

CECS 285: Lab1

Keil 8051 Basics – Due 1/21/2016

OBJECTIVES:

- Become familiar with the Keil Development environment.
- To observe how instructions are stored in ROM and data is stored in memory.

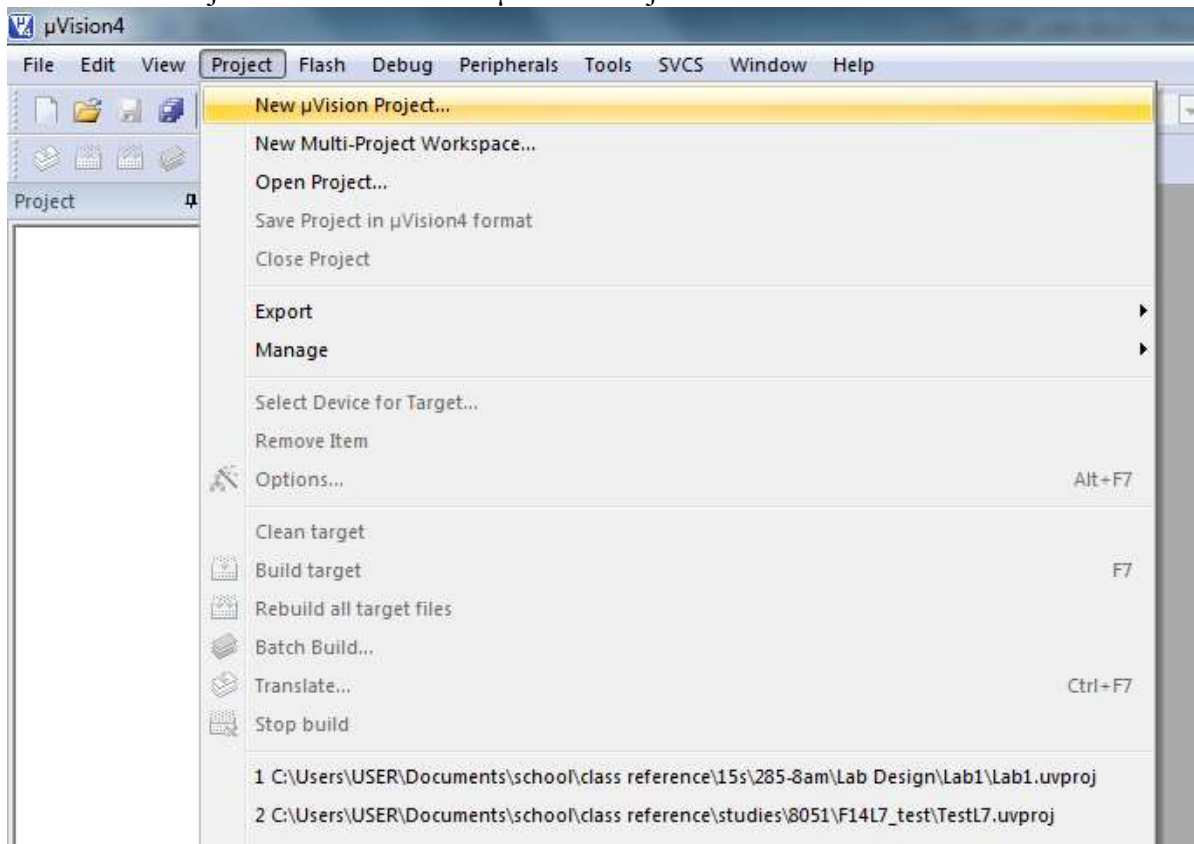
ACTIVITY:

Use the Keil development environment to run the following simple program and observe the contents of memory. This document will guide you through the steps of setting up a project and executing a simple program in debug simulation.

