



# FIRST Robotics Competition Software Setup Guide

Programming Subteam

Ubuntu

Written & Provided by FRC 1923: The MidKnight Inventors  
West Windsor-Plainsboro Regional School District  
[www.FIRSTrobotics1923.org](http://www.FIRSTrobotics1923.org) | [@FRC1923](https://twitter.com/FRC1923) | [facebook.com/FRC1923](https://facebook.com/FRC1923)

# Table Of Contents

#Introduction

#Java

#Eclipse

#WPILib

#CTRE Toolsuite

#Git

#GRIP

#FRC Drivers Station

# INTRODUCTION

This guide will give you the tools you need to set up all necessary software for FRC Programming on a FRC team. The following sets of instructions will make sure you have the tools required to contribute to the robot's code.

## YOU WILL NEED:

1. An internet connection for software downloads.
2. Recommended:
  - Logging in to your system as an administrator, this will save you time as you will not be required to enter a password for every install.

## THINGS TO REMEMBER:

- If you need help, you can always reach out to your team leadership on Discord or via email. We are here to help you! This document just serves to cover frequently asked questions and our normal instructions.
- Wait for a step to finish before moving on, as programs may perform unexpected actions.
- The **white on black text** represents file paths.
- The **grey highlighted text** represents shell commands; run them in a terminal.
- This guide focuses on Ubuntu.
  - We have plans to release a macOS-based document in the future. Stay tuned on our website, [www.FIRSTrobotics1923.org](http://www.FIRSTrobotics1923.org)!
  - Our Windows version is available here: [insert view-only link here]

## Installation Guide:

# Java (JRE and JDK)



## INTRODUCTION

Java is one programming language option for FRC robots. Note that you can run either Java or C++ on the robot; our team uses Java. Because it is the language for the AP Computer Science exam, many schools offer a Java course. Java's syntax and structure is also more concise and easier to understand than C++, making it easier to learn.

## INSTALLATION

### Java Runtime Environment (JRE)

This is the program that allows Java programs to be executed. This is a fairly easy installation.

1. First run this command (you need this for later):

```
sudo apt-add-repository ppa:wpilib/toolchain
```

2. Next add Java (warning: you will get a deprecation notice):

```
sudo add-apt-repository ppa:webupd8team/java
```

3. Run the following command - you should see all of the packages that you just installed:

```
sudo apt-get update
```

4. Now, paste the following command as is:

```
sudo apt-get install \  
git \  
libc6-i386 \  
curl \  
jstest-gtk gradle oracle-java8-installer \  
frc-toolchain meshlab cmake libprotobuf-dev \  
libprotoc-dev protobuf-compiler
```

Most installations should have a \$HOME/Downloads directory, but this command will create it if it is not present: `mkdir -p $HOME/Downloads`

5. Now we will edit the file /etc/environment.  
`sudo nano /etc/environment`
6. Append a line with the following. This will define a system-wide variable named JAVA\_HOME that references the install location of Java on your machine.  
`JAVA_HOME="/usr/lib/jvm/java-8-oracle"`
7. Immediately load the /etc/environment configuration file you just created.  
`source /etc/environment`
8. List the contents of your Java installation directory using your new \$JAVA\_HOME variable.  
`ls $JAVA_HOME`

You should see content like this, which means that your system is properly referencing the Java installation directory we set above.

```
bin          LICENSE
COPYRIGHT    man
db           README.html
include      release
javafx-src.zip src.zip
jre          THIRDPARTYLICENSEREADME-JAVAFX.txt
lib          THIRDPARTYLICENSEREADME.txt
```

## Notes

For information on how the compile and execute process works in Java check out the following:

- How compilers work: <https://youtu.be/QXjU9qTsYCc>
- Java Program Execution: <https://youtu.be/G1ubVOl9IBw?t=1m2s>

You should now have java installed. Now you can begin setting up your development environment.

