FIRST Tech (FTC) Robotics:
New Programming Platform Workshop

FTC Team 9901
Techie Titans

FTC Team 9901

Techie Titans

- A rookie FTC Team from Nova Labs Robotics
- Formed with 10 Members from 1 FTC and 4 FLL teams
- Each member and coach had 2-3 years of successful FLL competition at state level
- 3 Team members with some Java and Android experience

FTC Team 9901

Techie Titans

Aneesh, 8th

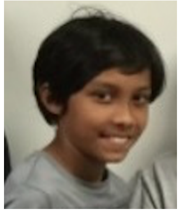
Faraaz, 7th

Sonika, 8th

Rahul, 8th

Vinay, 7th

Viraj, 7th


Ridha, 7th

Natasha, 9th

Jasmine, 10th

FTC Team 9901

Coaches and Mentors

Ram Boreda

https://www.linkedin.com/in/ramboreda

Tauhid Hossain Rahman

https://www.linkedin.com/in/tauhid-hossain-rahman-pmp-pmi-acp-csm/

Jagan Manickam

https://www.linkedin.com/in/jagan-manickam

Ruhul Chowdhury

https://www.linkedin.com/in/ruhul-chowdhury

FTC Team 9901

Techie Titans

→ Events **hosted or contributed** to:

- Discovery Studio: Summer Adventure
- Geared for Girl Makers: Sugo Bots
- First Lego League (FLL) Success!! – Best Practices for Coaches and Kids
- FIRST Lego League Maker Mix

FTC Team 9901

Techie Titans

http://techietitans-ftc.com

https://www.facebook.com/TechieTitans9901

https://www.youtube.com/channel/Techietitansftc

https://github.com/techietitansftc

https://drive.google.com/drive/techietitansftc

https://techietitans.slack.com

Agenda

- Brief overview of New platform
- Software development Environment
  - ZTE Speed Phone
  - FTC Apps (Driver Station and Robot Controller)
  - Android Studio
  - Our lesson plan to ramp up without HW
- FTC SDK
- Set up with Git Hub
- Event driven and linear programming model
- Example Op Modes
- Build, Deployment and Drive!!

Non Goals

- → Robot design and build
- → Best Practices and Strategy..We are rookie :)
- → App Inventor
- → Compatibility/reusability with legacy HW
- → Advanced Topics (If time permits)
  - ◆ Parallel threads
  - ◆ Motor Calibration, Stalling
  - ◆ Autonomous techniques (i.e. Line tracker, IR Beacon follower)

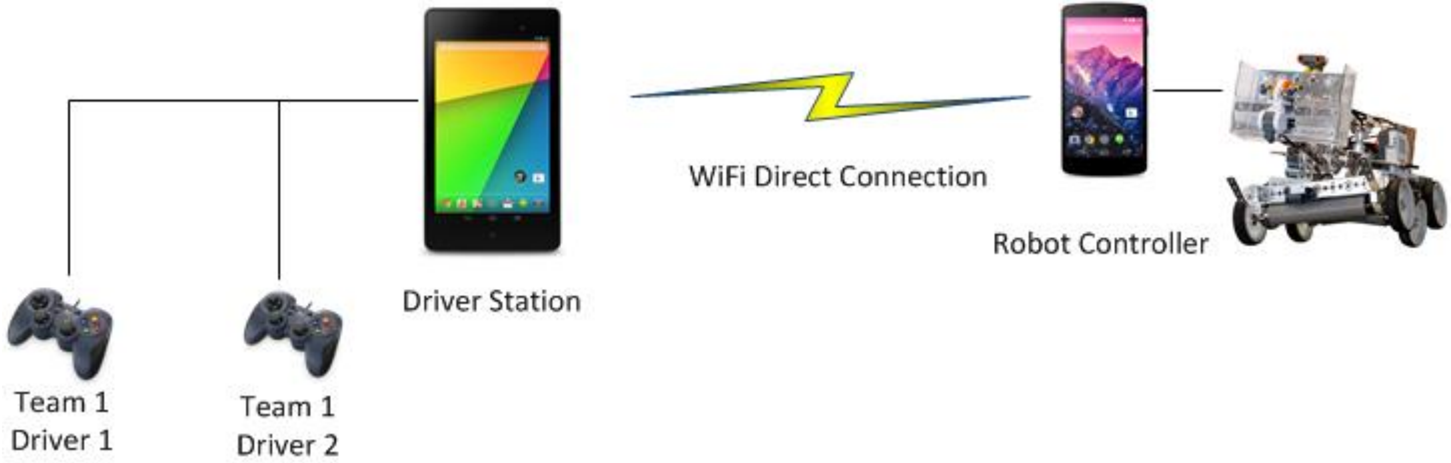
Attendee Poll

- Experience: Rookie Team? 1-2 years? More than 2 year?`
- Received the Kit? Tetrix? Matrix?
- Installed Android Studio? Built an app?
- Installed the FTC App?
- Ran a OpMode?
- Wrote and tested an OpMode?

New Platform

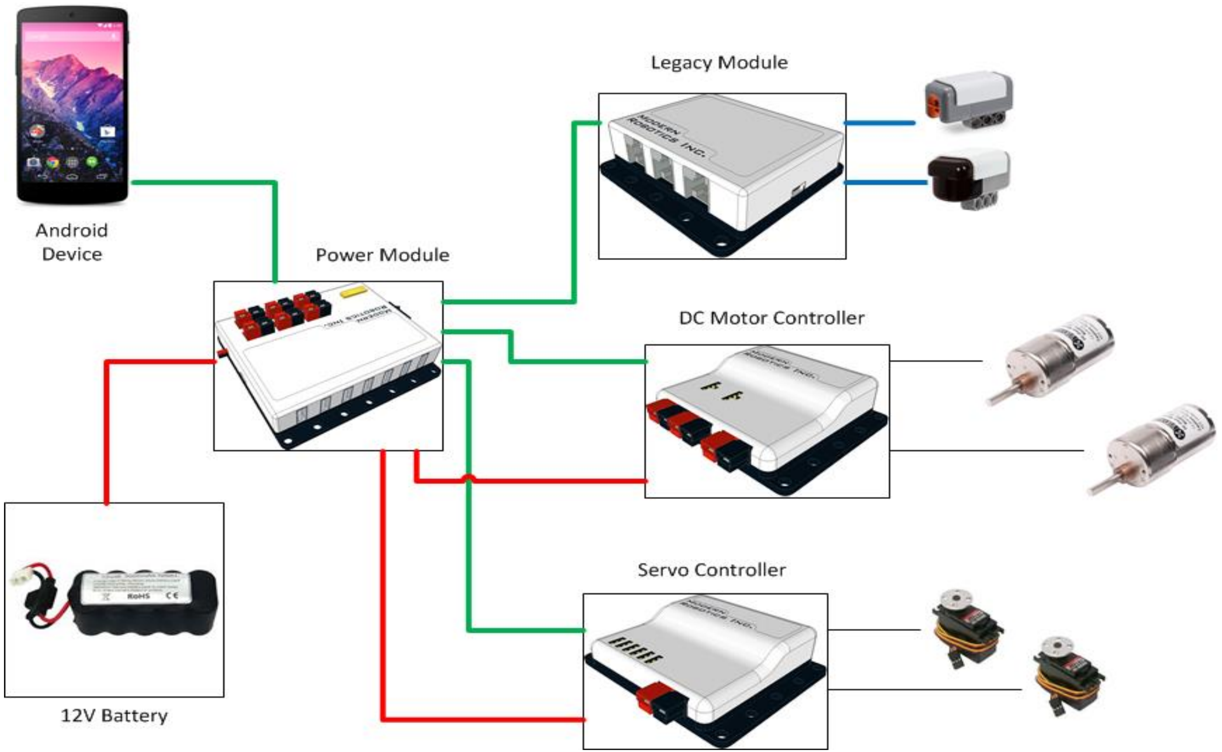
Overview

- Based on the Android OS and Java
- Uses two phones/tablets: Robot Controller and Driver’s Station
- Devices are connected over USB



