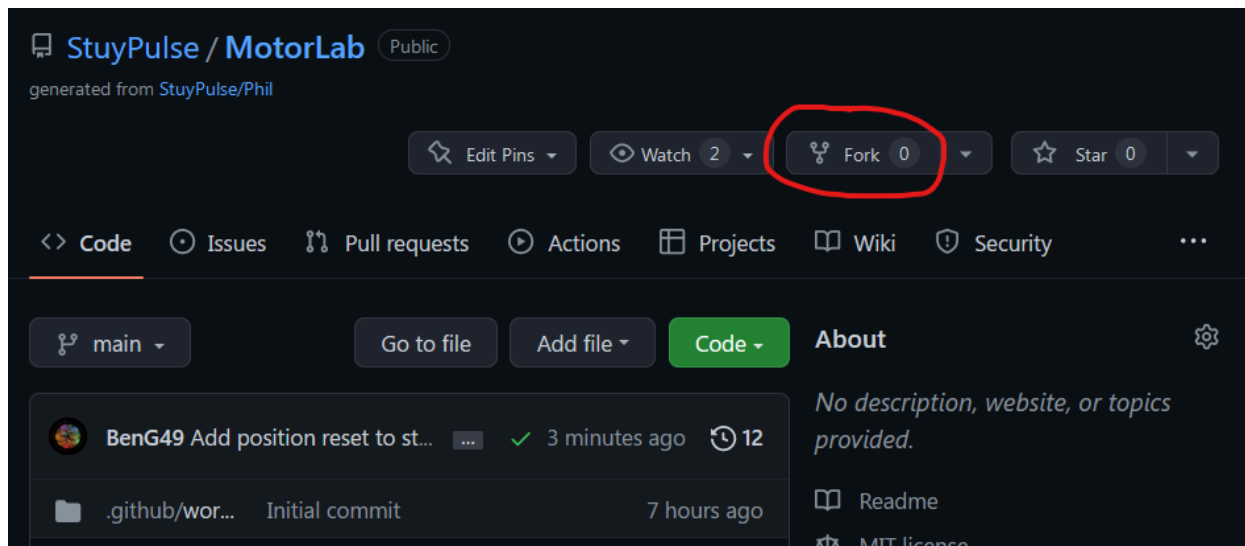


Motor Lab

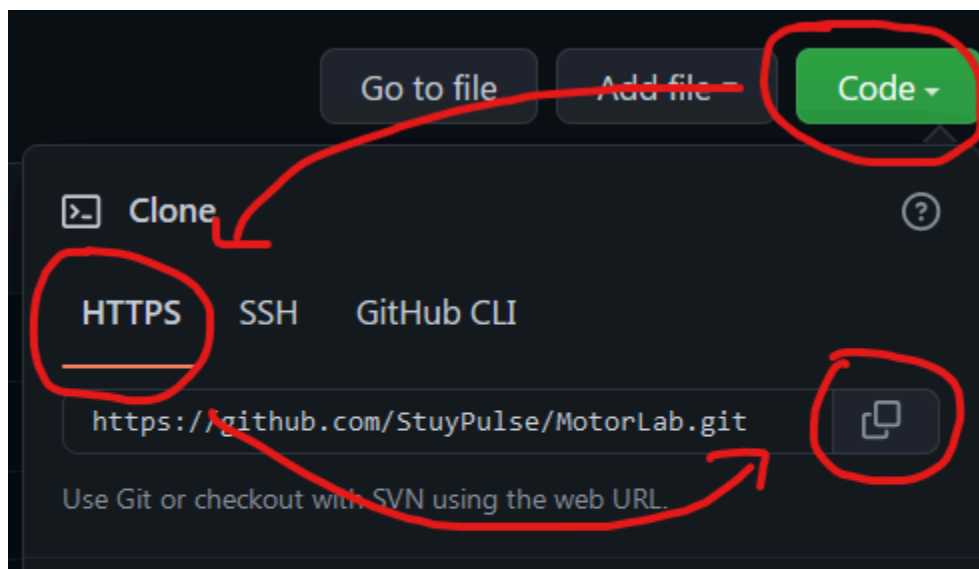
In today's activity, we'll ask you to code some simple commands to get some intuition for how controlling a robot with a tank drive (a robot with two sets of wheels that don't turn) works.

Using Github

Unlike the last activity, we'll be using github to put your changes online.



To start making changes go to <https://github.com/StuyPulse/MotorLab> and click on "Fork". Leave all of the settings and names as is and click on "Create fork".



To download your code go back to <https://github.com/> and find the new MotorLab. Click on “Code”, then “HTTPS”, and then copy the link.

```
Ben@BEN-DESKTOP MINGW64 ~
$ git clone https://github.com/BenG49/MotorLab.git
Cloning into 'MotorLab'...
remote: Enumerating objects: 160, done.
remote: Counting objects: 100% (160/160), done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 160 (delta 46), reused 145 (delta 37), pack-reused 0
Receiving objects: 100% (160/160), 82.60 KiB | 3.44 MiB/s, done.
Resolving deltas: 100% (46/46), done.
```

Open up Git Bash and type `git clone <link here>`. You're set!

