

CUTTLEFISH
ROBOTICS
PRESENTS



31663

ANDROID STUDIO GUIDE PART 2
**GETTING AN
OPMODE RUNNING**



Kiwanis
CLUB OF CALGARY CH NOOK



GETTING AN OPMODE RUNNING

1. Understanding Android Studio

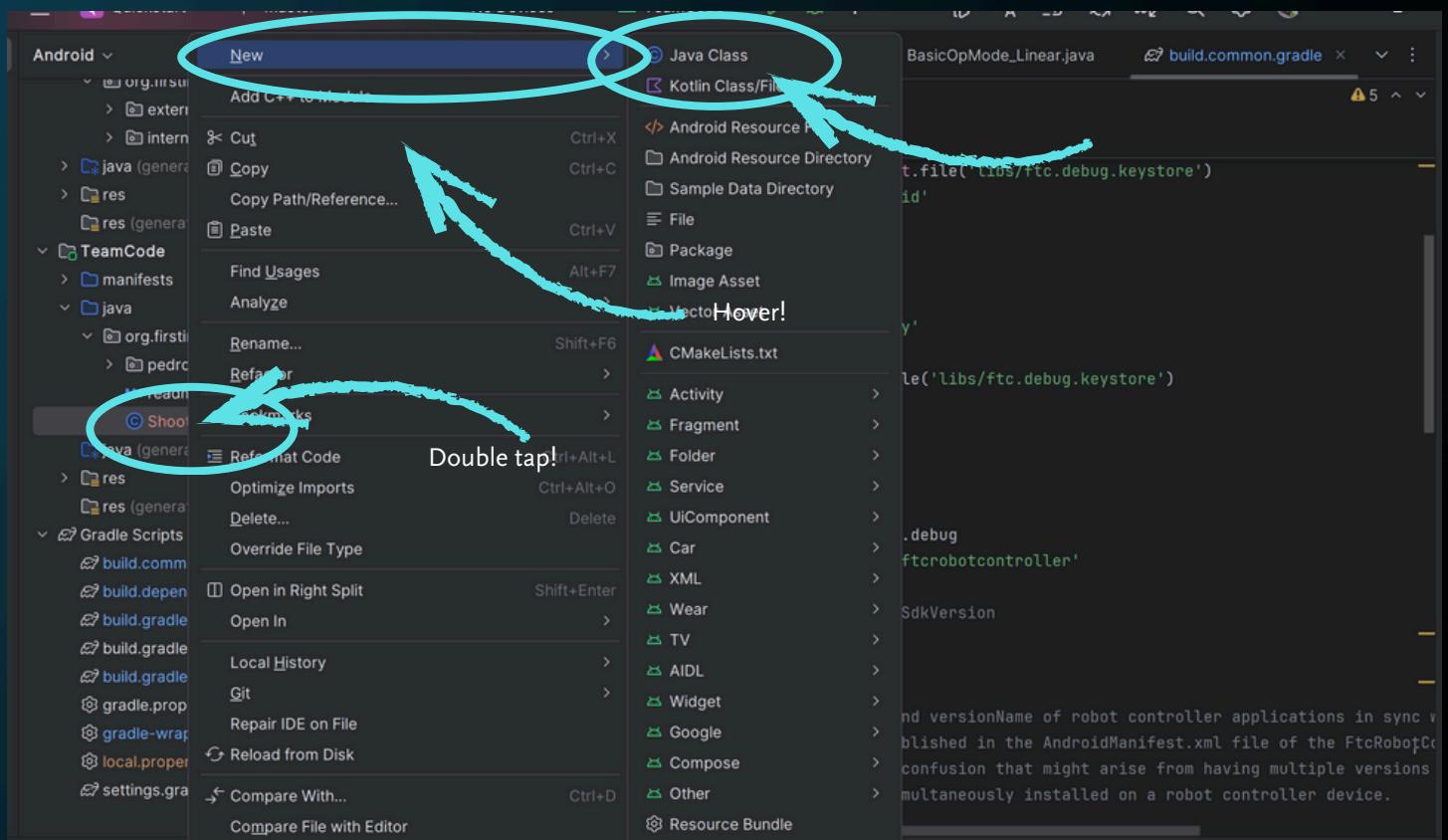
This is where your sample code is!

This is where you put your team's code!

