C/C++ Programming For 2015 FRC Teams (includes Java Setup)

Mike Anderson
(robot_maker12@verizon.net)

Herndon High School
FRC Team #116

What We'll Talk About

- Goals
- Why C/C++?
- The development environment
- Talking to the RoboRIO
- Making it move
- Resources
- Summary

Goals

- The goal of this presentation is to help you understand how to use C/C++ in the development of your robot
- We clearly can't explain all of the aspects because we only have 75 minutes
- But, you should leave here with a better understanding of the process

Why C/C++?

- C/C++ is a standard in embedded systems programming for over 30 years
  - Used in many real-world robotics frameworks
    - ROS, Orinoco, OpenJAUS, etc.
- It's still the most predominant language in embedded Linux and the real-time operating system (RTOS) world
  - This gives your team valuable real-world experience
- It's compiled to native machine code
  - No virtual machine interpreters
    - No pausing due to garbage collection
  - It's fast
- It's the native language of the new RoboRIO's Linux-based operating system
  - The environment is written in C and Assembler
  - You get easy, direct access to the underlying O/S
- C++ is object oriented
  - Full support from WPILib

Top 10 Languages – Dec 2014

Dec 2014	Dec 2013	Change	Programming Language	Ratings
1	1		C	17.588%
2	2		Java	14.959%
3	3		Objective-C	9.130%
4	4		C++	6.104%
5	5		C#	4.328%
6	6		PHP	2.746%
7	10	^	JavaScript	2.433%
8	8		Python	2.287%
9	11	^	Visual Basic .NET	2.235%
10	12	^	Perl	1.826%

- LabVIEW was #45 on this list
- Many of the top 10 were all derived from C/C++

Why Not C/C++?

- C/C++ is compiled
  - This adds complexity to the build
- C/C++ is textual
  - There are no GUIs with lots of pallets of obscure symbols and squiggly lines ☺
- There is no hand-holding VM to catch your mistakes
  - You are responsible for memory management
- BTW, the Java VM is written in C/C++
- C/C++ has pointers
  - Objects can be referenced in many different ways
  - This concept can be complicated for some developers

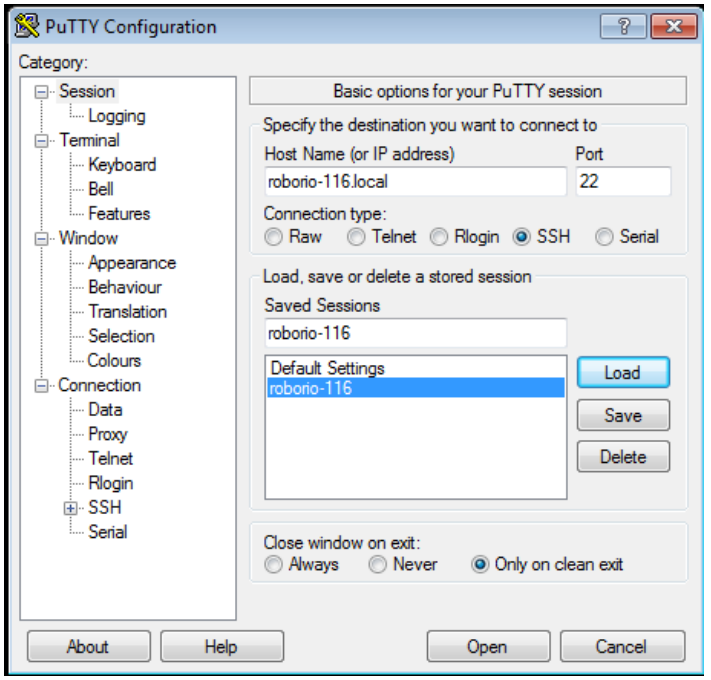
Getting Your RoboRIO Ready

- Before you can start development, you'll need to make sure that your RoboRIO has the proper operating system image on it
  - This is accomplished using the RoboRIO imaging tool or it can be done through LabVIEW

- Java developers will need to deploy the Java 8 JDK to the RoboRIO
  - There will be special instructions for this from FIRST

Some Useful Info…

- The RoboRIO runs Linux
  - SSH server is available
    - Use Putty on Windows to get to SSH shell
  - File transfers from IDE use SFTP
- Addressing is via mDNS
  - roborio-<team #>.local
- The web server requires Microsoft Silverlight plug-in
- Do not delete “admin” account and don’t add a password to it
  - All program transfers require it



The Development Environment

- The FIRST supported platform for C/C++ and Java is the Eclipse IDE tool
  - The beta teams are using the Luna release
  - The compiler is the open-source GNU 4.9 compiler
    - Supports C++11 extensions
- The compiler is actually a *cross-compiler*
  - We are building on an x86 for an ARM-based system
    - Again, this is a standard approach for commercial, embedded development
- During the beta, FIRST is only supporting Windows
  - However, libraries and compilers are available for OS/X and Linux