# Installation Guide:

# Eclipse



## INTRODUCTION

Eclipse is an Integrated Development Environment (IDE). Eclipse assists with development by providing features such as error reporting, suggestions, warnings and code autocompletion.

## INSTALLATION

1. Install eclipse:  
<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/oxygen/R2/eclipse-inst-linux64.tar.gz>
2. Open the folder in Nautilus (files)
3. Right click the folder with the .tar.gz extension and click 'extract here'
4. Open the eclipse-installer directory; double-click the eclipse-inst installer to begin the installation process.
5. When prompted, select Eclipse IDE for Java Developers as the flavor of Eclipse that we want. Note that Eclipse will be installed to \$HOME/eclipse/java-oxygen by default.
6. Follow the prompts to install Eclipse, and launch Eclipse after installation is complete
7. When prompted to choose a workspace, use the default value and click OK. The workspace is the directory where Eclipse will save your programming projects, and defaults to \$HOME/workspace.
8. Check on a couple of configuration options in Eclipse.  
Go to: Window -> Preferences -> Java -> Installed JREs.
9. Make sure the Oracle JDK8 is listed.  
Go to: Window -> Preferences -> General -> Workspace
10. Check Save automatically before build. Note that, if you used all the default settings, Eclipse should now be installed at **\$HOME/eclipse/java-oxygen/eclipse**

## Notes

Once Eclipse is set up you can start writing in java.

The following resources will help you learn the Java language:

- Java Beginners: <https://www.youtube.com/playlist?list=PLFE2CE09D83EE3E28>
- Object Oriented Java: [https://www.youtube.com/playlist?list=PLonJJ3BVjZW6\\_q8gh7XoLUhRlyBcYJLP](https://www.youtube.com/playlist?list=PLonJJ3BVjZW6_q8gh7XoLUhRlyBcYJLP)
- Java AP review: <https://www.youtube.com/watch?v=3Ky9MZyL8r4>
- Online Guide: <https://www.tutorialspoint.com/java/>
- Online Practice: <http://codingbat.com/java>

The following resources will help you write useful, concise and effective code that is easily understood:

- Java Documentation: <https://docs.oracle.com/javase/8/docs/api/>
- Reading javadocs: [https://www.tutorialspoint.com/java/java\\_documentation.htm](https://www.tutorialspoint.com/java/java_documentation.htm)
- UML diagrams: <https://www.ibm.com/developerworks/rational/library/769.html>
- Standardized java code style: <https://google.github.io/styleguide/javaguide.html>

You should have Eclipse up and running. Now you can write basic Java code!

Try printing “Hello, World!” to the console to make sure that everything works!