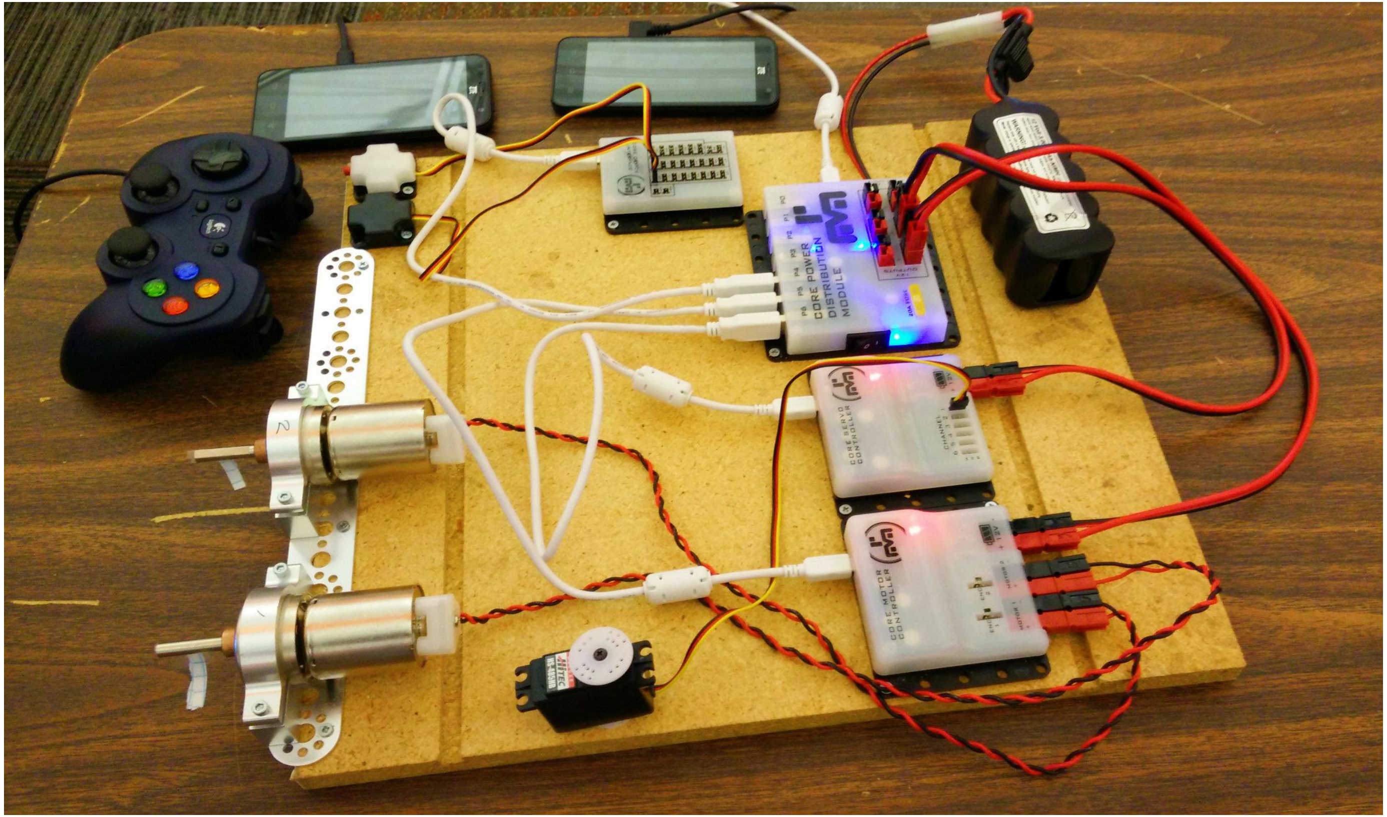
New Platform

Controllers



FTC Hardware

Our First learning setup



Android Device

2015-2016 *FIRST*® Tech Challenge

Game Manual Part I

FTC

FIRST® Tech Challenge

- There are two (2) allowed Android devices that Teams will use to control their Robot:
  - ZTE Speed.
  - Motorola Moto G (2nd Generation)

Android Device

BETA RELEASE

FTC *FIRST*® Tech Challenge

Next Gen Platform: Team & Mentor Guide

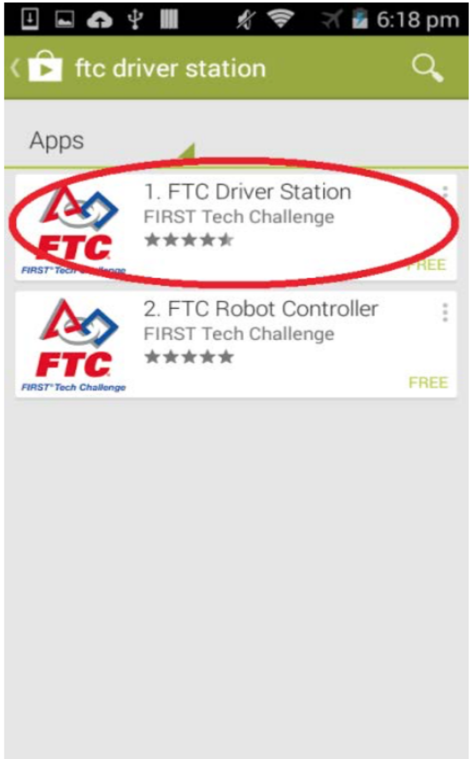
- The ZTE Speed is the recommended Android device.
- It has been carefully tested with the FTC software and hardware and has performed well.

- In most regions outside of the US, the Motorola Moto G (Kit Kat) is available for international teams to use for FTC.

FTC Apps

Overview

- → Download Driver and controller App from Google app store
- → <<Add Screencast >>



6:18 pm

ftc driver station

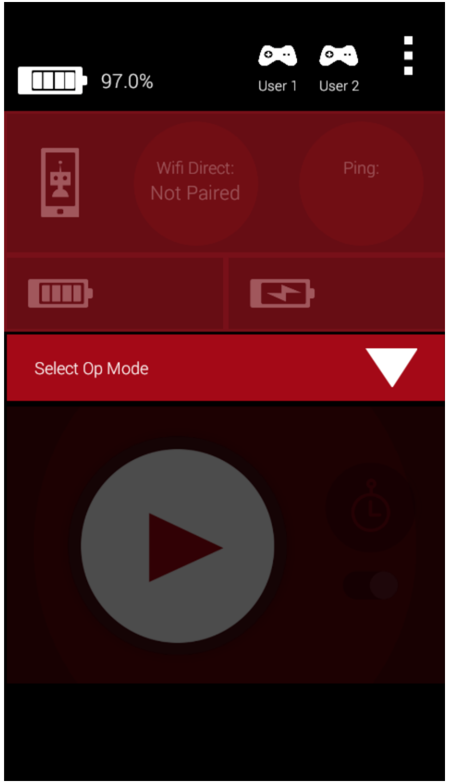
Apps

1. FTC Driver Station
FIRST Tech Challenge
FREE

2. FTC Robot Controller
FIRST Tech Challenge
FREE

FTC Apps

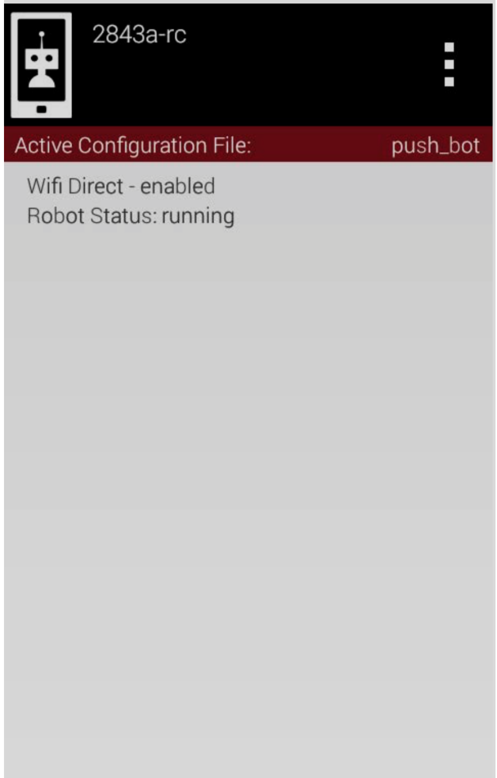
Driver Station



- Closed Source
- No code is deployed
- Front end app:
  - Program(OpMode) selection, Start and Stop
  - Gamepad is connected via micro USB
  - Telemetry (message from robot) is displayed
- Setup for Wifi communication to Robot Controller

FTC Apps

Robot Controller



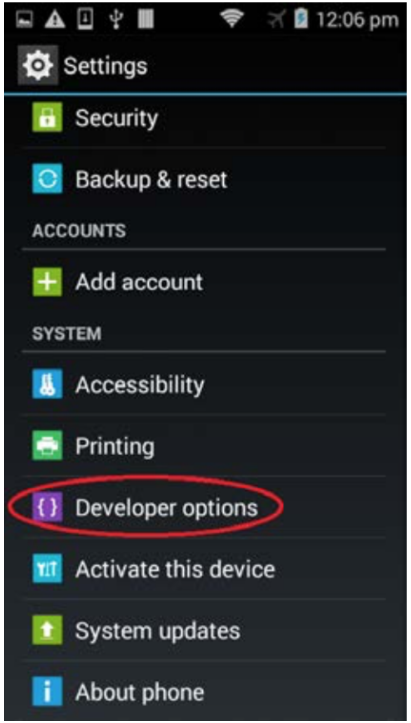
2843a-rc

Active Configuration File: push_bot

Wifi Direct - enabled
Robot Status: running

- Partial Open Source
- Part of Robot Assembly:
  - Connects to Power module/USB Hub
  - Integrates and executes programs (OpModes)
  - Broadcasts Telemetry messages to Driver station
- Setup and configuration of HW (i.e. Motors, sensors).

