

# Writing A Single File Assembly Program in Keil 5.28

11/12/2018

Sepehr Naimi



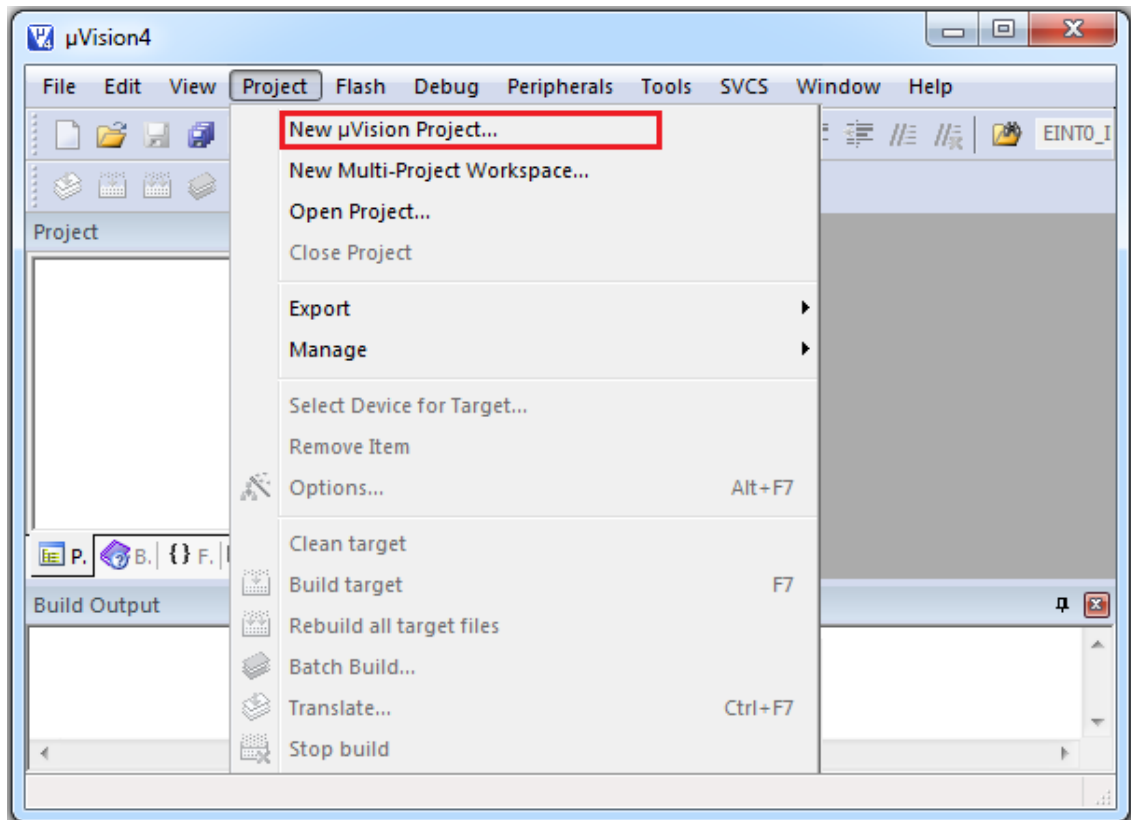
[www.NicerLand.com](http://www.NicerLand.com)

## Contents

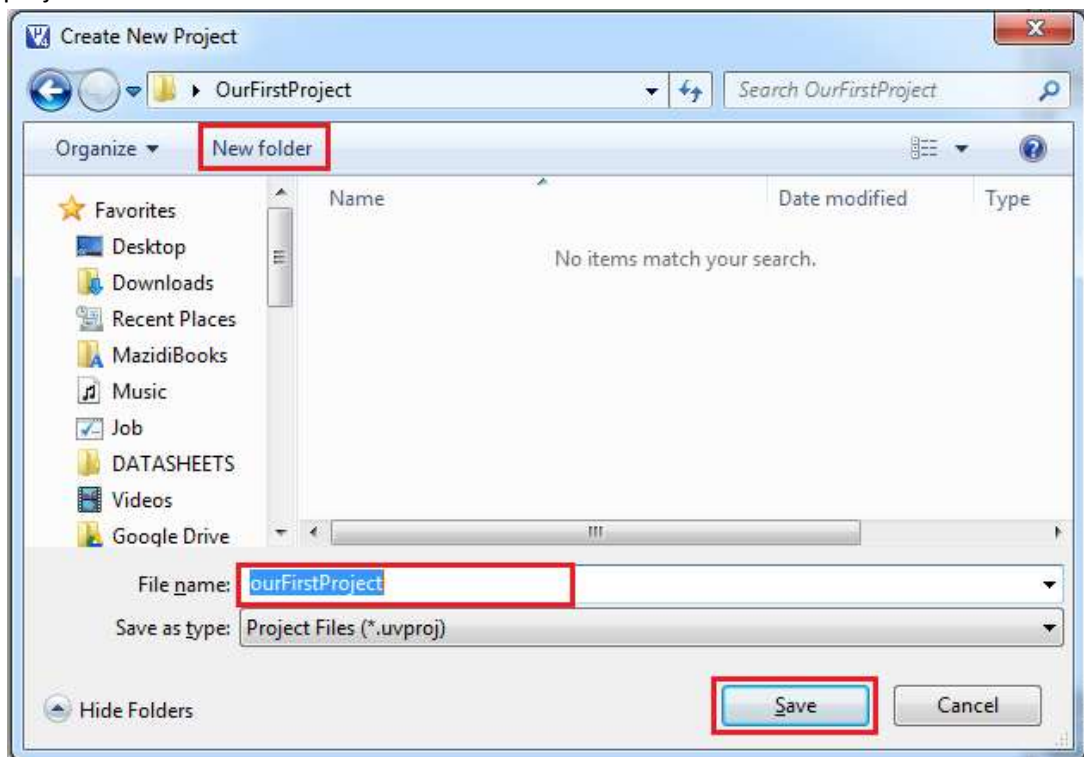
Creating an Assembly project in Keil .....	3
Building.....	6
Choosing Simulator as the Debugger .....	7
Debugging and Tracing.....	8