# Installation Guide:

# WPILIB



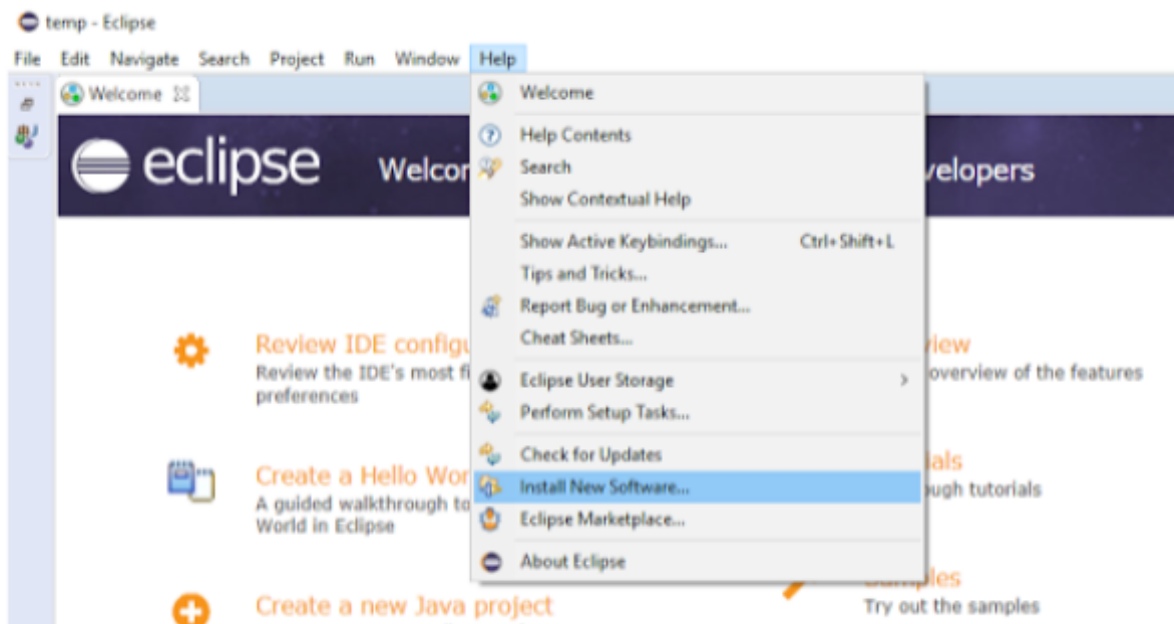
## INTRODUCTION

WPILIB is the tool that enables communication with the robot hardware. This tool allows communication to the drivers station and controllers. WPILIB is a plugin for eclipse. Plugins are small add ons that enable new features. WPILIB provides code, as well as debugging tools and a way to easily deploy your code to the robot.

## INSTALLATION

Installation is the same for all operating systems, all you need is eclipse and an internet connection.

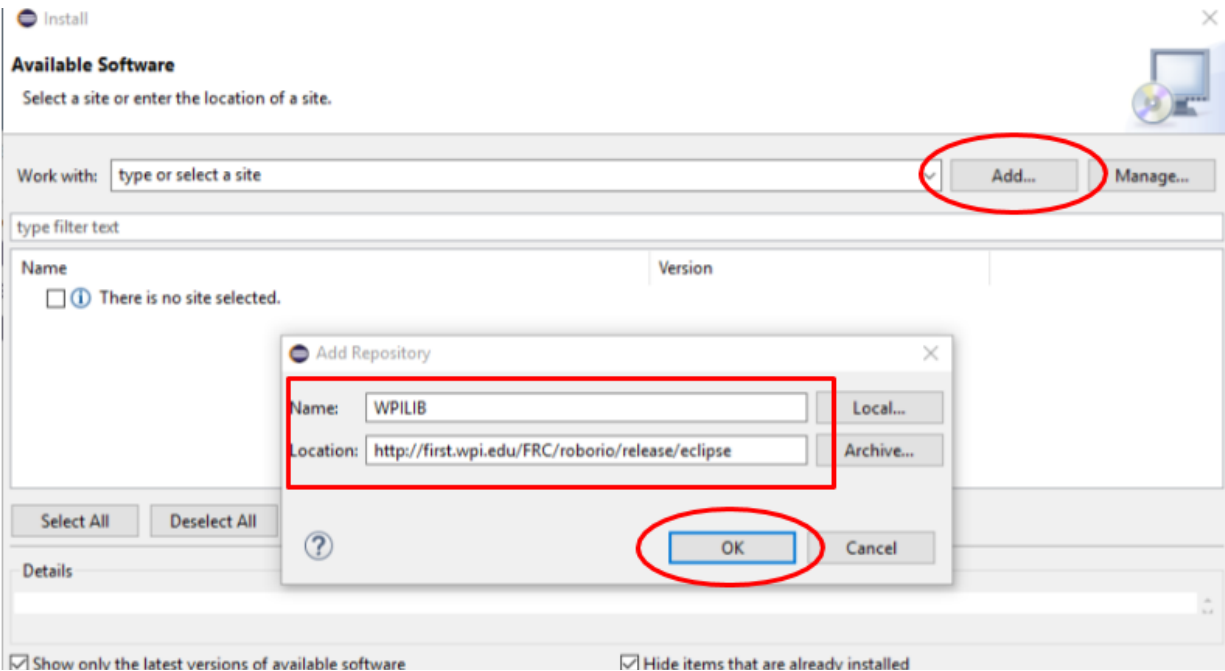
1. Open Eclipse.
2. Go to **Help -> Install New Software**



