Knowing Your Way Around Your Mbed LPC1768

It's not necessary to purchase your own Mbed LPC1768 for this class. We're focusing on Unit Testing, not System Testing. But we ARE building a real workable project and it might even be useful to you when we're finished. If you DO decide to purchase one from the manufacturer, here are a few notes we put together to make your life easier when using it:

Powering It Up & Connecting to Your PC / Mac / Linux box / Whatever

Powering it up is as simple as plugging in the USB connector… if you plug it into your development computer instead of a power adapter, you'll have established communication at the same time.

The LPC1768 should connect to your machine as two virtual connections. It will show up as a Mass Storage Device (like a USB Drive) and it will connect as a Virtual Comm Port. We'll be using both of these in this class.

Building the Latest Release

Building a release is fairly straightforward with Ceedling. Simply run the following from the command prompt while in your project's main directory:

```
ceedling release
```

This will produce a number of build artifacts in the `build/release` directory. The file `unit4.bin` is the release build itself. If running on an LPC1768 directly and programming with a JTAG interface, this is the file you want.

You'll also find a `unit4_crc.bin`. This file is the unit4.bin with the addition of a crc calculation, whose purpose is to work with the Mbed bootloader for LPC1768. Unfortunately, those maintaining the project are no longer publishing the method used to calculate this crc, and it has changed. Therefore, this build does not currently run. We've included the instructions below, just in case someone reverse engineers it and updates the toolchain. For now, however, you'll have to go with the next option.

Our last option is to use the online mbed compiler. Instead of needing to update ALL these files each time, though, we've attempted to make your life more convenient. When you generate a Ceedling release build, it generates a file in your `build` directory called `hot_melted.c`. You'll use this with the MBed compiler.

How to Program (Using the Online Mbed Compiler)

Go to `ide.mbed.com`.

The first time you go to this site, there is some initial setup. You may have to sign up for an account. Then, you can create a new project. Select `mbed LPC1768` as the platform. Then `Empty Program` as the Template. Feel free to name it whatever you want… possibly `MadScience`.

Each time you want to program your device, we're then going to take the hot_melted.c and copy its contents and paste them into `Main.c` in this sample project. Hit compile and a `hot_melted.LPC1768.bin` file will automatically be generated for you. You can then drag this file onto the "drive" that appears when you connect your LPC1768 to your laptop. Voila. Updated!

We know, it's cumbersome. If we are able to figure out the special sauce, we'll be updating the next method, because it's oh-so-much-simpler!

How to Program

Programming the device should be simple. In the build/release directory, locate `unit4_crc.bin`. This is the binary that has been updated to have a valid CRC calculated over the entire range. It's this copy that you want to use.

To program the LPC1768, all you need to do is copy `unit4_crc.bin` to the root directory of the mass storage device. Once it is fully copied, you can eject the device. Reboot. The firmware is magically updated!

How to Talk To It

As a Virtual Comm Port, it should show up on your list of standard serial ports. You can connect to it using whatever terminal application you like (ucon? minicom?). If asked about baudrate, it really doesn't matter because you're a VIRTUAL com port, but you can select 115200 and it will run just fine.

You will want to enable local echo of characters, because it's not going to echo characters back to you when you type them.

Remember to frame the messages properly according to the spec… all messages have been defined to be ascii, so you shouldn't have to worry about sending anything too tricky.