Carleton University: Saturday Engineering and Design Session 2011

Computer Systems Engineering – Where Hardware meets Software

There are three programs that you can build and run in this lab. All are posted on the following website:

http://sce.carleton.ca/~mfloyd/SEDS2011/

About your account

Your computer monitor (or HC12 board) should have a yellow sticker with your computer number on it. This computer number will tell you what your username and password are for the computer:

username: *edcXX*
password: *edc*09*

For example, if your computer is number 03, your username would be "*edc3"*. If your computer number if 25, your username is "*edc25"*. All accounts use the same password, "*edc*09*".

All files to be saved must be saved on your **M**: drive. Make a folder there, called "**working**". Anytime you create any file, save it in this folder. **Start** by downloading the three programs from the webpage: LED.c, stepper.c and Dcmotor.c (right-click on the link and "Save Target As...").

General Instructions for Building and Running Any HC12 Program

With embedded programming, there are two stages to developing a program:

1. **Building**: **Edit** (write) the program and **compile** it (change it from source code to executable code)
2. **Running**: **Download** the executable (send it from the PC to the HC12) and **run** the program (execute).

In the lab, these two stages require the use of two different **tools** (programs)

Building a HC12 Program: Use the **ICC** compiler:

1. Start the ICC compiler (All Programs -> SYSC Software -> ImageCraft Development Tools -> ICC12)
2. **Create a Project**: Under the Project menu, select New. Use the browser to navigate to your working directory (M:\working). Type in the name of your project (Names provided below for each program).
   - *When saving the project, make sure that you add the file extension ".prj".*
3. **Add a File to the Project**. : Under the Project menu, select "Add File(s)...", selecting one of the files that you saved in your "**working**" folder.
4. Edit the file as directed.
5. When ready, **compile** : Under the Project menu, select "Make Project" (or press F9). Wait until Done. If there are any errors, ask for help.

Running a HC12 Program: Use the **NoICE** debugger:

1. Turn the power on your HC12 board **first (**switch on side of board**)**.
2. Start the NoICE debugger (All Programs -> SYSC Software -> NoIce -> NoIce for 68HC12)
   - *Important Check for the first time: Under the Options menu, select Target Communications. Ensure that the Interface textbox contains "NoICE Serial Protocol" and the Port is COM2 (for most boards; some will be COM1).*
   - *If at any time, you get an Error "No response from target" or if your cursor is a grey shaded circle, you have lost communications with the HC12 board. You must turn off-and-on the power for the HC12 board, and wait. If needed, repeat the process. If needed, close NOICE completely and start again, remembering to turn on the power for the HC12 board first.*
3. Download your executable. Under the File menu, select "Load...". Select the file type as Hex Files. Then browse to your working directory. Select the executable file (.S19) to be run, and click on Open. Wait until NOICE finishes.
4. Run your program. Under the Run menu, select "Go" (or press F5)
5. Your program will not stop. You will have to reset your board (turn the power off, then back on) and selecting "Halt" (from the Run menu) before downloading another program for execution.

Program 1: Turning on an LED

1. Study the HC12 board. Find the LEDs. Note that there are four LEDs, each of a different colour.
2. Build the LED project (See instructions above for Building a HC12 Program)
   a. Create a new project called LED.prj
   b. Add LED.c to this project.
   c. The code for LED.c will not automatically show. You must open it by double-clicking on the file's name shown in the top-right box. Look at the code and see if you can predict which coloured LED will turn on.
   d. Compile the program
3. Load the LED program (See instructions above for Running a HC12 Program)
4. Run it. What colour turns on?
5. Edit the program. Turn on another LED of your choice. You will need to recompile the code and reload the LED.S19 file.
6. In the same program, turn on one LED at a time, pausing between each one so that you can actually see the LEDs change.

   To pause, simply call a subroutine. Type: delay();
7. Close the project, to clear things before the next part. Under the Project menu in ICC12, select Close.

Program 2 : Moving the Stepper Motor

1. Study the HC12 board. Find the stepper motor. It has a little black knob on top with a white stripe drawn across it so that you can see its movement.
2. Build the Stepper project (See instructions above for Building a HC12 Program)
   a. Create a new project called stepper.prj
   b. Add stepper.c to this project.
   c. The code for stepper.c will not automatically show. You must open it by double-clicking on the file's name shown in the top-right box.
   d. The main code has two **for-loops** that each repeat one time. The **first loop** should turn the stepper motor **clockwise** and the **second loop** should turn the stepper motor **counter-clockwise**. **Your job is to complete the code** within the loop by activating the stators in the proper sequence. **Use the lecture notes to remind you.** Compile the program
3. Run the Stepper program (See instructions above for Running a HC12 Program)
   a. Download the program. How far did the stepper motor actually turn in either direction?
4. Edit the program: Make the stepper motor turn 2 complete revolutions in either direction.
5. Close the project, to clear things before the next part. Under the File menu, elect Close.

Program 3: Controlling the Speed of the DC Motor

1. Study the HC12 board. Find the DC motor. It should be next to the stepper motor and have a wheel attached to it.
2. Build the DCMotor project (See instructions above for Building a HC12 Program)
   a. Create a new project called DCmotor.prj
   b. Add DCmotor.c to the project.
   c. The code for DCmotor.c will not automatically show. You must open it by double-clicking on the file's name shown in the top-right box.
   d. The main code first slowly turns on the DC motor, revving it up to a "maxSpeed". It then lets the motor run for a bit at this speed. Finally, if slowly brakes the motor. The acceleration and deceleration take place within two for-loops. **Leave the program as is for now**.
3. Compile the program.
4. Download and run the program. What direction does the motor turn?
5. Edit the program: Make the motor turn in the opposite direction.
6. Optional: Change the maxSpeed of the motor although PLEASE do not go outside the values specified in the program's comments (20 – 35).