Keil Development Software

Some of what you need to know to run the Keil development software is covered in Chapter 9 of **C and the 8051**. Some is covered in this handout. The complete manuals for Keil are available online (under the books tab in the project window). I strongly recommend you look over chapters 4 and 5 of the Tools User's Guide - uVision2 Getting Started manual under this tab. Most of the basic operations of the software are covered here. These chapters will answer your most basic questions.

The following is a (very) brief outline of what you need to do to write and debug a program with Keil.

- Start uVision from the network in the CLC (Open Application Explorer - UMR - Run KeilPro6.01 - KeilPro601 Launch. This may change.) or from your harddrive (if the evaluation version was installed on your harddrive, there should be a Keil icon on your desktop), or from your CD (¡CD-Drive¿:/Keil/UV2/Uv2.exe). The full network version has greater capability than the evaluation version on your CD. I recommend installing the evaluation version, if you use it, rather than running from the CD.

- Create project (in your home directory, on a:, or on *your* hard-drive) using the menu: "Project - New Project" and then entering the name of the project.

- Select the Device for the target. This is the 8051 family member for which you are trying to create a program. For most of our projects, I'd recommend the Generic-8051 device.

- *ASIDE.* Now that you've specified the target device, you can view the Keil manuals associated with that device. Pick the "Books" tab in the project window and double click on the manual of interest to view that manual.

- Create new file from menu: "File-New".

- Write your C or ASM program within the code window that pops up.

- Save program with .a51 extension (filename.a51) if an ASM program and with a .C extension (filename.C) if a C program. Menu item "File-Save As".

- Add program to project. Hit menu item "Project-Targets, Groups, Files", select the "Groups/Add Files" tab, select the group "Source Group 1", press the button "Add Files to Group", select your file, and then press "Add". If done correctly, you should see your file appear under the "Target-Group" file in the project window (You must have the "files" tab selected to see this).

- *ASIDE.* Keil has provided special startup code that clears memory, sets memory banks, and the like if you wish to include it (for most of our projects you shouldn't have to). To include it, copy the file "STARTUP.a51" from /Keil/C51/Lib into your directory and then add STARTUP.a51 to your project. In most cases, you will need to edit variables in STARTUP.a51 for your particular application.

- *ASIDE.* If you have special options to set for your project, now is a good time to set them. Use the “Project - Options for Target” menu item. The default selections should be appropriate for most of our projects. If you are generating a hex file to download to a chip, you will need to select the Output tab and check the “Create HEX File” box.

- Compile your code using the “Project - Build target” menu item. If any errors occur, correct them, save program, and re-compile. You can see where the error occurred in your code by double clicking on the error message in the output (build) window.

- Run the simulator/debugger by pressing the “Debug - Start/Stop Debug Session” menu item. Here are a few hints on using the debugger:

  - Some good windows to have up are listed below. You can bring them up under the “View” menu item.

    - Project Window with the “Regs” tab selected – this will show you the value of any special function registers
    - Output Window – this will show you results of a compilation and also acts as a window in which you can enter commands to the debugger.
    - Watch and Call Stack Window – you can watch individual variables here. Your local variables (local to the function that is currently being run) are shown under the “Locals” tab. You can put your own variables under the “Watch” tabs. Just select the watch tab, hit F2 to enter a variable, and then enter the variable whose value you want to watch.

    Some other semi-useful windows include:

    - Disassembly window – this will show your code along with the assembled machine code that is fed to the 8051. Useful to see the compiled results of your code.
    - Symbol window – will show all the variables used in your code and where and how they are defined.
    - Memory window – shows contents of memory. You can specify code memory (Ex. C:0x0000,C:0x0042), internal direct data memory (Ex. D:0x00,D:0x7F), external data memory (Ex. X:0x5280,X:0x52FF), internal bit-addressable memory (Ex. B:0x00,B:0xFF), or internal indirect data memory (Ex. I:0x7F,I:0xFF).

  - “Reset” the 8051 for each run through your program using the “RST” button or the “Peripherals - RESET CPU” menu item.
  - Set breakpoints in your program by either double clicking to the right of the instruction where you’d like to set a breakpoint or by selecting the instruction, hitting the right mouse button, and selecting “Insert/Remove breakpoint”.
  - You can run your code in 3 ways: you can RUN until the next break point, you can RUN until the program hits the instruction highlighted by your cursor, or you can single step through the code. These options can be chosen from the “Debug” menu item or by pressing buttons on the debug toolbar (see other Keil handout).
  - Values for program variables can be entered by selecting the variable in the register or local window, waiting a second, and then pressing the left mouse button a second time.

This will allow you to edit the current value of the variable. You may also change a variables value by entering it in the output window. For example, to make the variable i contain the value 42, enter “i=42”. You can also do simple math, for example “i=a+42”.