ZTE Phone Setup



12:06 pm

Settings

Security

Backup & reset

ACCOUNTS

Add account

SYSTEM

Accessibility

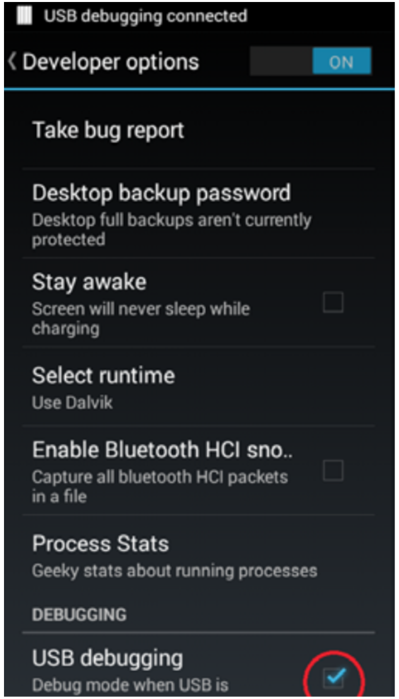
Printing

Developer options

Activate this device

System updates

About phone



USB debugging connected

Developer options ON

Take bug report

Desktop backup password
Desktop full backups aren't currently protected

Stay awake □
Screen will never sleep while charging

Select runtime
Use Dalvik

Enable Bluetooth HCI sno.. □
Capture all bluetooth HCI packets in a file

Process Stats
Geeky stats about running processes

DEBUGGING

USB debugging ☑
Debug mode when USB is

→ Remove the SIM card

→ Turn ON:

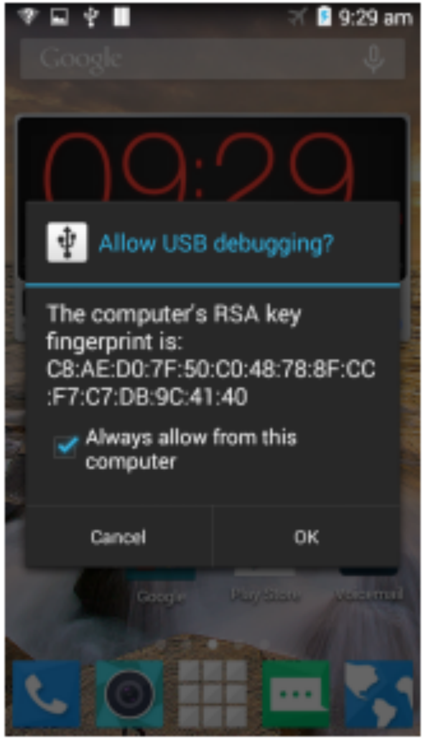
- Airplane mode
- Wifi

-- Robot controller phone only --

- Developer mode
- USB debugging
- Wifi Direct

ZTE Phone Setup

Connect to PC/Mac

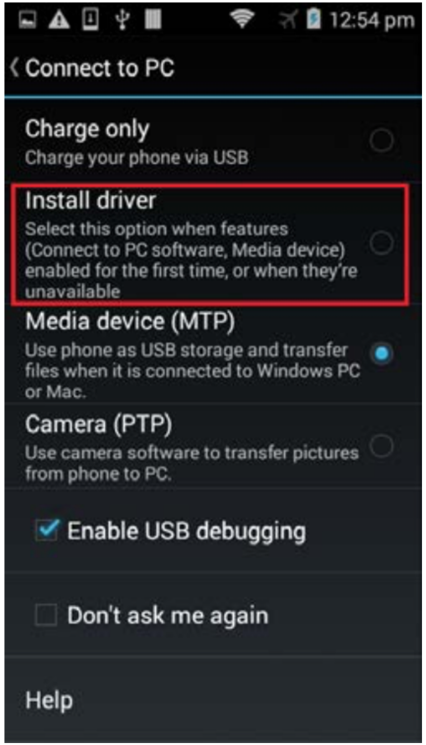


Allow USB debugging?

The computer's RSA key fingerprint is:
C8:AE:D0:7F:50:C0:48:78:8F:CC:F7:C7:DB:9C:41:40

☑ Always allow from this computer

Cancel | OK



‹ Connect to PC

Charge only
Charge your phone via USB ○

Install driver
Select this option when features (Connect to PC software, Media device) enabled for the first time, or when they're unavailable ○

Media device (MTP)
Use phone as USB storage and transfer files when it is connected to Windows PC or Mac. ●

Camera (PTP)
Use camera software to transfer pictures from phone to PC. ○

☑ Enable USB debugging

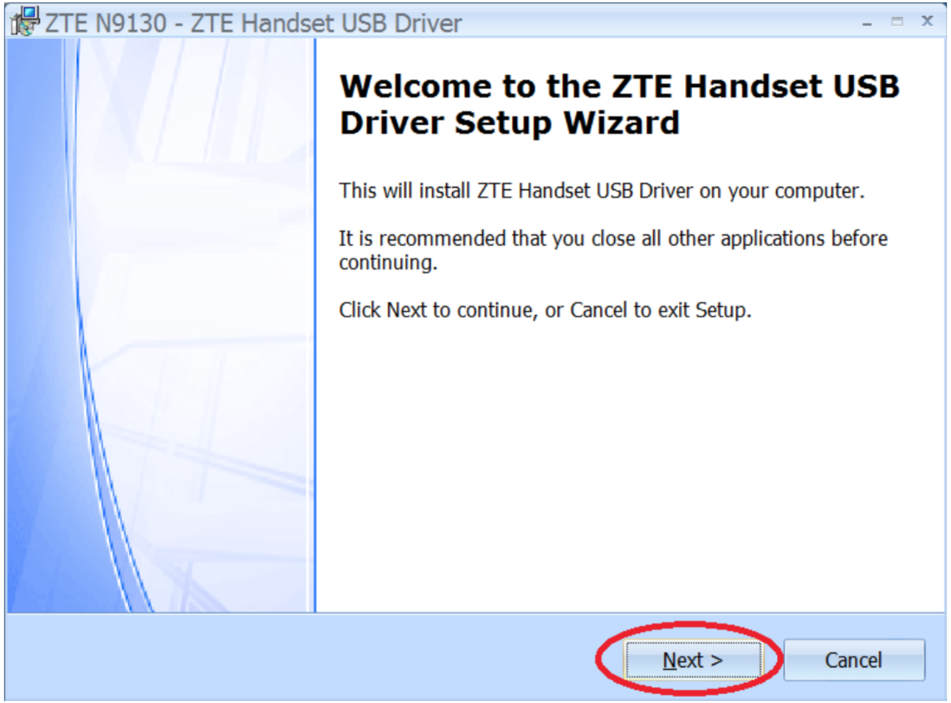
□ Don't ask me again

Help

- → Connect ZTE phone to the development PC/Mac via USB
- → Accept the RSA Key of PC/MAC in the phone

ZTE Phone Setup

Connect to PC/Mac



ZTE N9130 - ZTE Handset USB Driver

Welcome to the ZTE Handset USB Driver Setup Wizard

This will install ZTE Handset USB Driver on your computer.

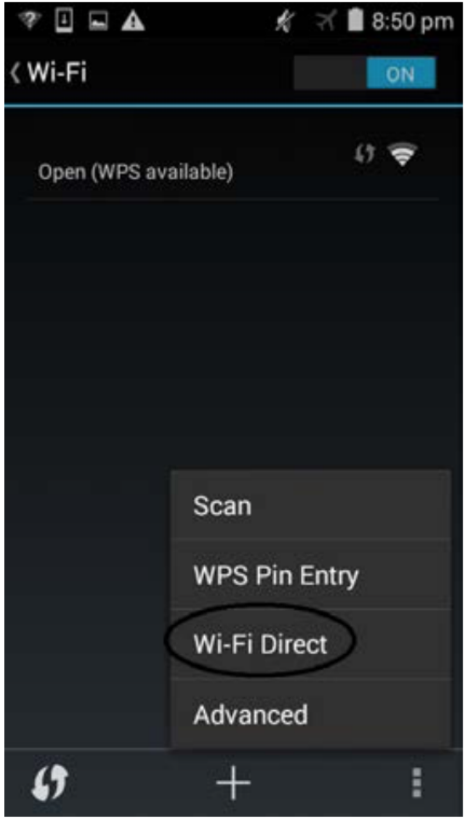
It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

→ Install the USB Driver on PC/Mac

ZTE Phone Setup

WiFi Direct



8:50 pm

Wi-Fi ON

Open (WPS available)

- Scan
- WPS Pin Entry
- Wi-Fi Direct
- Advanced

→ Robot controller acts as group owner

→ Give a unique name to your channel (i.e. FTC9901)

Summary

Q & A and Breakpoint 1

➔ ZTE phones are set with Driver Station and Robot controller app

➔ We should be able to run built in Op Modes

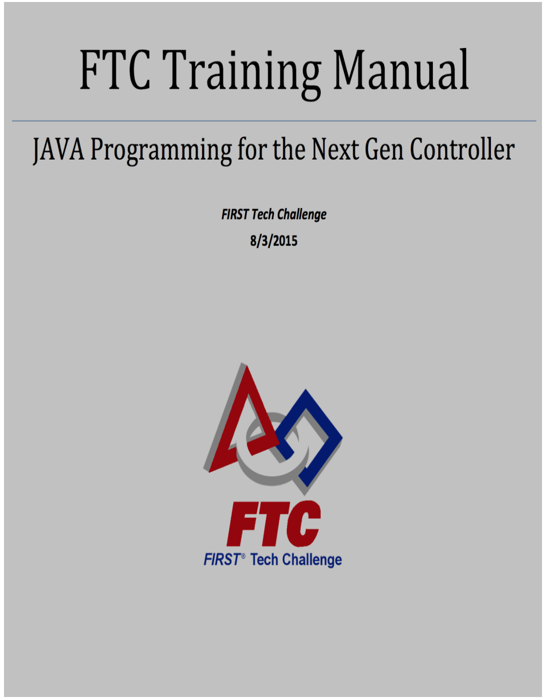
Android and Java

Our learning

- Early summer we spent some time learning Java Basics and Android Studio development environment.
- Our programming coach developed a lesson plan for the team members to follow.
- Leveraged online resources. We liked the following:
  - http://stackoverflow.com/
  - http://www.tutorialspoint.com/java/index.htm
- 2 projects we did:
  - Simple Calculator
  - Tic Tac Toe