1. Open Keil



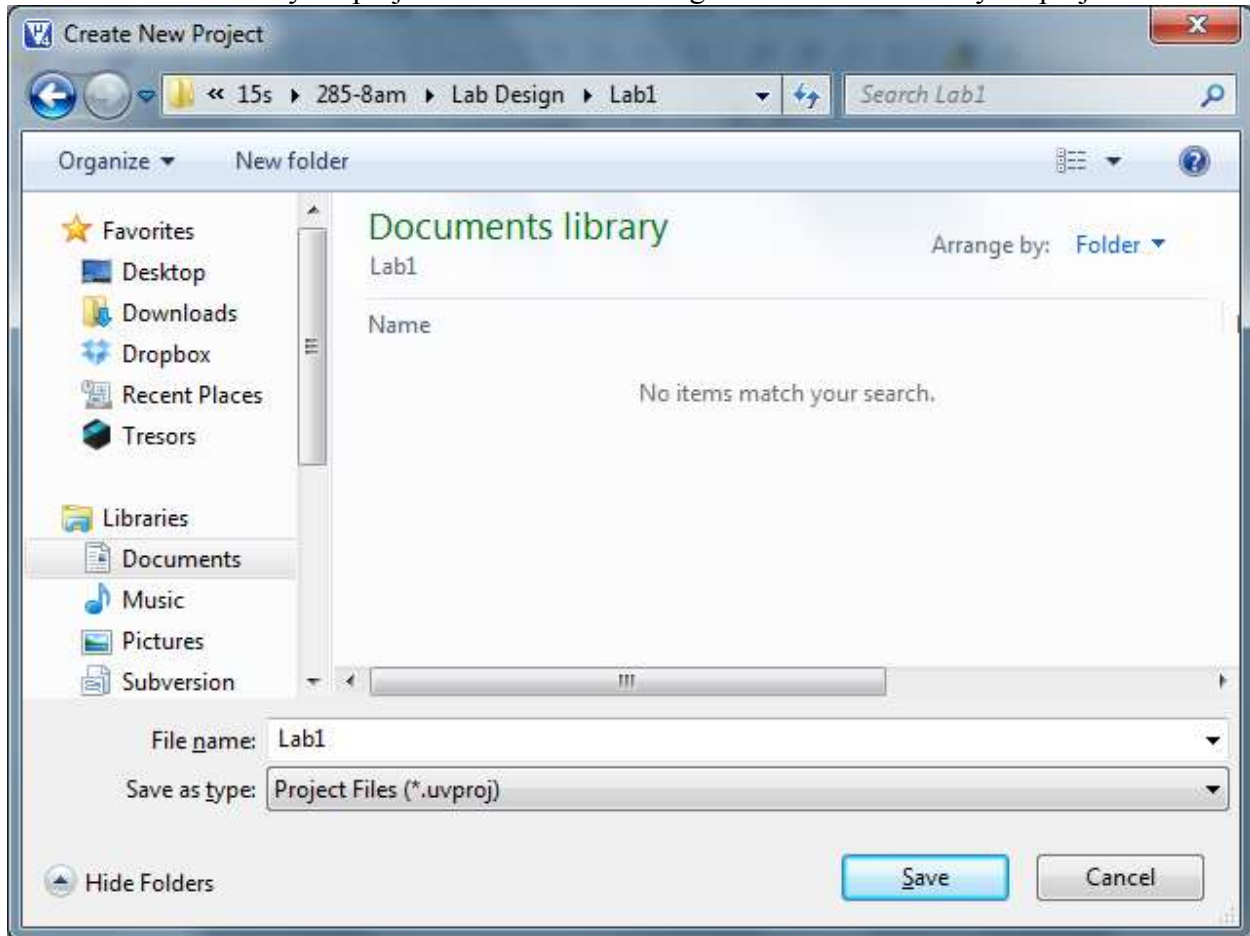
2. From the Project menu select “New μ Vision Project”



