ECE471: Embedded Systems – Homework 2
Raspberry Pi and Linux

Due: Friday, 15 September 2023, 1:00pm

This Homework is meant to get you started with the Raspberry Pi so that you will be prepared for future homework assignments.

1. **Get Linux Working**

   * Install some form of Linux onto your Raspberry Pi. I recommend Raspbian, which has recently been renamed to "Raspberry Pi OS".
     - If you have an SD card with Linux already installed, you are all set.
     - If you need to install Linux on an SD card you can find instructions here:
       `https://www.raspberrypi.com/software/`
       You can get a raw image and install it, but they also have some imaging software there that can do advanced things like pre-configure the image before installing it.
   * Power up the Pi and ensure you can login.
     - To power it up, hook up a usb-micro cable to the proper port on the Pi (or if you have a Pi4, it will need a USB-C power supply)
     - As described in the class notes there are many ways to use the Pi. The most straightforward is to have a USB keyboard and a TV/monitor connected via HDMI.
     - Be sure the SD card with Linux is in the SD slot, plug in the power supply/connector, and it should boot up.
     - If you run into trouble there should be many tutorials on this online, or feel free to ask someone (including me) for help.
   * The first boot it might ask you to configure things, see the lecture notes if you have any issues with that.
   * Once you are configured, you might want to reboot the system.
   * Now it will boot up into either a GUI or text interface (you can configure that and other things with the *raspi-config* tool.
   * Use the username/password you picked to login. If it's an old version of Linux instead you might have to use the default `pi` / `raspberry` login. If that's the case, change the password as soon as possible with the `passwd` command.
   * If the GUI doesn't start by default, often you can use `startx` after logging in to get a graphics interface.
   * If you have trouble with any of these steps and need help, let me know as soon as possible!

2. **Copy the Assignment to your Raspberry Pi**

   * Download the code from:
     `https://web.eece.maine.edu/~vweaver/classes/ece471/ece471_hw2_code.tar.gz`
   * There are various ways you can do this.

- – If your pi is hooked up to the internet
    - ∗ At a command prompt you can use *wget* or *curl* to download the file. Just type the command and then type or paste the URL above after it.
    - ∗ Instead if you are running the GUI you can just use a web-browser to download it
    - ∗ You can also download the file on your desktop/laptop and copy it to your Pi using *scp*. This is part of the *ssh* remote login utils.
      There are also a number of GUI scp or sftp clients you can use to connect if you don't like the command line.
- Using a USB key
  - – Ideally you just plug the USB key in and if you are running the GUI it will pop up a window and let you find the file that way.
  - – If you are in text mode the USB key will appear somewhere in the /media directory tree.
  - – If you're really unlucky you might have to mount the USB key by hand, something like
    `sudo mount /dev/sda1 -o rw /media/usbkey`
  - – To copy `ece471_hw2_code.tar.gz` to your home directory type something like the following (it will vary based on exactly where the USB key got mounted)
    `cp /media/usbkey/ece471_hw2_code.tar.gz ~`
  - – When you are done with the usbkey, **before** removing it you have to unmount it; you should be able to do this from the GUI, or else you can manually
    `sudo umount /media/usbkey`

3. **Unpack the homework files**

   Uncompress/unpack it with the command
   `tar -xzvf ece471_hw2_code.tar.gz`

4. **Build the C files**

   - Change into the ece471_hw2_code directory
     `cd ece471_hw2_code`
   - Run `make` to build the code.
   - Run `./hello_world` and it should run!

5. **Modify the** `hello_world.c` **file** (5 pts)

   - Change the file so the output is 15 lines, each line looking something like this:
     `#1:  ECE471 MY_MESSAGE`
     Where the number after the # increments each line, ECE471 is always printed, and MY_MESSAGE is any message you want to print. Start numbering from 1. Full credit requires using some sort of loop (not just cut/paste).
   - You can use any text editor you want to do this coding. A simple one that is available on the Pi is `nano`, you can start it by running `nano hello_world.c`
   - Once you have updated the code, you can simply run `make` and it should recompile your code. Then test by running `./hello_world`
   - Be sure to comment your code! Also fix all compiler warnings!

6. **"Something Cool"** (1 pt)

   Copy your working code on top of the `something_cool.c` file.
   `cp hello_world.c something_cool.c`

   Then do one of the following.

   - Change the number of times your printing code loops based on a command line argument. The `atoi()` or `strtod()` functions might be useful for this.
   - Modify your `hello_world.c` file further to print the lines in different colors.
     HINT: Look up "ANSI escape codes"
   - (harder) Instead of printing everything one color, print each line a different color.
   - (even harder) Print each individual letter of the output a different color.

7. **Answer the following questions** (4pts total)

   Short answers are fine. Put your answers in the `README` file using a text editor, it will be automatically included in the submission process.

   (a) If you want to know more about the `ls` program, what command can you run?

   (b) What does the `-a` option to the `ls` program do?

   (c) List one reason why C is often used on embedded systems.

   (d) What method are you using to connect to your pi? (monitor/keyboard, network, serial, etc.?) How are you copying files back and forth?

8. **Submit your work**

   - Be sure your name and answers to questions are in the `README` file.
   - Run `make submit` which will create a `hw2_submit.tar.gz` file containing `README`, `Makefile`, `hello_world.c` and `something_cool.c`.
     You can verify the contents with `tar -tzvf hw2_submit.tar.gz`
   - e-mail the `hw2_submit.tar.gz` file to me by the homework deadline. Be sure to send the proper file!