Introduction

Programming robots in Java can be divided into four parts:

- Configuring the Android devices (usually, phones) which will be used on the robot and for the driver station
- Configuring the computer with the FTC software development kit (SDK). The FTC SDK consists of apps which run on the devices and Application Programming Interfaces (APIs) which will be used to write the programs (also known as “op modes”) which give commands to the robot.
- Installing the FTC apps to the devices
- Writing the op modes using the FTC SDK and re-downloading the robot controller app

The configuration tasks may need to be repeated when additional devices or computers are used by a team.

Parts of this document are based on the FTC training manual at https://github.com/ftctechnh/ftc_app/blob/master/doc/tutorial/FTCTraining_Manual.pdf . The training manual contains step-by-step instructions for each of the parts below. It should be read from start to finish because it is the best overview of the FTC SDK and includes a lot of information not shown here. However, the training manual was last updated in August 2015, so some of it is outdated:

- Section 7’s instructions are somewhat outdated; use the instructions at Downloading, Installing, and Configuring the FTC SDK instead.
- You can skim section 7.7.
- the part about registering op modes in sections 9.3.1 and 9.3.2 can be ignored; the procedure in Creating Your First Op Mode should be used instead.

Part I: Configuring Android Devices / Phones

ZTE Speed Phone Setup

The instructions below are somewhat specific to the ZTE Speed; the directions will need to be adjusted for other devices.

1. Start with the phone off. If you are using the phones from the FTC kit, remove the bumper.
2. Remove the back cover by sliding your fingernail into the small slot near the bottom right of the phone, then slide it around to complete the removal.
3. Remove the white SIM card by pressing in the slot (roughly in the middle of the phone) until the SIM card releases. Discard the SIM card.
4. Replace the back cover of the phone.
5. Power on the phone by pressing the Power button on top for a few seconds.
6. The Phone will prompt for activation - click Next three times to get to the WiFi screen
7. Click Next to bypass the WiFi screen.
8. Hands-Free Activation screen will appear and disappear on its own.
9. Will get to Self-Service screen: "Are you swapping this device...?" Click No.
10. Hands-Free Activation screen will reappear. To avoid waiting, press Activate and No again.
11. At the "Make Yourself at Home" screen, press the 9 squares button.
12. You will be at the Home screen. Press the 9 squares button again.
13. Press anywhere at the "View Applications" popup.
14. Scroll to Settings (not Google Settings)
15. You will be using the Settings button a lot, so press and hold the Settings button and drag it to the bottom middle of the screen, then let go.
16. Press the Home button - you should see the Settings button on your Home screen now.
17. Press the Settings button.
18. Press the "Airplane mode" slider to turn it from Off to On.
19. Inside the Settings app, scroll all the way down to About phone and press it.
20. Scroll down to "SW Version" and press it 7 times. After 3 presses, it will count down to tell you the number of presses needed to become a developer. Just keep pressing until it says you're a developer.
21. Click the Back button to return to the Settings app. Under "System", you should now see a "Developer options" item. Press it.
22. Inside Developer options, check the box next to "USB Debugging". Press "OK" at the "Allow USB Debugging" screen.
23. Scroll down to "Verify apps over USB" and uncheck the box. This saves you some time when you are downloading new versions of the app.
24. Optional: Scroll down to "Drawing". Select each of the three items that end in "scale" and select "Animation off".

25. Press the Back button to leave Developer options and return to the Settings app.
26. Under "Device", click "Display".
27. Uncheck the box for "Auto-rotate screen"
28. Press the "Sleep" button and change it to a longer value (recommended: 10 minutes)
29. Press the Back button to leave Display and return to the Settings app.
30. Press Wi-Fi to to go into the WiFi activity
31. Press the "Off" slider to turn on WiFi.
32. Press the three dots menu and select Wi-Fi Direct
33. Press Rename Device, type in a new name, and press OK. The name should follow the format in the Game Manual: your team number, a hyphen, a letter (A, B, C), then "RC" for the Robot Controller or "DS" for the Driver Station. E.g., for team 1234, the robot controller phone should be named 1234-A-RC and the driver station phone should be named 1234-A-DS.
34. Press the Back button twice to return to the Settings app
35. In the Settings app, under "Personal" select "Security".
36. Check the box for called "Unknown sources" (you will need to press OK to confirm).
37. Uncheck the box for "Verify apps"
38. Optional: Click Screen lock, then click None. This just saves some hassle when you're waking up the phone.
39. Press the Back button to return to the Settings app.