CECS 285: Lab1

Keil 8051 Basics – Due 1/21/2016

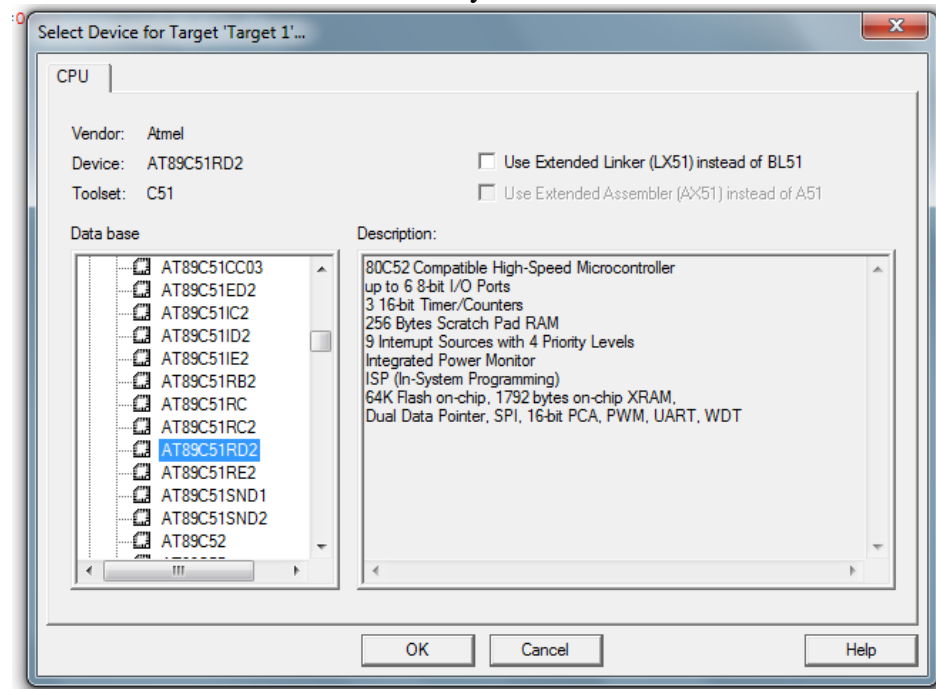
3. Create a Folder for your project named “Lab1” and give the same name to your project file.



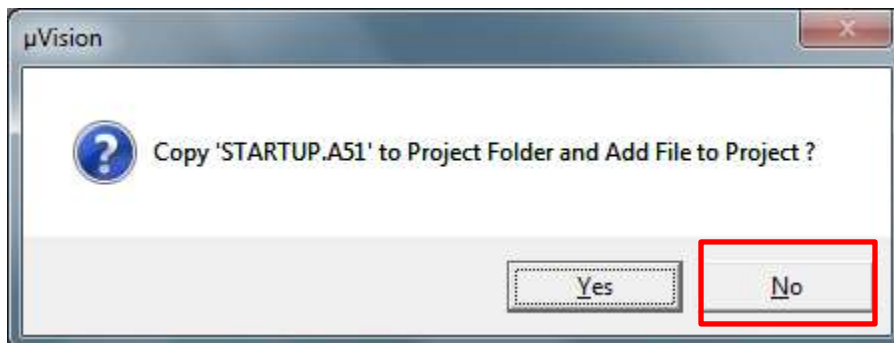
CECS 285: Lab1

Keil 8051 Basics – Due 1/21/2016

4. *The device you select will only affect debug timings.* For now, in Device Type selection choose Atmel Vendor and scroll down until you see AT89C51RD2. Select AT89C51RD2 and hit OK



