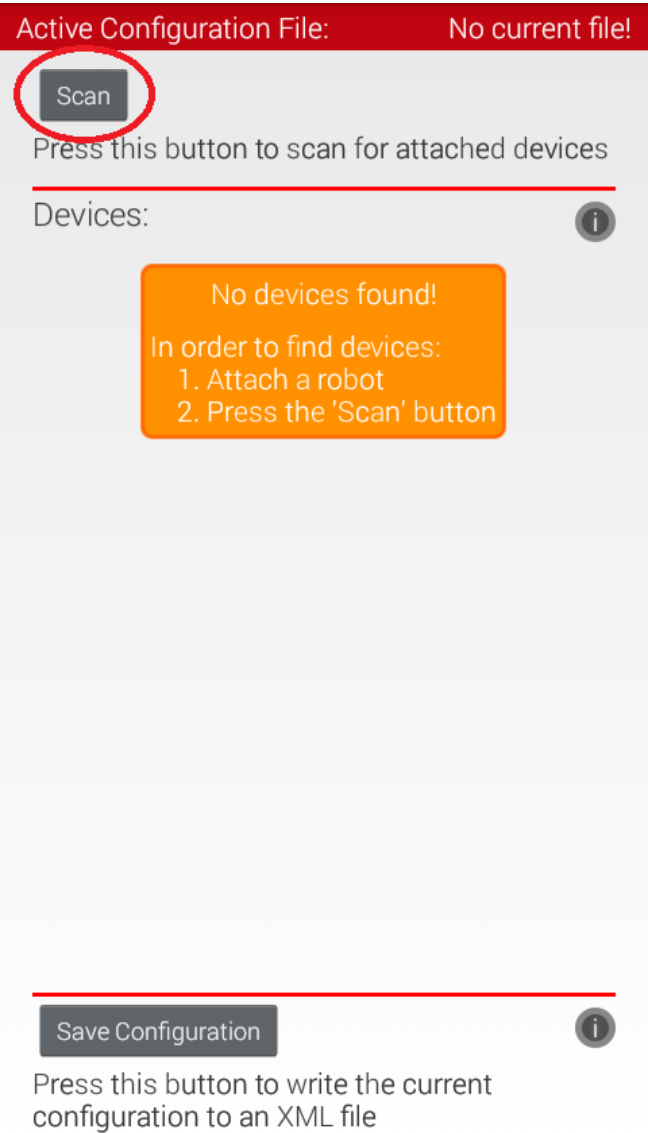
Step 5: Select 'Configure Robot'.



Step 6: Select 'New'.



Step 7: Select 'Scan'.



Active Configuration File: No current file!

Scan

Press this button to scan for attached devices

Devices: ⓘ

No devices found!

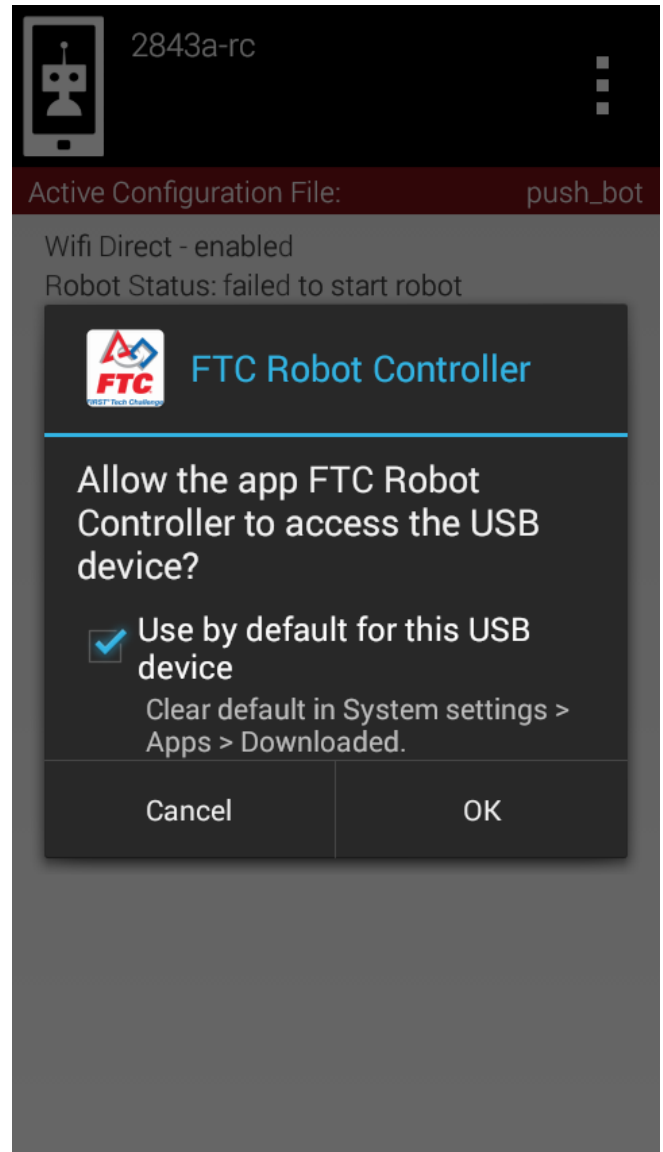
In order to find devices:

1. Attach a robot
2. Press the 'Scan' button

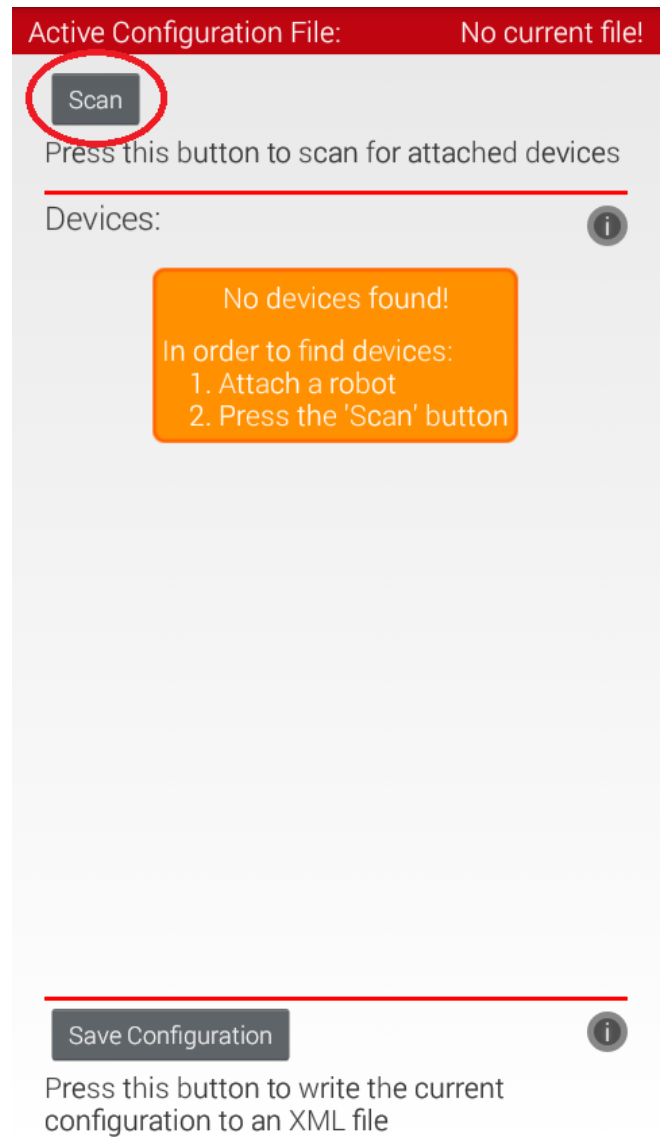
Save Configuration ⓘ

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

Step 8: If this dialog is shown, then check the box to allow the app to access the USB device by default and select 'OK'. If it isn't shown, then continue to the next step.



Step 9: Select 'Scan' and the app will build a list of connected devices. Multiple scans may be required until the two motor controllers and one servo controller appear in the list.



Step 10: Select one of the motor controllers; it doesn't matter which one.

Active Configuration File: No current file!

Scan

Press this button to scan for attached devices

Devices: i

Motor Controller 2

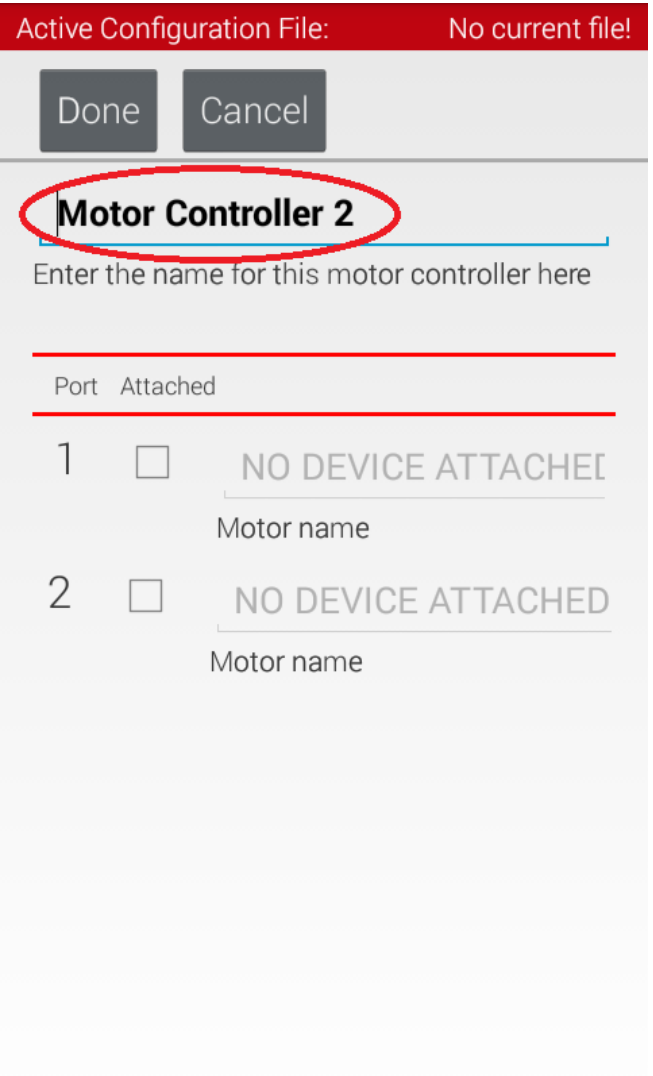
Servo Controller 1

Motor Controller 3

Save Configuration i

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

Step 11: Tap on the Motor Controller's name. It is the text just above "Enter the name for this motor controller here". Using the keyboard, enter 'drive_controller'. Long holding the 'f' character will insert the underscore between left and drive.



Active Configuration File: No current file!

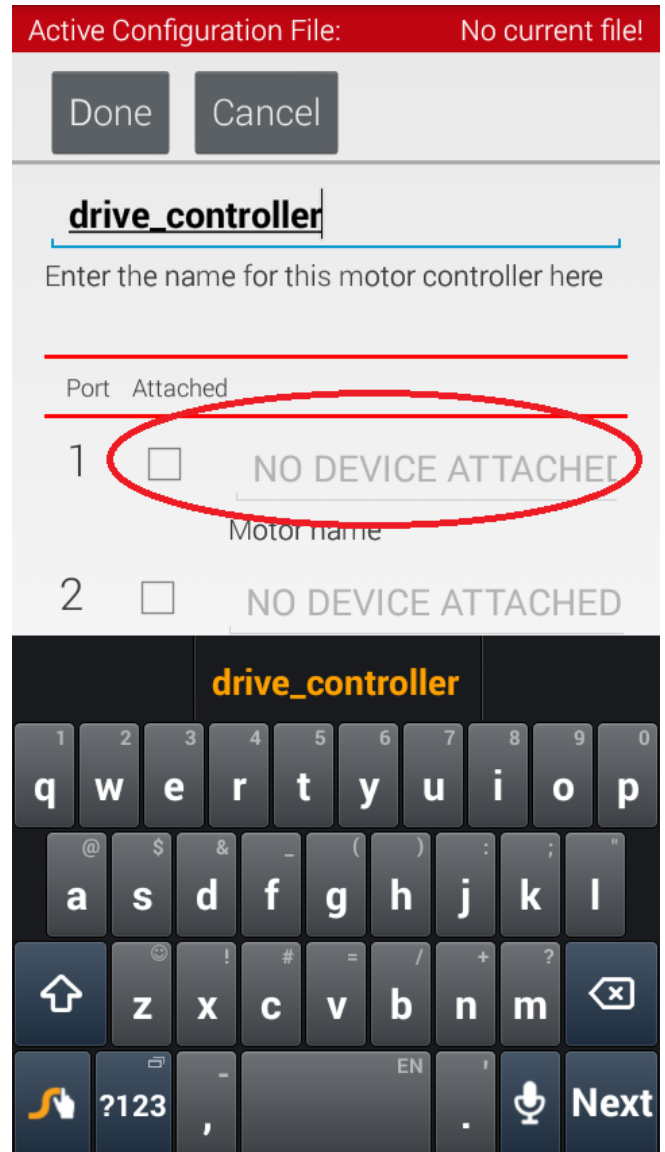
Done Cancel

Motor Controller 2

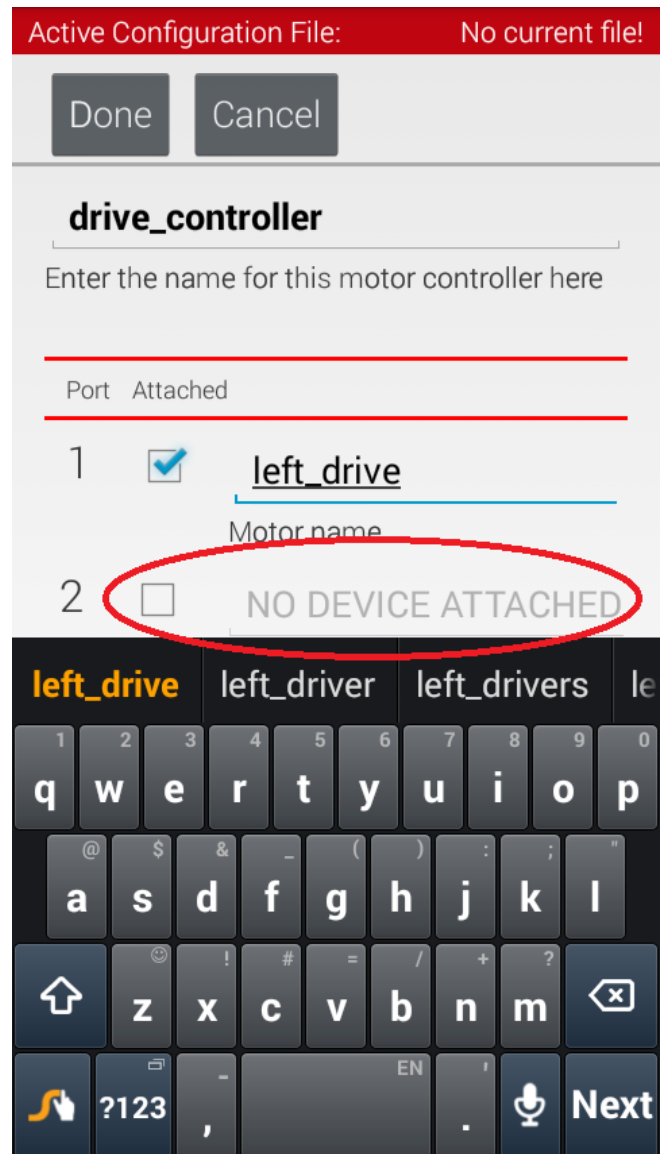
Enter the name for this motor controller here

Port	Attached
1	<input type="checkbox"/> NO DEVICE ATTACHED Motor name
2	<input type="checkbox"/> NO DEVICE ATTACHED Motor name

Step 12: Check the box next to 1. Tap on the text “NO DEVICE ATTACHED”. Using the keyboard, enter ‘left_drive’. This must be exact or the Op mode will not work.



Step 13: Check the box next to 2. Using the keyboard, enter 'right_drive'. This must be exact or the Op mode will not work.



Step 14: The screen should appear as it does below. Select 'Done'.

Active Configuration File: No current file!

