What is debugging things?

debug: (v.) to identify and remove errors from computer hardware or software

The debug tool allows you to run a program interactively while watching the source code and the variable during the execution.

Why is debugging things important?

To ensure your program is able to run smoothly without any errors, debugging the program is recommended to remove bugs from your code (such as ants, beetles, mosquitoes, flies, etc).

How do I debug a thing?

Begin by setting breakpoints in your code. Breakpoints are used to specify where the execution of the program should stop to allow you to investigate variables, change their content, etc. If you want to stop the execution only if a field is read or modified, use watchpoints. To debug your application (besides the obvious solution of utilizing bug spray), select a Java file which can be executed, right-click on it and select *Debug As → Java Application*. If, for some reason, you didn't read and follow the first sentence of this section, your breakpoint-less code will run as normal. Upon starting the debugger, Eclipse (or some other IDE) may ask you if you want to open the Debug perspective. Politely accept Eclipse's offer by answering *Yes*, unless, of course, you don't want to open the perspective. Then you should politely decline, and perhaps provide an explanation to Eclipse so as to not come off as rude.

Okay, cool. What are some commands I can use when debugging things?

Voilà:

Key	Description
F5	Executes the currently selected line and goes to the next line in your program. If the selected line is a method call the debugger steps into the associated code.
F6	F6 steps over the call, i.e. it executes a method without stepping into it in the debugger.
F7	F7 steps out to the caller of the currently executed method. This finishes the execution of the current method and returns to the caller of this method.
F8	F8 tells the Eclipse debugger to resume the execution of the program code until is reaches the next breakpoint or watchpoint.

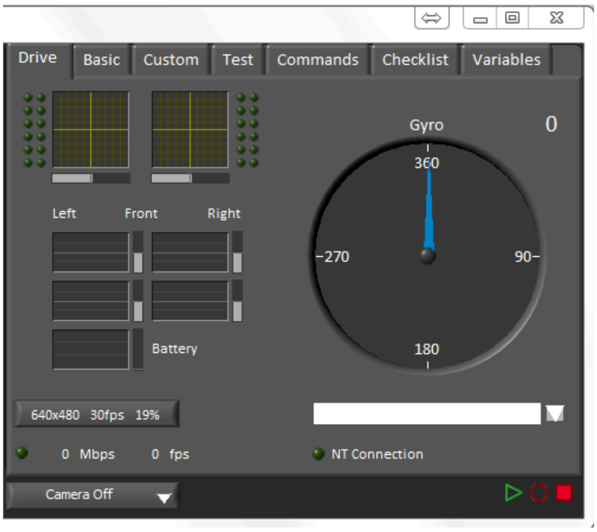
READ IT

Source: http://www.vogella.com/tutorials/EclipseDebugging/article.html

Yeah, umm, that's nice and all but I'm actually here to learn about how to fix any bugs with the driver station. Sorry.

Oh, uh, wow; this is awkward.

What are the camera settings I need to use?



640x480 30fps 19%

Camera Off

Ok, so you see this image on the left? And you see those settings that say "640x480 30fps 19%" and "Camera Off"? Those are settings for resolution, frame rate, compression, and camera mode. We don't use the settings shown in the picture here. Make sure those settings are set to **640x480** resolution, **30fps**, **46%** compression, and **IP Camera** mode.

What do I do when there's shorts?

Ok, I'm going to go ahead and assume you mean shorts in the code and not like shorts on your body because that's a whole different department. First, check the wires and confirm with the Electronics team that everything is wired correctly. If there isn't anything wrong with the wiring, click the "View Log File" button in the driver station, and change "Default" to "All Events, All Info". If brown circles appear, that means there's an error somewhere. If there's yellow bars, there is significant packet loss, which means it's not getting any information. If there's red spikes in CPU, that means there is a brownout or there are code problems. Diagnose the issue and solve accordingly.

For additional reference:
https://wpilib.screenstepslive.com/s/4485/m/24193/l/599678-driver-station-log-file-viewer

What IP configuration do I use?

Like other teams, we also follow the TE.AM IP Notation Format, thus our IP address is: **10.2.53.5**. Unless you lack sight or sufficient analytical knowledge, you can tell that our team number (253) is right there in the middle of the IP address. The numbers 10 and 5 at the ends do not have any significant symbolism. And also to prevent super bad things that are not good, do not put 9 at the end of the IP address. What's that? Did I stutter? **DO NOT PUT 9 AT THE END OF THE IP ADDRESS**, please. Doing so will avoid terrible stuff that are horrible and will not do nice things.

Hey, uhh, I can't connect to the robot.

Ok, ok, don't panic. Get your towel and go to the Control Panel on the laptop. Follow the following: Network and Internet > Network and Sharing Center > Change Adapter Settings > Right Click Wi-Fi > Properties. Check to see if Internet Protocol 4 is enabled because it should be.

I think I'm getting a "sticky fault". How do I solve this?

Ok, so first make sure your robot isn't completely sticky. This probably doesn't have anything to do with the sticky fault but if your robot is entirely submerged in maple syrup, I think you have bigger problems to worry about. In the case that your robot is nice and clean, first start by disabling Wi-Fi. Use Internet Explorer and go to **roboRIO-253-FRC.local**, or **172.22.11.2** if using USB. Log in with the username "admin" and no password. Select Self-Test and double or triple click the button until "Faults cleared!" appears. If it still isn't cleared, spam the hell out of that button. (No more useful information about sticky faults beyond this point.) Spam it like Cookie Clicker. Spam it like your trash email. Spam it like you're breaking your mouse but your only weapon is your finger. Spam it like you're a pro Starcraft gamer and have godly APM. Spam it like a can of fresh, salty, pre-cooked meat in World War II. Spam it like you're trying to quickscope with pre-nerf Widowmaker and you're preparing for a sick grappling hook headshot (which you won't hit). Spam it like the energy generated by your clicking single-fingeredly owers an entire continent. Spam it like you have 80 billion error messages because Windows screwed up and now you have to close them all manually. Spam it like your team overextends and you want them to retreat the hell back to spawn because no sane player should be pushed up that much at that low health but none of your teammates are responding to your pings and so naturally your team gets wiped and you get blamed for not being a part of that massacre which was a horrible idea in the first place and you are so tilted and salty right now.

What if the robot isn't getting code?

So you connect to the robot with the USB, and try running the code in Eclipse (or whatever IDE we're using) while the laptop is connected to the robot, which should upload the code. If you're receiving an error message that says "BUILD FAILURE", there is an error somewhere in the code. If you do not have sufficient knowledge of the robot's code to fix the error, seek help because code is pretty important to the robot.