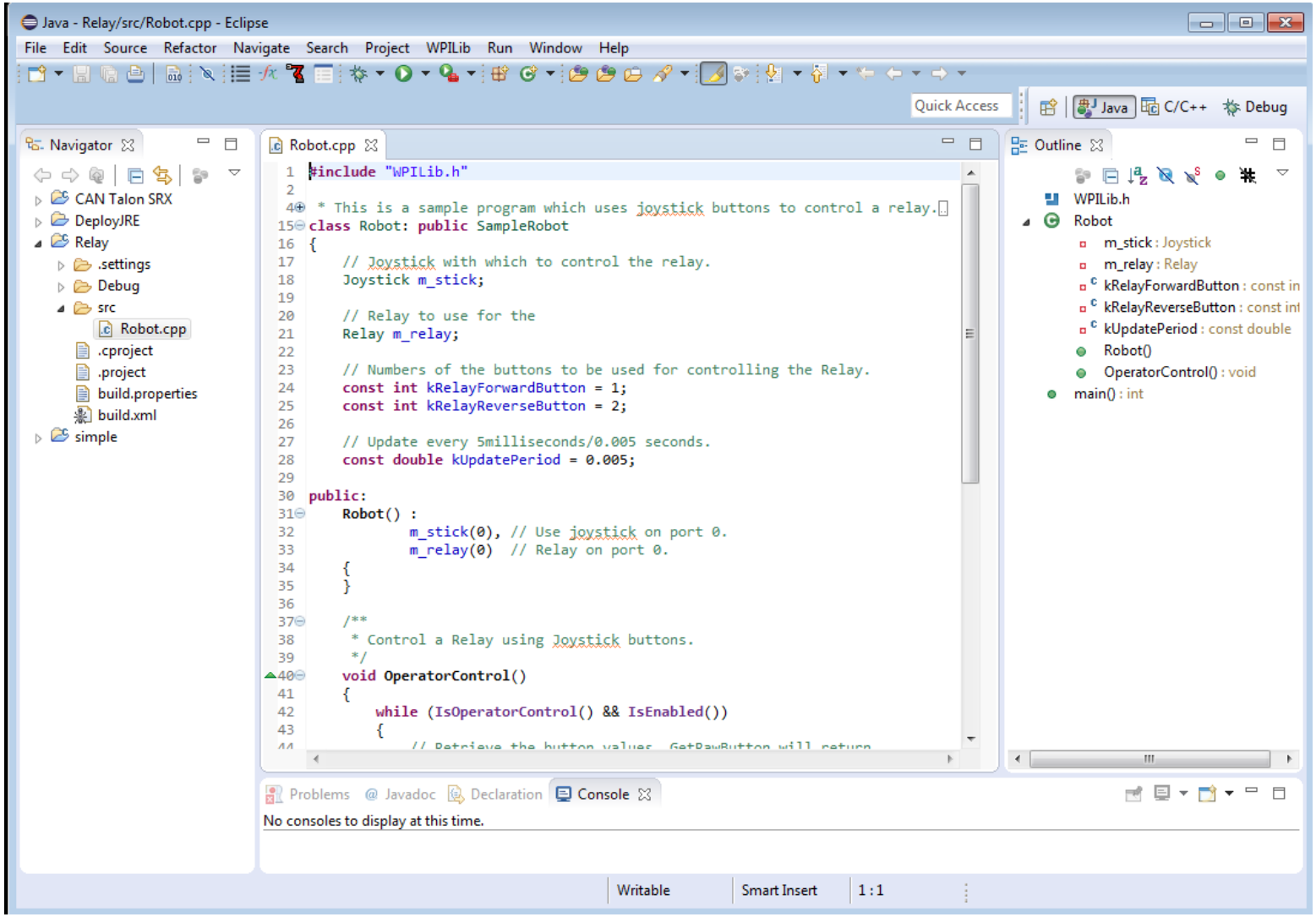
Development Environment #2

- The beta supports Windows 7 or Windows 8
  - You can still run it in a virtual machine
    - For all you Mac OS/X and Linux fans
- 64-bit Windows is supported
- During the beta, only Windows 7 is supported for the driver's station and dashboard

The Eclipse Luna IDE

- To get started, first install Oracle Java 8 on the Windows host
  - Either 32- or 64-bit depending on your host
- Next, go to Eclipse.org and download the Eclipse IDE for C/C++ developers
  - 32- or 64-bit version to match your Java install
  - Works for both C/C++ and Java development
- Once installed, you'll connect to FIRST to download the plug-ins for development
  - This may change for kick-off

