lab 2: prelab

In lab 2, you will practice modeling your embedded systems based on the example from lecture 3, and will learn some powerful new tools that bring us “closer to the metal.” You will then use these tools to build a tiny virtual world for your “firefly” to navigate.

In general, in order to build interesting embedded systems, you will want to become proficient at following online tutorials, reading documentation, installing new tools on your machine, and knowing where to ask for help when you get stuck. Getting other people’s code to run successfully on your machine can be a very annoying process, and it’s good mentally prepare for this. Fortunately, while you’re in this class you have support available to help if you get stuck.

For the prelab, you will install and set up a terminal, serial console, and text editor you can use for the remainder of the course. The goal is to turn your own laptop into a powerful portable development environment that you can use both in and outside the lab.

PRELAB: TERMINAL, SERIAL CONSOLE, TEXT EDITOR

A key embedded systems skill is to get comfortable using text-based “terminal” or “terminal emulator” programs, which offer lightweight, direct access to your computer. If you’ve never used a terminal, they may seem daunting at first, but they are essentially text-based versions of the graphical file explorers you are all familiar with.

For historical reasons, terminal support is much easier in UNIX-like environments like Linux or macOS. If using Windows I recommend experimenting with Windows Subsystem for Linux (Ubuntu) during this

course, or finding a Mac or Linux machine you can access as needed (even a Raspberry Pi!) Other tools to be aware of include PuTTY, Cygwin, PowerShell, Windows Terminal, and VS Code's integrated terminal.

For the prelab, your goal is to replace your Adafruit "Mu" editor with a more powerful text editor and serial console you can use for the rest of the course.

P2.1 INSTALL LOG

When setting up new tools or building software on your computer, it's a very good idea to keep track of the steps you took. If something doesn't work, or if it DOES work and you want to replicate it in the future, your notes will be invaluable to look back at once you've forgotten the exact details of your setup.

Create a folder on your machine (or however your prefer to organize your notes) to record what software you set up during this course and what steps you took to get it working succesfully. Open a new file in this folder and take notes as you proceed with the prelab. You will be asked to create a short guide to your setup as part of the final lab submission.

P2.2 TERMINAL

You may use any terminal program you like for the class, and you may want to research what options you think will work best for your machine. As a checkpoint, complete the following small task using whatever online resources you like:

- Install a terminal program of your choosing.
- Create a new ".txt" file on your computer, using any program you like.
- Using the terminal, navigate to the folder containing your file, then list the contents of the folder. You should see your text file in this list.

- Print the contents of your text file to your terminal window (in UNIX terminals and PowerShell you can use the “cat” command for this; in cmd.exe you can use “type”).

P2.3 TEXT EDITOR – VIM

Once you have a terminal installed, you will often want to edit files from the terminal. VIM is one good option for this; a lightweight text editor that’s extremely efficient at working with code, and very easy to install in resource constrained environments (e.g. a remote Linux terminal as part of an embedded system). It will also allow for some interesting visualizations in this course since the editor itself is programmable.

If not already available from your terminal, install Vim on your system. You may want to install a graphical version for ease of use outside the terminal window (see: gVim, Macvim, Neovim). Figure out how to run the “vimtutor” program on your system, and complete at least the first several slides.

P2.4 SERIAL CONSOLE – SCREEN, PUTTY, ETC.

Follow the Adafruit tutorial for your machine here, and connect to your RP2040’s REPL using a serial console.

- Advanced Serial Console on Windows
- Advanced Serial Console on Mac
- Advanced Serial Console on Linux

Once you’ve completed these steps, you are ready for Lab 2.