

# Cross Compilation Framework

*18–349 Embedded Real-Time Systems*

October 6, 2010

## 1 Introduction

For the remaining labs in this course, we have made a cross-compilation environment available to you. This tool chain and environment will allow you to store, develop and maintain your code on any linux machine (if you cannot access any other linux machine, you should be able to use the ECE color cluster) and use the same to compile your code.

**You will necessarily have to use the cross-compiler tool chain for labs 3 and 4.** The C libraries on verdex-pro were compiled to use a hardware FPU for all floating point computations (which would have been ok except for the fact that the PXA270 does not have an FPU). So if you are using any floating point operation (like a divide operation) your code will not compile using the gumstix compiler.

One of the biggest advantage of the cross-compiler tool chain is that you can use your favorite editor (no longer stuck with vi) to do the development and also the make process is quite fast on the cross-compiler tool chain. Moreover you won't have to wait for 10 minutes for your verdex-pro board to reboot in order for you to execute your application from U-Boot or develop your application in linux.

This is a guide on how to get this infrastructure working for you.

## 2 Cross Compiler Setup

To set up your workspace to correctly use the cross-compilation toolchain, please follow these steps:

1. Download the toolchain `opt.tar.gz` from the course website to your linux development machine.
2. If you don't have a linux development machine, log in to an ECE cluster machines that run Linux like `ece000.ece.cmu.edu` (you can use `ece000-ece031` but **NOT** the color cluster) using your ECE account and add `/afs/ece/class/ee349/gumstix/opt/bin` to your path. On bash or derivative shells (execute `/bin/bash` if the default shell is not bash), this can be performed with:

```
export PATH=/afs/ece/class/ee349/gumstix/opt/bin:$PATH
```

You may also add this to your `.bashrc` or your `.bash_profile`.

3. If you do have a linux development machine, un-tar `opt.tar.gz` in some folder (say the full path of the folder is `/canonical/path/to/folder`). Set your path variable to include the tool chain. On bash or derivative shells, this can be performed with:

```
export PATH=/canonical/path/to/folder/opt/bin:$PATH
```

You may also add this to your `.bashrc` or your `.bash_profile`.

4. Test the path — try to which `arm-linux-gcc` or try `arm-linux-gcc -v`. If these commands succeed, you know that you have set up the path to the cross-compilation toolchain correctly. The toolchain contains standard gcc and binutils programs such as `gcc`, `ld`, `as`, `ar`, `objdump` and `strip`.
5. Edit the Makefile at the root of your project directory to use the cross-compilation toolchain.

```
CCPREFIX = arm-linux-
```

Take one of your existing projects and compile it using:

```
make clobber && make
```

You now have a working cross-compilation environment on your/andrew machine. You may use this to develop and compile programs.

### 3 Local Setup

You now need to setup your local machine so as to be able to download the binaries that you generate on your/andrew machine to the gumstix. You will be using U-boot to download the binaries for you. U-boot requires files to be send over using the kermit protocol. Here is how you would transfer files:

On Windows Machines:

1. Use an scp program or an AFS mount to transfer the required bin files to your machine.
2. Power up the gumstix and interrupt the boot process. You will now be presented a GUM> prompt.
3. At the prompt, enter:

```
loadb a3000000
```

The gumstix should now print a message informing you that it is waiting for a file to be transfered.

4. Using Hyperterm's Send File feature, send over the required bin file, after selecting the kermit protocol in the drop box.
5. Repeat as necessary.
6. Type `go a3000000` to start your kernel.

On Unix-like Machines:

1. if you are using the ECE cluster machines then scp or rsync your binaries or use an AFS mount to transfer the bin files to your machine otherwise the binaries should already be on your machine.
2. If you don't already have it, install the ckermit package on your system using your favorite package manager.
3. You may be able to use minicom and integrate ckermit with minicom — we have not tried this out. Instead, in this section, we shall use kermit to connect to your gumstix.
4. Download the provided kermrc from course web site. Place it in your home directory as `/.kermrc`. Now invoke `kermit`.
5. Power up the gumstix and interrupt the boot process. You will now be presented a GUM> prompt.
6. At the prompt, enter:

```
loadb a3000000
```

The gumstix should now print a message informing you that it is waiting for a file to be transferred.

7. Use Ctrl-backslash C to escape from the gumstix to the local machine. Use `cd` to navigate to the directory with the required bin file. Once you have found the file:

```
send file_name.bin  
c
```

The last `c` was to tell kermit to “continue” listening to the gumstix.

8. You should be back to the `GUM>` prompt. To load one of your application programs, enter:

```
loadb a0000000
```

The gumstix should again print a message informing you that it is waiting for a file to be transferred.

9. Use Ctrl-backslash C to escape from the gumstix to the local machine. Again `cd` to the directory that contains the program and:

```
send app.bin  
c
```

This will load the program `app.bin` at address `0xa0000000`.

10. Type `go a3000000` on `GUM>` prompt to start your kernel (which, if written appropriately, should execute the app).

## 4 Warning

Some students might notice that programs that may have earlier worked when compiled on the gumstix have failed to work when compiled using the cross-compiler. Almost all of these cases have stemmed from a bug in the program and not in the cross-compile toolchain. The compiler provided here is newer and provides many more optimizations. This increases the chance that poorly written C will break. Please use all of the tools at your disposal to diagnose and remedy your problems.