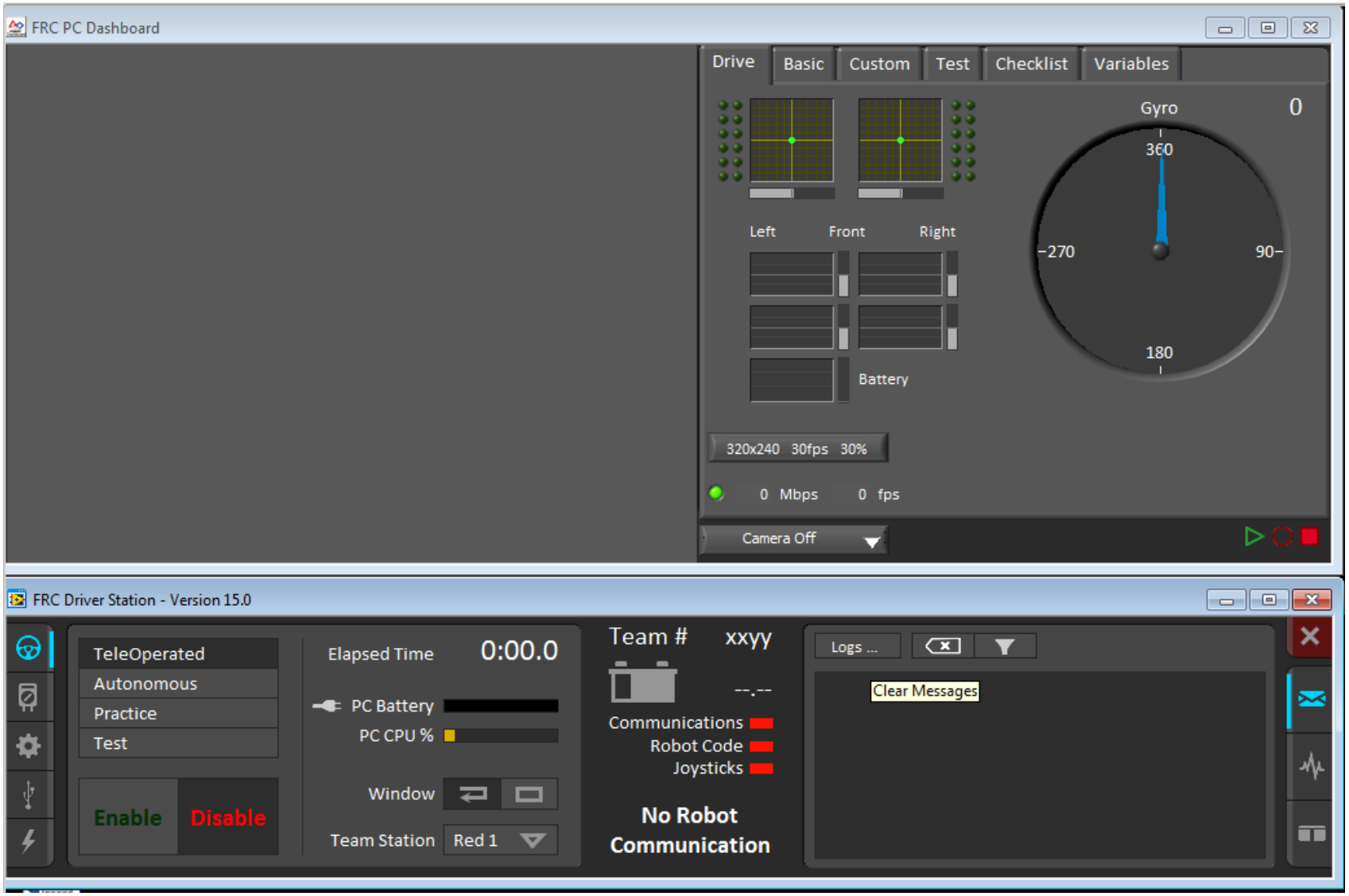
Installed Eclipse



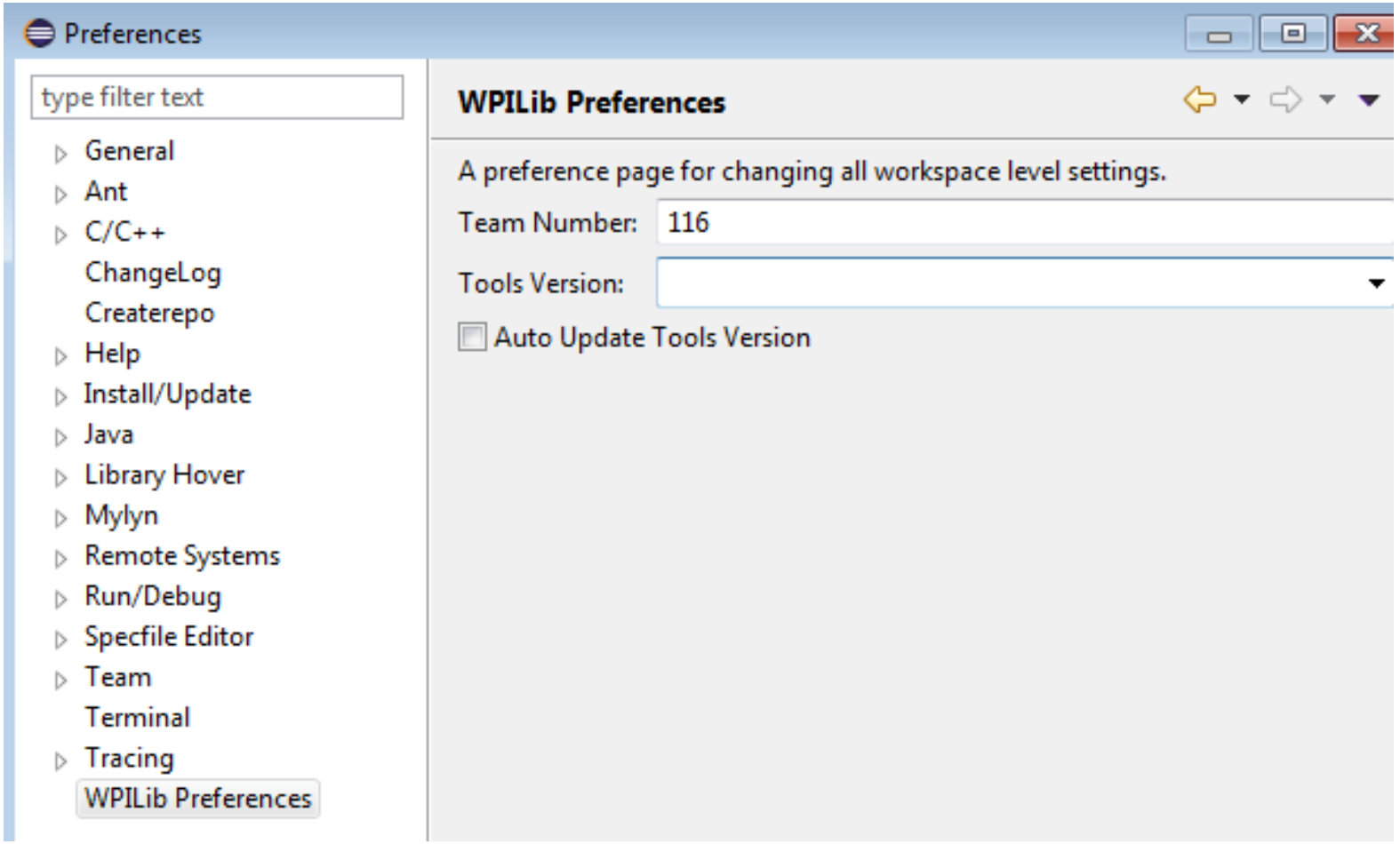
Install FRC 2015 Update Suite

- First, make sure you remove any old LabVIEW or driver station code
- Run the update suite installer
  - This will install the RoboRIO imaging tool and the latest firmware release
  - Installs the latest driver station and smartDashboard