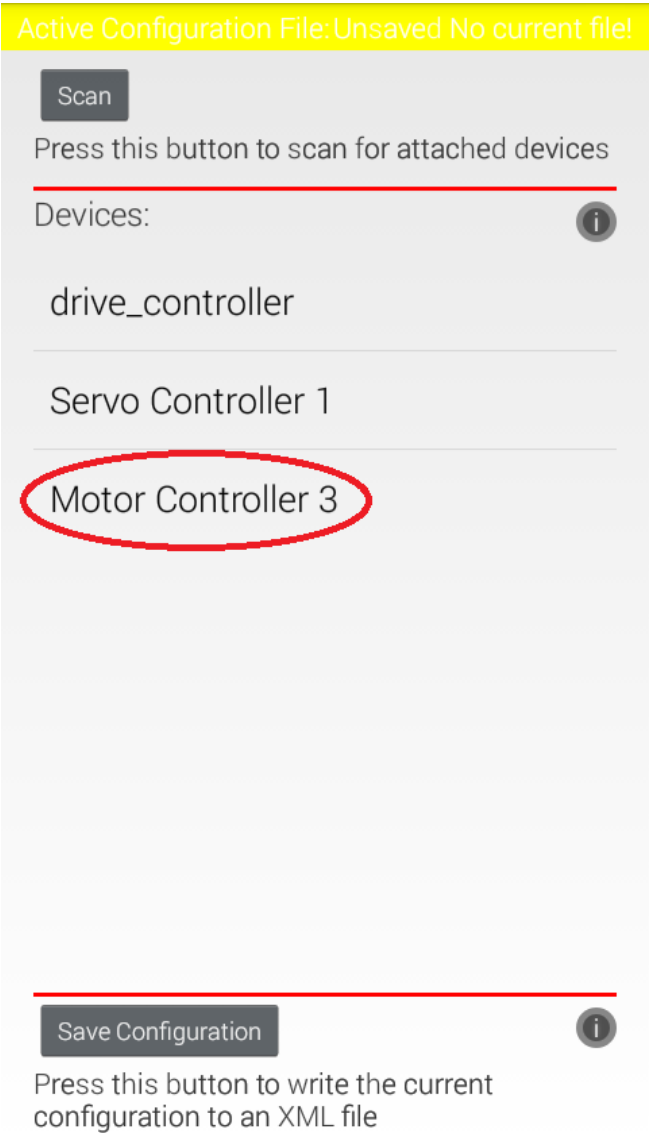
Done **Cancel**

drive_controller

Enter the name for this motor controller here

Port	Attached	
1	<input checked="" type="checkbox"/>	<input type="text" value="left_drive"/> Motor name
2	<input checked="" type="checkbox"/>	<input type="text" value="right_drive"/> Motor name

Step 15: Select the other motor controller.



Active Configuration File: Unsaved No current file!

Scan

Press this button to scan for attached devices

Devices: ⓘ

drive_controller

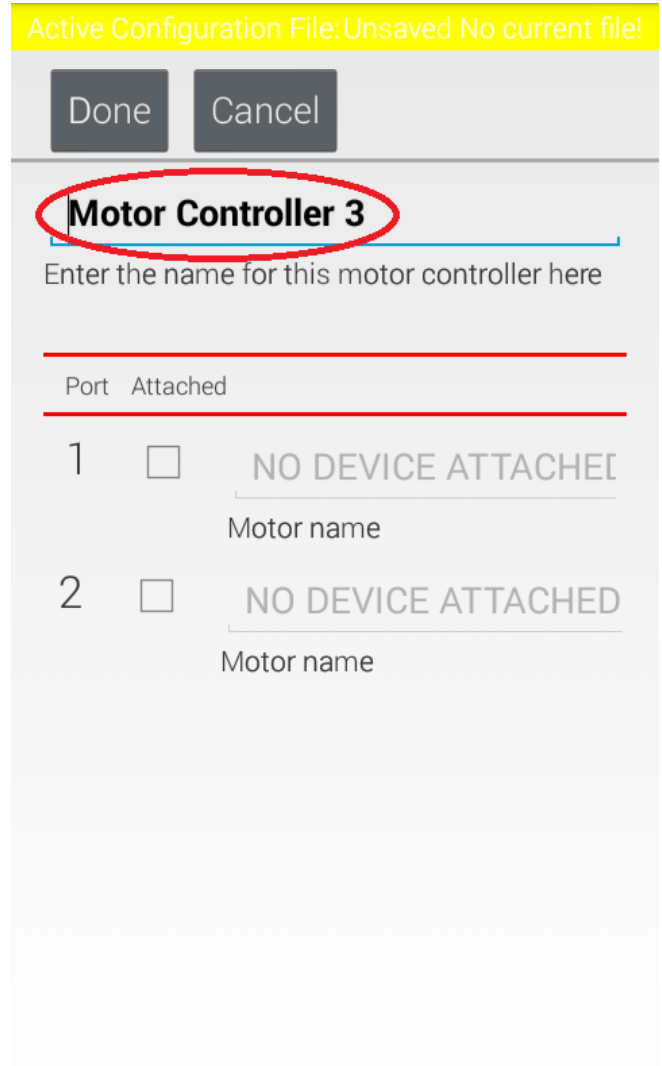
Servo Controller 1

Motor Controller 3

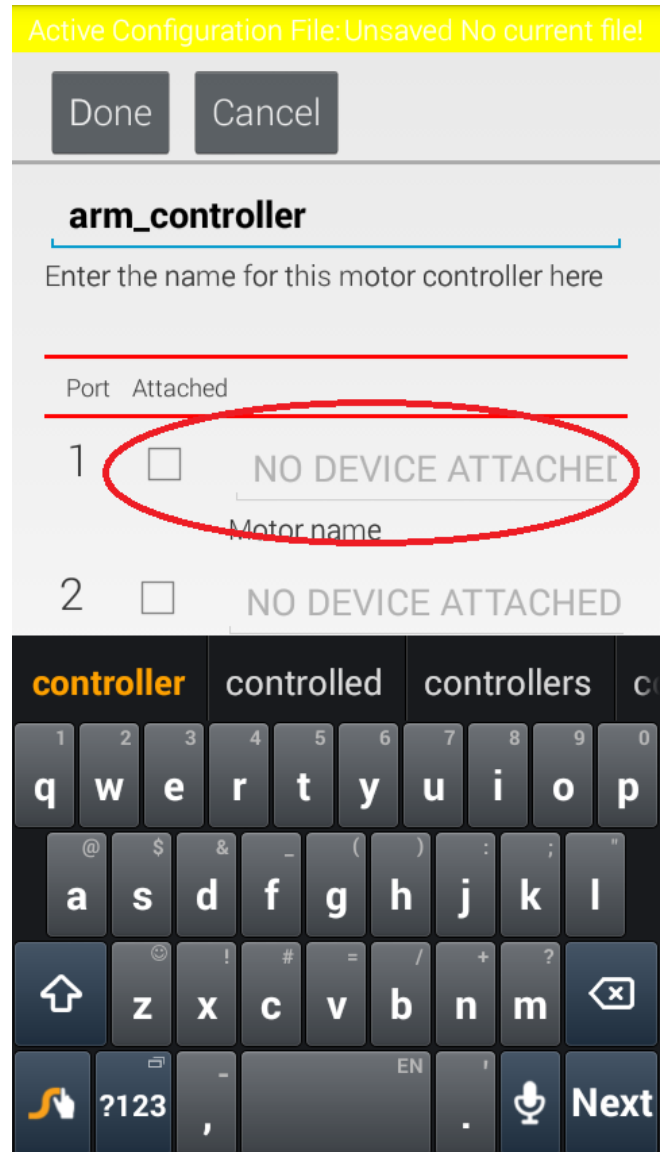
Save Configuration ⓘ

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

Step 16: Tap on the Motor Controller's name. It is the text just above "Enter the name for this motor controller here". Using the keyboard, enter 'arm_controller'.



Step 17: Check the box next to 1. Using the keyboard, enter 'left_arm'. This must be exact or the Op mode will not work.



Step 18: Select 'Done'.

Active Configuration File: Unsaved No current file!

Scan

Press this button to scan for attached devices

Devices: i

drive_controller

Servo Controller 1

arm_controller

Save Configuration i

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

Step 19: Tap on the Servo Controller's name. It is the text just above "Enter the name for this servo controller here". Using the keyboard, enter 'servo_controller'.

Active Configuration File: Unsaved No current file!

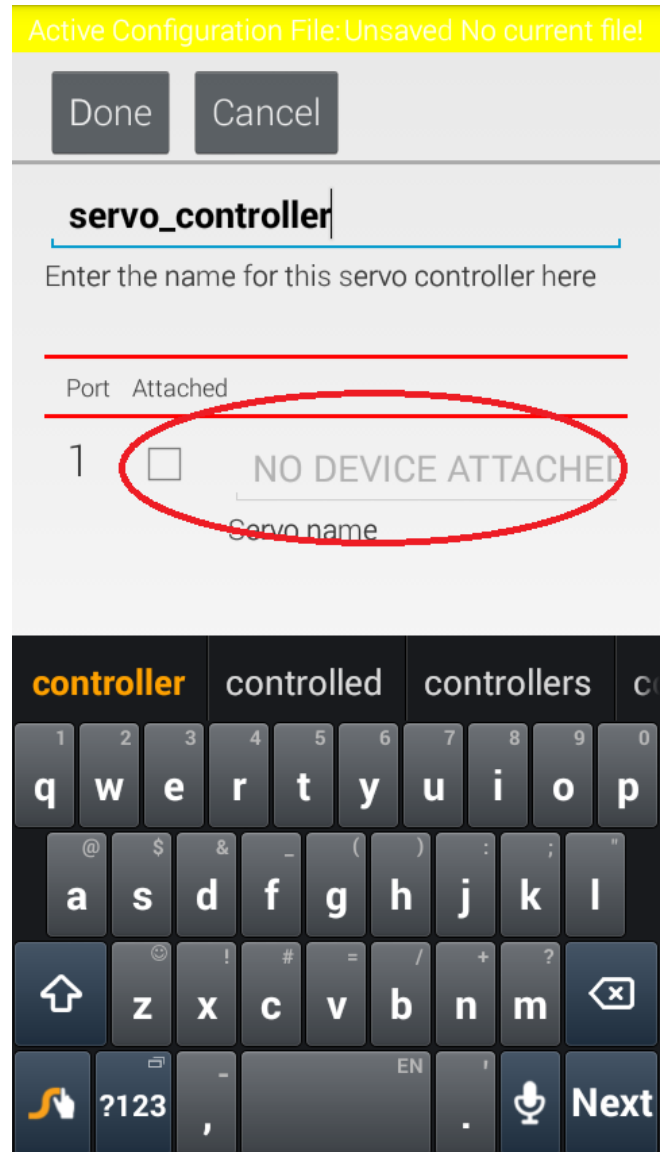
Done Cancel

Servo Controller 1

Enter the name for this servo controller here

Port	Attached
1	<input type="checkbox"/> NO DEVICE ATTACHED Servo name
2	<input type="checkbox"/> NO DEVICE ATTACHED Servo name
3	<input type="checkbox"/> NO DEVICE ATTACHED Servo name
4	<input type="checkbox"/> NO DEVICE ATTACHED Servo name

Step 20: Check the box next to 1. Using the keyboard, enter 'left_hand'. This must be exact or the Op mode will not work.



Step 21: Check the box next to 2. Using the keyboard, enter 'right_hand'. This must be exact or the Op mode will not work.

Active Configuration File: Unsaved No current file!

Done Cancel

servo_controller

Enter the name for this servo controller here

Port	Attached	
1	<input checked="" type="checkbox"/>	<input type="text" value="left_hand"/> Servo name
2	<input checked="" type="checkbox"/>	<input type="text" value="Enter name"/> Servo name
3	<input type="checkbox"/>	<input type="text" value="NO DEVICE ATTACHEI"/> Servo name
4	<input type="checkbox"/>	<input type="text" value="NO DEVICE ATTACHEI"/> Servo name

Step 22: Select 'Done'.

Active Configuration File: Unsaved No current file!

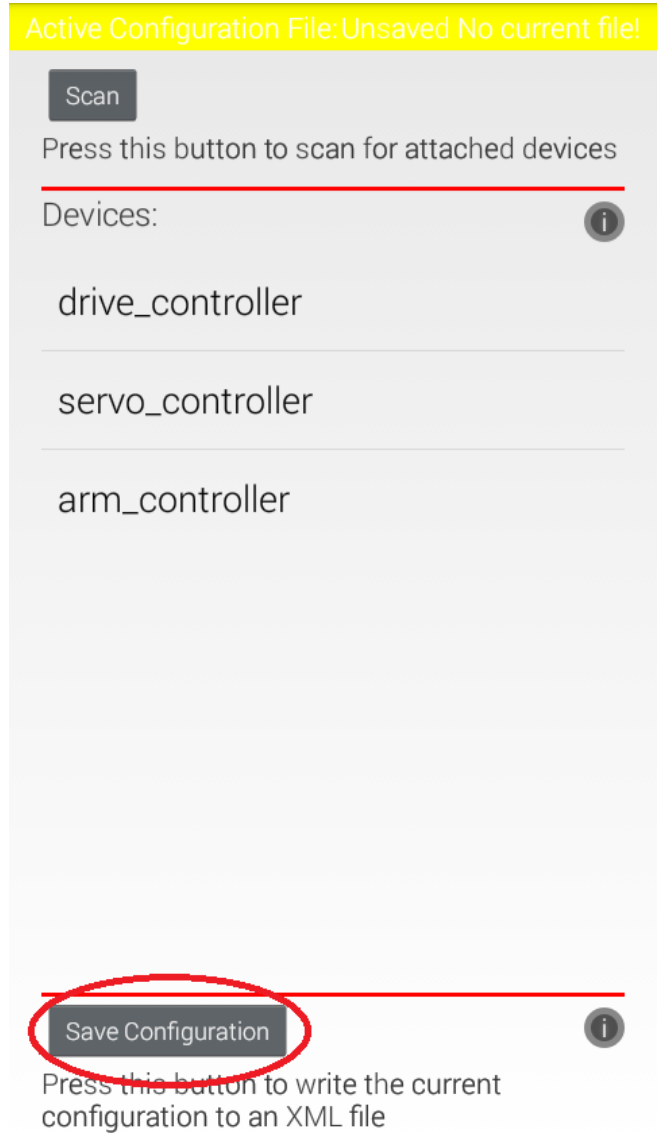
Done **Cancel**

servo_controller

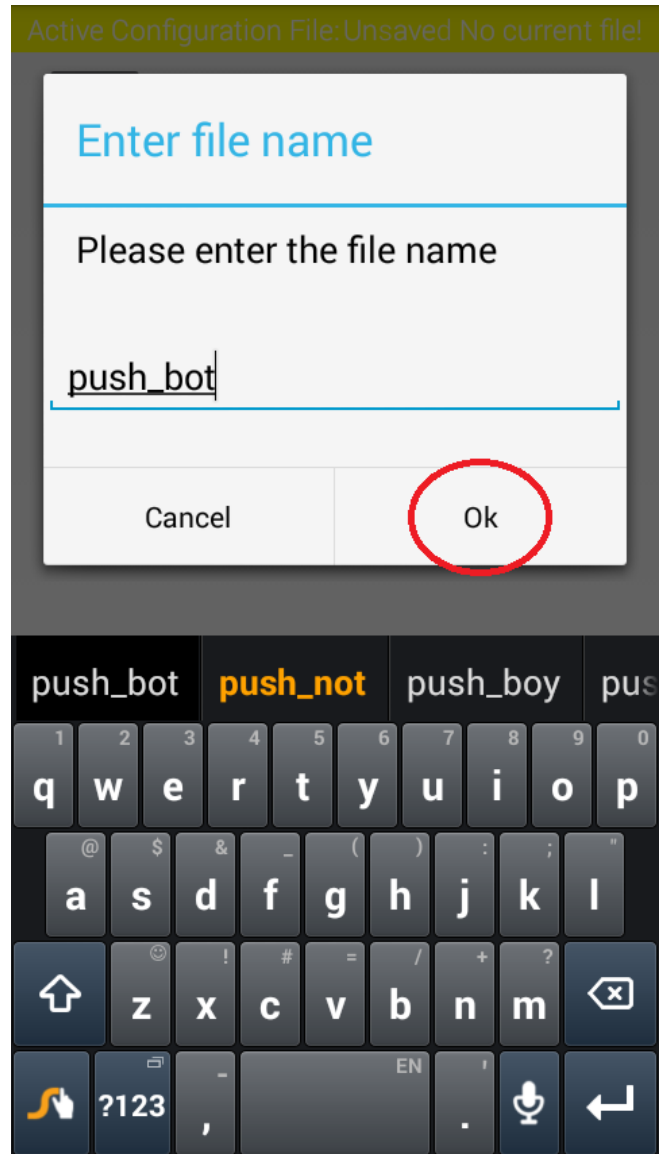
Enter the name for this servo controller here