Part II: Configuring the Computer

You will need to download a couple of applications in order to program in Java. These steps below should be used to setup the 2016-17 Java-based development environment.

Downloading, Installing, and Configuring the FTC SDK

1. Download "Java SE Development Kit 8u102" for your computer from http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html .

2. Install the Java 8 SDK (just click Next -> Next, but if you're short on disk space, you can de-select the "Source Code" and "Public JRE" icons during the installation)

3. Download the FTC SDK, release version, from https://github.com/ftctechnh/ftc_app . Click “Clone or download”, then ”Download Zip”. This will create a file named ftc_app-master.zip. If you downloaded the SDK before September 9, 2016, download it again and use the new version.

4. Unzip the FTC SDK's ftc_app-master.zip file to a convenient directory.

5. Download Android Studio 2.2 from https://developer.android.com/studio/index.html

6. Install Android Studio. You can follow the directions at https://developer.android.com/studio/install.html or just click Next -> Next -> I Agree -> Next -> Install. If you're short on disk space, you can uncheck the "Android Virtual Device" option on the "Choose Components" page.

7. Run Android Studio (the last panel of the installation may have started Android Studio for you). The first time you run Android Studio, it will prompt for some configuration options, but the default settings do not need to be changed. If you get the message, "Unable to access Android SDK add-on list", see Note 1 below.

8. At the "Welcome to Android Studio" screen, choose Configure -> SDK Manager. In the dialog which appears, click “Launch Standalone SDK Manager”.

9. Click "Deselect All", then click the checkboxes to add these items:

- Tools -> Android SDK Build-tools, Rev 23.0.3
- Android 6.0 (API 23) -> SDK Platform. API 23, Rev 3
- Android 6.0 (API 23) -> Google APIs. API 23, Rev 1
- Extras -> Google USB Driver. Rev. 11 (not needed or selectable on Mac OS)

10. Click “Install 4 packages”, then click the "Accept License" button, then click Install.

11. After the installation has finished, exit Android Studio.

12. Run Android Studio again and choose "Import project (Eclipse ADT, Gradle, etc)".

13. Find the "ftc_app-master" directory where you unzipped the FTC SDK, then select the "build.gradle" file and click OK.

14. Android Studio will compile. This will take a while, and it will require network access to get some additional files which aren't included in the SDK. When it's done, click the "Gradle Console" button on the bottom right and you should see a line at the end which says, "BUILD SUCCESSFUL".

15. You may need to wait a few minutes before the next two steps will become available; Android Studio indexes the FTC SDK and may not allow you to do any more builds until the indexing is completed. Once the indexing has completed (the status bar line that says "Indexing" will disappear), select the Build -> Make Project menu item and make sure the Gradle Console also says BUILD SUCCESSFUL.

- ❖ Note 1: Some networks (typically those in schools or large organizations) require a proxy server in order to connect to the Internet. Browsers can usually determine how to connect to a proxy server, but Java applications such as the Android Studio usually need to be configured separately. Ask the organization's network administrator for how to connect to the proxy server. At a minimum, you will need to know the host name and port number of the proxy server; some proxy servers will require a username and password, too. Once you have this information, you can go through the rest of the instructions, but you will need to do some steps differently:
    - ➢ In step 7, when prompted to set up a proxy server, click Cancel (don't set up the proxy server yet!) and go through the rest of the set up steps. You will receive a message, "Android SDK is up to date." Continue with the next step.
    - ➢ At the end of step 8 (after launching the SDK Manager), select Tools -> Options and fill in the information for the proxy server. Close the SDK Manager. In the "Welcome to Android Studio" screen, choose Configure -> Settings. Under "Appearance & Behavior" -> "System Settings" -> "HTTP Proxy", enter the proxy information and close the dialog. Close the Settings dialog and Android Studio. Re-open Android Studio and re-launch the SDK Manager, then continue with the next step.
    - ➢ At the end of step 10, double-check that all the installations worked OK. Repeat them again for any items which failed. Continue with the next step.
    - ➢ During step 13, Android Studio will prompt you, "Would you like to have the IDE's proxy configuration be set in the project's gradle.properties file?", choose Yes.