New Driver Station



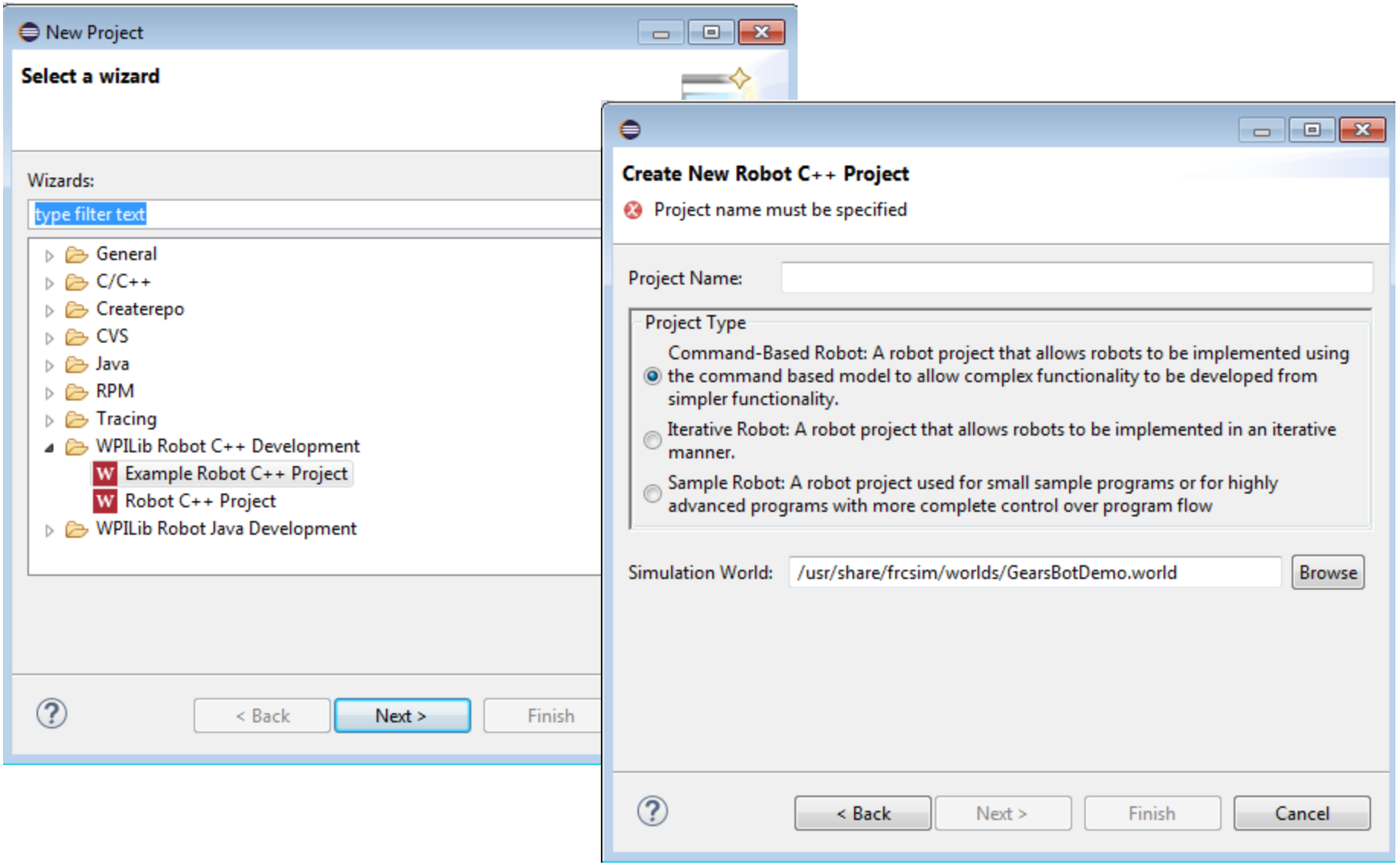
Set Team # in Preferences



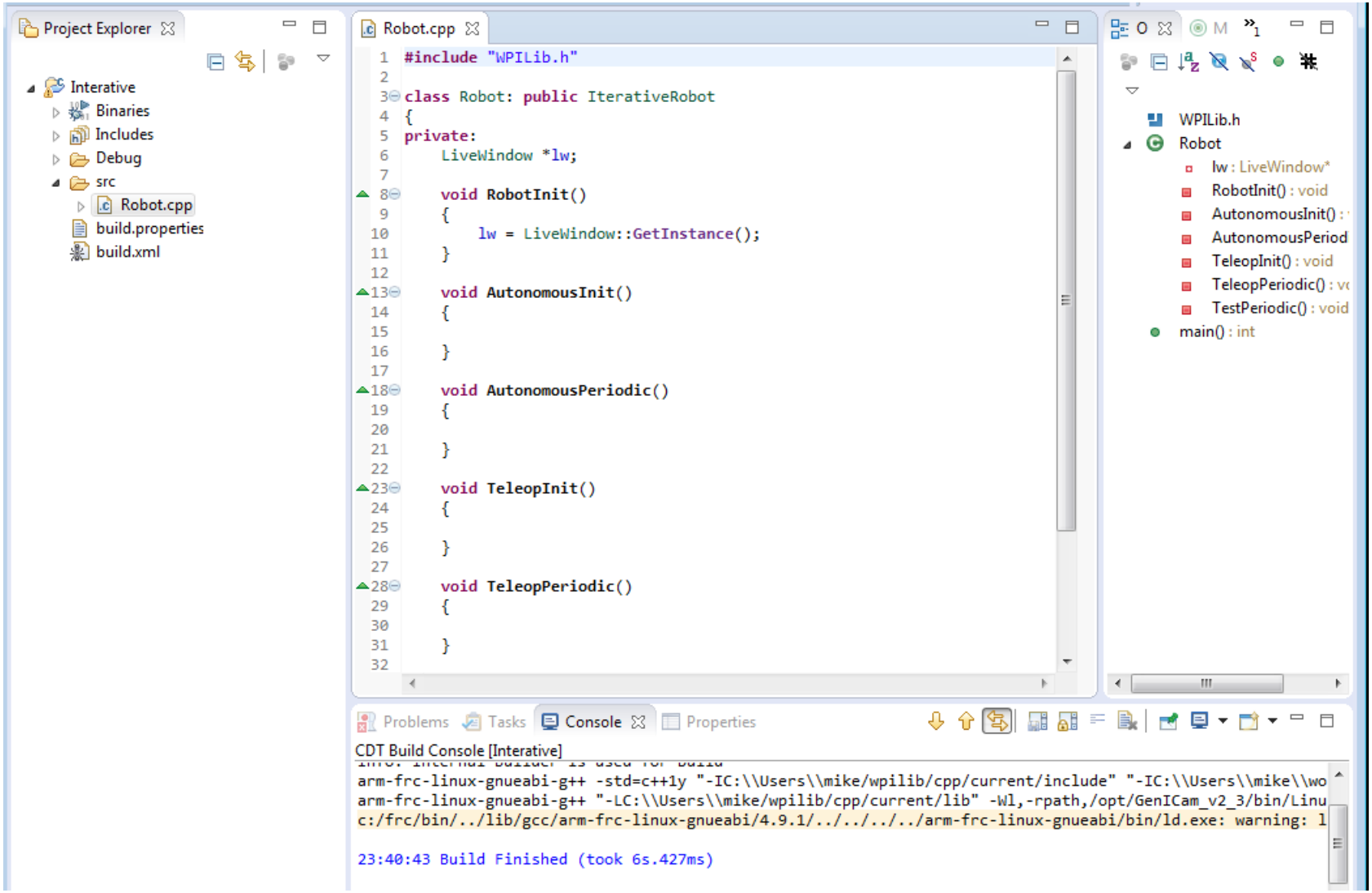
Creating A Project

- Eclipse collects all of the files related to building a piece of code into a subdirectory called a project folder
  - It's normally stored in your C:\user account
    - You can put the project elsewhere when you open Eclipse
- You can also import and export projects
  - This allows you to create a .zip of the project for archival purposes
- To create a new project use:
  - File->New->Project and select WPILib Project