Port	Attached	
1	<input checked="" type="checkbox"/>	<u>left_hand</u> Servo name
2	<input checked="" type="checkbox"/>	<u>right_hand</u> Servo name
3	<input type="checkbox"/>	<u>NO DEVICE ATTACHEI</u> Servo name
4	<input type="checkbox"/>	<u>NO DEVICE ATTACHEI</u> Servo name

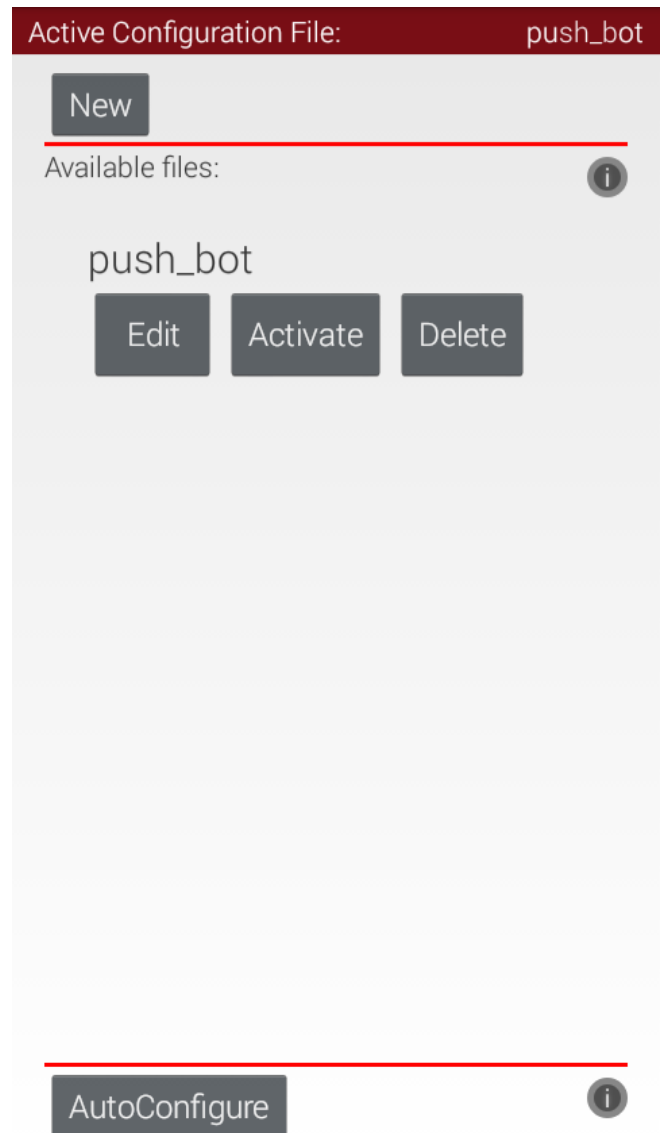
Step 23: Note that all of the devices have new names. Select 'Save Configuration'.



Step 24: A pop-up will be displayed (left). Enter 'push_bot' as the file name. Select 'OK'.



Step 25: Use the phone's back button to return to the page shown on the left. Notice that the push_bot configuration is the 'Active Configuration File' (top right on the page). Also note that it is displayed as an available file.



Step 26: Use the phone's back button to show the page on the right.

WIFI CHANNEL SELECTION

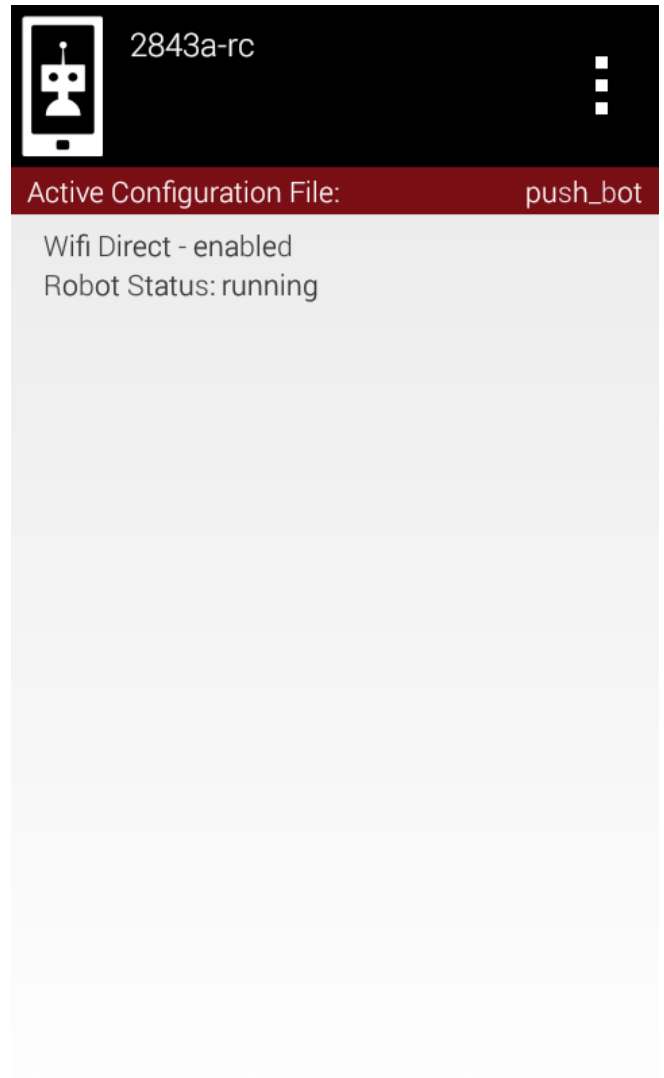
Change Wifi Channel

ROBOT CONFIGURATION SETTINGS

Configure Robot

Autoconfigure Robot

Step 27: Use the phone's back button to show the page on the left.



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.

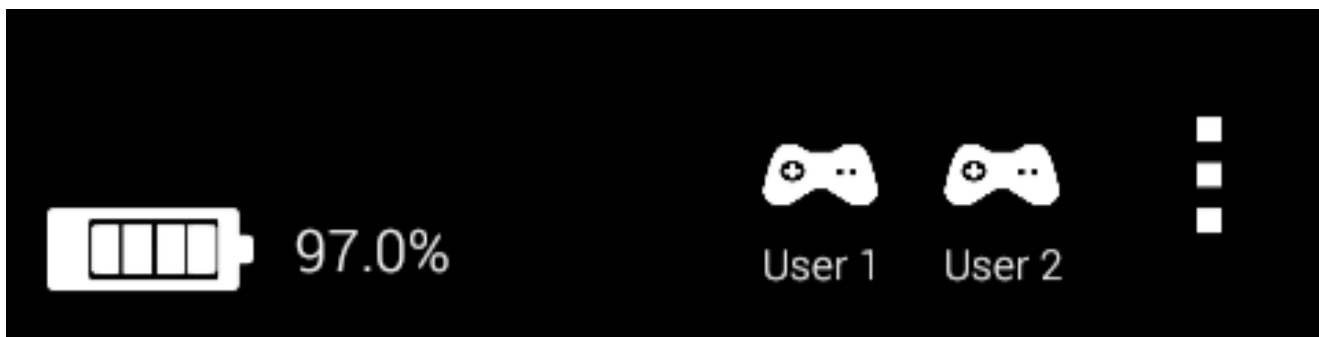
Step 2: Select a gamepad that will be used as driver 1. Push the **Start** key on the gamepad while simultaneously pressing the **A** button.



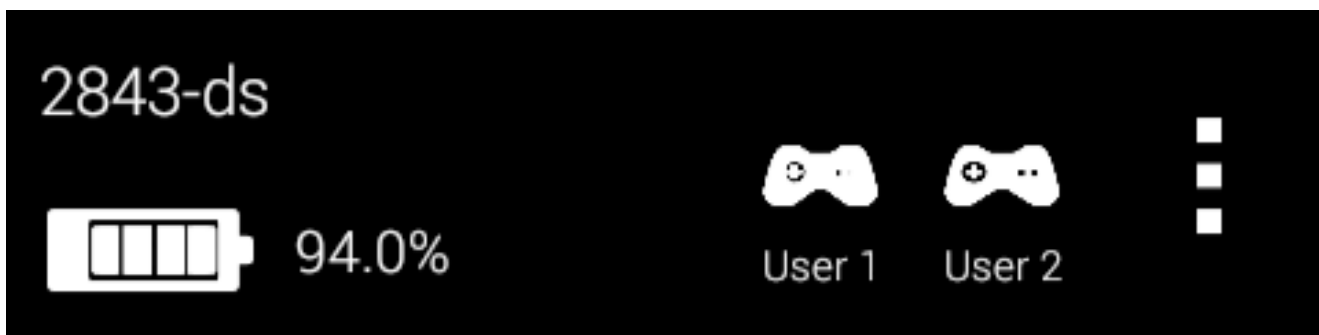
Step 3: Select the other gamepad. It will be used as driver 2. Push the **Start** key on the gamepad while simultaneously pressing the **B** button.



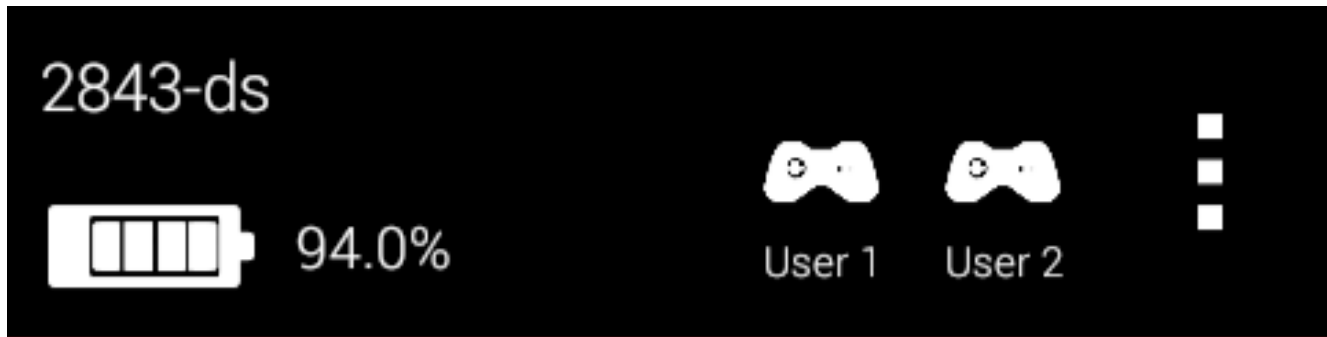
Step 4: Observe that small gamepad icons appear at the top of the screen. User 1's information should be on the left. User 2's information should be on the right. This image shows that neither gamepad is assigned.



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.

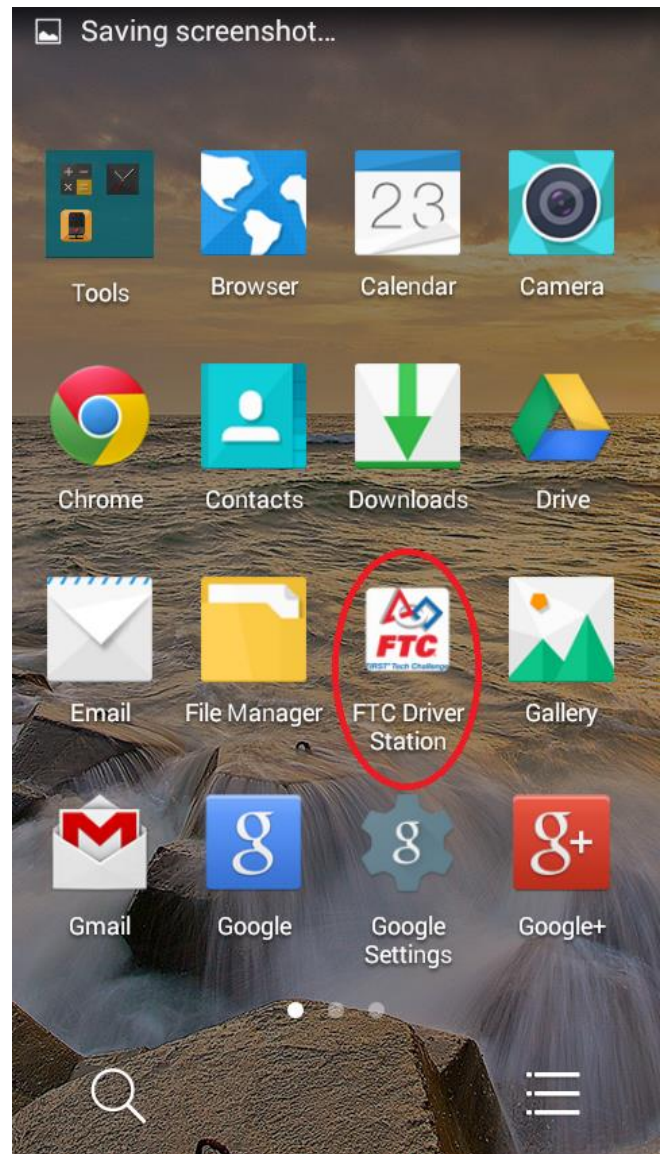
Pairing the Driver Station to the Robot Controller

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

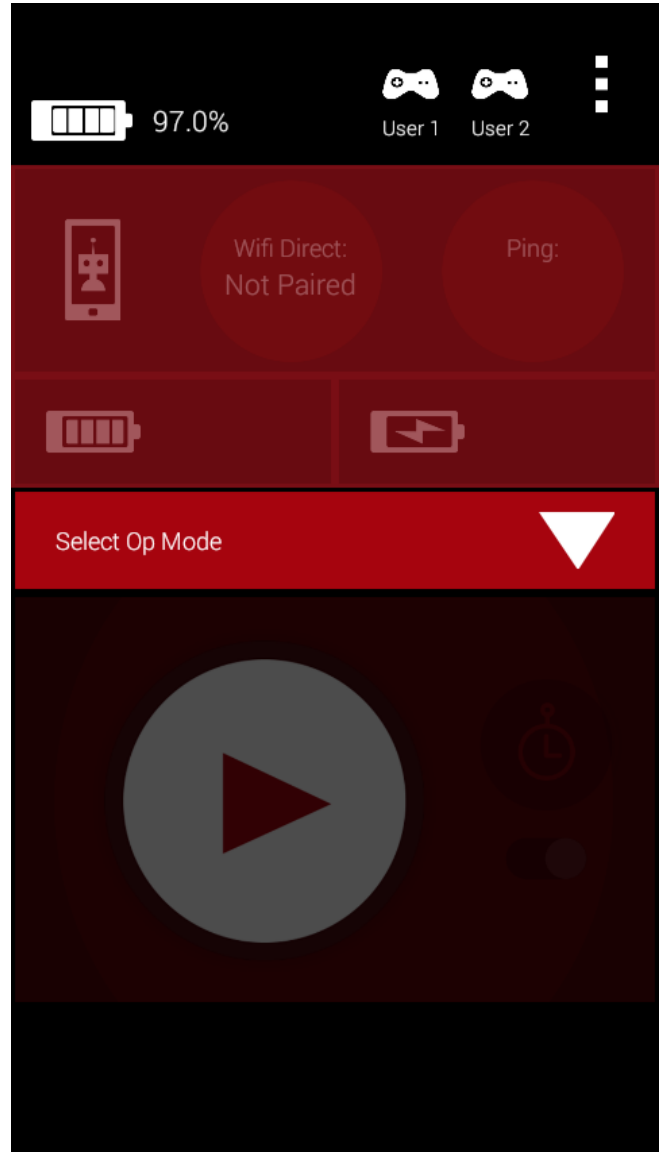
Step 1: Power both cell phones on. Open the app drawer if the FTC Driver Station icon is not present on the main screen.