Adding Your Code

In the folder MotorLab, the only file that you'll need to edit is `DriveFunctions.java` in `MotorLab/src/main/java/com/stuypulse/robot/commands/`.

```
1  package com.stuypulse.robot.commands;
2
3  import com.stuypulse.robot.util.Motor;
4
5  public class DriveFunctions {
6
7      public static void driveForwards(Motor left, Motor right) {
8
9      }
10
11     public static void driveBackwards(Motor left, Motor right) {
12
13     }
```

This is what your file should look like when you first open it (some lines have been cut off). You'll be coding inside of the brackets of each section, and one section will be run at a time depending on which command you choose to run.

```

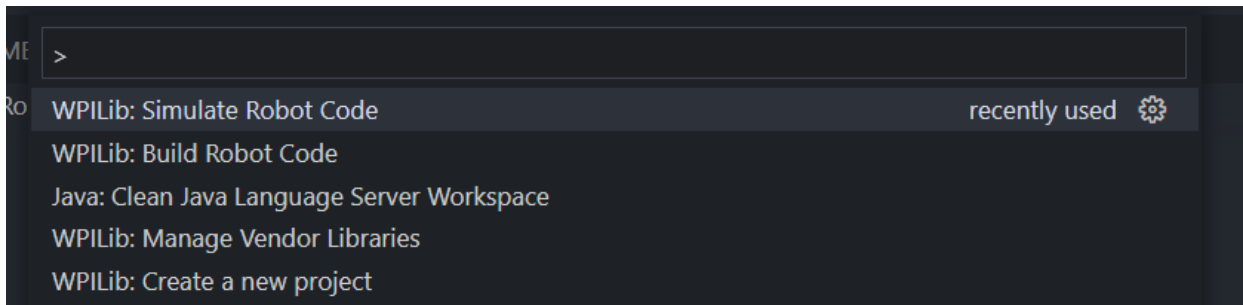
1  package com.stuypulse.robot.commands;
2
3  import com.stuypulse.robot.util.Motor;
4
5  public class DriveFunctions {
6
7      public static void driveForwards(Motor left, Motor right) {
8          left.set(1.0);
9      }
10
11     public static void driveBackwards(Motor left, Motor right) {
12
13     }

```

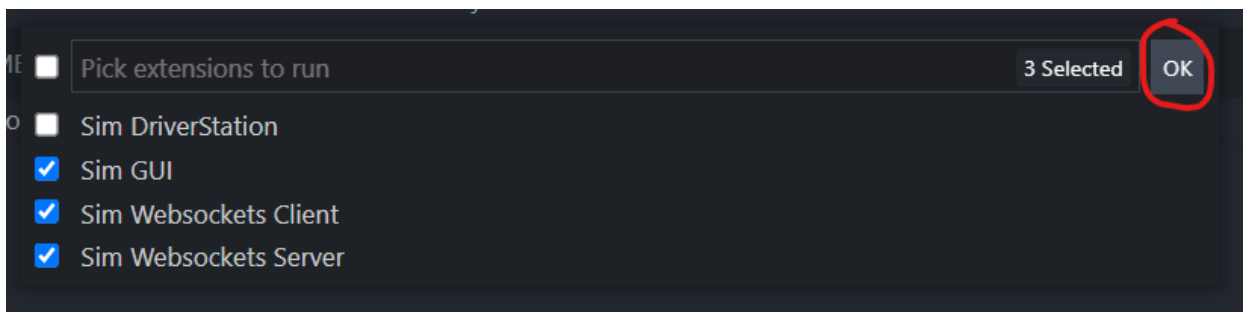
For example, the above code will run the left motor at 100% forever when the **Drive Forward** command is being run.

Running Your Code

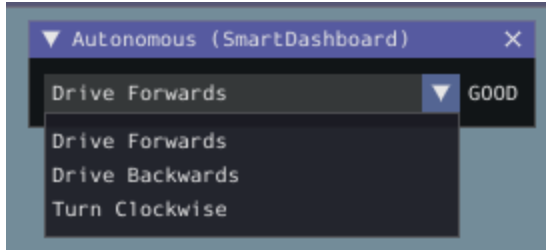
To run your code, press **Ctrl+Shift+P** or click on the WPILib logo at the top right.



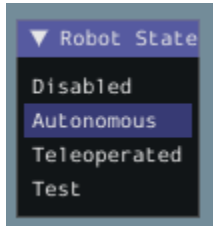
Then select/type **WPILib: Simulate Robot Code on Desktop**.



Hit it **OK** and you're set.



To select which command to run, use the Autonomous selector after you simulate your code and then press “Autonomous” in the robot state selector to run the command.



Motor Functions		
Function	Description	Returns
left.set(x)	Set the motor speed to x, a value from 1.0 to -1.0 (1.0 is full speed forwards, -1.0 is full speed backwards).	nothing (void)
right.set(x)		
left.getSpeed()	Returns the speed in inches per second of the motor.	double
right.getSpeed()		
left.getDistance()	Returns the distance in inches the motor has traveled.	double
right.getDistance()		

- **RED METHODS ARE DEPRECATED** (no longer supported, you wont need to use them for this assignment)