New Project -- Simple Robot

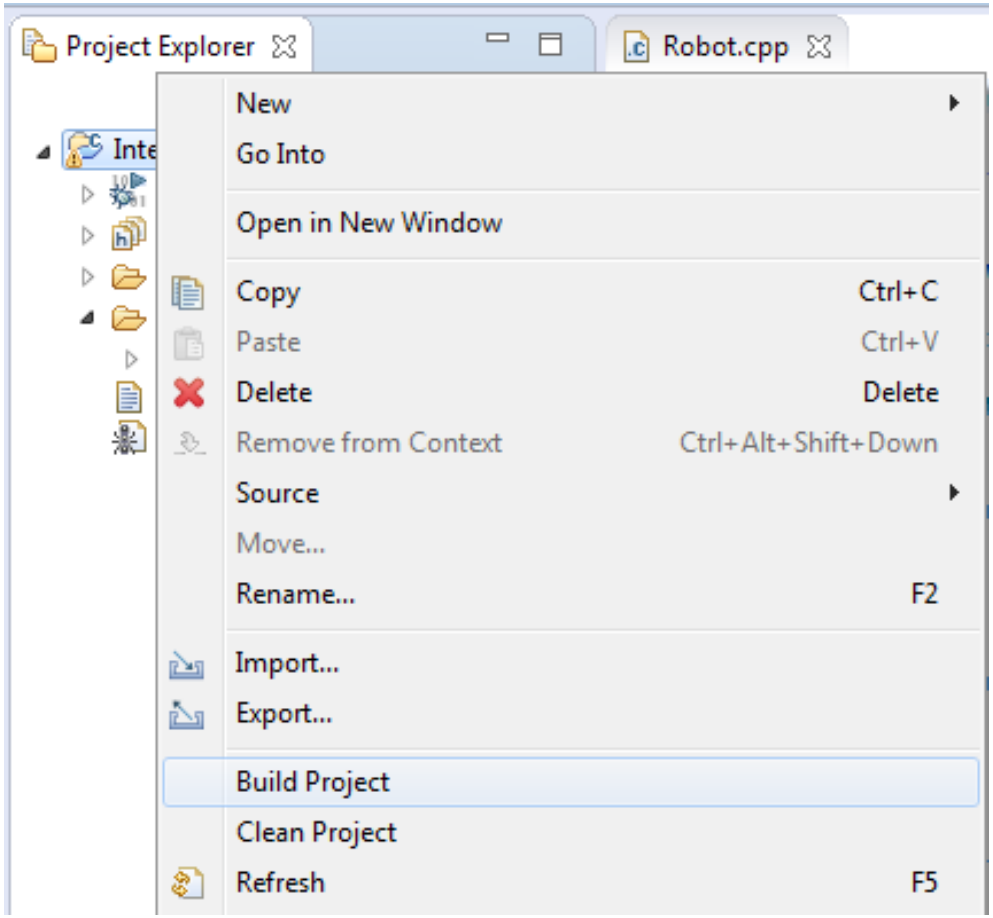


New Project Result



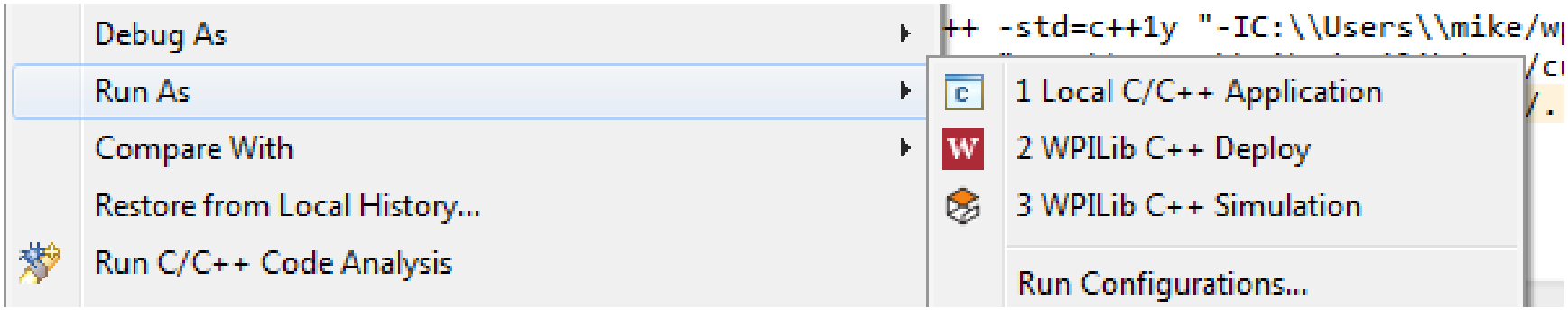
Build the Project

- Eclipse will default to building the project automatically
- However, you can clean and build the project manually
- Use the Project menu to configure the auto-build feature