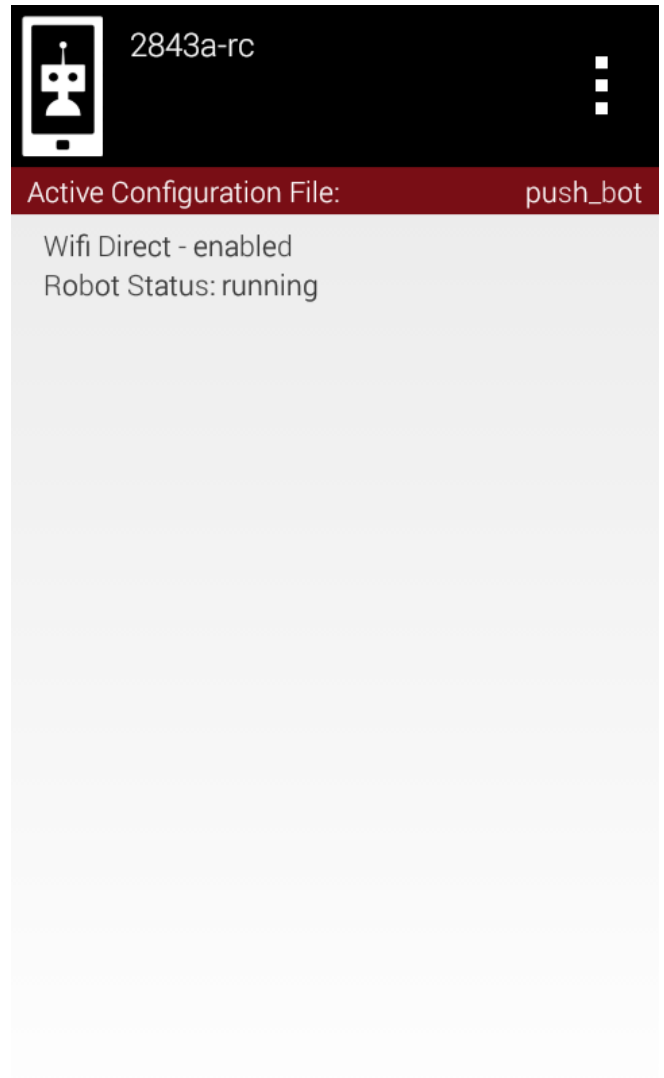
Step 2: Start the Driver Station app on the Driver Station cell phone.



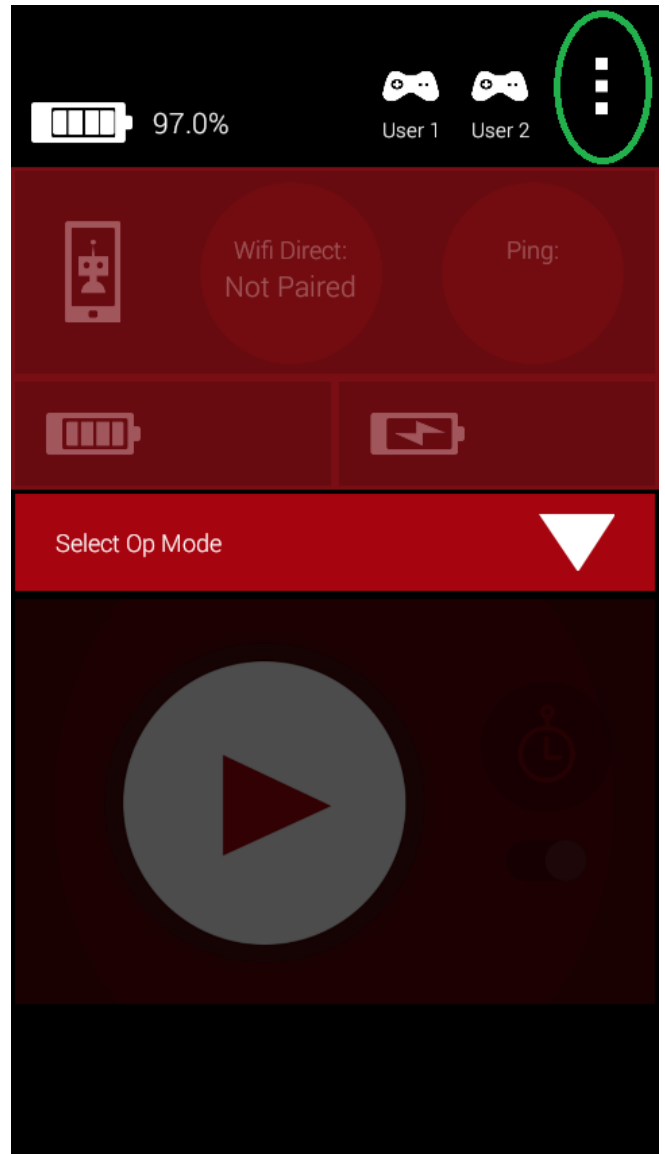
Step 3: The screen on the right should be displayed on the Driver Station cell phone.



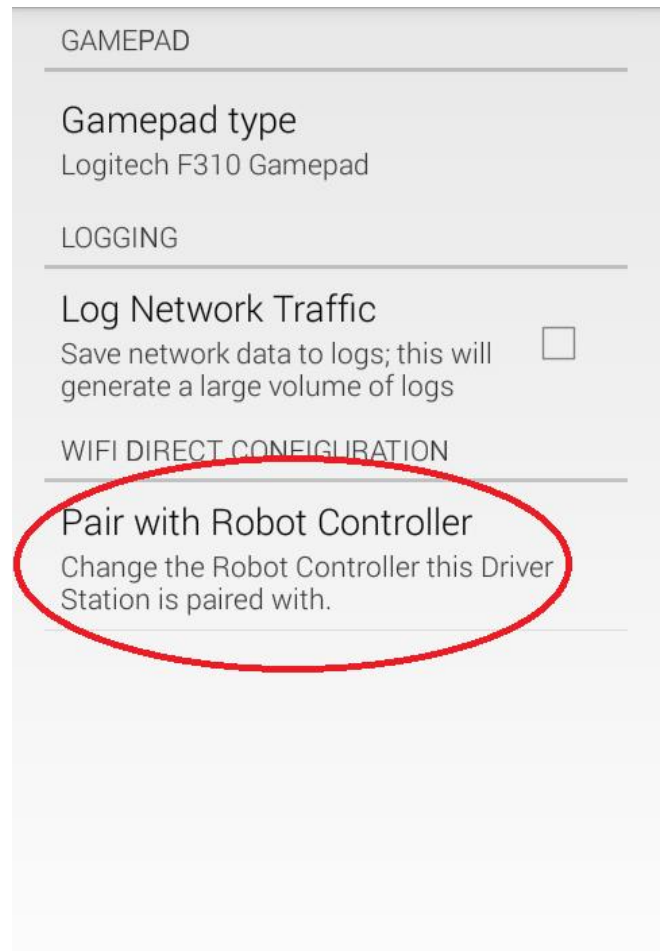
Step 4: Make sure the Robot Controller app is running on the Robot Controller cell phone.



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.

Make sure that the FTC Robot Controller app is open on your other device, and that both devices have wifi enabled

Wifi Direct Devices:

- None
- Do not pair with any device

- 2843a-rc
01:00:00:00:00:00

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

GAMEPAD

Gamepad type

Logitech F310 Gamepad

LOGGING

Log Network Traffic

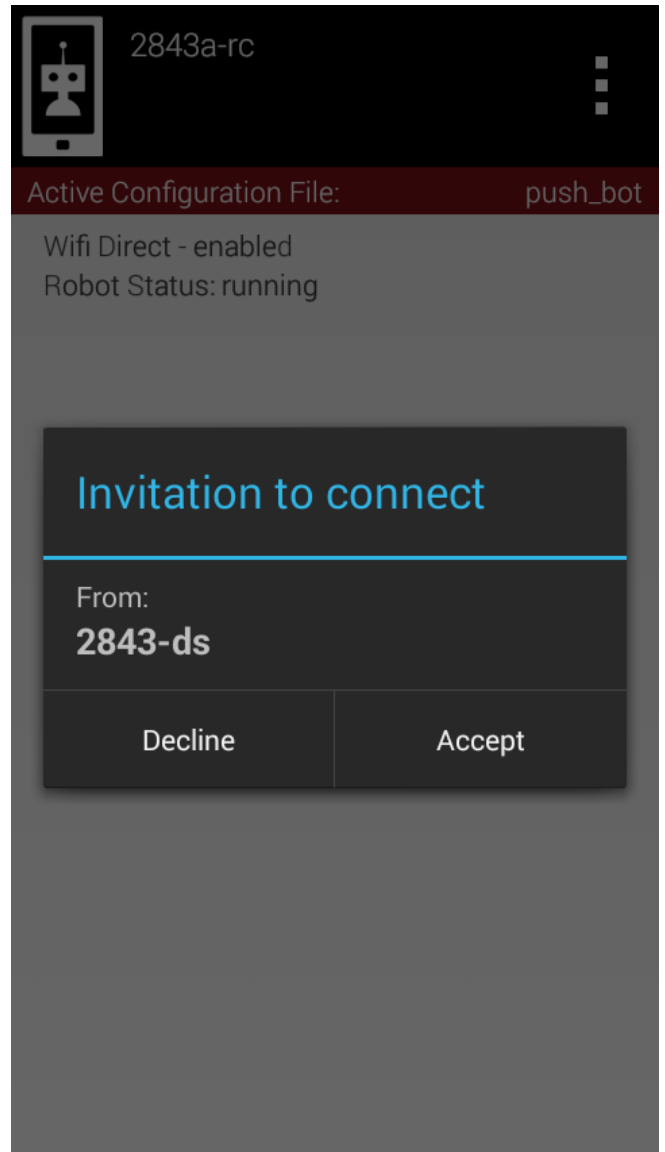
Save network data to logs; this will generate a large volume of logs

WIFI DIRECT CONFIGURATION

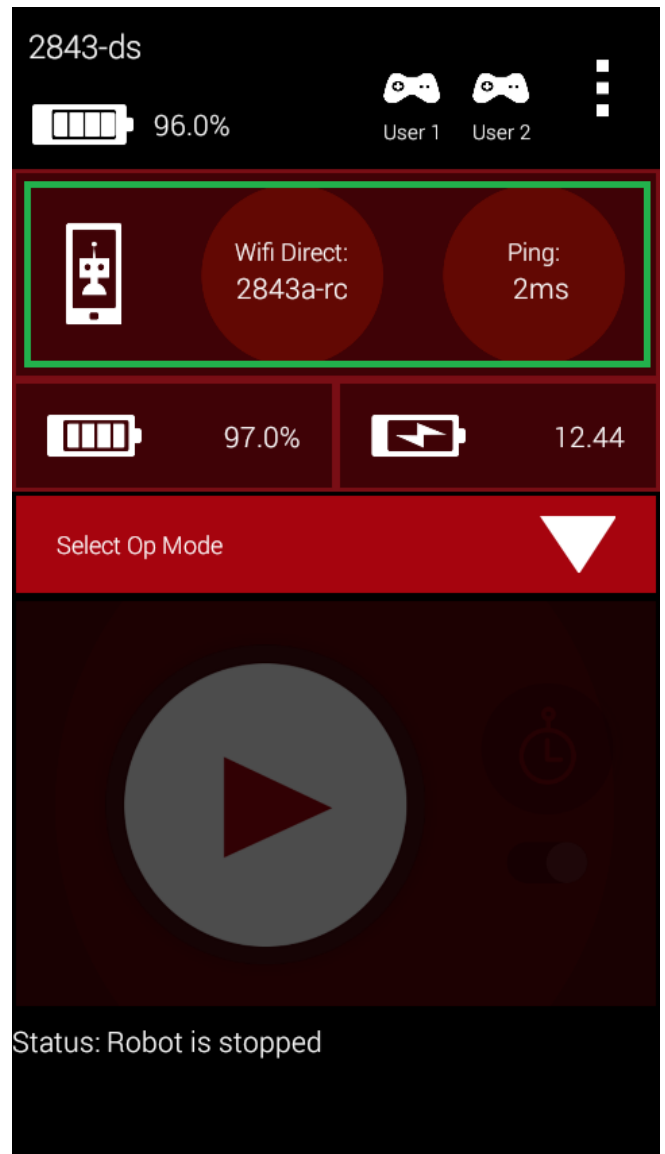
Pair with Robot Controller

Change the Robot Controller this Driver Station is paired with.

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'.



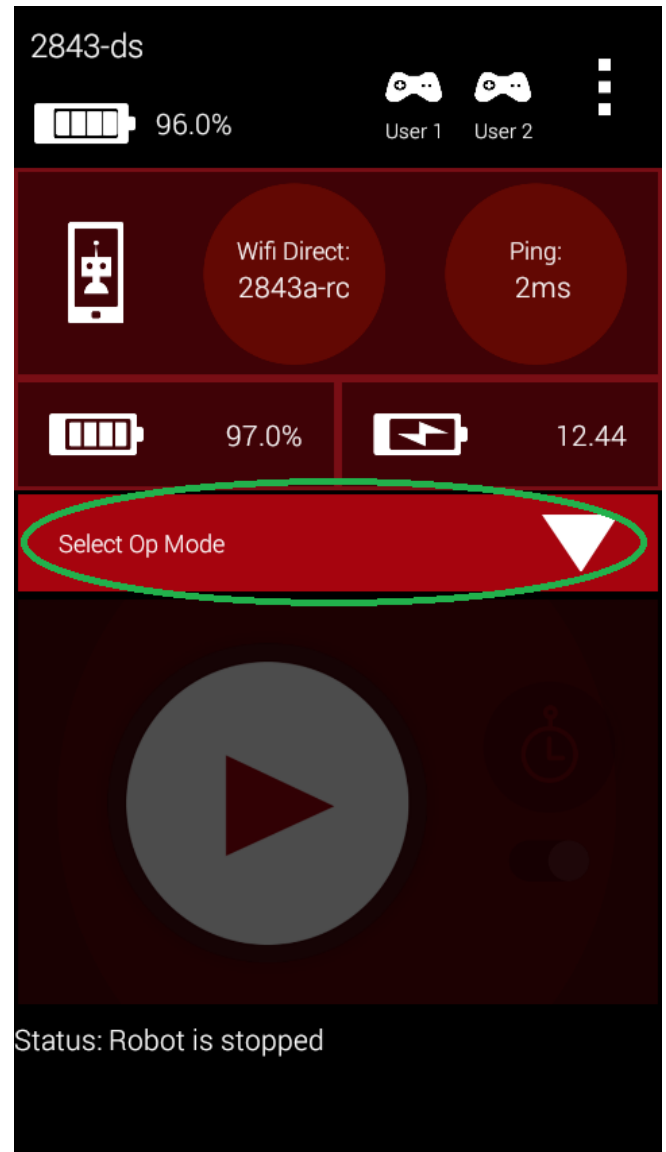
Step 9: Observe that some of the controls on the display have changed indicating that the Driver Station is now connected to a Robot Controller (green rectangle).



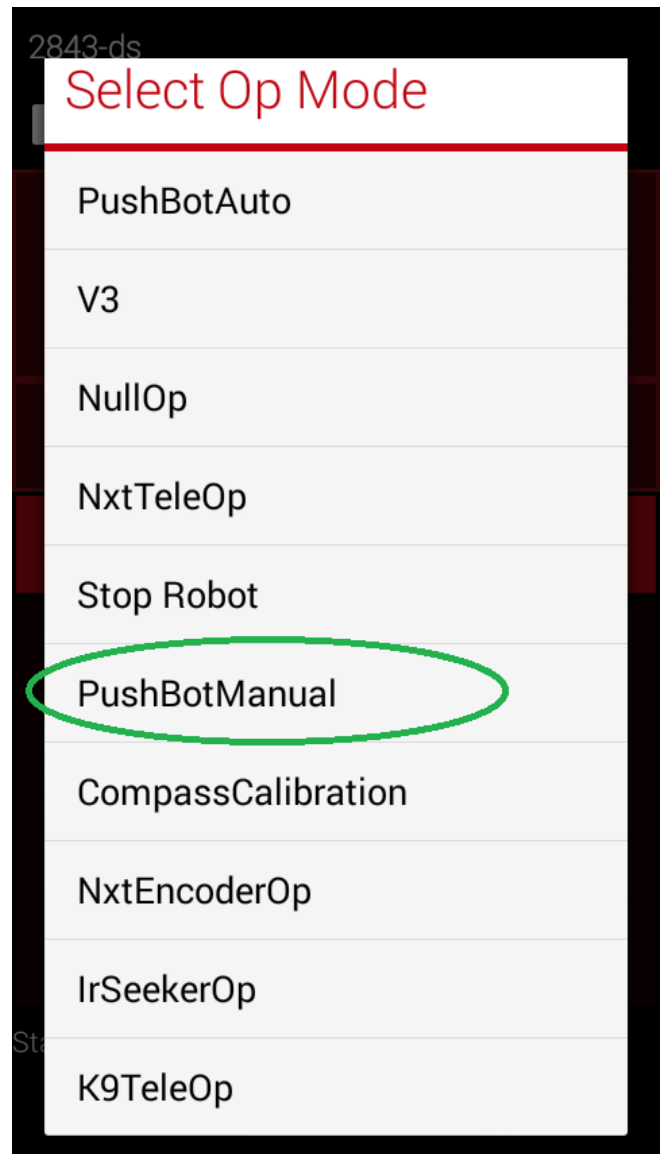
Determining the drive wheel motor controller

This section shows how to determine which motor controller will be used to power the drive wheels and which will be used to power the arm motor. Both the Robot Controller and Driver Station need to be running for this section.