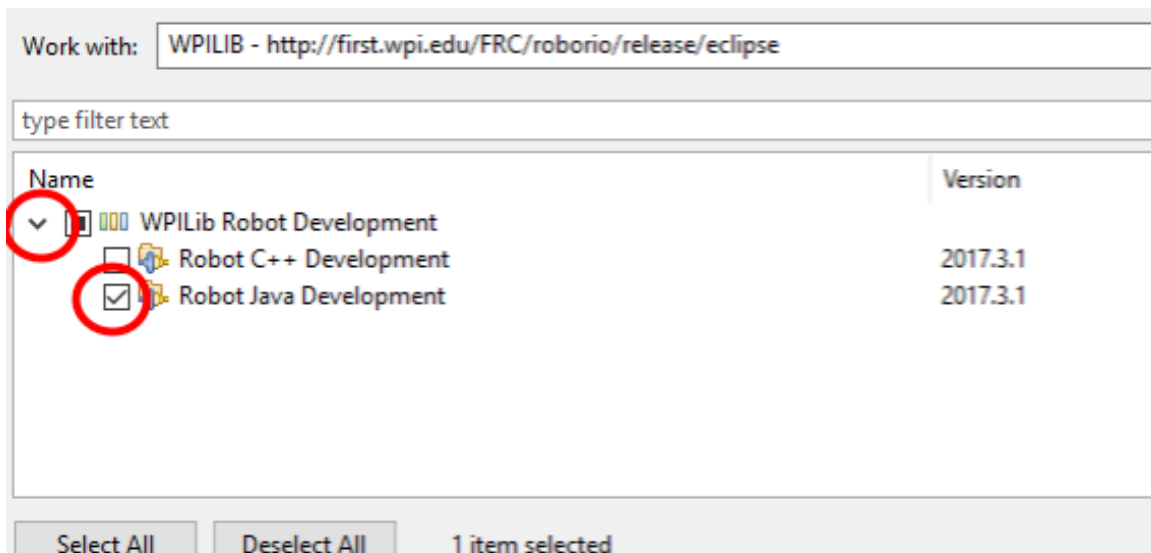
3. Click the "Add" button. A popup should appear



4. In the popup:
  - a. Type "WPILIB" in the "Name" textbox
  - b. Type <http://first.wpi.edu/FRC/roborio/release/eclipse>
  - c. Click "Ok"



5. The center area of the window should now have "WPILib Robot Development"
  - a. Click the arrow on the left to expand the selection
  - b. Select "Robot Java Development"



6. Click the "Next" button on the bottom of the window.
7. A new window will open. Click the "Next" button on the bottom.

8. Accept the license agreement and click “Finish”
9. You might see a security warning. If you see this warning click “Install Anyway”.
10. Leave eclipse alone while it installs WPILib in the background, this may take a while depending on your internet connection.
11. Click the “Restart Now” button when prompted.

Validating the install:

