1 Overview

This document will describe how to run the ratload program on GNU/Linux and OS X. You will need the source for `ratload`, which is available at at `http://github.com/jhladky/ratload/tree/master/src/`. This document will also describe how the Windows version of ratload is built for those interested. Please note this is a more technical document than the README. If you are only interested in running ratload then skip the next section, otherwise read on...

2 How ratload is Built

The ratload "program" actually consists of two programs, a CLI Java program and a python GUI that just calls the CLI program. The CLI program can be used totally independently. Each is fully self-contained. To achieve this independence both programs have to be compiled as native binaries, and statically linked with any dependent libraries.

For the CLI Java program this is acomplished with gcj, the GNU Compiler for Java. Gcj treats Java like gcc treats C, which obviates the need for the normal JVM. Additionally, since this is Windows, we compile with --static-libgcj and --static-libgcc, which bundles the gcj and gcc libraries into the executeable. The binary this process produces is about 40M. We can use *strip* to remove debug symbols and reduce that size by more than half, to 18M. A non-static version of the binary is about 400K, for comparison.

The python GUI program is made native with py2exe. Compared to using gcj this process is simple. A helper python script tells py2exe where to look and where to enter the program.

This process can be repeated on Linux and OS X if you want. You can install gcj and compile the CLI ratload program non-statically. You can even compile ratload statically on Linux if you really wanted to, although this is not recommended, and is not really necessary anyway.

3 Running ratload on OS X and Linux

Run ratload like you would any Java project of your own. No jar is provided, just compile Ratload.java with *javac* and run it with *java*. Make sure to include `jssc.jar` in the classpath when compiling and running.

Ratload can be run in the following configurations. You can mix and match the short and long forms of the options as you please; both are included here.

- `ratload -h|--help` Display a help message
- `ratload -l|--list` List available serial devices (only on the Windows command-line version)
- `ratload -d|--device <serial device> -t|--test` Test communication with the Nexys board through the specified serial device. This invocation of ratload is used in Section **??**
- `ratload -d|--device <serial device> -f|--file <prog_rom file>` The most common invocation of the ratload program. Send the specified prog_rom file to the Nexys board through the specified serial device.

A few other notes:

- To figure our your serial device, list the contents of the `/dev` folder. It will be called something like "ttyUSB0" in Linux, and "tty.usbserial" in OS X.
- You'll probably have to run ratload as root in order to access the serial device.