Step 1: On the Driver Station cell phone start the Driver Station app. Push the 'Select Op Mode' button.

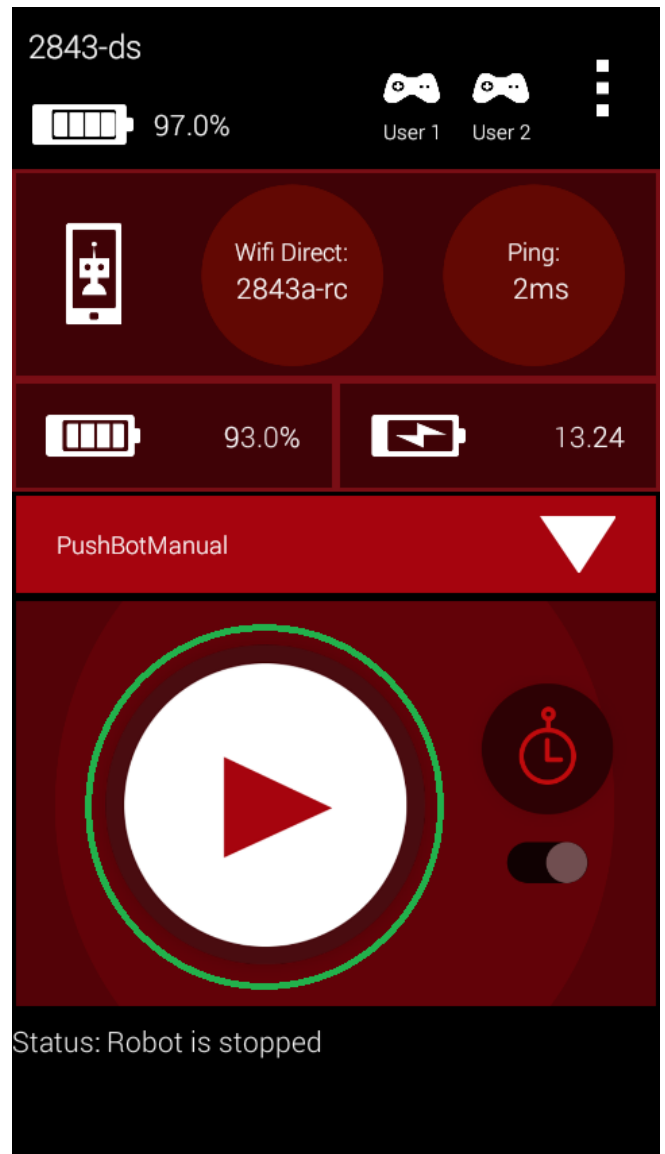


Step 2: The list of available Op modes are displayed.
Select 'PushBotManual'.



Step 3: Push the 'Start' button.

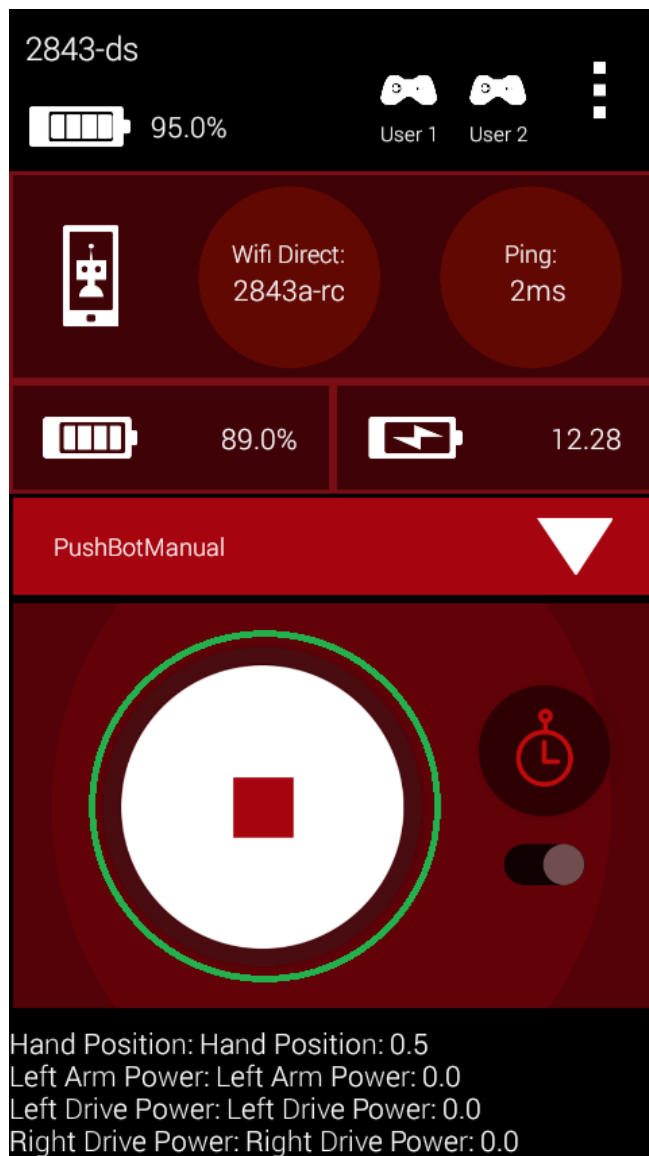
Step 4: Plug a DC motor into port 1 of a motor controller; it doesn't matter which one. Hold the motor so it doesn't roll while operating the joysticks. Push the left joystick of gamepad one up or down. If the motor moves, mark the motor controller as the drive motor controller (ex, tape with 'Drive Controller' written on it). If it doesn't move, push the left joystick of gamepad two up or down. If the motor moves, mark the motor controller as the arm motor controller (ex, tape with 'Arm Controller' written on it).



Step 7: Notice that the telemetry data being sent from the the Robot Controller app to the Driver Station app (lower portion of the display). On the Driver Station, press 'Stop'.

Step 8: Turn the core power distribution module off.

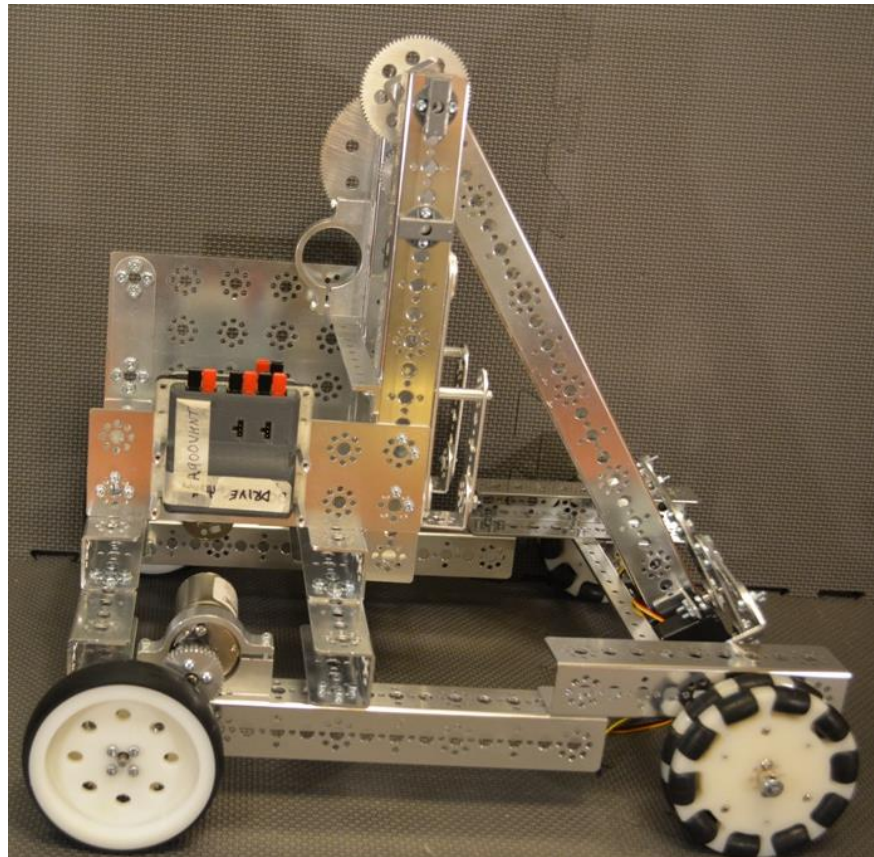
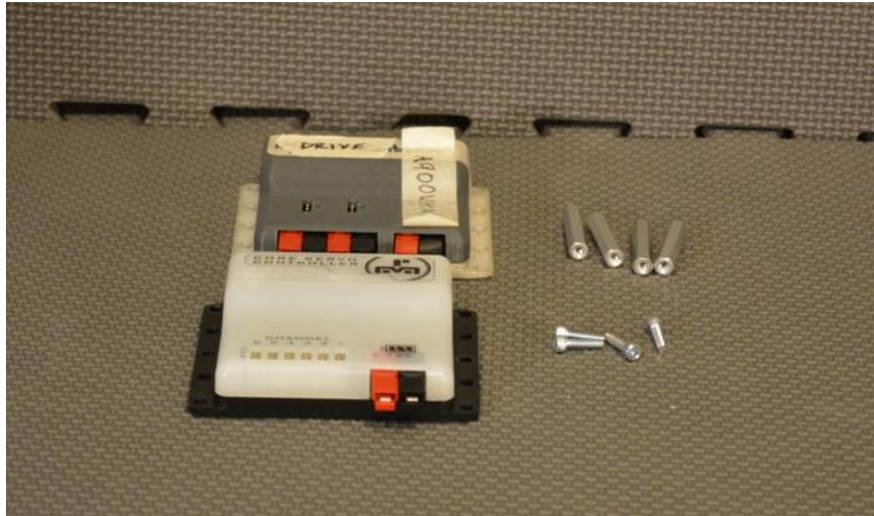
Step 9: Exit the Driver Station and Robot Controller applications by using the over flow settings (the three dots in the top right corner).

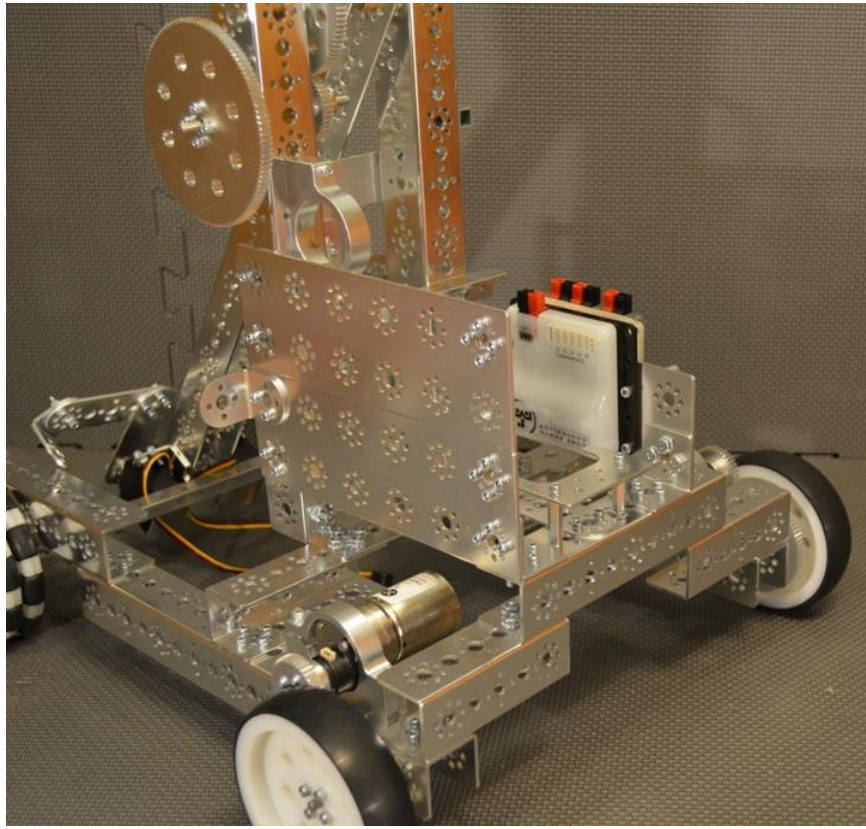


Wiring the Robot

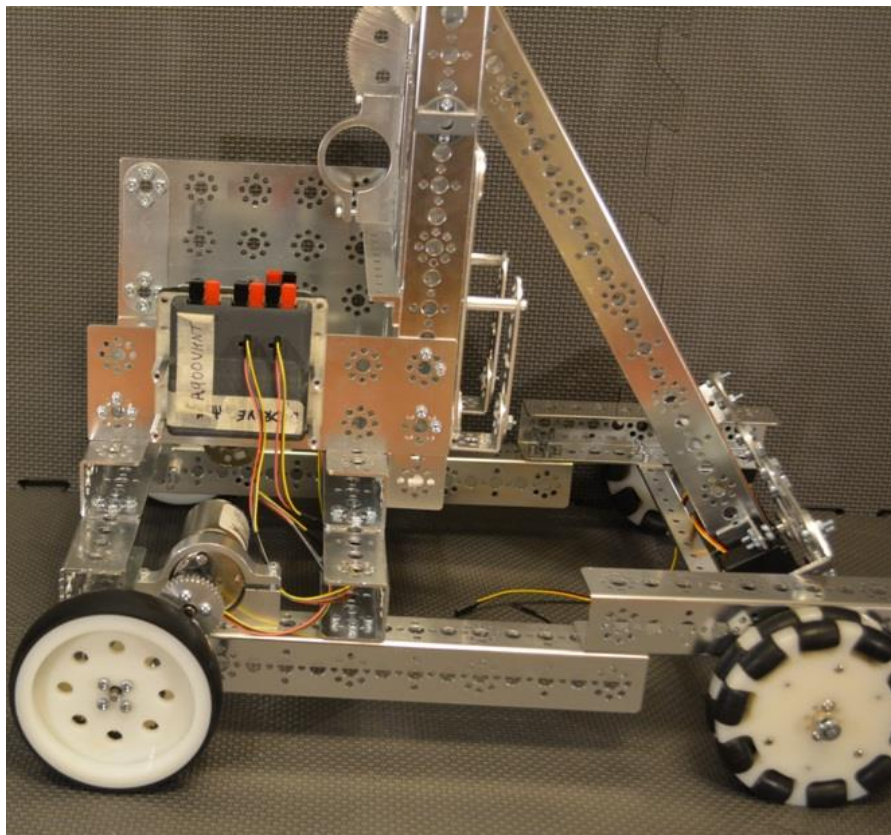
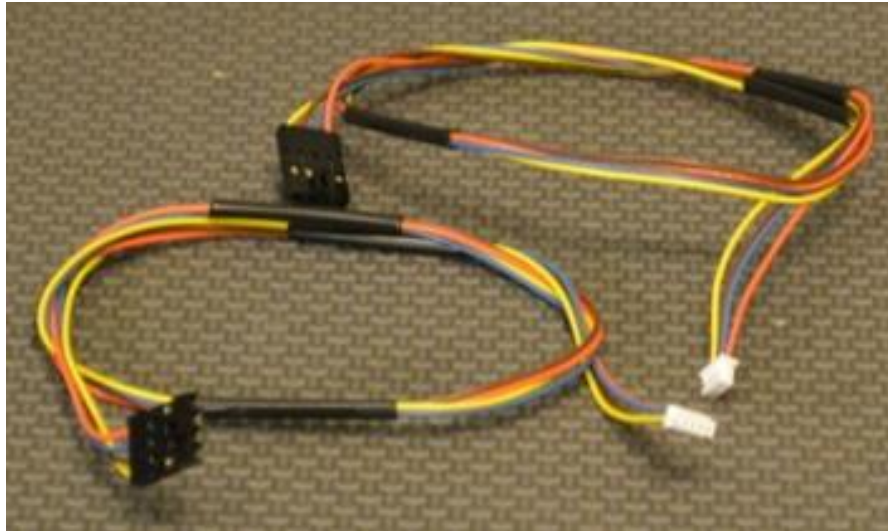
This section will outline the installation of the electronics on the Robot. It will also include finishing the last motor and gear assembly and installing it on the Robot.