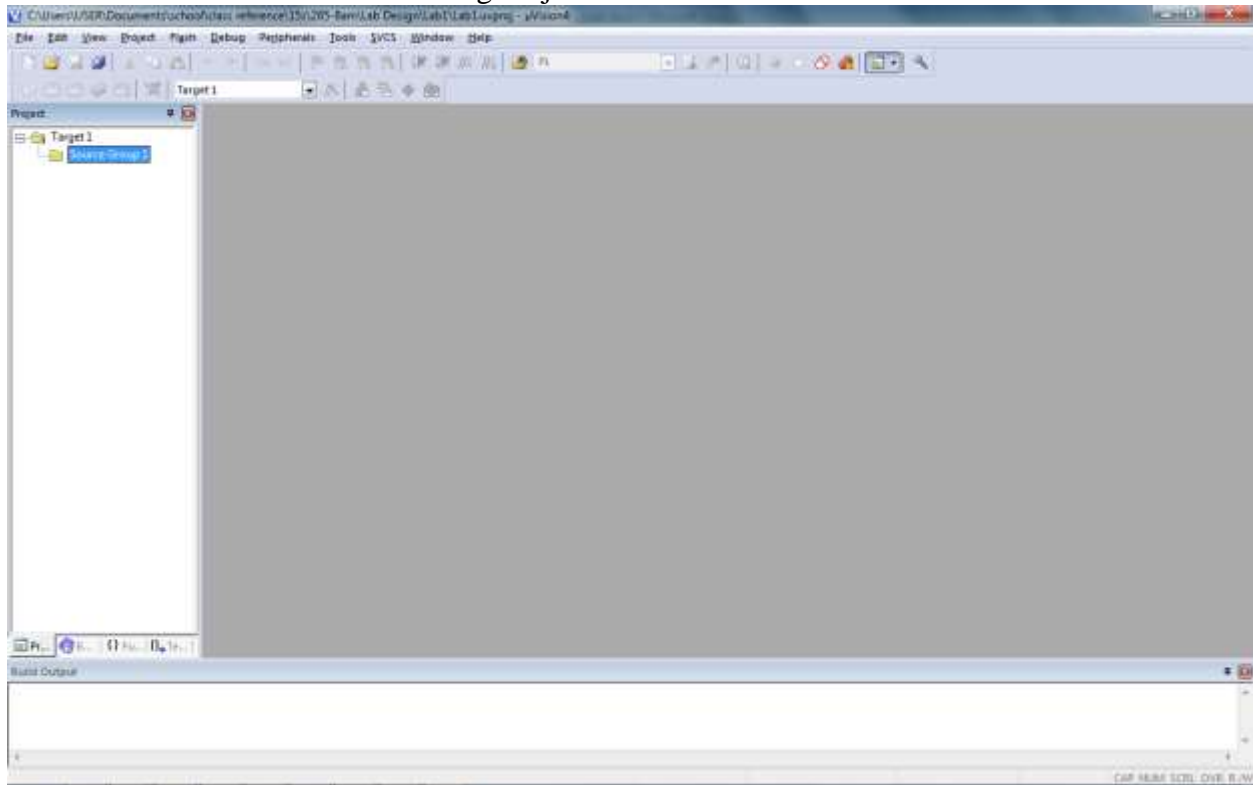
5. Click No on the following prompt



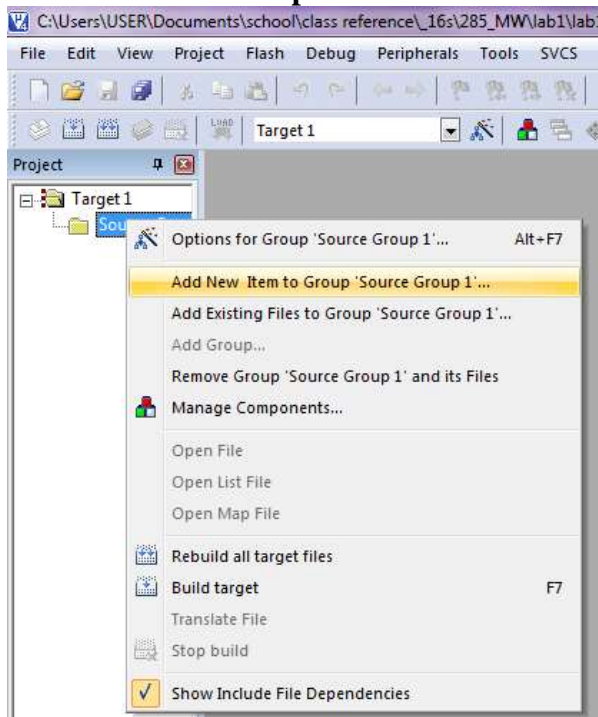
CECS 285: Lab1

Keil 8051 Basics – Due 1/21/2016

6. You should now have the following Project screen:



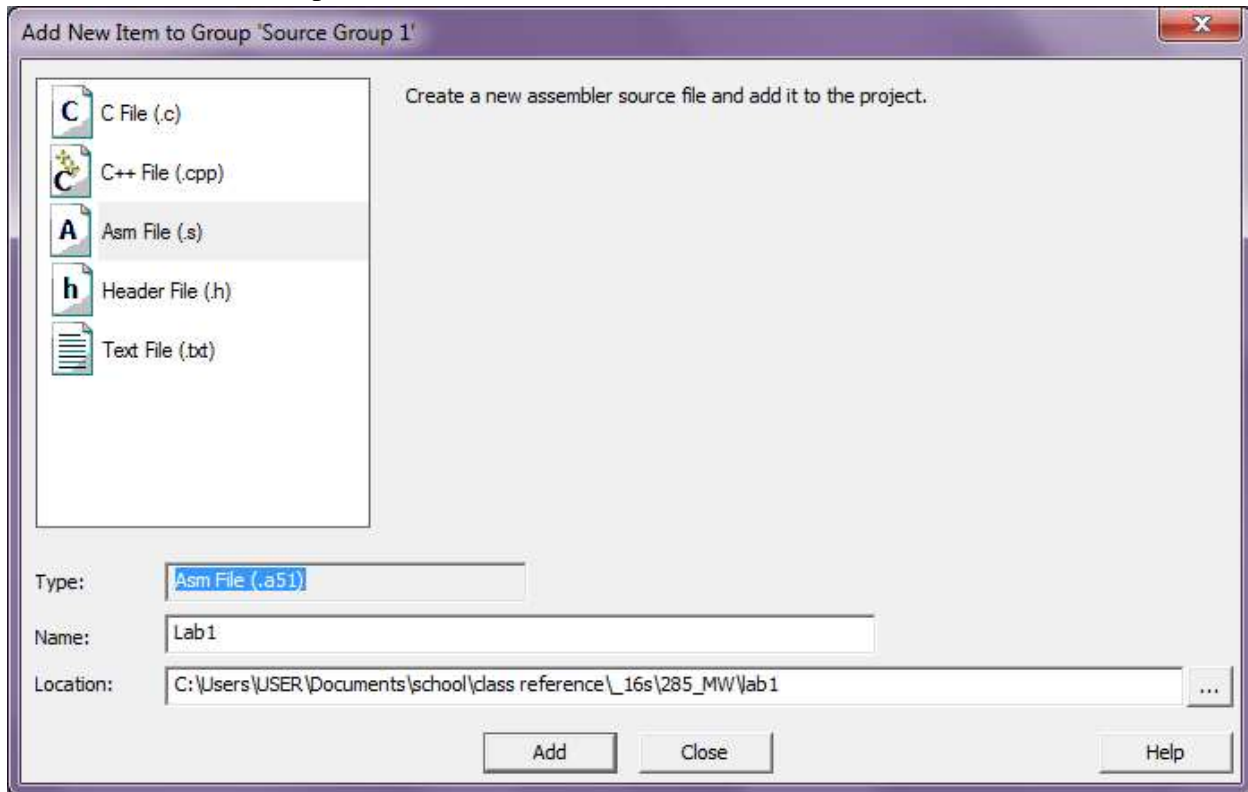
7. Expand the **Target 1** project tree item, then right click on **Source Group** and choose **Add New Item to 'Source Group 1'...**