This is where you put your dependencies!

```
public class BasicOpMode_Linear extends LinearOpMode {  
    public void runOpMode() {  
        // Setup a variable for each drive wheel to save power level for telemetry  
        double leftPower;  
        double rightPower;  
  
        // Choose to drive using either Tank Mode, or POV Mode  
        // Comment out the method that's not used. The default below is POV.  
  
        // POV Mode uses left stick to go forward, and right stick to turn.  
        // - This uses basic math to combine motions and is easier to drive straight.  
        double drive = -gamepad1.left_stick_y;  
        double turn = gamepad1.right_stick_x;  
        leftPower = Range.clip(number: drive + turn, min: -1.0, max: 1.0);  
        rightPower = Range.clip(number: drive - turn, min: -1.0, max: 1.0);  
  
        // Tank Mode uses one stick to control each wheel.  
        // - This requires no math, but it is hard to drive forward slowly and keep stra:  
        // leftPower = -gamepad1.left_stick_y;  
        // rightPower = -gamepad1.right_stick_y;  
  
        // Send calculated power to wheels  
        leftDrive.setPower(leftPower);  
        rightDrive.setPower(rightPower);  
  
        // Show the elapsed game time and wheel power.  
        telemetry.addData(caption: "Status", value: "Run Time: " + runtime.toString());  
        telemetry.addData(caption: "Motors", format: "left (%.2f), right (%.2f)", LeftPower
```

2. Creating a basic OpMode



GETTING AN OPMODE RUNNING

```
samples\BasicOpMode_Linear.java    build.common.gradle    teamcode\BasicOpMode_Linear.java
```

You should get this!

3. Copy and pasting sample code

Android

- FtcRobotController
 - manifests
 - java
 - org.firstinspires.ftc.robotcontroller.external.samples
 - BasicOmniOpMode_Linear
 - BasicOpMode_Iterative
 - BasicOpMode_Linear
 - ConceptAprilTag
 - ConceptAprilTagEasy
 - ConceptAprilTagLocalization
 - ConceptAprilTagMultiPortal
 - ConceptAprilTagOptimizeExposure
 - ConceptAprilTagSwitchcaseCameras
 - ConceptBlackboard
 - ConceptExploringIMUOrientation
 - ConceptExternalHardwareClass

Find this sample code!

```
BasicOmniOpMode_Linear.java    samples\BasicOpMode_Linear.java    build.common.gradle
```

1 > /.../
29
30 package org.firstinspires.ftc.robotcontroller.external.samples;
31
32 import ...
33
34 /*
 * This file contains a minimal example of a Linear "OpMode". An OpMode is a 'program'
 * the autonomous or the teleop period of an FTC match. The names of OpModes appear on t
 * of the FTC Driver Station. When a selection is made from the menu, the correspondin
 * class is instantiated on the Robot Controller and executed.
 *
 * This particular OpMode just executes a basic Tank Drive
 * It includes all the skeletal structure that all linear OpModes contain.
 */
48
49 Use Android Studio to Copy this Class, and Paste it into your team's code folder with
50 Remove or comment out the @Disabled line to add this OpMode to the Driver Station Op
51

Copy
(Cmd + A, Cmd + C) or
(Ctrl+A, Ctrl+C)

Android

- SensorTouch
- UtilityCameraFrameCapture
- UtilityOctoQuadConfigMenu
 - internal
- java (generated)
- res (generated)
- TeamCode
 - manifests
 - java
 - org.firstinspires.ftc.teamcode
 - pedroPathing
 - BasicOpMode_Linear
 - readme.md
 - ShooterAngle
 - java (generated)
 - res (generated)
- Gradle Scripts
 - build.common.gradle (Project: Quickstart)
 - build.dependencies.gradle (Project: Quickstart)
 - build.gradle (Project: Quickstart)
 - build.gradle (Module :FtcRobotController)
 - build.gradle (Module :TeamCode)
 - gradle.properties (Project Properties)
 - gradle-wrapper.properties (Gradle Version)
 - local.properties (SDK Location)

```
samples\BasicOpMode_Linear.java    build.common.gradle    teamcode\BasicOpMode_Linear.java
```