Step 1: Core motor controller for the drive (1), core servo controller (1), 32mm stand-off (4), and 1/2" socket head cap screw (4). The second and third photos show the parts installed.

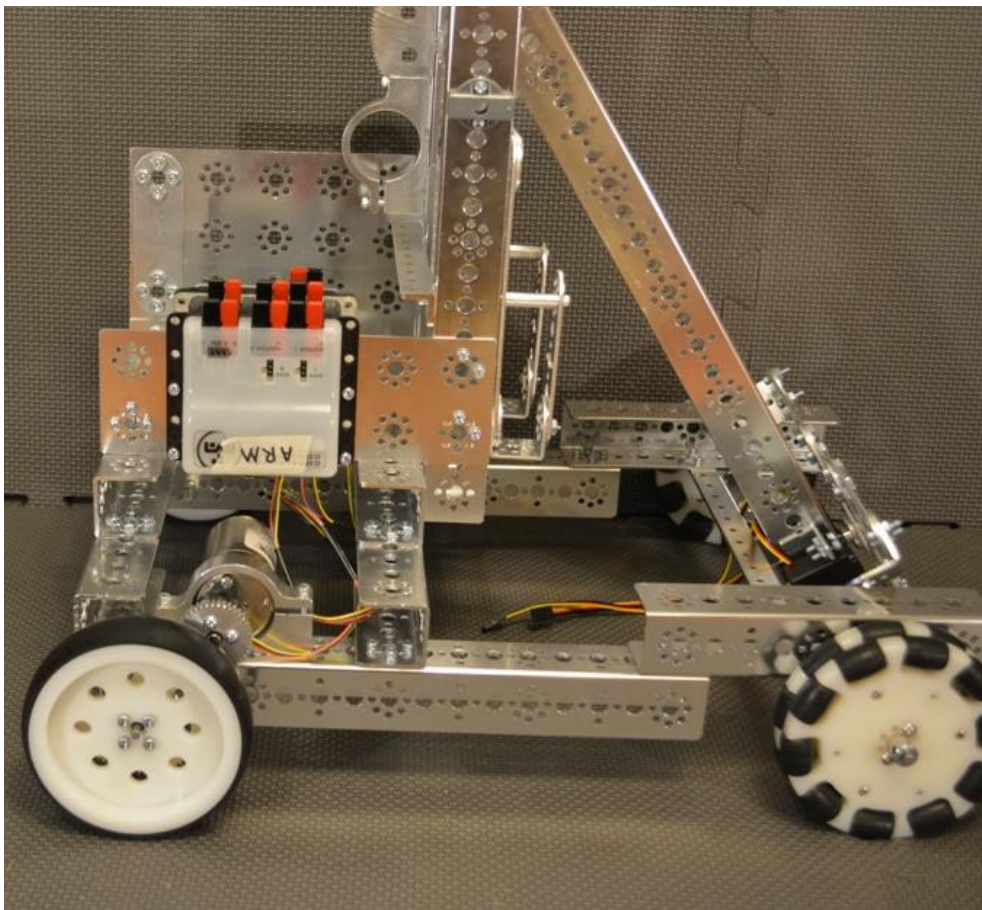




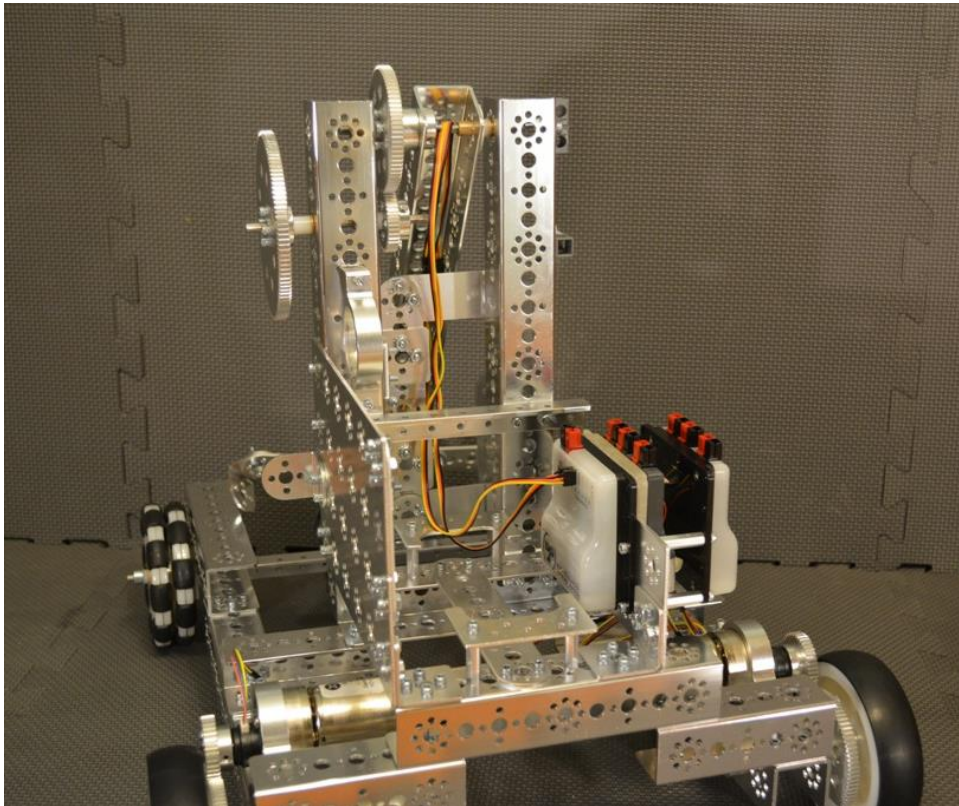
Step 2: Motor encoder wires (2). Plug the end of one wire into the encoder port for motor port 1. Plug the other end into the encoder on the left drive motor. Plug the end of the other wire into the encoder port for motor port 2. Plug the other end into the encoder on the right drive motor. The second photo shows the wires installed.



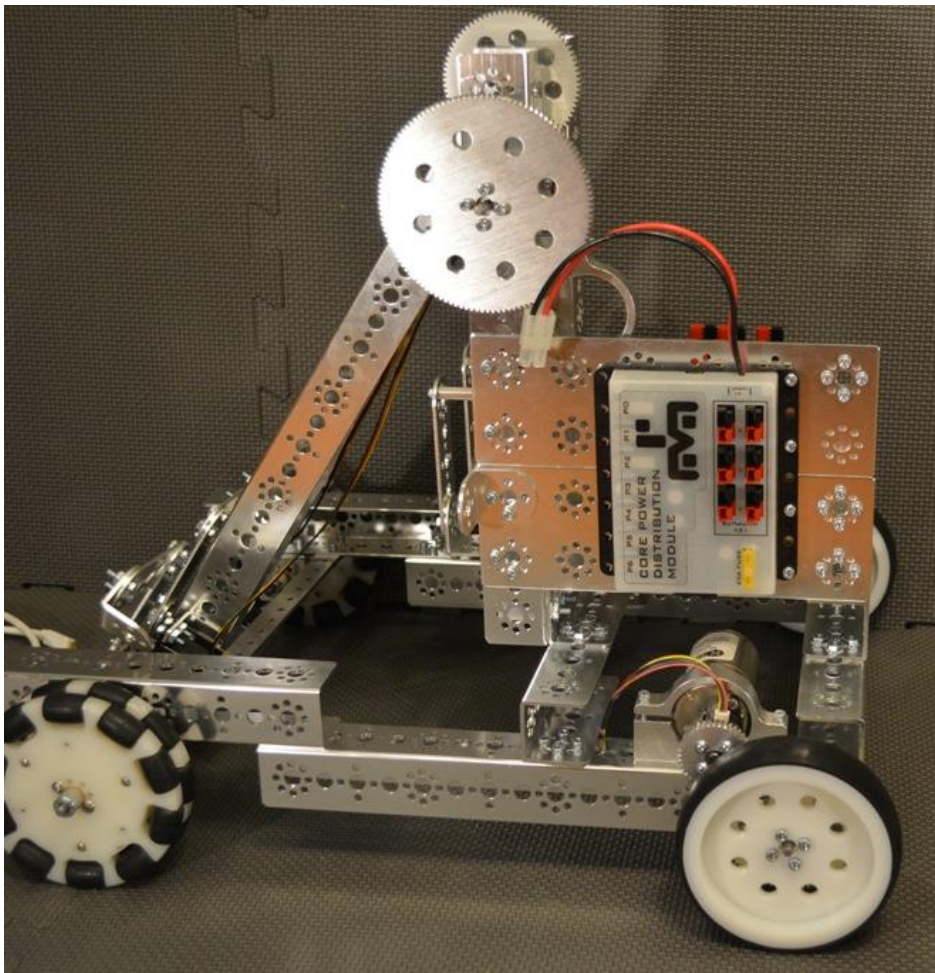
Step 3: Core motor controller for the arm (1) and 1/2" socket head cap screw (4). The second photo shows the parts installed.



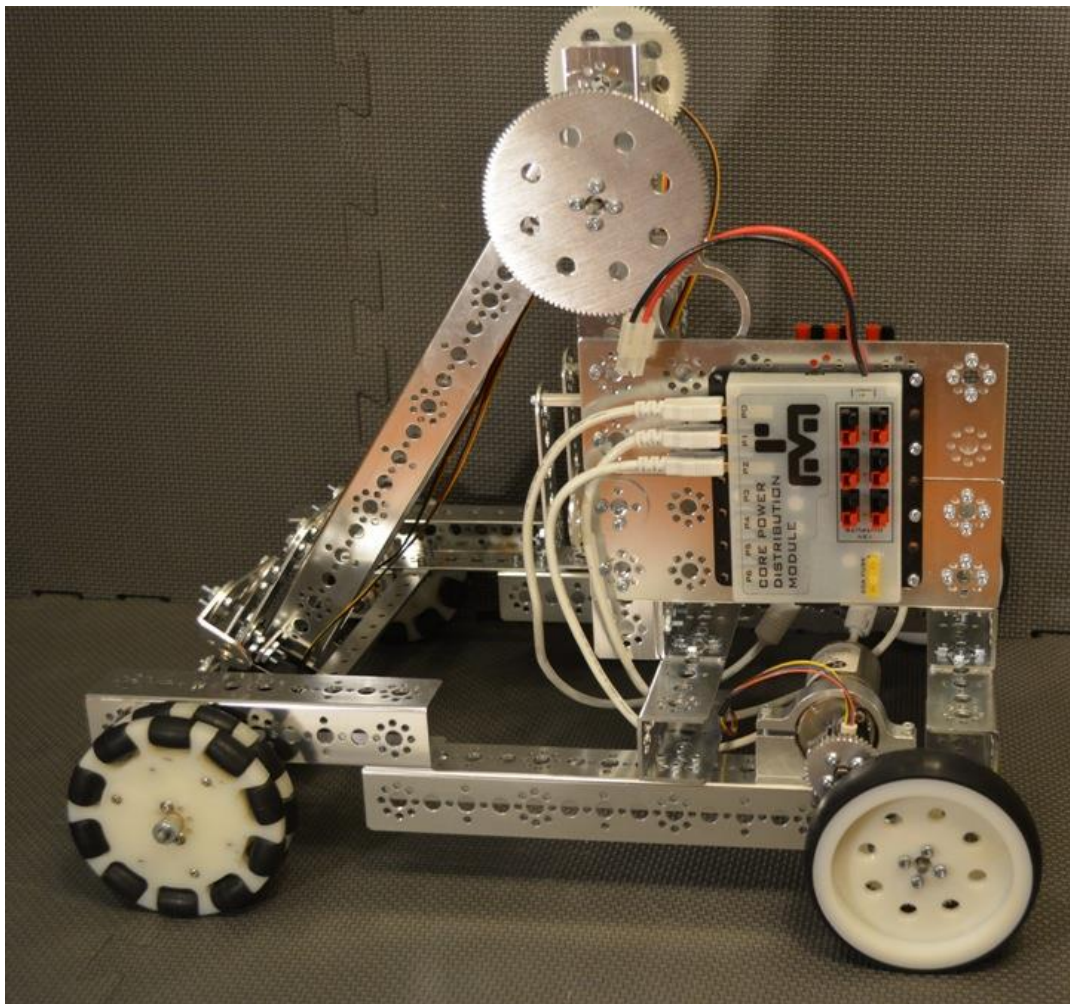
Step 4: Servo extension wire (2). Make sure that when you connect these to the servo wires that you match the colors. Also observe the orientation when installing in the core servo controller (black - b, red - r, yellow - w). The wires will be less likely to get damaged if they are run down the center of the arm channel and over the axel. The second photo shows the wires installed.



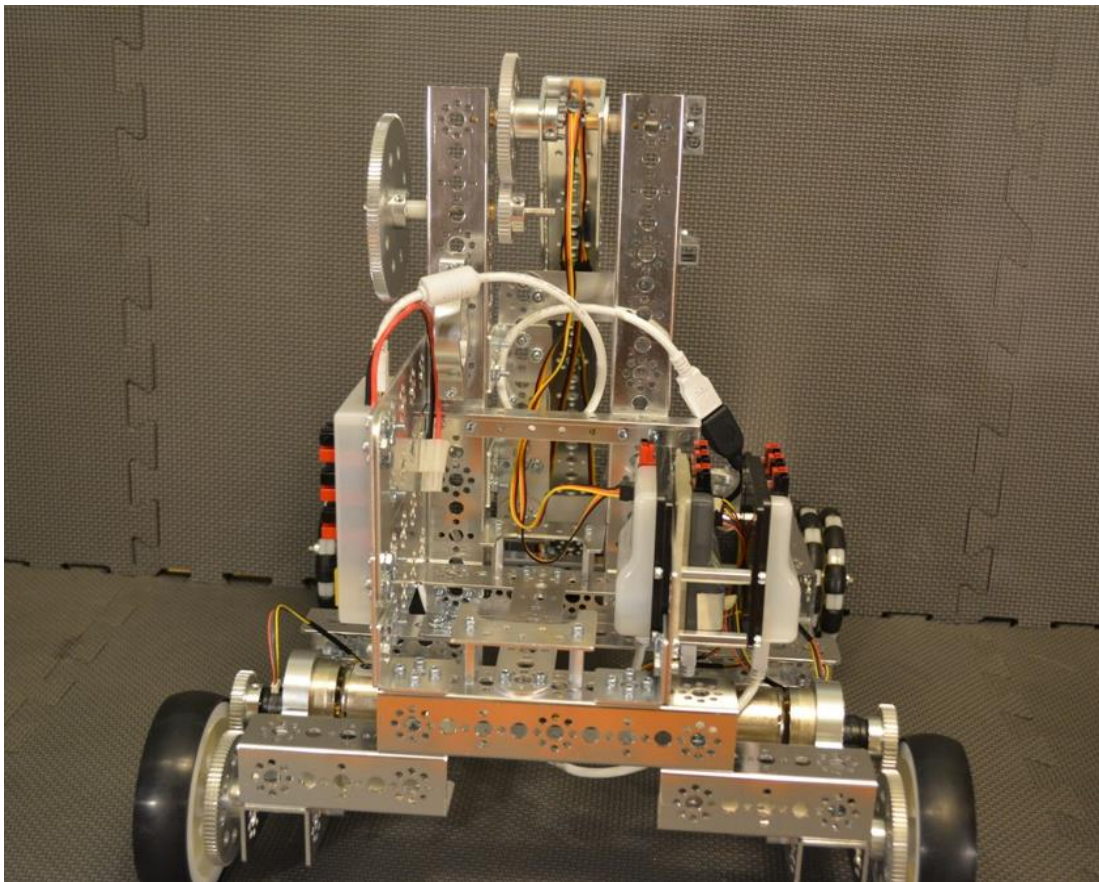
Step 5: Core power distribution module (1), 1/2" socket head cap screw (4), and keps nut (4). The second photo shows the parts installed.



Step 6: USB A male to mini-B USB cable (3). Connect one wire each between the core power distribution module and each of the two core motor controllers and the core servo controller. The second photo shows the cables installed.