CECS 285: Lab1

Keil 8051 Basics – Due 1/21/2016

8. Select the Asm File option, name source file Lab1, then click the Add button



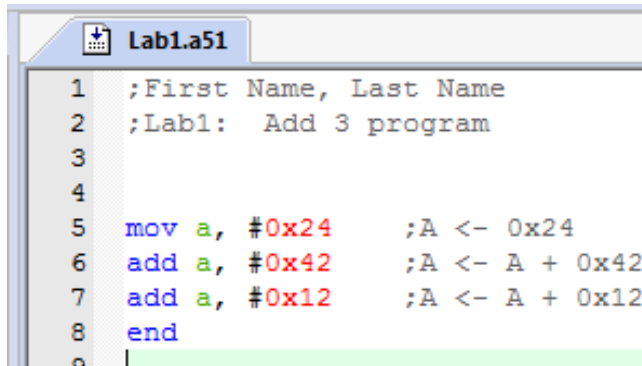
9. The Source File should now appear in your project window pane:



CECS 285: Lab1

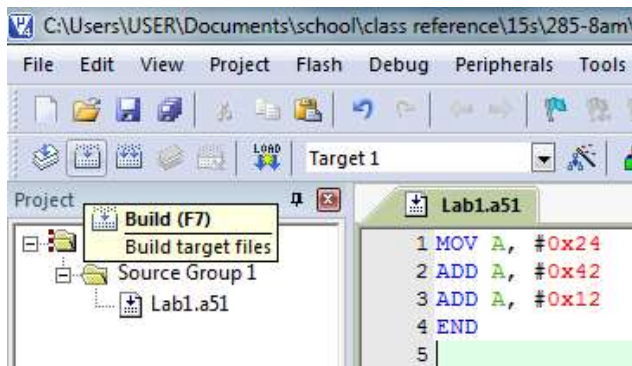
Keil 8051 Basics – Due 1/21/2016

10. Enter the following Instructions and into the Lab1.a51 source file and hit save. The instructions are for a simple program that adds 3 1-byte hex values. Comments are preceded with a semicolon. Replace **First Name, Last Name** with your name to serve as a basic document heading.

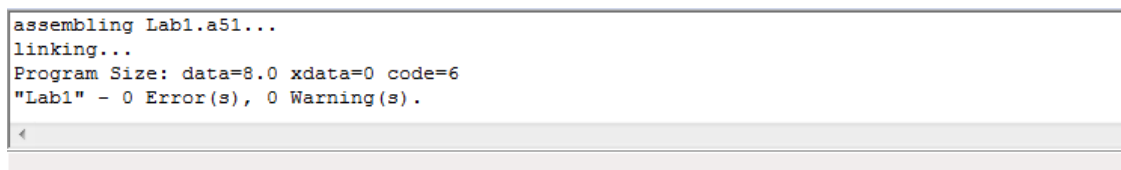


```
1 ;First Name, Last Name
2 ;Lab1: Add 3 program
3
4
5 mov a, #0x24 ;A <- 0x24
6 add a, #0x42 ;A <- A + 0x42
7 add a, #0x12 ;A <- A + 0x12
8 end
9
```

11. Click the Build button or press the F7 button to Build your project



12. The console window at the bottom of your screen should indicate 0 Errors and 0 Warnings

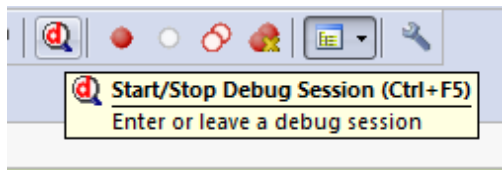


```
assembling Lab1.a51...
linking...
Program Size: data=8.0 xdata=0 code=6
"Lab1" - 0 Error(s), 0 Warning(s).
```

