EE458 - Embedded Systems
Intro to the Netburner

• Outline
  – Introduction to the MOD5270 Board
  – Getting Started with Development

• References
  – NNDK Getting Started Guide
  – NNDK Programmers Guide
  – NB Eclipse Getting Started
  – Netburner Runtime Libraries
Intro to the Netburner
Introduction: Hardware

- The Netburner MOD5270 module that we will be using is based on the 32-bit Freescale (Motorola) ColdFire MCF5270 processor.

- The processor runs at 148 MHz has 64 kB of internal SRAM, 8 kB cache, embedded 10/100 Ethernet, 3 UARTs, four 32 bit timers, four 16 bit programmable interrupt timers, four DMA channels, 47 digital I/O, watchdog and much more.
Intro to the Netburner
Introduction: Hardware

• The Netburner module board adds 2 MB of **SDRAM**, 512 KB of Flash memory, two 50 pin connectors and an RJ-45 Ethernet jack.

• The carrier board provides two RS-232 connectors, 8 programmable LEDs, 8 switches, a power connector and voltage regulation.
Intro to the Netburner

Introduction: Software

- The IDE is based on Eclipse.
- The GNU C/C++ tools are used for cross-compilation and debugging.
- The target real-time OS is uC/OS.
- The standard C libraries are the open source Cygnus Support and Math libraries.
- Many other libraries (serial, TCP, UDP, HTTP, HTML, email, RTC) are provided by Netburner.
By default, all software is installed to an **nburn** subdirectory under the root directory.

**NBEclipse** contains the Netburner customized version of Eclipse.

**gcc-m68k** contains all of the cross-compilation tools, standard C libraries and header files.

**lib** contains the **uC/OS** and Netburner libraries.
Intro to the Netburner
Nburn Directory Structure

- **include** contains the header files for the uC/OS and Netburner libraries.
- **system** contains the source code to the uC/OS and Netburner libraries.
- **pcbin** contains all of the Netburner host (Windows) tools. These tools are used to configure and download applications to the target board.
Intro to the Netburner

Nburn Directory Structure

- **pctools** contains the source code to all of the host tools.
- **examples** contains complete source code to several example applications.
- **docs** contains all of the documentation in PDF format. There is numerous documentation provided. It is described on the following slides.
Intro to the Netburner Nburn/docs Directory Structure

- Documentation is under the `docs` directory.
- The three most important manuals are the `uC/OS` and `Netburner Library References` in the `NetBurnerRuntimeLibrary` directory and the `NNDK Programmers Guide` in the `NetworkProgrammersGuide` directory.
- The `libc.pdf` and `libm.pdf` files in the `GNU` directory describe the standard C support (`printf`, `strlen`) and `math` (`cos`, `sqrt`) libraries.
Intro to the Netburner
Nburn/docs Directory Structure

• Typically one would read the *NNDK Getting Started* (*DocumentationOverview.pdf*) and the *NBEclipse Getting Started* (in the *Eclipse* directory) guides first, BUT these documents primarily assume that you are going to download applications over the Ethernet link. We will (initially) use only a serial link. I recommend looking over these documents, but keep in mind that we are doing things slightly differently.
Also of interest are the MOD5270 Hardware Manual (Mod5270.pdf) in the platform subdirectory and the ColdFire Programmers Reference Manual (CFPRM.pdf) and MCF5270 Reference Manual in the FreescaleManuals directory. (These are over 300 and 600 pages respectively.)

Embedded and External Flash memory programming guides are in the EFFS subdirectory.
Intro to the Netburner Nburn/docs Directory Structure

- The *Netburner Tools User's Manual* in the NetBurnerPcTools directory describes the provided host (Windows) tools. There is also information here on in-circuit debugging over either a serial or network connection.

- Finally, the Netburner website contains several application notes related to timer, interrupt, etc. programming.
void UserMain (void *pd)
{
    int n = 0;
    OSChangePrio (MAIN_PRIO);
    iprintf ("Application started\n");
    while (1) {
        iprintf ("Hello there ... %d\n", n++);
        OSTimeDly (20);
    }
}
Our initialization task is named `UserMain()`.