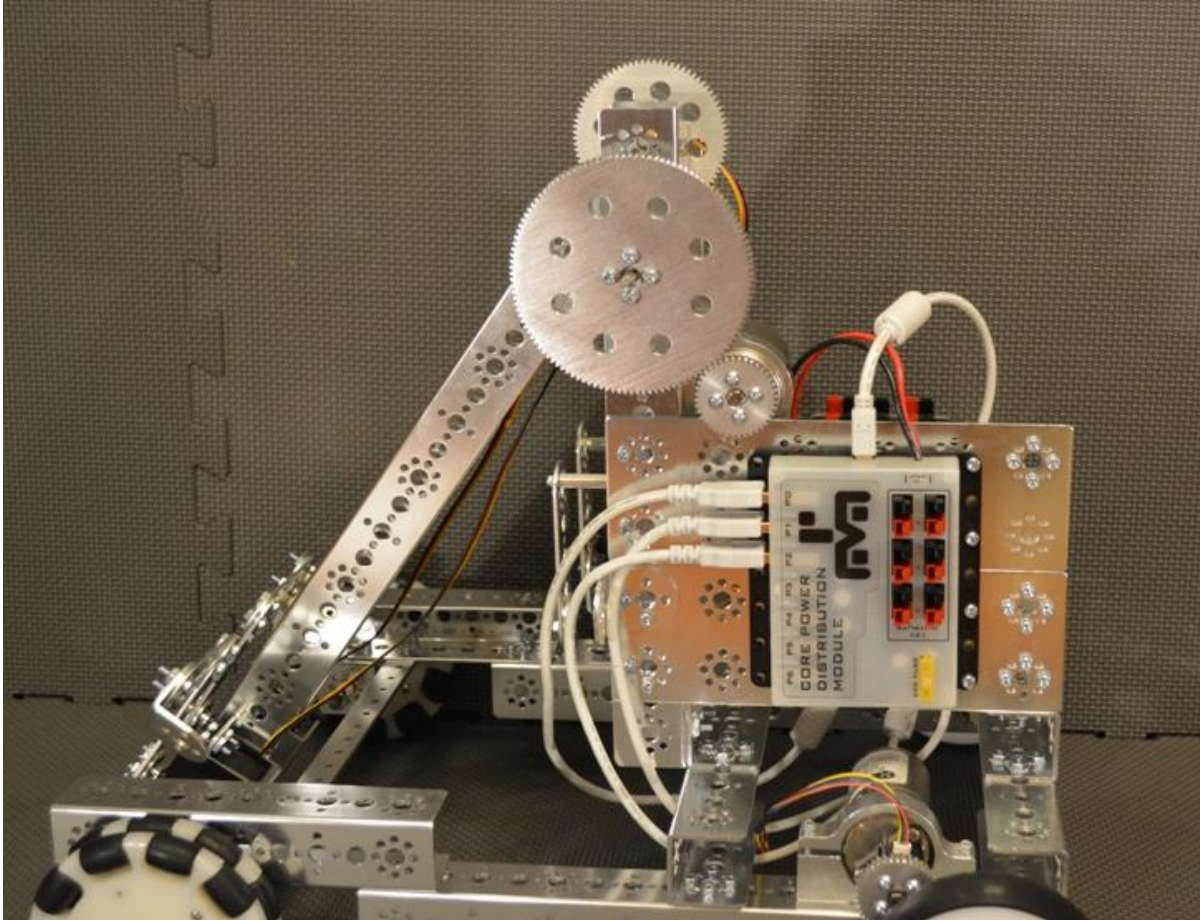
Step 7: USB OTG cable (1), USB A male to mini-B USB cable (1). Connect the min-B end of the cable to the core power distribution module. Connect the other end to the USB OTG cable. Route the other end of the USB OTG cable to the right side of the phone holder assembly. The second photo shows the cables installed.



Step 8: Motor hub and gear assembly (1) (made in the Robot construction section, TETRIX DC gear motor (1). The second photo shows the parts assembled.

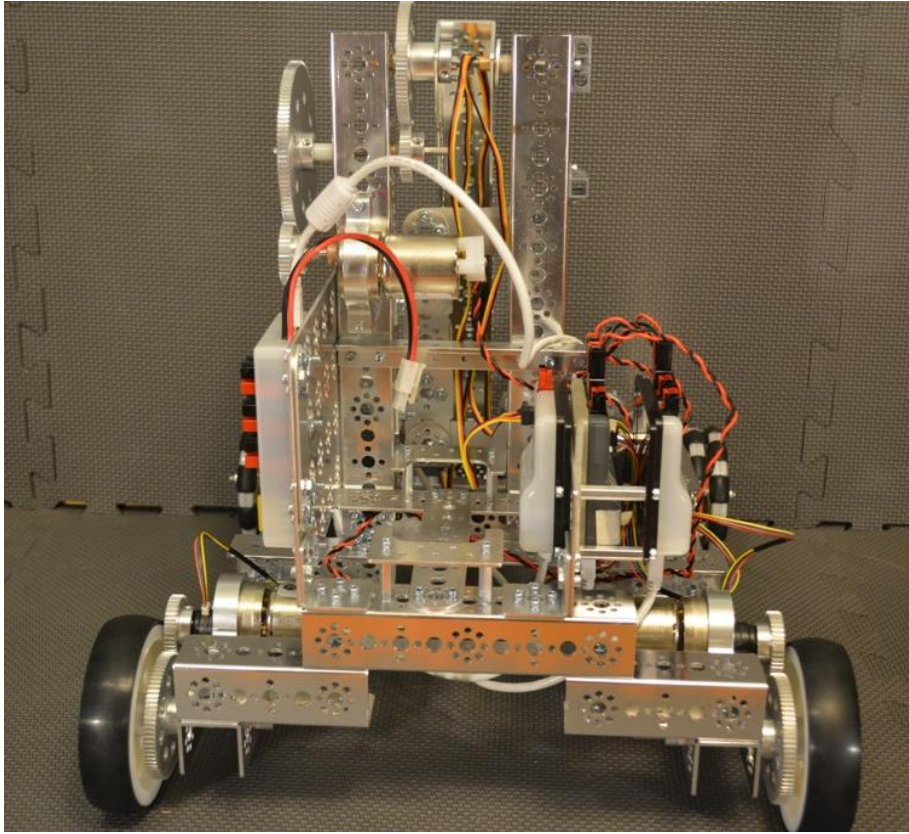


Step 9: Install the motor in the motor mount on the Robot tower. Rotate the motor in the mount to mesh the gears. Do not mesh the gear too tight or the motor will bind. Do not mesh the gears too loose or the teeth will slip.

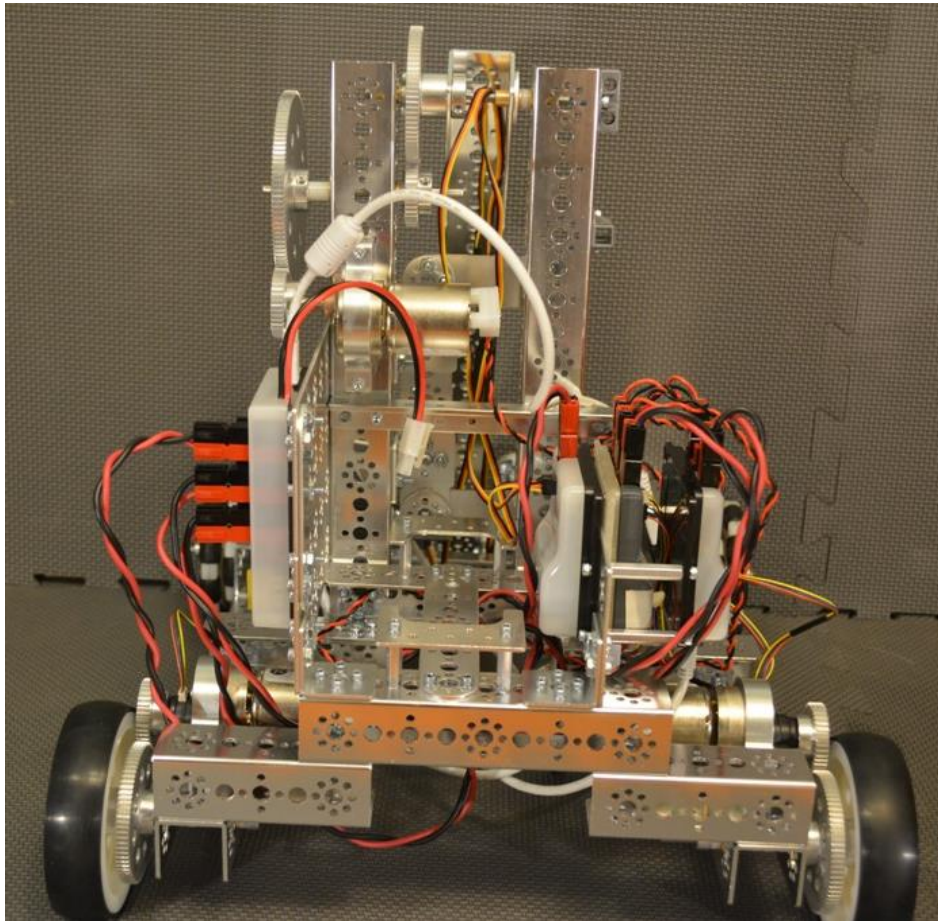


Step 10: DC motor power wire (3). Connect the motor from the left drive to motor port 1 on the drive motor controller. Connect the motor from the right drive to motor port 2 on the drive motor controller. Connect the motor on the arm to motor port 1 on the arm motor controller. The second photo shows the wires installed.

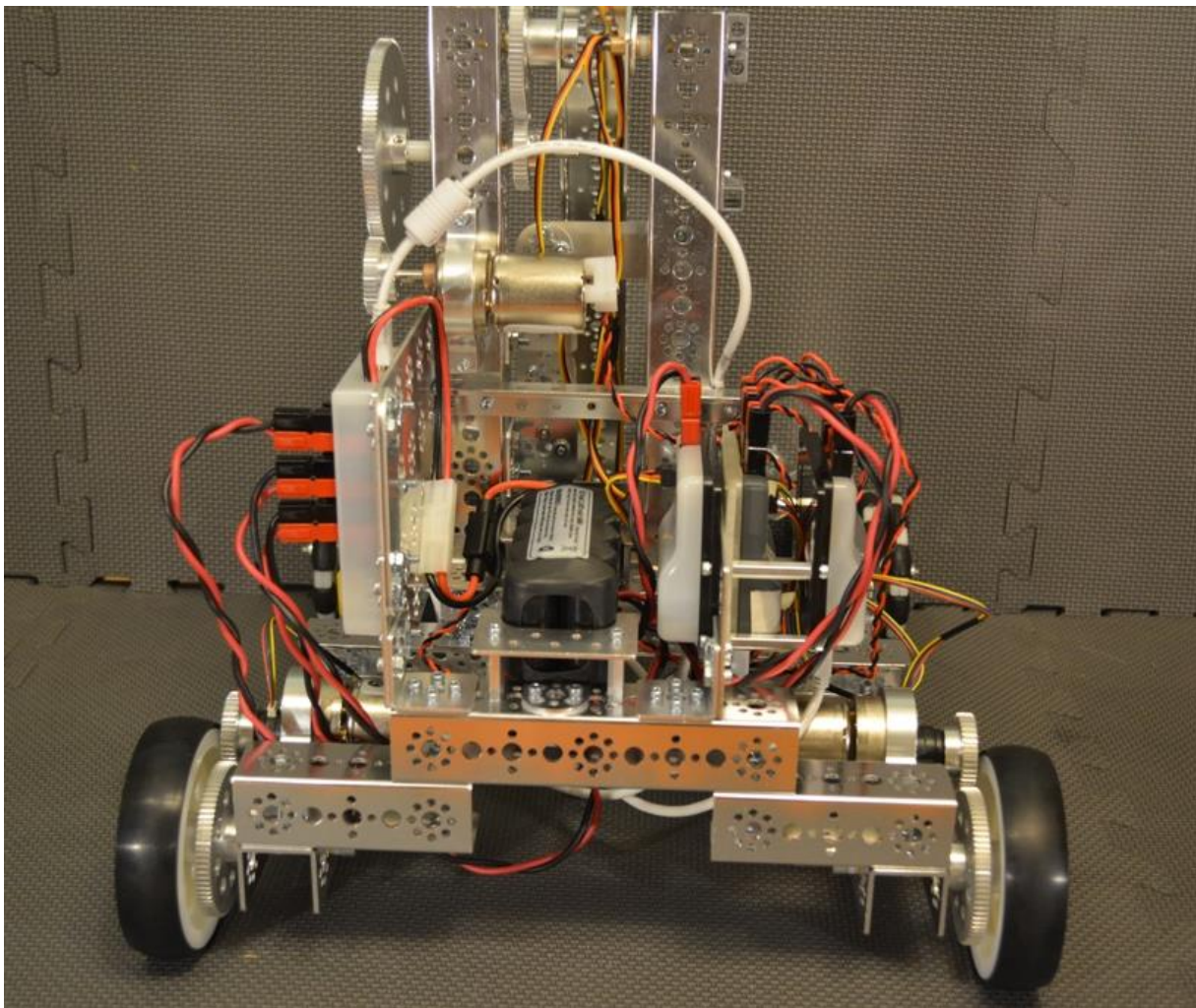




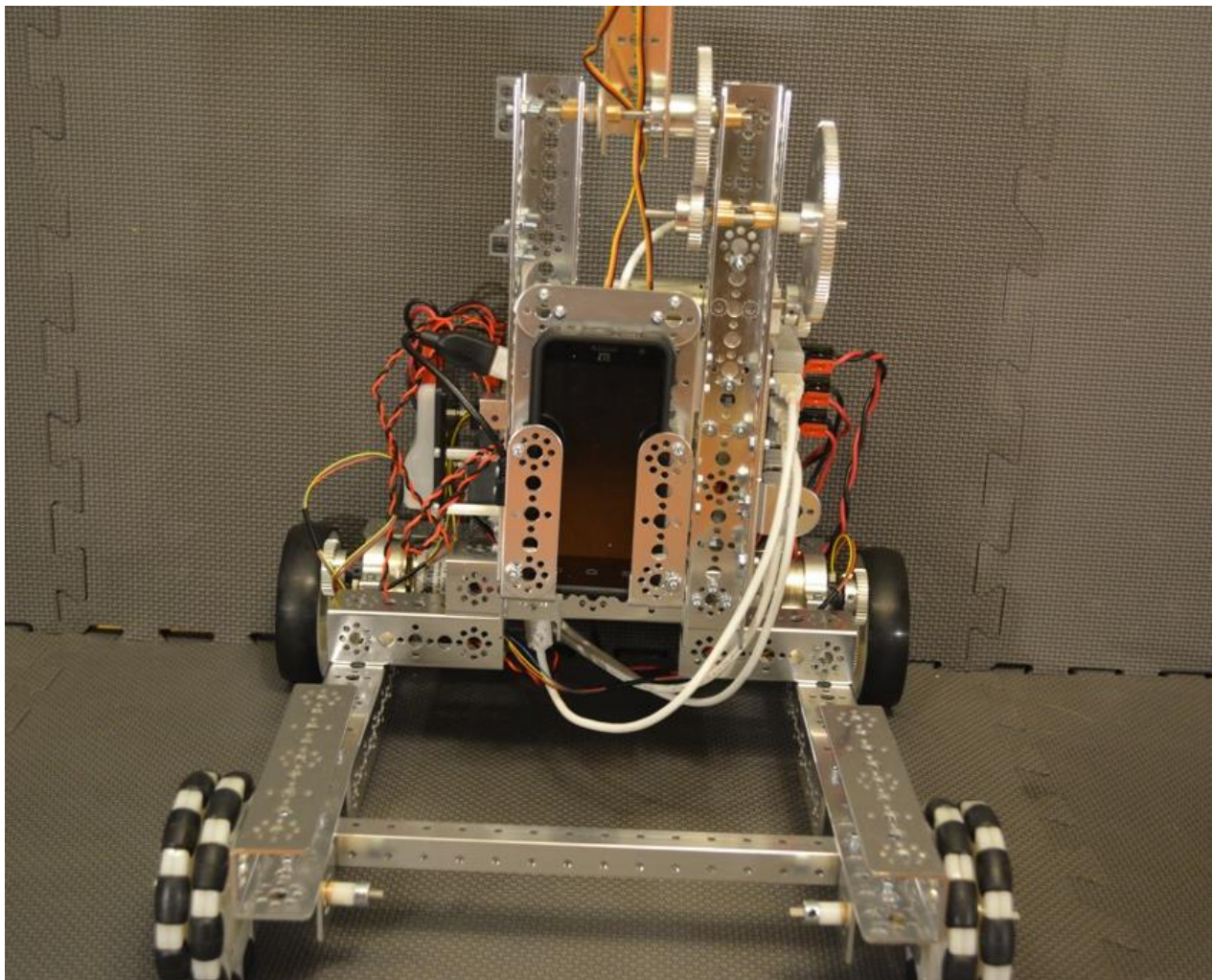
Step 11: DC power wire (3). Connect one end of each DC power wire to any port on the core power distribution module. Connect the other end to the power input port on the core servo controller and the core motor controllers. The second photo shows the wire installed.