Installing the ZTE USB Handset Driver on the Computer

The ZTE Speed device needs drivers in order to be usable by Windows-based operating systems. Fortunately, the driver is included on the phone and can be installed with the steps below. Other devices may also need drivers; Motorola device drivers are here, for example.

Note: you will need Administrator access to the computer. This needs to be done for each computer that will run Android Studio.

1. In the phone’s Settings app, under "Device", click "Connect to PC"
2. Click the circle (it’s called a radio button) next to "Install Driver"
3. Plug the phone into the computer.
4. Your computer may show a prompt "Installing device driver software". If you’re using a Windows computer, you may see an “Autoplay” dialog; click the link to “Run AutoRun.exe”. If you don’t see any prompts, you may need to open My Computer, find the CD drive which just appeared (it should be called “USB_Driver), and run the installation program in the folder for your operating system. Follow the prompts (just click Next a few times) to install the driver. Click the Finish button when it appears.
5. On the phone, click Settings (if necessary), then "Connect to PC".
6. Check the "Don't ask me again" box, then change the radio button to "Charge only".
7. A prompt, "Allow USB Debugging" should appear on the phone. Check the "Always allow from this computer" checkbox, then press OK.

Note: you may need to repeat these steps whenever you need to program from a different computer.

Part III: Installing apps onto the devices

Each phone used by a team can be used on the robot or as part of the driver station, but switching back and forth is not recommended because the names of the devices will have to be changed if you use the device in a different role.

Downloading and Running the Robot Controller App

1. Open Android Studio. Open the FTC app project, if necessary.
2. In Android Studio, click the "Run" menu, then "Run TeamCode".
3. After a few seconds, the "Choose Devices" dialog should appear and the phone should be listed. For the ZTE Speed, the prompt should look like, "ZTE 9030 Android...". If the dialog does not list any devices, make sure that the device has USB debugging enabled and the driver for the device has been installed.
4. Click OK in the Choose Devices dialog.
5. The phone should install and run the Robot Controller app.
6. On the phone, press the three dots menu in the upper right and select "Exit"
7. Press the Home button, then the 9 squares button.
8. Find the FTC Robot Controller app, press and hold it, then drag it to the bottom like you did with the Settings app.
9. Press the Home button again and you should see the FTC Robot Controller icon on your home screen.

Installing or Upgrading the Driver Station App

The Driver Station app can be installed in one of two ways - through the Google Play store or by using the version included in the FTC SDK zip file. The latter is preferred because it doesn't require that the phone be connected to the internet and because the app will not automatically update when a new (and possibly incompatible) version of the app is released. To install the Driver Station app:

1. Enable USB Debugging on the Driver Station device using steps 20 through 23 under the ZTE Speed Phone Setup.
2. Plug the Driver Station device into a computer with Android Studio installed and accept the prompt on the phone to allow USB debugging from this computer.
3. Open a command prompt and change to the directory where the FTC SDK was unzipped
4. Run these commands:

```
cd doc\apk
adb install FtcDriverStation-release.apk
```

You should see messages like these (the parts in bold may vary):

```
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
3165 KB/s (2392558 bytes in 0.738s)
        pkg: /data/local/tmp/FtcDriverStation-release.apk
Success
```

If you get a message such as, “`'adb' is not recognized as an internal or external command, operable program or batch file` “, follow the steps at http://www.howtodroid.com/2014/02/how-to-add-adb-command-prompt-windows.html and try again.

If you get a message “`Failure [[INSTALL_FAILED_ALREADY_EXISTS]]` ”, then run these commands:

```
adb uninstall com.qualcomm.ftcdriverstation
adb install FtcDriverStation-release.apk
```

Optional apps

Loading up the phone with a bunch of apps is not recommended. However, some apps might be useful for debugging:

Publisher	**App Name**	**Why**
Droid Kitchen	File Manager	List XML files
Xavier Gouchet	Axel	View and edit XML files
Makoto NARA (Mc. N)	USB Host Viewer	See what’s on USB bus (e.g., that the Robot Controller app should find)
Chainfire	USB Host Diagnostics	More detailed information on

		USB bus

Part IV: Writing Op Modes

Creating Your First Op Mode

Note: https://www.youtube.com/watch?v=TKPscPqsz8s is a video walkthrough of these steps provided by GEARSinc. https://www.youtube.com/watch?v=9437L9upnpE is a follow-up that goes into more detail about creating op modes.

