

## Virtual\_Robot and Roadrunner Quickstart

Virtual\_Robot includes the AcmeRobotics Roadrunner core “.jar” file as a dependency, and can be used to learn and test roadrunner. A good place to start is the teamcode from the AcmeRobotics [roadrunner-quickstart](#) project. The following are instructions for using the quickstart teamcode with virtual\_robot.

1. Download the road-runner-quickstart .zip from github, and extract the contents.
2. Find the teamcode package (org.firstinspires.ftc.teamcode) in the quickstart project, and copy its “drive”, “trajectorysequence”, and “util” folders into the teamcode package (also org.firstinspires.ftc.teamcode) of virtual\_robot.
3. Delete the following files from the teamcode.util package in virtual\_robot:
  - a. Logfiles.java
  - b. LoggingUtil.java
  - c. LynxModuleUtil.java
  - d. AssetsTrajectoryManager.java
4. Delete the following file from the teamcode.drive.opmode package in virtual\_robot: AutomaticFeedForwardTuner.java.
5. Comment out some code in several quickstart files:
  - a. In trajectorysequence.TrajectorySequenceRunner.java, comment out line 26:

```
//import org.firstinspires.ftc.teamcode.util.LogFiles;
```
  - b. In trajectorysequence.TrajectorySequenceRunner.java, comment out lines 212-217:

```
// if (targetPose != null) {  
// LogFiles.record(  
// targetPose, poseEstimate, voltage,  
// lastDriveEncPositions, lastDriveEncVels, lastTrackingEncPositions, lastTrackingEncVels  
// );  
// }
```
  - c. In drive.SampleMecanumDrive.java, comment out line 34:

```
//import org.firstinspires.ftc.teamcode.util.LynxModuleUtil;
```
  - d. In drive.SampleMecanumDrive.java, comment out line 87:

```
// LynxModuleUtil.ensureMinimumFirmwareVersion(hardwareMap);
```
  - e. In drive.SampleTankDrive.java, comment out line 34:

```
//import org.firstinspires.ftc.teamcode.util.LynxModuleUtil;
```

- f. In `drive.SampleTankDrive.java`, comment out line 81:

```
// LynxModuleUtil.ensureMinimumFirmwareVersion(hardwareMap);
```

6. In `drive.SampleMecanumDrive.java`, make the following changes to work with the MecanumBot configuration of `virtual_robot`:

- a. Change the configuration names of motors in lines 101-104 as follows:

```
leftFront = hardwareMap.get(DcMotorEx.class, "front_left_motor");  
leftRear = hardwareMap.get(DcMotorEx.class, "back_left_motor");  
rightRear = hardwareMap.get(DcMotorEx.class, "back_right_motor");  
rightFront = hardwareMap.get(DcMotorEx.class, "front_right_motor");
```

- b. After the “TODO” comment at line 119, reverse the left-sided motors:

```
leftFront.setDirection(DcMotorSimple.Direction.REVERSE);  
leftRear.setDirection(DcMotorSimple.Direction.REVERSE);
```

7. Now, some “tuning” is required to make roadrunner trajectories work properly with the MecanumBot configuration of `virtual_robot`. The following is based on a combination of trial-and-error, and using the tuning opmodes provided with road-runner-quickstart:

- a. In `drive.DriveConstants.java`, make the following assignments:

1. `TICKS_PER_REV = 1120;`
2. `MAX_RPM = 160;`
3. `RUN_USING_ENCODER = true;`
4. `TRACK_WIDTH = 17.91;`
5. `MAX_VEL = 21;`
6. `MAX_ACCEL = 21;`
7. `MAX_ANG_VEL = Math.toRadians(170);`
8. `MAX_ANG_ACCEL = Math.toRadians(170);`
9. `kV = 1.395;`
10. `kA = 0.002;`

- b. In `drive.SampleMecanumDrive.java`, set the values of `TRANSLATIONAL_PID` and `HEADING_PID` for Follower Tuning:

1. `TRANSLATIONAL_PID = new PIDCoefficients(5, 0, 0);`

```
2. HEADING_PID = new PIDCoefficients(5, 0, 0);
```

Now, when you run the project, using the MecanumBot robot configuration, the quickstart opmodes should operate as expected.