Deploying to the Target

- When the code is built, you can select Run As->WPILib C++ Deploy

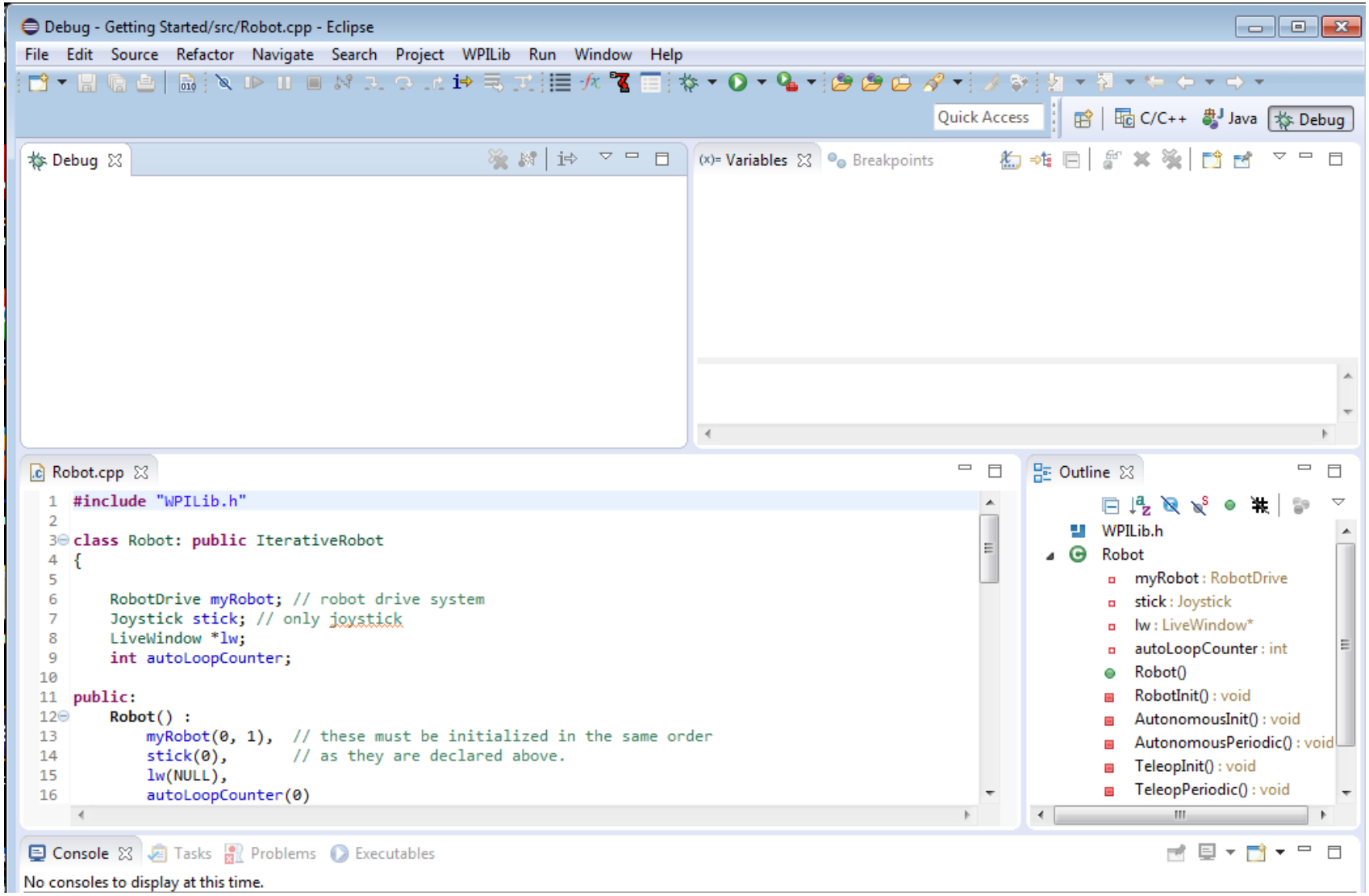


- This will open an SFTP connection to the RoboRIO (as “admin”) and copy the application to the file system
- The application will then start running
  - Waiting for the driver station