1. Open Android Studio
2. If the “1: Project” panel isn’t open, open it using the button on the left side of the screen. Expand the FtcRobotController node, then browse to the src -- java – org.firstinspires.ftc.robotcontroller – external.samples folder. This has the sample opmodes which you can customize to your needs. Select an op mode (e.g., “SensorMRColor”), right-click and choose Copy.
3. Expand the TeamCode node, then browse to and select the src – main – java – org.firstinspires.ftc.teamcode node. Right click and choose Paste. In the CopyClass dialog, enter a new name (e.g., Team1234SensorTest) and press OK.
4. In the Team1234SensorTest.java file, find the three lines which look like:

```
@TeleOp(name = "Sensor: MR Color", group = "Sensor")
@Disabled
public class Team1234SensorTest extends LinearOpMode {
```

5. Remove the middle line (“@Disabled”). You can also change the "Sensor: MR Color" to a better name such as “Sensor Test”. Removing this line tells the FTC SDK to allow this op mode to appear.
6. Plug in your Robot Controller Android device to your computer's USB port. In Android Studio's Run menu, select Run 'TeamCode'.
7. This will prompt you to run the FtcRobotController app on a device; select your device and run your new app!
8. Start the Driver Station app on your Driver Station Android device and you should see your new op mode using the name from Step 5.

Learning Java and Android Programming

Java is a mature language and has lots of tutorials available. You can start with the Java Tutorials from Oracle at https://docs.oracle.com/javase/tutorial/ . Under the heading “Trails Covering the Basics”, go through each of these trails from start to finish unless otherwise noted:

- Getting Started

- Learning the Java Language
- Essential Java Classes – you can skim or skip the part about Regular Expressions
- Collections – this can be skimmed, but it might be helpful if you plan on working with a lot of data

Android apps are programmed in Java but do not use the Java runtime. Read this page about the Android application fundamentals to get an overview of how Android apps work. You don't need to understand everything on this page, because you won't be building an app from scratch, but you may find it useful. In addition to understanding the basics of an Android application, you may want to understand how the Android Studio works so that you can be more productive while programming - read this page to get an overview of Android Studio.

The FTC SDK includes support for external sensors, but you can also use the phone's internal sensors to guide the robot. See http://developer.android.com/guide/topics/sensors/sensors_overview.html if you want to learn about using the phone's built-in sensors such as the accelerometer.

Learning the FTC SDK

To learn the FTC SDK, the best sources are:

- The FTC training course at http://ftc.edu.intelitek.com/course/view.php?id=7 . This training course is broken up into several modules and covers how to create several different types of op modes. Module 2 covers programming with App Inventor and can be skipped if your team will be programming in Java. This course was written for the 2015-16 season and some parts, such as the part about registering Op Modes in 3.4 are outdated.
- The sample op modes. The sample op modes cover much of the functionality available in the SDK. Most of the sample op modes include detailed comments explaining their operation. The samples are under the directory where the FTC SDK was unzipped in the FtcRobotController\src\main\java\org\firstinspires\ftc\robotcontroller\external\samples\ directory.
- The JavaDoc for the FTC SDK. In the directory where you unzipped the FTC SDK, there is a "doc\javadoc" directory which has the documentation on all the classes and methods in the FTC SDK in JavaDoc format. JavaDoc is a Java standard documentation format that describes all of the classes and methods available to the programmer, but it doesn't have a lot of start-to-finish examples of how to use it.

Where to Get Help

Some suggested sources of help are:

- FIRST's FTC forum under the "New Technology" heading
- The FTC subreddit on reddit.com is a good place to ask questions about any part of FTC robotics, including programming
- Chief Delphi's FTC forum
- The FTC Android Network on Google Plus
- The FTC TEC network
- The ##ftc channel on IRC (web interface here)

Your affiliate partner may be able to connect you with other teams closer to you who can offer help.