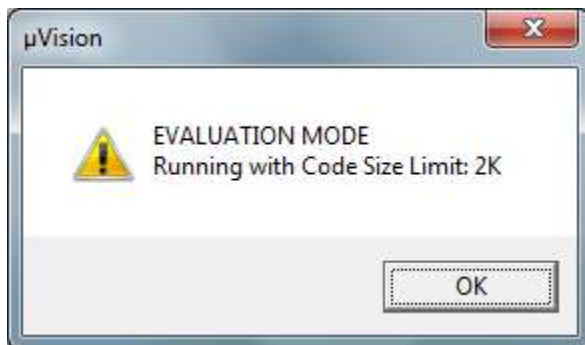
CECS 285: Lab1

Keil 8051 Basics – Due 1/21/2016

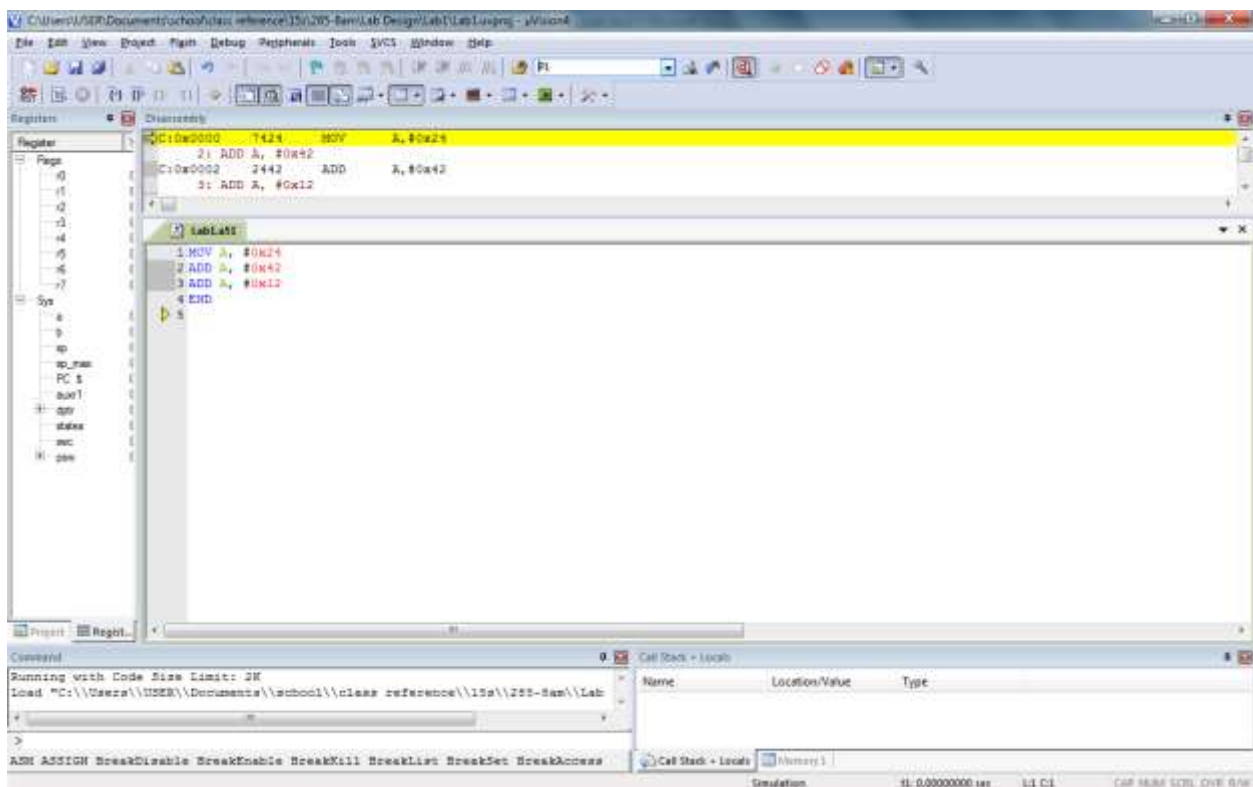
13. Start the Debugger by clicking the Debug icon or by pressing Control+F5