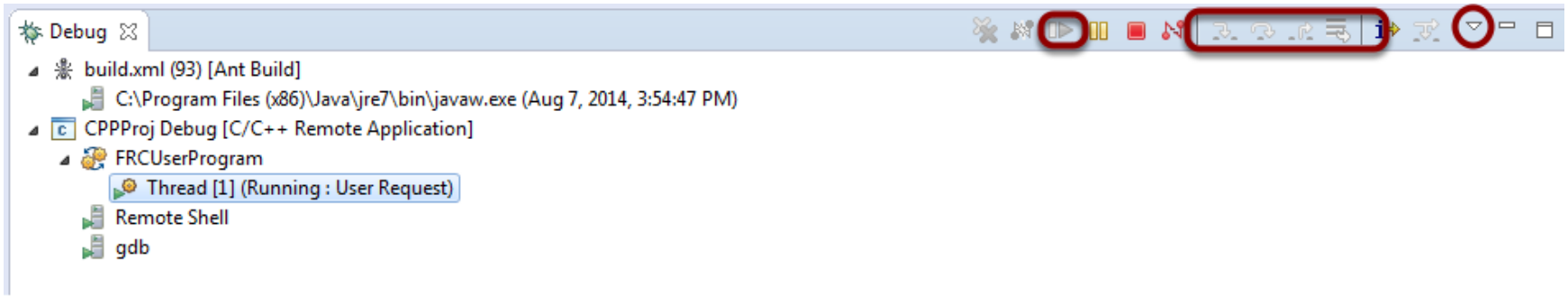
Debugging Code

- The project is usually automatically created with debugging enabled
- Select Debug As->WPILib C++ Deploy
- The Eclipse will automatically switch to the Debug perspective
- You can then set breakpoints
- Once the driver station enables the application, your debug session will begin

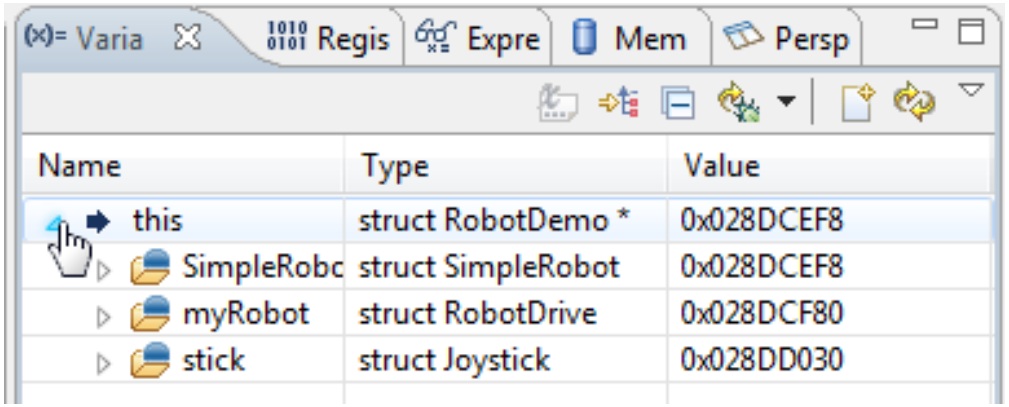
The Eclipse Debug Perspective



The Debug Windows




```
void Autonomous(void)
{
        myRobot.SetSafetyEnabled(false);
        myRobot.Drive(-0.5, 0.0);    // drive forwards half speed
        Wait(2.0);                   //    for 2 seconds
        myRobot.Drive(0.0, 0.0);     // stop robot
}
```



Resources

- Chief Delphi
  - http://www.chiefdelphi.com
- FIRST forums
  - http://forums.usfirst.org
- NI Community Forums
  - http://ni.com/FIRST
- WPI / *FIRST* NSF Community site (ThinkTank)
- These sites are monitored members of:
  - WPI
  - NI
  - *FIRST*
- All source code available for team-team assistance
- Phone support through NI
  - 866-511-6285 (1PM-7PM CST, M-F) ?

Summary

- C/C++ can be very challenging to new developers
  - C/C++ is similar enough to Java that Java developers can adapt to it quickly
    - However, pointers will require some explaining
- The rewards include:
  - Faster operation of the robot
  - Fine-grain control of the robot's behavior
  - The ability to leverage native Linux facilities
  - Taking a step towards future full employment ☺