Step 12 TETRIX 12V battery (1). Install the batter between the battery clips and connect to the core power distribution module. The second photo shows the battery installed. It would be a good idea to use a zip tie or some other mechanism to hold the battery in place during competition.



Step 13: ZTE speed phone (1) configured as Robot controller. The second photo shows the phone installed (the arm is moved out of the way to show installation).



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 is powered on, the cell phones are powered on and the FTC apps are running.

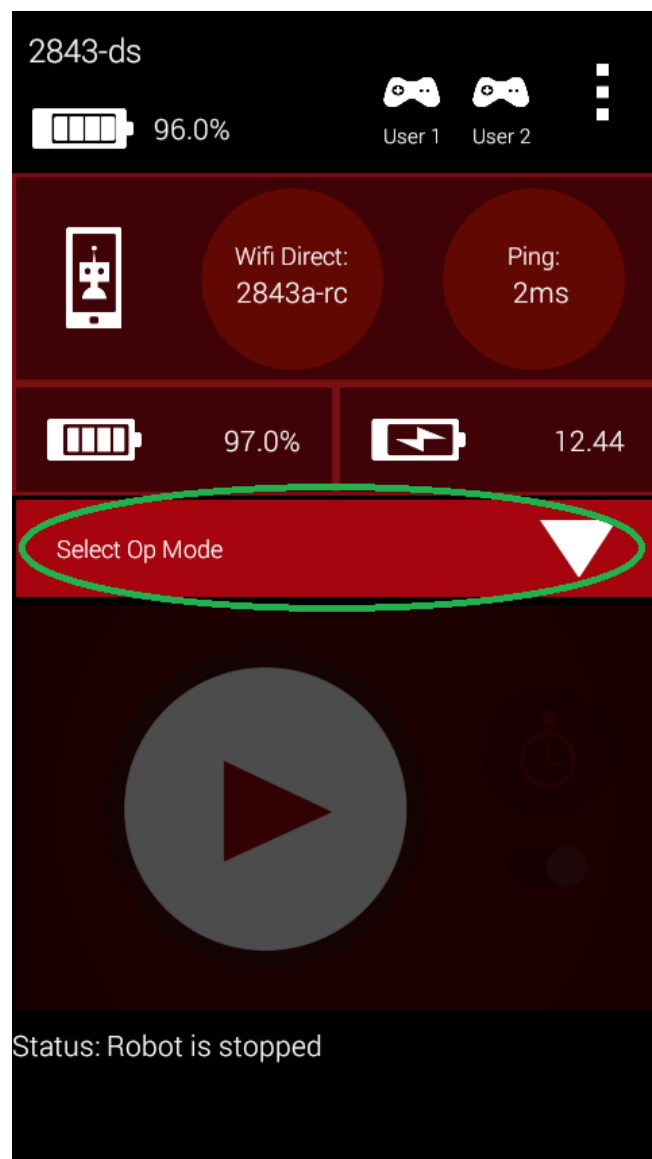
Running an Op mode

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 FTC 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 left hand corner of the Driver Station screen. In the image above the ping statistics reads “4 ms”. This means it took 4 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.

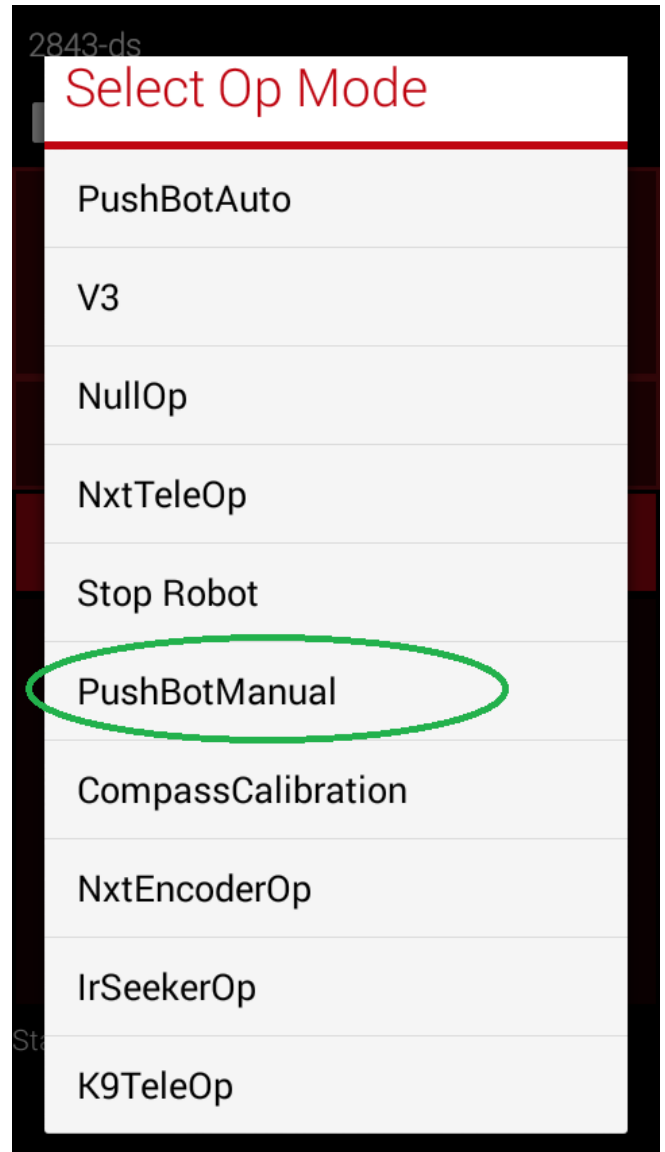
Step 2: Touch the **Select Op Mode** button to display a list of the available Op Modes sent to the Driver Station app from the Robot Controller app. Select “PushBotManual” from the list and the round play button should become brighter.



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 FTC 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.

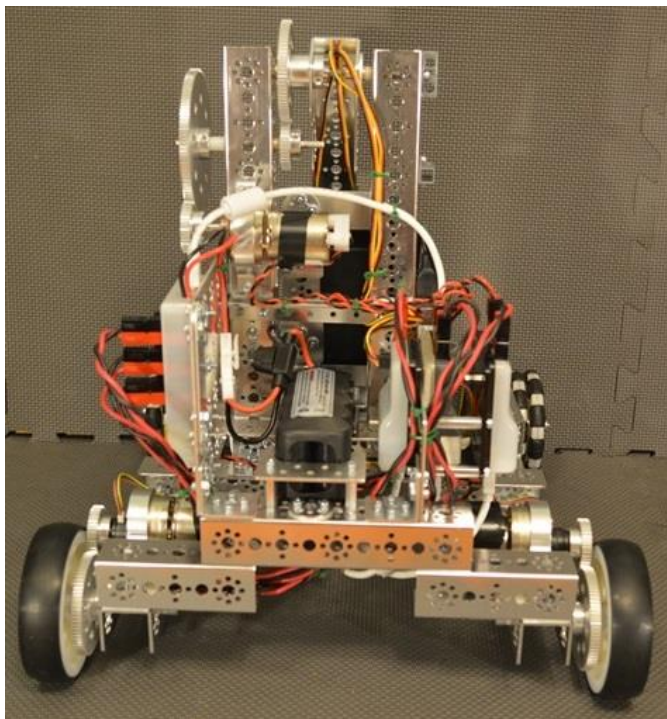
If the **Start** button is pressed, then the Op mode will run indefinitely until the program terminates or until the **Stop** button is pressed. Press the start button and proceed to the next section.



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 left joystick controls the 'left' drive motor and the right joystick controls the 'right' drive motor. The image above 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 33" above the ground. The arm uses a 1:6 gear-ratio to enable the arm to lift moderately heavy items. Push the left joystick away from the body to raise the arm; toward the body to lower the arm.

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.



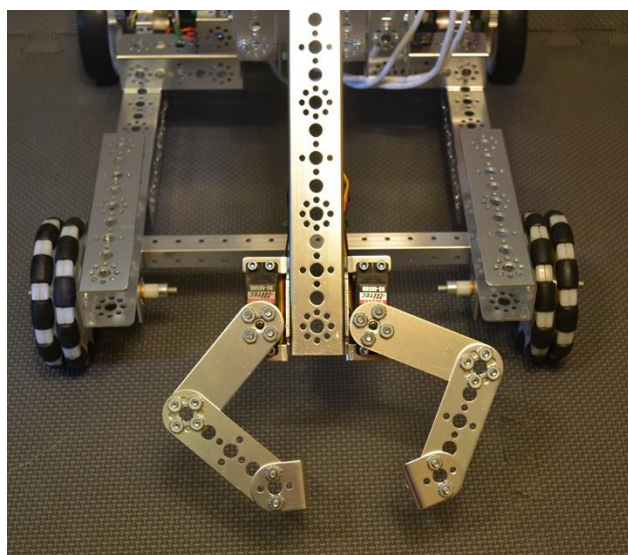
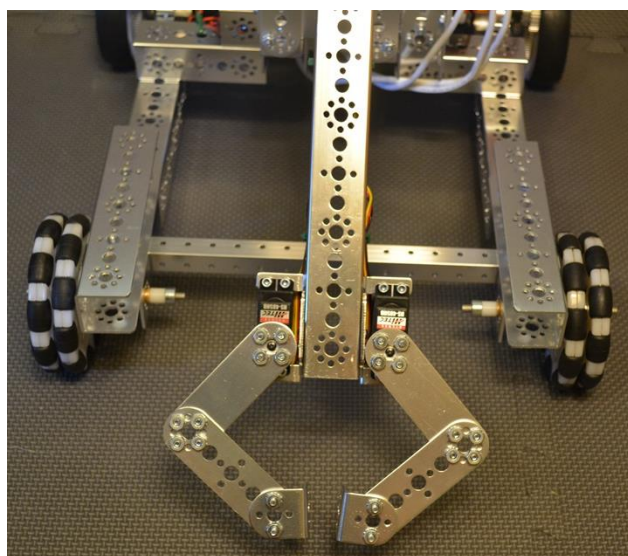
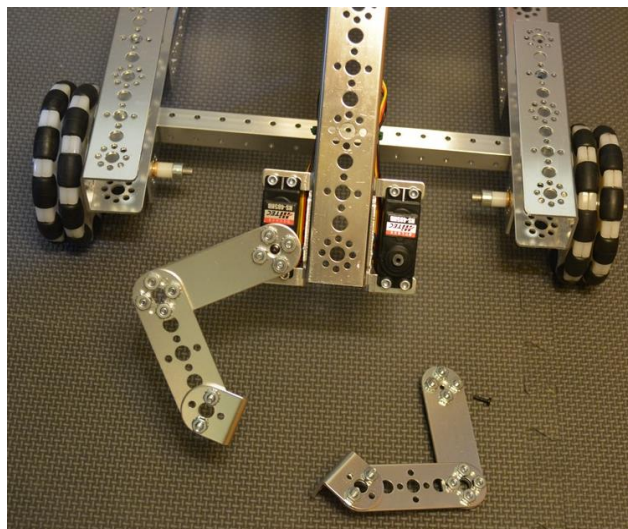
Adjusting the Servos

If the servos don't 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.



Repeat step 1-3 for the other servo.

Final steps

- 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.
- 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!

***FIRST*[®] Tech Challenge PushBot Build Guide**

Appendices

Appendix A: Bill of Materials 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
12	SCREW_SHCS_6-32_X_1_25_2011	Gear Hub Screw
1	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
2	TETRIX_738009_BATPACK_BRKT_2013	Battery Clip
4	TETRIX_739028_2012	40-Tooth Gear
2	TETRIX_739055_2012	4" Wheel
1	TETRIX_739057_2012	Battery Pack
2	TETRIX_739060_2012	Single Standard Scale Servo Bracket
5	TETRIX_739061_2012	Flat Bracket
3	TETRIX_739062_2012	L Bracket
2	TETRIX_739065_2012	32mm Channel
4	TETRIX_739066_2012	96mm Channel
4	TETRIX_739067_2012	160mm Channel
4	TETRIX_739068_2012	288mm Channel
1	TETRIX_739069_2012	416mm Channel
1	TETRIX_739071_2012	288mm Angle
1	TETRIX_739072_2012	144mm Angle
3	TETRIX_739073_2012	Flat Building Plage
3	TETRIX_739079_2012	Motor Hub
3	TETRIX_739083_2012	TETRIX DC Motor
3	TETRIX_739086_2012	80-Tooth Gear
6	TETRIX_739088_2013	Axel
3	TETRIX_739089_2012	DC Motor Mount
3	TETRIX_739090_2012	Gear Hub Spacer
17	TETRIX_739091_2013	Bronze Bushing
6	TETRIX_739092_2012	Axel Collar
162	TETRIX_739094_2013	Keyp Nut
64	TETRIX_739097_2013	1/2" Socket Head Cap Screw
86	TETRIX_739098_2012	5/16" Socket Head Cap Screw
11	TETRIX_739100_2012	1/8" Axel Spacer
3	TETRIX_739101_2012	3/8" Axel Spacer
6	TETRIX_739102_2012	1" Stand-off
4	TETRIX_739107_2013	32mm Stand-off
42	TETRIX_739111_2012	Button Head Cap Screw
2	TETRIX_739120_2012	Hard Point Connector
5	TETRIX_739172_2012	Axel Hub
6	TETRIX_739195_2012	Motor Mount Screw
2	TETRIX_739197_2012	Standard-Scale HS-485HB Servo Motor
2	TETRIX_739270_2013	Inside C Connector





3	TETRIX_739272_2013	160mm Flat
4	TETRIX_739273_2013	96mm Flat
4	TETRIX_739281_CORNER_BRA_2013	Inside Corner Bracket
1	TETRIX_739387_2013	2mm Spacer
2	39081	Servo Extension Wire

Appendix B: 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 with the consent of *FIRST*.

Appendix C: Project Team Profiles

	<p>Isa Year 9 student at Great Mills High School. Student for 1 year on Team #2843, Under the Son</p>
	<p>Alanis Year 9 student at Great Mills High School. Student for 1 year on Team #2843, Under the Son</p>
	<p>Nanette Year 7 student at Esperanza Middle School. Student for 1 year on Team #2843, Under the Son</p>
	<p>Laura Year 9 student at Leonardtown High School. Student for 7 years on Team #2843, Under the Son</p>

	<p>Mary Year 12 student at Leonardtown High School. Student for 7 years on Team #2843, Under the Son Volunteer for 7 years, MD FIRST</p>
	<p>Claudia Mentor for 1 year on Team #2843, Under the Son</p>
	<p>Lydean Coach for 7 years on Team #2843, Under the Son Volunteer for 7 years, MD FIRST ssi@the-spanglers.net</p>
	<p>David Mentor for 7 years on Team #2843, Under the Son Volunteer for 7 years, MD FIRST ssi@the-spanglers.net</p>

Appendix D: Resources & Support

FTC Game Manuals – Part I and II - <http://www.usfirst.org/roboticsprograms/ftc/Game>

FIRST Headquarters Support

Phone: 603-666-3906 or toll free 800-871-8326

Email: FTCTeams@usfirst.org

USFIRST.ORG

[FIRST Tech Challenge \(FTC\) Webpage](#) – For everything FTC.

[FTC Season Timeline](#) – Find FTC Events in your area.

[FTC Forum](#) – to ask questions about the annual game challenge and participate in information sharing.

[FIRST Championship Conferences](#) webpage – for archived presentations and materials

FIRST Tech Challenge Social Media

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

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

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

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

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

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

Android-Based Technology Support

[FTC Technology Forum](#)

[FTC Technology Webpage](#)

[FTC Team Resources Webpage](#)

Product Support

FIRST will handle questions about Team registration, grants, Events, and Affiliate Partners. Pitsco will handle questions about ordering, payment, and delivery of competition sets and materials.

- U.S. & Canada: 800-835-0686 FREE
- Outside U.S. and Canada: 620-231-0100
- U.S. Fax: 800-533-8104

Feedback

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