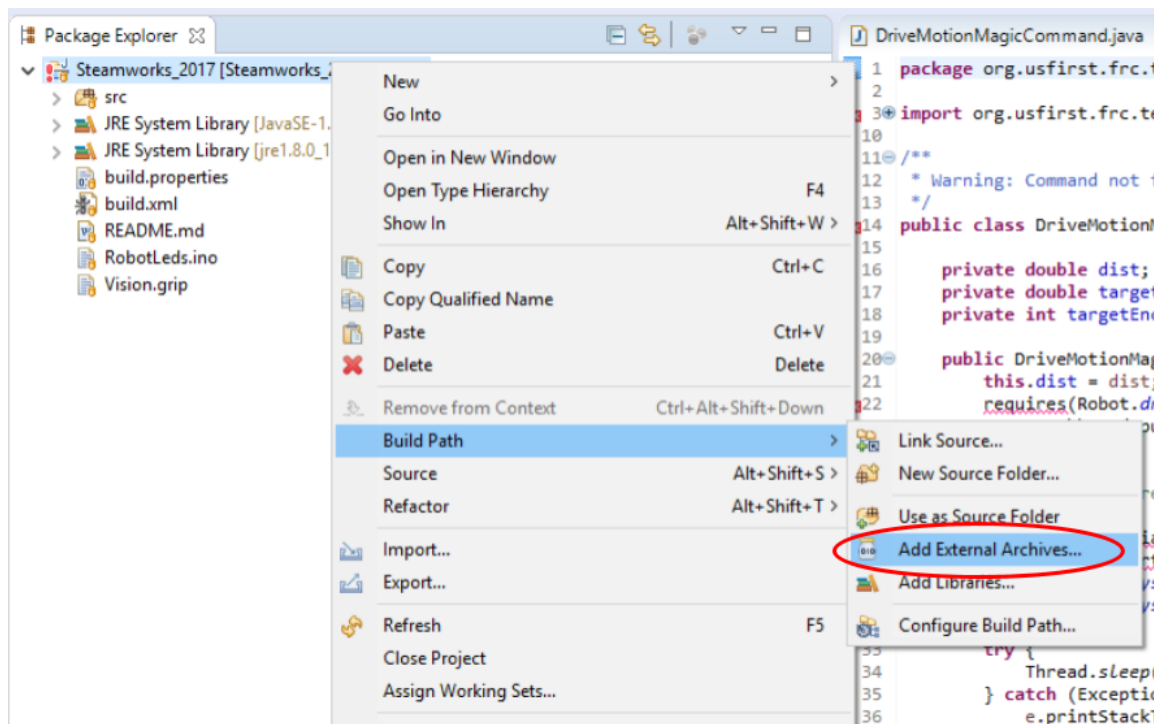
1. If you see “WPILib” on the menu bar (the one with “File”, “Edit”, etc) then you have correctly installed WPILib.

## SETUP

### Adding WPILib to a robot project

Even though you have the plugin installed, you may still encounter errors in your project. If you know your code should work, then the primary issue is probably that your project does not reference the correct jar files. If you have no errors, then you can skip this section.

1. Right click your robot project, go to **Build Path -> Add External Archives**



2. Navigate to `$HOME/wpilib/java/current/lib`
3. Highlight all the jar files, then click “Open”

# Notes

For information on using WPILib check out the following resources:

- WPILib Documentation: <https://wpilib.screenstepslive.com/s/4485>
- WPILIB API: <http://first.wpi.edu/FRC/roborio/release/docs/java/>
- Working project, 1923 Steamworks:  
[https://github.com/Team1923/Steamworks\\_2017](https://github.com/Team1923/Steamworks_2017)

You should have WPILib up and running. Now you can write and test robot code!

Try writing a basic program, like moving the robot forward, to make sure everything's running.

## Installation Guide:

# CTRE Toolsuite



## INTRODUCTION

In addition to WPILib, you'll need to install the CTRE Toolsuite. As of October 13 2016, WPILib no longer provides direct CAN Talon SRX support. This means that CTRE provides the interface for the talons as a 3rd party integration. We will be downloading the CTRE installer which places their libraries into the correct location on your system. WPILib tools will automatically detect the libraries and allow them to be used in FRC programs.

## INSTALLATION

Please note that this installer is only for Non-Windows computers.