Android and Java

Android Studio



FTC Training Manual

JAVA Programming for the Next Gen Controller

FIRST Tech Challenge

8/3/2015

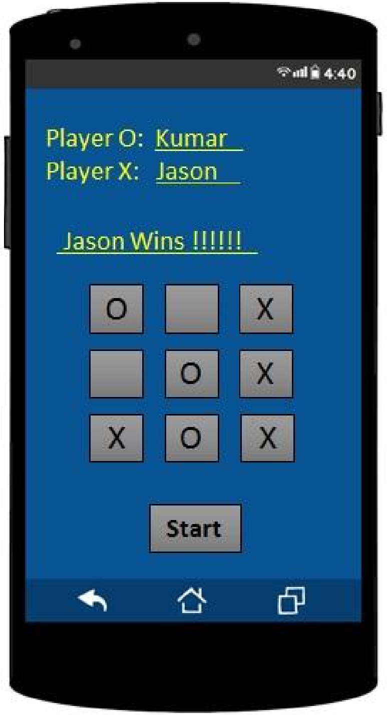
FTC

FIRST® Tech Challenge

- Team members installed Java and Android Studio using installation instructions in FTC Manual (Page 14 - 21)
- We deployed and tested Apps in:
  - Built in Emulator/Geny Motion
  - ZTE phone

Android and Java

Our learning: Tic Tac Toe



Player O: Kumar
Player X: Jason

Jason Wins !!!!!!

O		X
	O	X
X	O	X

- 3 Team members and Coach came up with 4 different and working solutions!

- We learned:
  - Class - its structure and pieces
  - Member Variables. Explore different types of Variables and where they are used.
  - Methods - Its structure and pieces
  - Java Data Types
  - Controls (if, for, while, switch/case etc.)

Android and Java

Our learning: Tic Tac Toe

[TicTacToe DEMO]

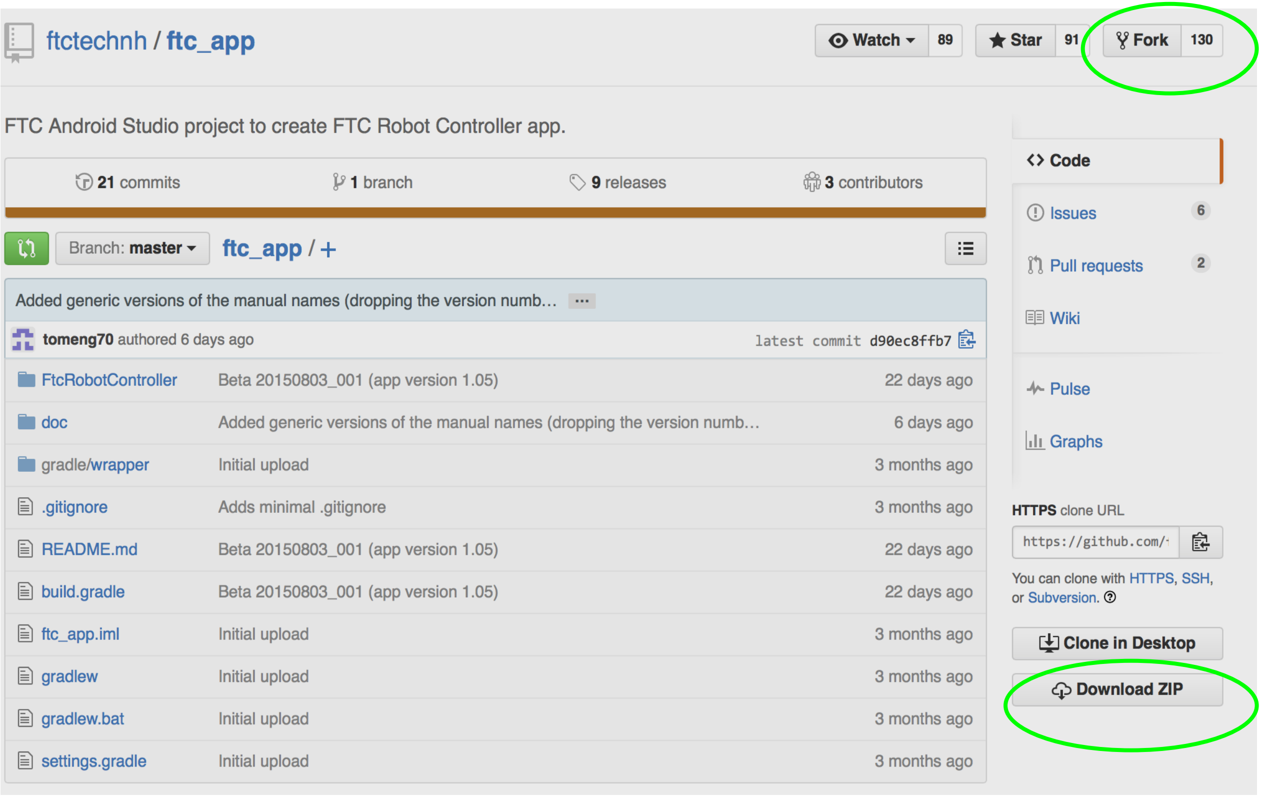
FTC Software

Basics

- Developed by QualComm for the FTC program
- Published in GitHub: https://github.com/ftctechnh/ftc_app
- Latest Version: FTC SDK (20150803_001) . Still in Beta.
- Includes:
  - Robot Controller Source Code in an Android Studio Project that teams will use to create their own programs(Op Modes)
  - Sample programs (Op Modes)
  - Documentation

FTC Software

Download



ftctechnh / **ftc_app**

FTC Android Studio project to create FTC Robot Controller app.

21 commits · **1** branch · **9** releases · **3** contributors

Added generic versions of the manual names (dropping the version numb… ···

tomeng70 authored 6 days ago — latest commit d90ec8ffb7

FtcRobotController	Beta 20150803_001 (app version 1.05)	22 days ago
doc	Added generic versions of the manual names (dropping the version numb…	6 days ago
gradle/wrapper	Initial upload	3 months ago
.gitignore	Adds minimal .gitignore	3 months ago
README.md	Beta 20150803_001 (app version 1.05)	22 days ago
build.gradle	Beta 20150803_001 (app version 1.05)	22 days ago
ftc_app.iml	Initial upload	3 months ago
gradlew	Initial upload	3 months ago
gradlew.bat	Initial upload	3 months ago
settings.gradle	Initial upload	3 months ago

HTTPS clone URL

https://github.com/

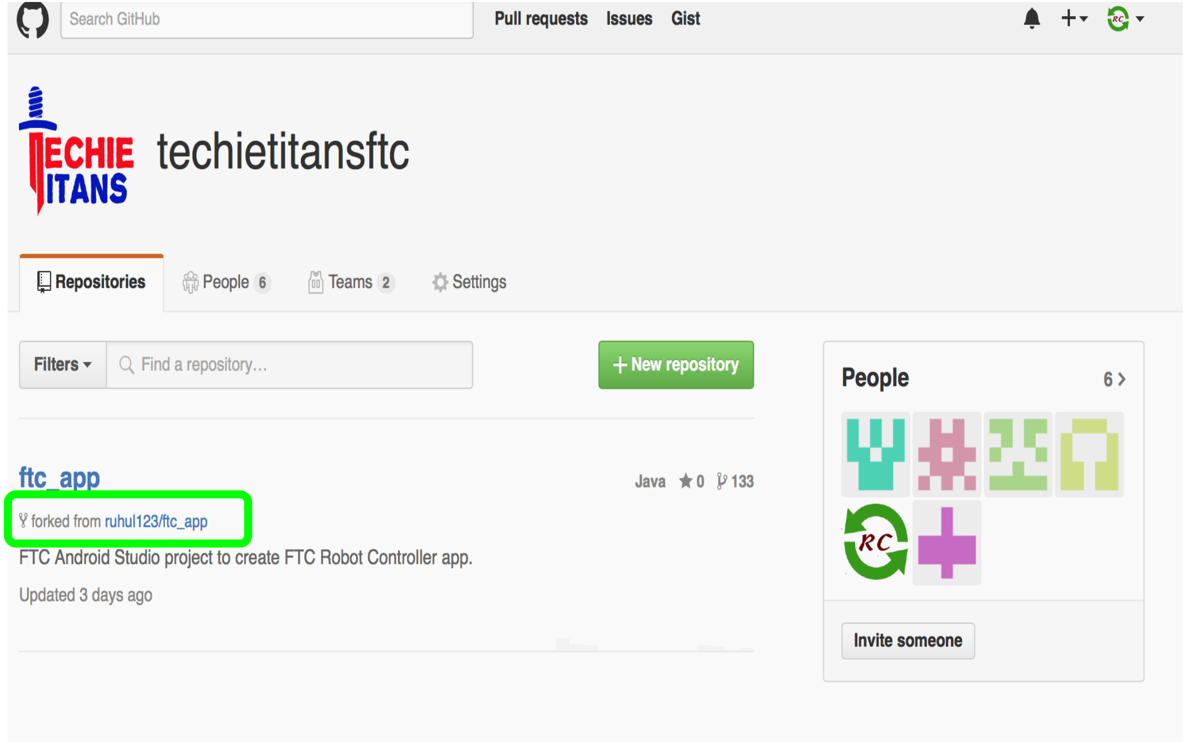
You can clone with HTTPS, SSH, or Subversion.

