The Basics

An Intro To FRC Java Programming

Keegan Bruer

January 23, 2019

Contents

1 Introduction

Are you new to FRC or to programming with the WPI Library (WPILib)? This paper is a great place to start learning how to program using the WPI Library. This paper will mainly go over how to use the WPI Library with Java but almost all of the concepts discussed can be recreated in the language of your choosing. This paper does assume at least a beginning knowledge of the Java language.

2 Downloading and Setting Up Eclipse

Before we can get started programming, we need to download and setup Eclipse first. After installing Eclipse, we will need to install the WPI Library plugin for Eclipse.

To Download and Install Eclipse

1. Go to the website https://www.eclipse.org/downloads/packages/installer and download the installer for Eclipse.
2. Launch the installer program and if prompted by a security warning select "run".
3. Select the package "Eclipse IDE for Java Developers" (or something similar)
4. Select the install location for your eclipse IDE.
5. Then launch Eclipse once installed.

To Install The WPI Library Plugin

1. At the top of Eclipse, click on an option called "help".
2. Then click on the option titled "Install New Software" (close to the end of the list).
3. At the top right of the new window click on the button displaying "Add".
4. In the "location" field, add the web address "http://first.wpi.edu/FRC/roborio/release/eclipse/". In the "name" field, type "FRC WPILib". Then click "add".
5. In the field below there should be a option called "WPILib Robot Development". Click the option and select the checkbox next to "Robot Java Development". then click next.
6. Finally just go through and accept the terms and click finish.

3 Setting Up A New Project

To get started programming with the WPI Library you first have to create a new "Robot Java Project". Using the Robot Java Project with the WPI Library plugin for Eclipse allows us to easily use WPI Library functions without including all of the libraries individually.

To create a "Robot Java Program":

1. In the top left of Eclipse mouse over "file".
2. Then mouse over the option called "new" (it should be the first one on the list).

3. In the list, click on the option called "project" (This option is different then the option "Java Project").

4. In the list of possible project types find the folder called "WPILib Robot Java Development" then click on "Robot Java Project".

5. First, select the option called "Iterative Robot". Second, enter in the name of your project. Finally, click "Finish".

After your project has been created a new project will appear on the left side of your screen displaying the name of your project. To start editing your project you have to open the "Robot.java" class. The "Robot.java" class can be found by navigating into the project, then the folder "src", then the package "org.usfirst.frc.team5854.robot", and double clicking on the file "Robot.java". When you first open the "Robot.java" class there will be auto generated code in it. This auto generated code is great as an example but is not needed. Below is a blank template that can just be copied and pasted into the "Robot.java" file.

```
package org.usfirst.frc.team5854.robot;

import edu.wpi.first.wpilibj.IterativeRobot;

public class Robot extends IterativeRobot {
   @Override
   public void robotInit() {
   }
   @Override
   public void autonomousInit() {
   }
   @Override
   public void autonomousPeriodic() {
   }
   @Override
   public void teleopInit() {
   }
   @Override
   public void teleopPeriodic() {
   }
   @Override
   public void testInit() {
   }
   @Override
   public void testPeriodic() {
   }
}
```

4 Motor Controllers

Controlling a motor is one of the, if not the, most important aspects of a robot. Without being able to control a motor the robot would be unable to drive and most of the mechanism would not work. Therefore it is very important to understand how to control a motor. On the physical robot each motor is wired to a motor controller allowing programming to control the motor from

the RoboRio. Each different motor controller has their own object type. These types include, VictorSP, PWMVictorSPX, TalonSRX, Jaguar, SPARK, SPARK Max. The most common controllers are **VictorSP and TalonSRX**.

4.1 How They Work

4.2 How To Program Them

Programming a motor controller is pretty simple but before you can program them you will need to know what type of controller you are trying to program. The type of the motor controller should be printed somewhere on the controller itself.

In this example I show how to program two of the most common motor controllers.

```
public class Robot extends IterativeRobot {
   VictorSP victorController = new VictorSP(0);
   TalonSRX talonController = new TalonSRX(0);
   @Override
   public void teleopPeriodic() {
      victorController.set(1);
      talonController.set(ControlMode.PercentOutput, 1);
   }
}
```

5 Creating a Drivetrain