1. Install CTRE Toolsuite: Since you are not on windows, you MUST use the following link:  
[http://www.ctr-electronics.com/downloads/lib/CTRE\\_FRCLibs\\_NON-WINDOWS\\_v4.4.1.14.zip](http://www.ctr-electronics.com/downloads/lib/CTRE_FRCLibs_NON-WINDOWS_v4.4.1.14.zip)
2. Unzip the file (We recommend unzipping to \$HOME)

## SETUP

1. Once installed, find where the CTRE jar is located in your filesystem, an example path is:  
`~/CTRE_FRCLibs_NON-WINDOWS_v4.4.1.14/java/lib/CTRLib.jar`
2. Open up eclipse, and right click on the project that uses Talons and add the CTRE.jar jar to the build path
3. Restart Eclipse (file -> Restart)

# Installation Guide:

# Git



## INTRODUCTION

Git is a version control system. It tracks changes and allows collaboration between large groups of people. Think Google Drive, but for code.

You might be confused between git and github. Git is a technology used for code collaboration. GitHub is a company that provides hosting services for git repositories. You can learn more about the difference here:

<https://www.codefellows.org/blog/git-and-github-what-s-the-difference/>

Before you start this installation we recommend that you create a GitHub account here:

<https://github.com/join>

For this guide we will install GitKraken: a GUI git client that makes it easy to use git, so we recommend it (GitHub Desktop isn't supported on Linux systems). If you are an advanced user, you can install the command line only version here: <https://git-scm.com/downloads>. Note that eclipse contains a git client as well.

## INSTALLATION

1. Visit <https://www.gitkraken.com/> and click "Download".
2. Run the file you just downloaded.
3. Let the installer run.
  - There should be nothing to do here, except to agree to a licence agreement if one appears. GitHub installation is very straightforward, and is already configured for your system.
4. Sign-in to your github account (or GitKraken account)

## Notes

- Some teams have their code on a private repository, to access the code you will need to contact your programming leadership. Make sure you include your GitHub username and email address in your request.
- For information on using Git and GitHub check out the following resources:
  - What is GitHub: <https://youtu.be/w3jLJU7DT5E>
  - Using GitHub Desktop: <https://help.github.com/desktop/guides/contributing-to-projects/>
  - Using git (basic): <https://www.git-tower.com/learn/git/ebook/en/desktop-gui/basics/what-is-version-control#start>
  - Command Line Tutorial: [https://youtu.be/SWYqp7iY\\_Tc](https://youtu.be/SWYqp7iY_Tc)
  - Git documentation: <https://git-scm.com/doc>
  - Using git (advanced): <https://www.git-tower.com/learn/git/ebook/en/command-line/basics/what-is-version-control#start>

You should have Git and GitHub setup and running. Now you can send off your contributions to your team!

Try creating a simple repository and tracking changes to it via git. Then try contributing to someone else's repository!

Once you feel comfortable with the GUI, move on to the command line!

# Installation Guide:

# GRIP



## INTRODUCTION

FIRST designed the field with elements that encourage teams to implement vision processing in their robot. GRIP is a program that enables teams to effectively use vision processing in their code.

## INSTALLATION

1. Visit <https://github.com/WPIRoboticsProjects/GRIP/releases> and download the appropriate file for your system.
2. Run the file
3. Click Install
4. Let the installer run.
  - There should be nothing to do here, except to agree to a licence agreement if one appears. This is a straightforward install.

## Notes

GRIP obviously requires a webcam, unless you are connecting to a live feed from the robot.

For information on using GRIP we recommend using the documentation and tutorials here: <https://github.com/WPIRoboticsProjects/GRIP/wiki>

You should have GRIP running setup. Now you can contribute to the forefront of computer vision programming! As a practice try isolating the image of a small brightly colored object.



## **FIRST Team 1923: The MidKnight Inventors**

West Windsor-Plainsboro Regional School District

[www.FIRSTrobotics1923.org](http://www.FIRSTrobotics1923.org) | [twitter @FRC1923](https://twitter.com/FRC1923) | [facebook.com/FRC1923](https://facebook.com/FRC1923)