→ Can be downloaded as:

- .zip
- Forked (to your github) and cloned to desktop

FTC Software

GIT Management Model



ftc_app

forked from ruhul123/ftc_app

Java ★0 133

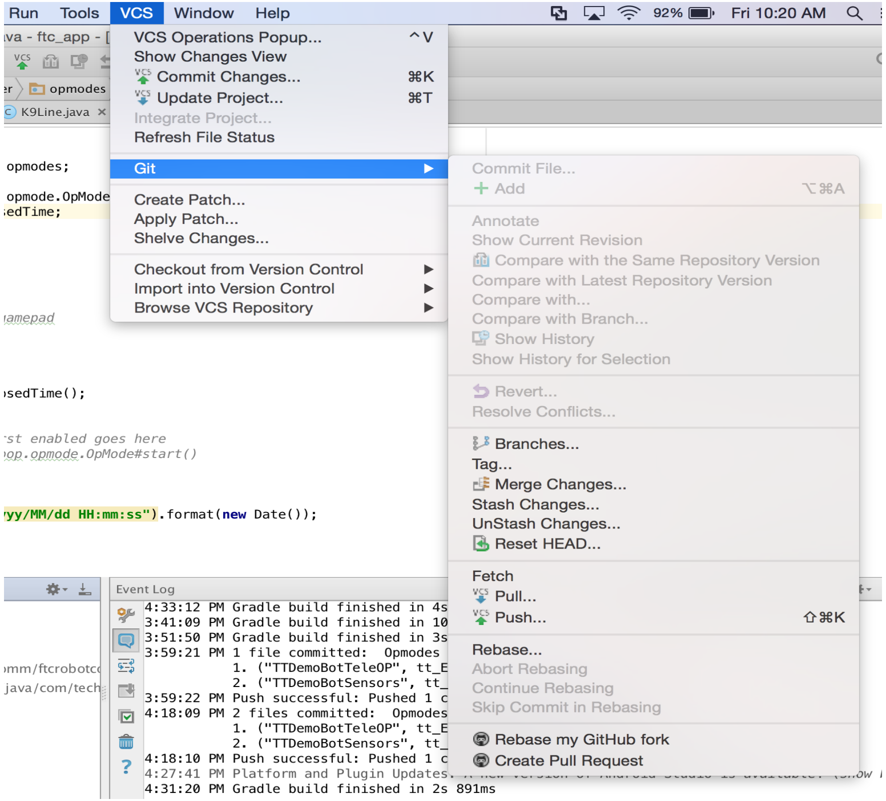
FTC Android Studio project to create FTC Robot Controller app.

Updated 3 days ago

- → Set a GROUP in github to manage and share code changes
- → Kids picked it up quickly !!

FTC Software

GIT Management Model



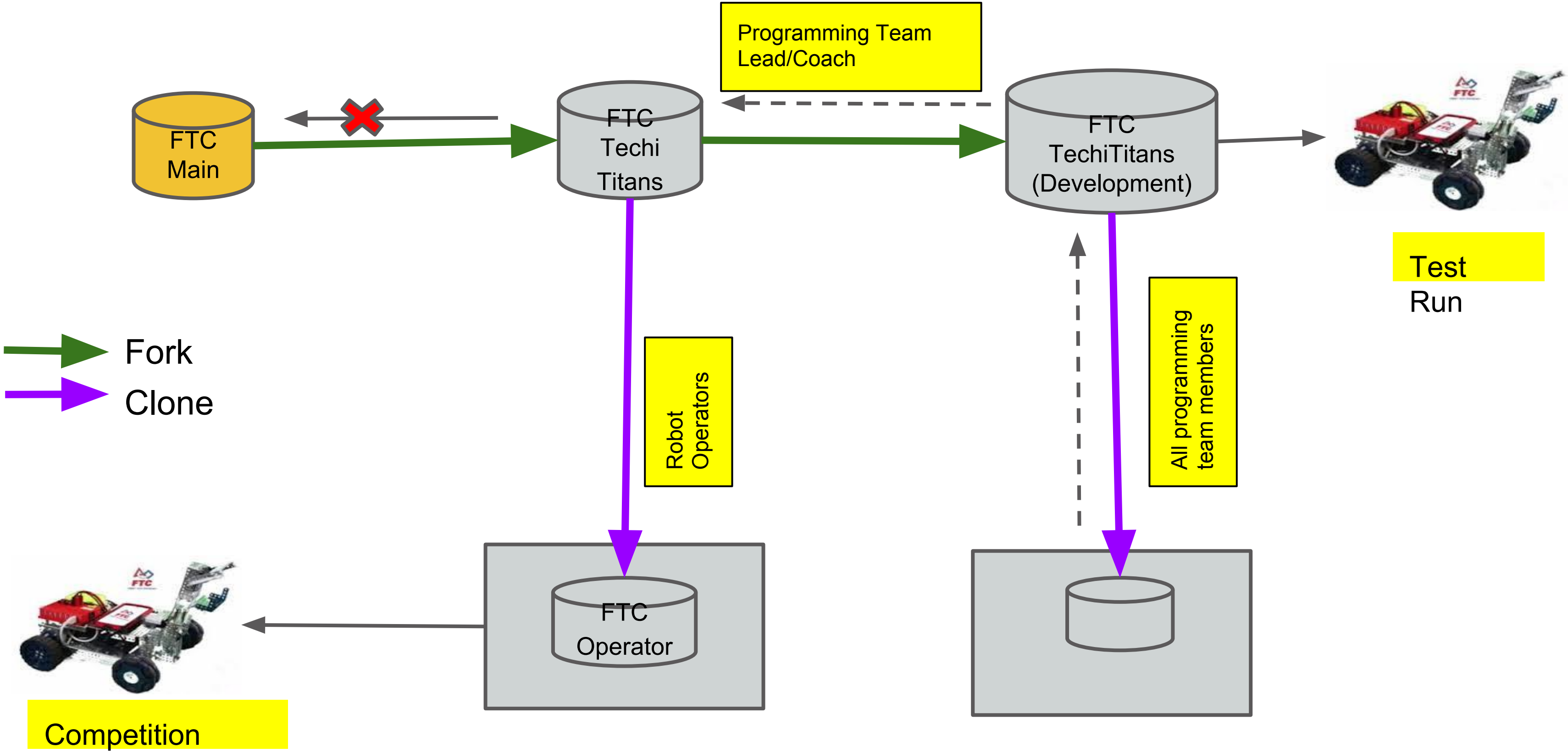
```
opmodes;

opmode.OpMode
sedTime;

gamepad

sedTime();

st enabled goes here
op.opmode.OpMode#start()

yy/MM/dd HH:mm:ss").format(new Date());

mm/ftcrobotc
java/com/tech
```

```
4:33:12 PM Gradle build finished in 4s
3:41:09 PM Gradle build finished in 10
3:51:50 PM Gradle build finished in 3s
3:59:21 PM 1 file committed:  Opmodes
           1. ("TTDemoBotTeleOP", tt_E
           2. ("TTDemoBotSensors", tt_
3:59:22 PM Push successful: Pushed 1 c
4:18:09 PM 2 files committed:  Opmodes
           1. ("TTDemoBotTeleOP", tt_E
           2. ("TTDemoBotSensors", tt_
4:18:10 PM Push successful: Pushed 1 c
4:27:41 PM Platform and Plugin Updates: A new version of Android Studio is available. (show
4:31:20 PM Gradle build finished in 2s 891ms
```

- ➔ All git(and github) features are in Android Studio
- ➔ Google did a great job!!