`OSChangePrio()` and `OSTimeDly()` are uC/OS functions. (Can you guess what they do from their names?)

`iprintf()` is similar to `printf()` but can only be used to display integer values (not floats). `printf()` could be used, but using `iprintf()` results in a smaller executable. Output is sent to the first serial port (UART 0).
Intro to the Netburner
NBEclipse Projects

- We will use NBEclipse to build our hello world program during the in-class exercise. These slides will briefly describe the process for you to reference later.

- Launch NBEclipse from the Start menu. If you are prompted for a Workspace I suggest you use an Nburn directory on a network drive. You can switch Workspaces later from the File menu.
Intro to the Netburner NBEclipse Projects

• To create a new project, from the File menu select New -> Project. In the Wizard that appears select NetBurner->NetBurner Project and click Next.

• Type in a Project name (a new directory of that name will be created in the Workspace directory) and click Next.
Intro to the Netburner NBEclipse Projects

- Choose to build both Release and Debug versions of the application and click Next.
- In the final window change the Target platform to MOD5270 and click Finish.
Intro to the Netburner
NBEclipse Quirks

- All source files (.c, .cpp, and .s files) in the project directory are automatically compiled and linked into the application when the project is built.

- You do not need to “add” files to a project, just copy source files into the directory. (You may need to refresh the project after copying files by right-clicking on the project folder in the C/C++ Project window and selecting Refresh from the menu.)
Intro to the Netburner
NBEclipse Quirks

- You can right-click on the project folder and select Import from the menu to import a file into the project. This will copy the file into the project directory.

- By default, a project will be rebuilt automatically whenever you save changes to a project file. This takes a little getting used to, but give it a try. (You can disable this feature from the Project menu.)
After building the project, the binary executable can be found in the Release subdirectory of the project directory. It will have a `.elf` extension.

The monitor program on the target requires that the executable be converted to Motorola S-record format (a text format). This is automatically done for you when you build the project. You will find files with `.s19` and `_APP.s19` extensions.
Intro to the Netburner
Download the Application

- The `.s19` file is intended to be downloaded directly into SDRAM. Any program in SDRAM will be lost when power is removed from the target.

- The `_APP.s19` file is a compressed version of the `.s19` file. The `_APP.s19` file should be downloaded to **Flash**. The monitor will automatically decompress and transfer the program to SDRAM when power is applied. (We will typically use the `_APP.s19` file.)
Intro to the Netburner
Download the Application

- Connect the power and serial cables. Connect the serial cable to the UART0 connector (just next to the power connector).
- Start the Mtty utility from the NBEclipse menu. Set the Port appropriately. Other settings: Baud – 115200, Parity – None, Data Bits – 8, Stop Bits – 1. Click Connect.
- Press the RESET button on the board and type the letter A into the Mtty utility within two seconds. (It must be a capital 'A'.)
Intro to the Netburner
Download the Application

- You should now see a prompt from the monitor program running on the target board (nb>). Type “help” to see a list of monitor commands. Type “fla” to download an application to Flash memory on the board.

- From the Mttyy Transfer menu select “Send File (Text) ...” (or press the F5 key). In the window that appears, browse to the project _APP.s19 file and select it. The download should start.
Intro to the Netburner
Running the Application

- After the download is complete the application should start automatically.
- Pressing RESET or removing and applying power should also start the application automatically. (The monitor program will decompress the application and transfer it from Flash to SDRAM.)
- You can also start an application loaded into Flash from the monitor prompt by typing "boot" at the prompt.
You can also use Putty/TeraTerm to connect (and even download files with TeraTerm) to the target board.

You can avoid NB Eclipse and use makefiles instead. This requires setting the PATH and several other environment variables correctly. I recommend using NB Eclipse.

Bring the board, serial, power and RED network cables to class each time. Please return everything at the end of the course.