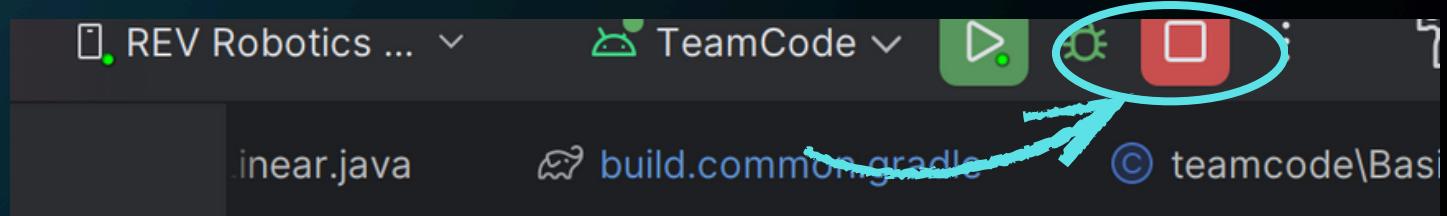
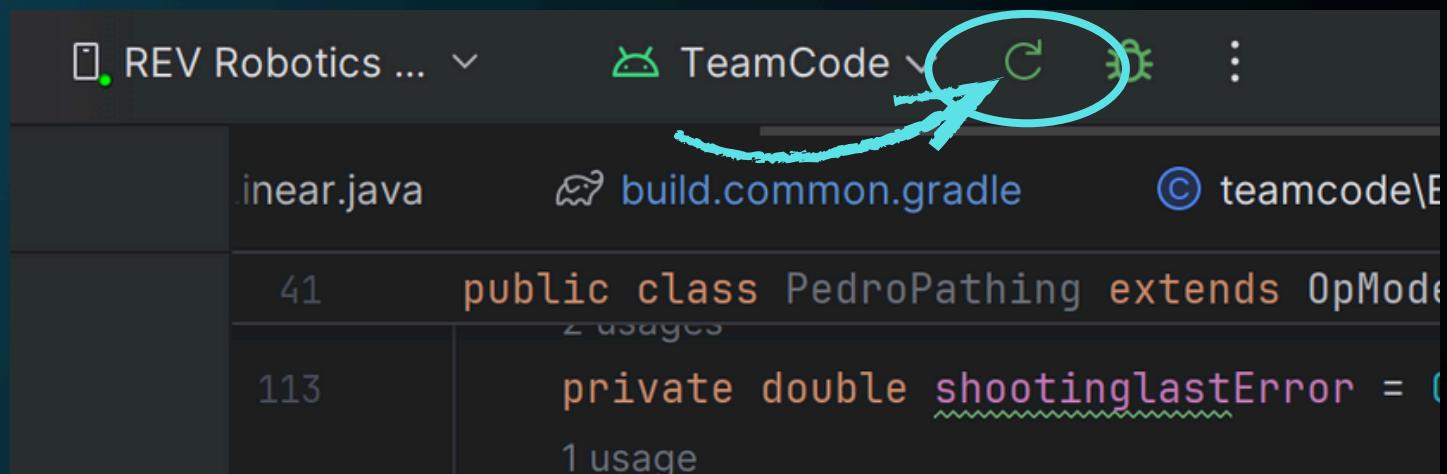
1 > /.../
2
3 /* Copyright (c) 2017 FIRST. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modification,
 * are permitted (subject to the limitations in the disclaimer below) provided that
 * the following conditions are met:
 *
 * Redistributions of source code must retain the above copyright notice, this list
 * of conditions and the following disclaimer.
 *
 * Redistributions in binary form must reproduce the above copyright notice, this
 * list of conditions and the following disclaimer in the documentation and/or
 * other materials provided with the distribution.
 *
 * Neither the name of FIRST nor the names of its contributors may be used to endorse or
 * promote products derived from this software without specific prior written permission.
 *
 * NO EXPRESS OR IMPLIED LICENSES TO ANY PARTY'S PATENT RIGHTS ARE GRANTED BY THIS
 * LICENSE. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
 * CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY
 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Paste it (Cmd + V) or (Ctrl + V) into your teamcode OpMode

GETTING AN OPMODE RUNNING

```
1 > /.../
29
30 package org.firstinspires.ftc.teamcode;
31
32 > import ...
33
34 /*
35 * This file contains a minimal example of a Linear "OpMode". An OpMode is a 'program'
36 * the autonomous or the teleop period of an FTC match. The names of OpModes appear on the
37 * of the FTC Driver Station. When a selection is made from the menu, the corresponding
38 * class is instantiated on the Robot Controller and executed.
39 *
40 * This particular OpMode just executes a basic Tank Drive Teleop for a two wheeled robot.
41 * It includes all the skeletal structure that all linear OpModes contain.
42 *
43 * Use Android Studio to Copy this Class, and Paste it into your team's code folder with
44 * Remove or comment out the @Disabled line to add this OpMode to the Driver Station OpMode
45 */
46
47
48
49
50
51
52
53 @Override(name="Basic: Linear OpMode", group="Linear OpMode")
54 @Disabled
55 > Direct Surround // : LinearOpMode {
56
57     // Declare OpMode members.
58     private ElapsedTime runtime = new ElapsedTime();
59     private DcMotor leftDrive = null;
60     private DcMotor rightDrive = null;
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78     leftDrive = hardwareMap.get(DcMotor.class,
79                                 Change "left_drive" and
80                                 "right_drive" to your
81                                 motor names
82
83     rightDrive = hardwareMap.get(DcMotor.class,
84                                  deviceName: "left_drive");
85
86
87
88
89
90     deviceName: "right_drive");
```

4. Connect to the Control Hub & run your code!



ANDROID STUDIO SET UP

Now you should have a basic Java OpMode that runs a 2-motor tank drive for your robot!

Be sure to check out our other PDFs and Tutorials to learn how to code in Java, how to set up PedroPathing, and how to use a LimeLight 3a!

END OF GUIDE