FTC Software

GIT Management Model


Programming Team Lead/Coach
FTC Main
FTC Techi Titans
FTC TechiTitans (Development)
Test Run
Fork
Clone
Robot Operators
All programming team members
FTC Operator
Competition

FTC Software

GIT Management Model

[Git DEMO]

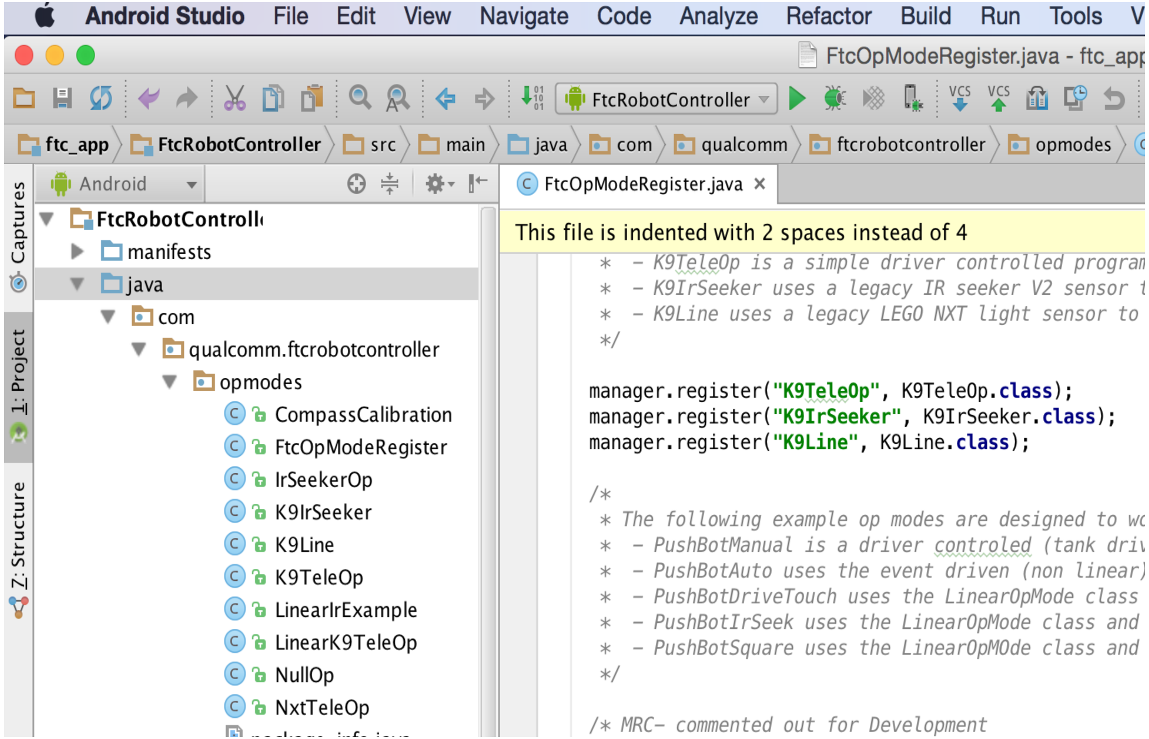
FTC Software

Op Mode

- An Op Mode is what teams use in order to create custom behavior for their robot. It is a Java class.
- Op Modes are similar to the tele-op and autonomous programs that teams wrote for their LEGO NXT controllers
- During a match, Op Modes are executed on the robot controller, but are selected by the team from the driver station
- Two Types: Event based and Linear

FTC Software

Op Mode



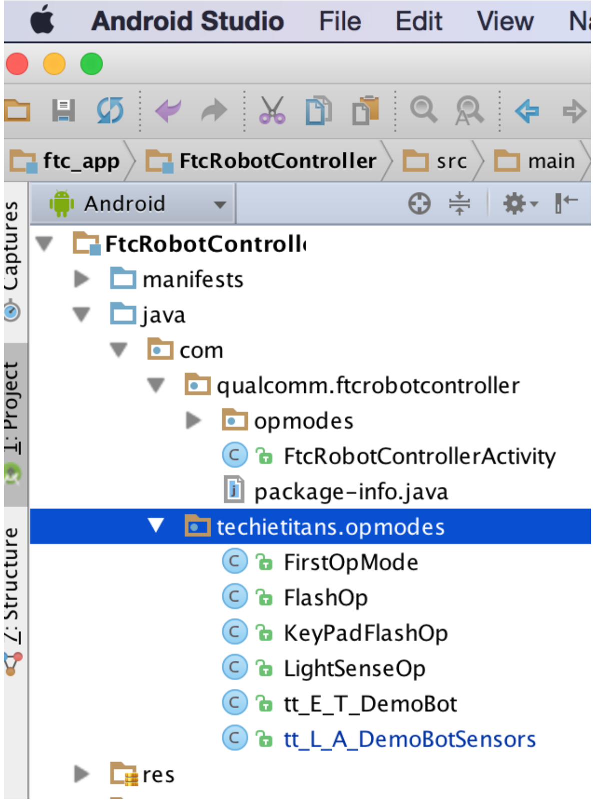
This file is indented with 2 spaces instead of 4

```
 *  - K9TeleOp is a simple driver controlled program
 *  - K9IrSeeker uses a legacy IR seeker V2 sensor t
 *  - K9Line uses a legacy LEGO NXT light sensor to
 */

manager.register("K9TeleOp", K9TeleOp.class);
manager.register("K9IrSeeker", K9IrSeeker.class);
manager.register("K9Line", K9Line.class);

/*
 * The following example op modes are designed to wo
 *  - PushBotManual is a driver controled (tank driv
 *  - PushBotAuto uses the event driven (non linear)
 *  - PushBotDriveTouch uses the LinearOpMode class
 *  - PushBotIrSeek uses the LinearOpMode class and
 *  - PushBotSquare uses the LinearOpMOde class and
 */

/* MRC- commented out for Development
```

- → Your code (Op Modes) are integrated in same project with FTC code
- → Only registered Op Modes will be available to Driver station

FTC Software

Code Separation



```
▼ FtcRobotControll
  ▶ manifests
  ▼ java
    ▼ com
      ▼ qualcomm.ftcrobotcontroller
        ▶ opmodes
          FtcRobotControllerActivity
          package-info.java
      ▼ techietitans.opmodes
          FirstOpMode
          FlashOp
          KeyPadFlashOp
          LightSenseOp
          tt_E_T_DemoBot
          tt_L_A_DemoBotSensors
  ▶ res
```

→ Our own namespace: *`com.techietitans.opmodes`*

→ Easier to manage future updates to FTC software

FTC Software

Event based Op Mode

- Inherited from `OpModes` base class
- Loop() method is continuously executed until the program terminates
- HW communication (Sensor reads, Motor control) is done at the end of each loop() execution
- Useful for program with distinct states and state transition

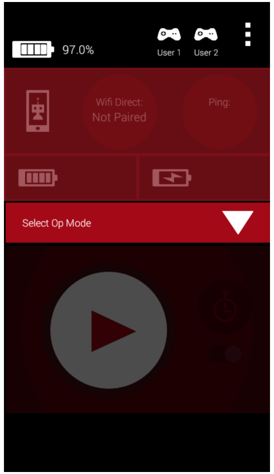
FTC Software

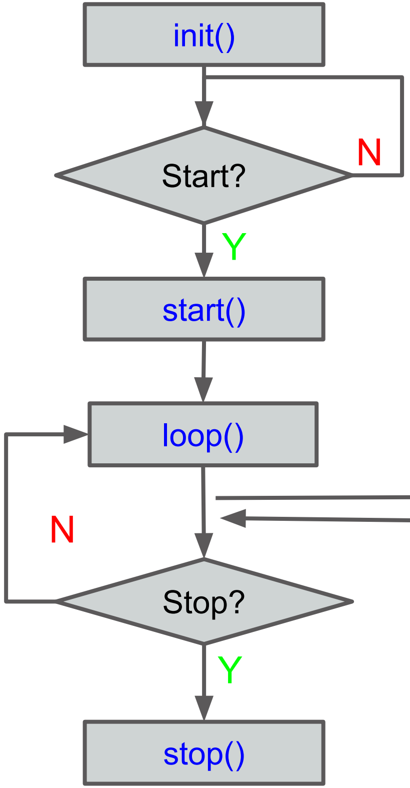
Event based Op Mode life cycle

Op Mode selected

Start Button click ▶

Stop Button click ■





init()

Start?

N

Y

start()

loop()

N

Stop?

Y

stop()

FTC Software

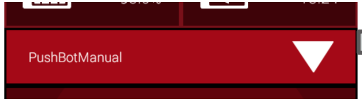
Linear Op Mode

- → Commands are executed sequentially one after the other
- → Similar to the model used to program a LEGO NXT with a tool like RobotC
- → Inherited from `LinearOpModes` base class
- → Can use blocking statements like `Thread.sleep()`
- → HW communication (Sensor reads, Motor control) is done on demand, as needed
- → Useful for Autonomous

FTC Software

Linear Op Mode life cycle

Op Mode selected



Start Button click ▶

Stop Button click ■

```
public class tt_L_A_DemoBotSensors extends
LinearOpMode {
    @Override
    public void runOpMode() throws
InterruptedException {
            // Wait for the start button to be pressed
            waitForStart();



            while(opModeIsActive()) {


            }
        }
}
```

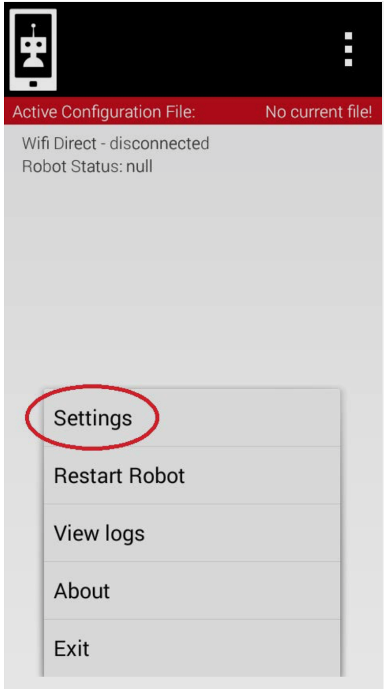
FTC Software

Op Mode

[CODE DEMO, BUILD and DEPLOY]