14. Press OK when this prompt appears:



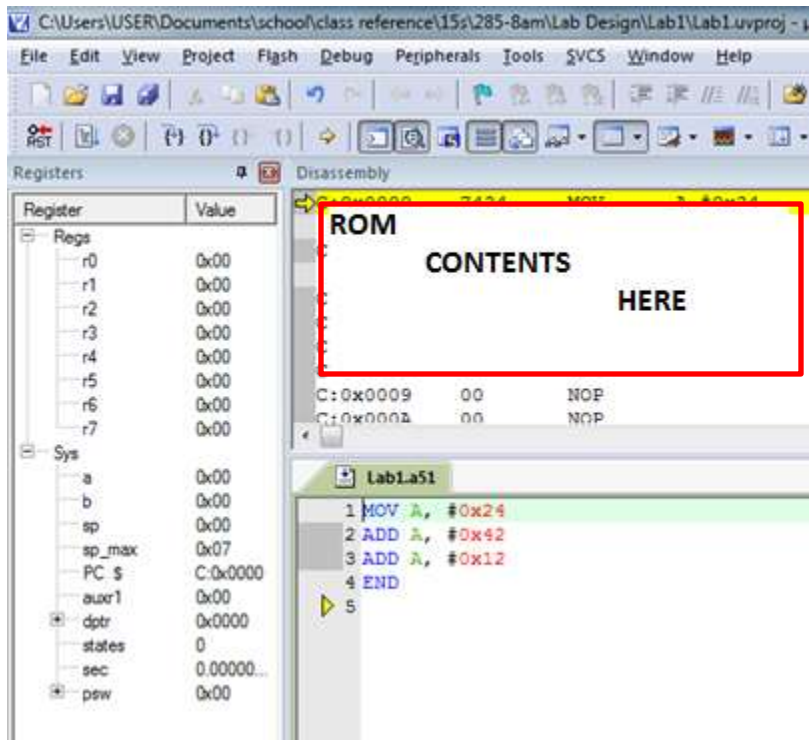
15. Keil should now show a debug interface:



CECS 285: Lab1

Keil 8051 Basics – Due 1/21/2016

16. Slide the right side of the far left window Pane over to clearly view the contents of Registers and slide the bottom of the top window Pane down to view the contents of ROM which is also labeled as Disassembly (The contents have been purposefully obstructed in the image below)



17. Observe the contents of ROM. On each line the first value you see is a ROM location indicated as 0x?????. To the right of each ROM location is machine code which corresponds to the Assembly language instruction contained in your source file. The Assembly language instruction contained in your source file is listed to the right of the Opcode. Hard coded data or constants are contained as part of the opcode.

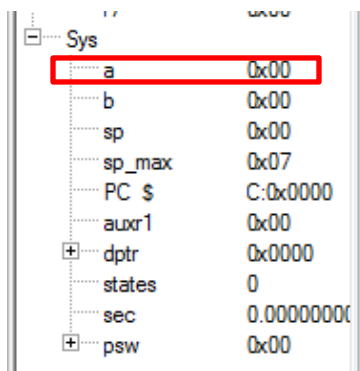
CECS 285: Lab1

Keil 8051 Basics – Due 1/21/2016

19. The debugger allows you to execute an Assembly program line by line so you can observe how the contents of Registers and Memory are updated one instruction at a time. This can be done by pressing the Step button or pressing the F11 button.



The A register is used for each instruction in the program you are working with.



Single Step through the program and observe the contents of the A register as each instruction executes.

Lab1 Deliverable:

Submit a single word doc or pdf to the Lab1 beachboard dropbox that contains a screenshot of your debug window showing:

- Your Lab1.a51 source file with header
- The expanded left window pane clearly displaying register A final contents
- The Disassembly window pane resized to show ROM location, machine code, and each instruction of your program
 - *Be sure to Resize your window or crop and resize your image accordingly to eliminate empty space in your screenshot. Submissions which are difficult to view will not receive full credit. Ask the instructor if you are unsure.*

Exercise completed, submit your document to the beachboard dropbox and have a great day!