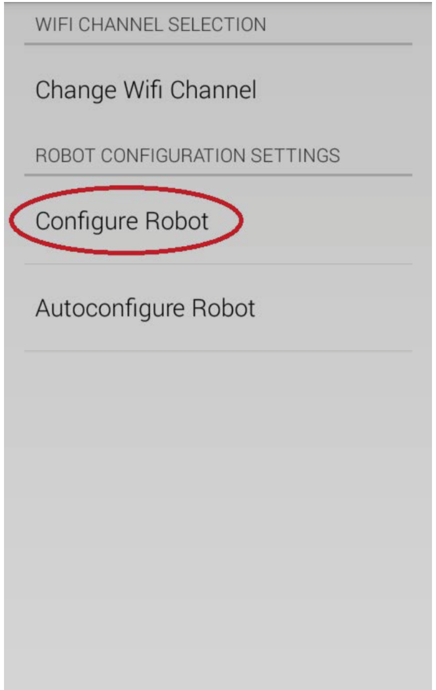
HW configuration

Robot Controller App



Active Configuration File: No current file!

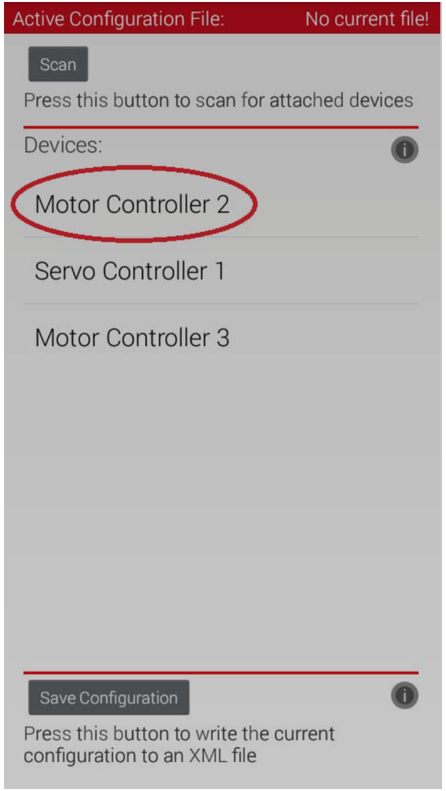
Wifi Direct - disconnected
Robot Status: null



- → Open your App Drawer and Run the “FRC Robot Controller”
- → Create a New Configuration

HW configuration

Robot Controller App



Active Configuration File: No current file!

Scan

Press this button to scan for attached devices

Devices:

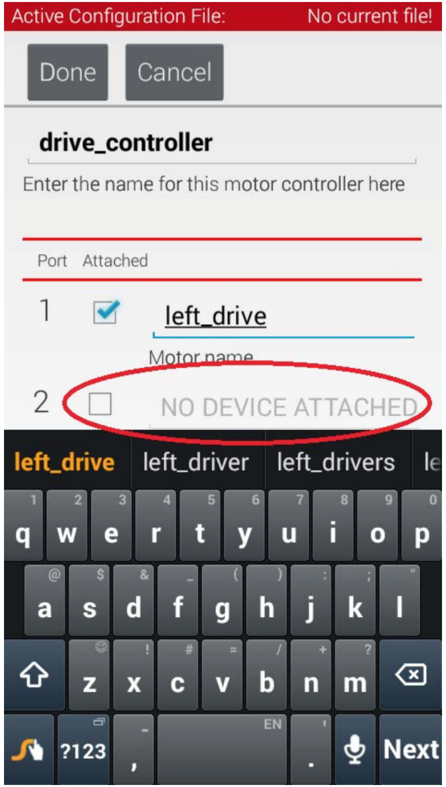
Motor Controller 2

Servo Controller 1

Motor Controller 3

Save Configuration

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



Active Configuration File: No current file!

Done Cancel

drive_controller

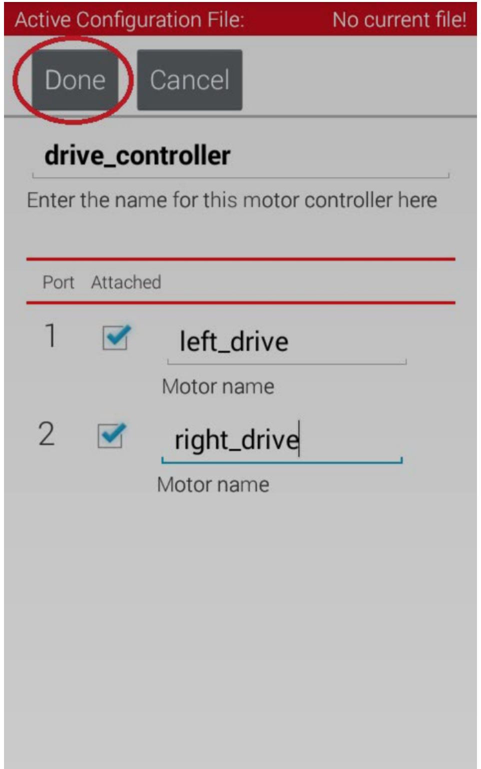
Enter the name for this motor controller here

Port	Attached	
1	☑	left_drive Motor name
2	□	NO DEVICE ATTACHED

- → **<u>Scan</u>** to auto discover connected controllers
- → Select a controller
- → Enter name for devices (motor, sensor)

HW configuration

Hardware map in the Op Mode

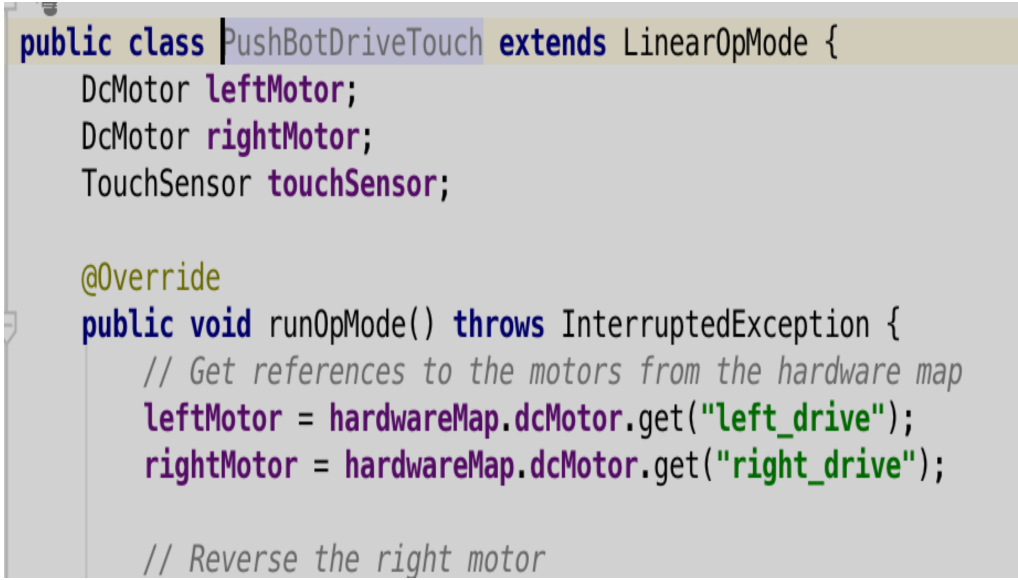


Active Configuration File: No current file!

drive_controller

Enter the name for this motor controller here

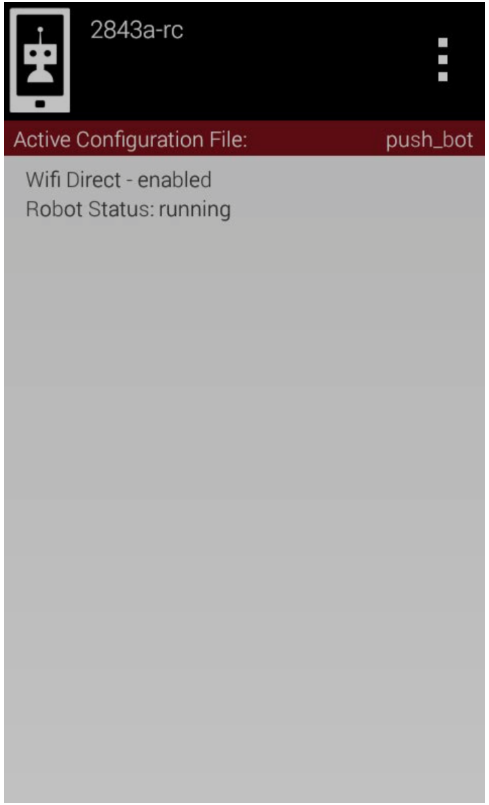
Port	Attached	
1	☑	left_drive Motor name
2	☑	right_drive Motor name



```
public class PushBotDriveTouch extends LinearOpMode {
    DcMotor leftMotor;
    DcMotor rightMotor;
    TouchSensor touchSensor;

    @Override
    public void runOpMode() throws InterruptedException {
        // Get references to the motors from the hardware map
        leftMotor = hardwareMap.dcMotor.get("left_drive");
        rightMotor = hardwareMap.dcMotor.get("right_drive");

        // Reverse the right motor
```

Execution Preparation

Robot Controller - Ready


2843a-rc

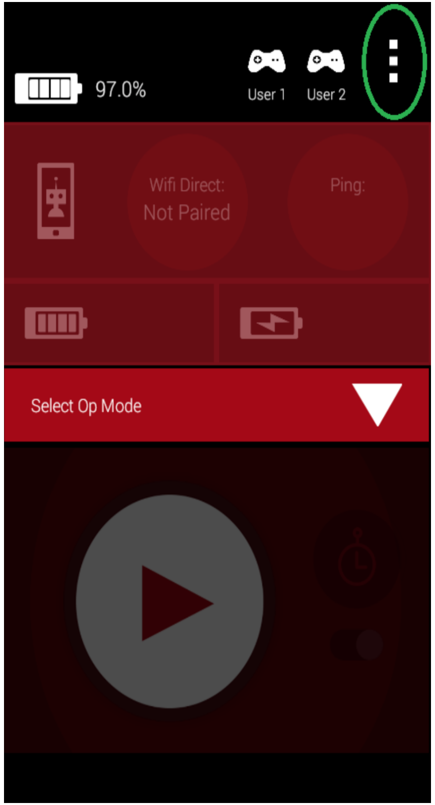
Active Configuration File: push_bot

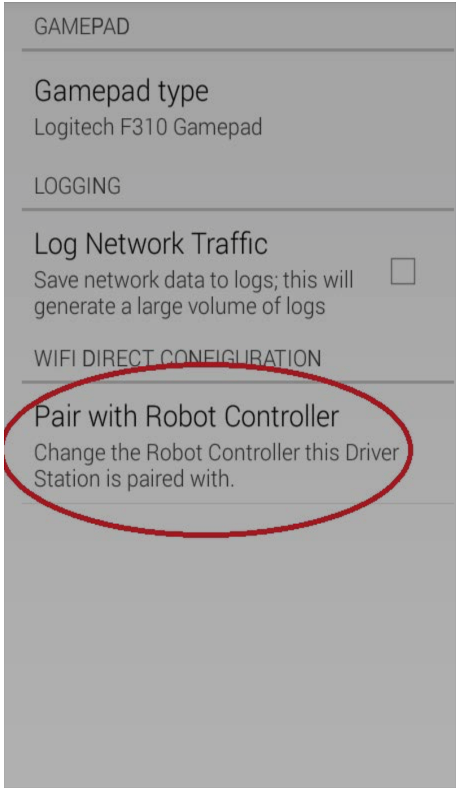
Wifi Direct - enabled
Robot Status: running

- → With our desired configuration file active - we are ready to start
- → We can have multiple configuration files, and swap between them as needed
- → An error here indicated that there is a mismatch between the file and attached HW

Execution Preparation

Driver Station - Pair Wifi





GAMEPAD

Gamepad type
Logitech F310 Gamepad

LOGGING

Log Network Traffic □
Save network data to logs; this will generate a large volume of logs

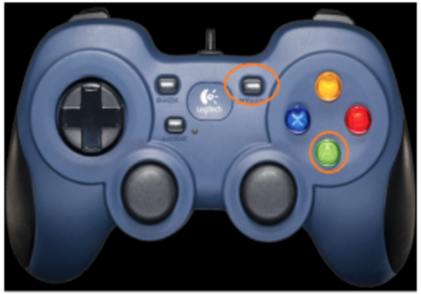
WIFI DIRECT CONFIGURATION

Pair with Robot Controller
Change the Robot Controller this Driver Station is paired with.

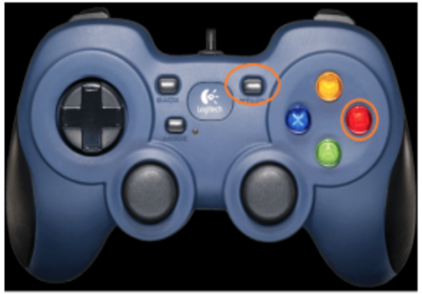
- → Initiate Pairing from FTC Driver Station app
- → Accept on Robot Controller

Execution Preparation

Driver Station - Joystick



Joystick 1



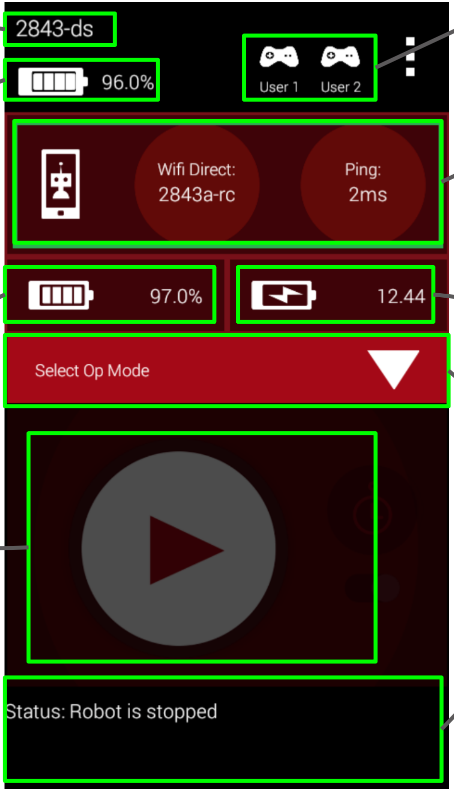
Joystick 2

→ Connect Gamepad/Joysticks to driver station ZTE phone using OTG USB Hub

→ Enable Driver 1 with START+A, Driver 2 with START+B

Execution Preparation

Driver Station - Ready



Status: Robot is stopped

Driver station Wifi Direct Name

Driver station battery level

Robot controller battery level

START/STOP controls

Gamepad activity

Network info

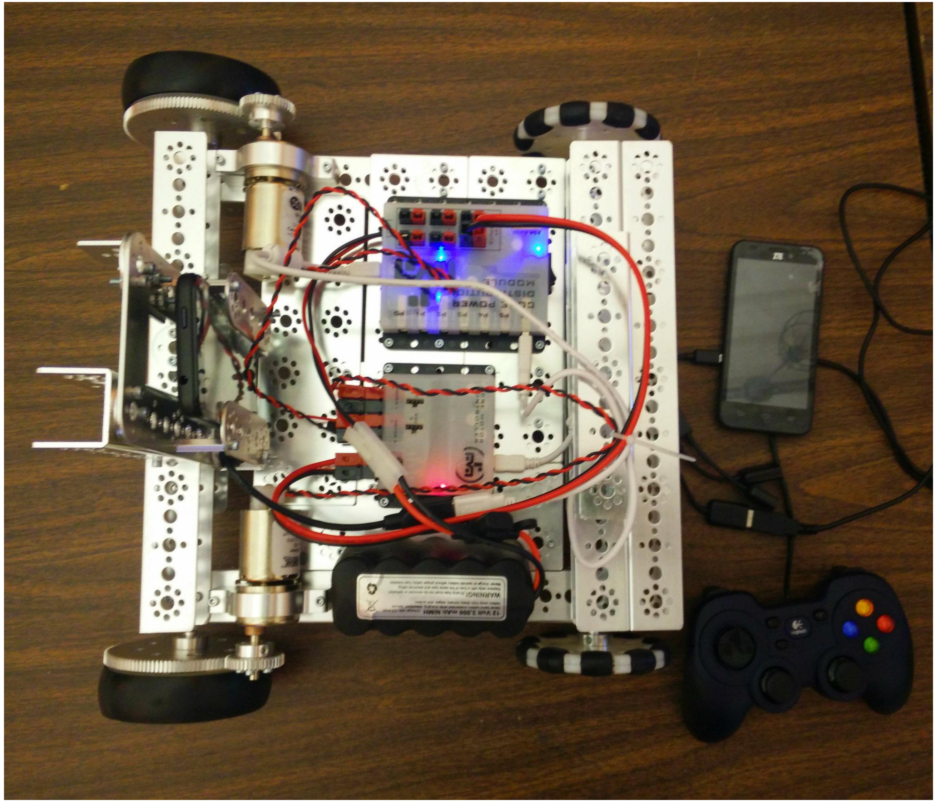
Robot battery level

Select Op Mode spinner

Telemetry Info

Execution Preparation

Techie Titan Demo Bot



- A Modified version of FTC Pushbot
- Hardware:
  - 2 DC Motor
  - Core Power Distribution Module
  - DC Motor Controller
  - 12 V Battery

Drive

[Drive DEMO]

Troubleshooting

- Check for “Allow USB Access” (prompt may be in background)
- Unplug phone and reboot, USB stack can get “stuck”
- When you upload new code/code changes, the phone sees it as a new program (all “allows” must be done again)
- Make sure the device names in your configuration file match the names exactly in your op mode
- Use debug statements (DbgLog), telemetry, and log files to help debug your op modes
- Refer to the javadoc for information about the classes and their member variables and methods

Q & A