## Creating an Assembly project in Keil

1. Open the Keil IDE by clicking on its icon on the desktop.
2. Choose **New uVision Project** from the **Project** menu.



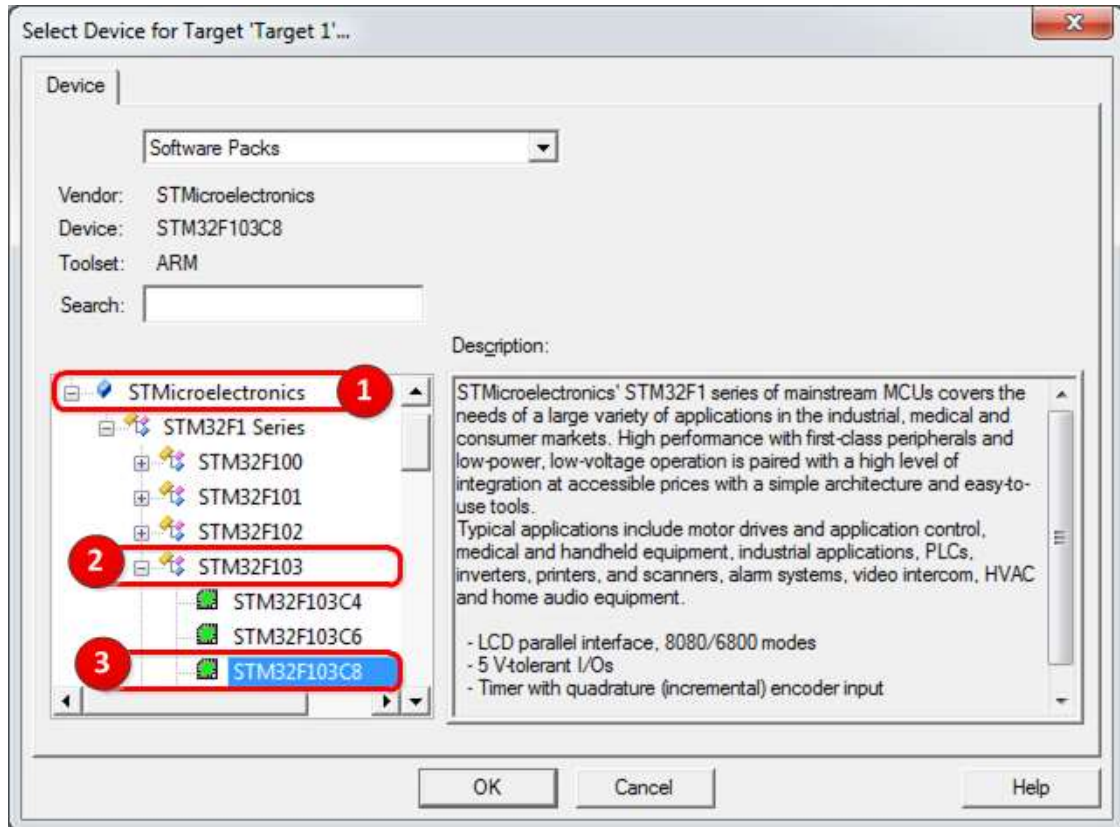
3. Create a new folder and Name it **OurFirstProject**. Type the name **ourFirstProject** for the project name and click **Save**.



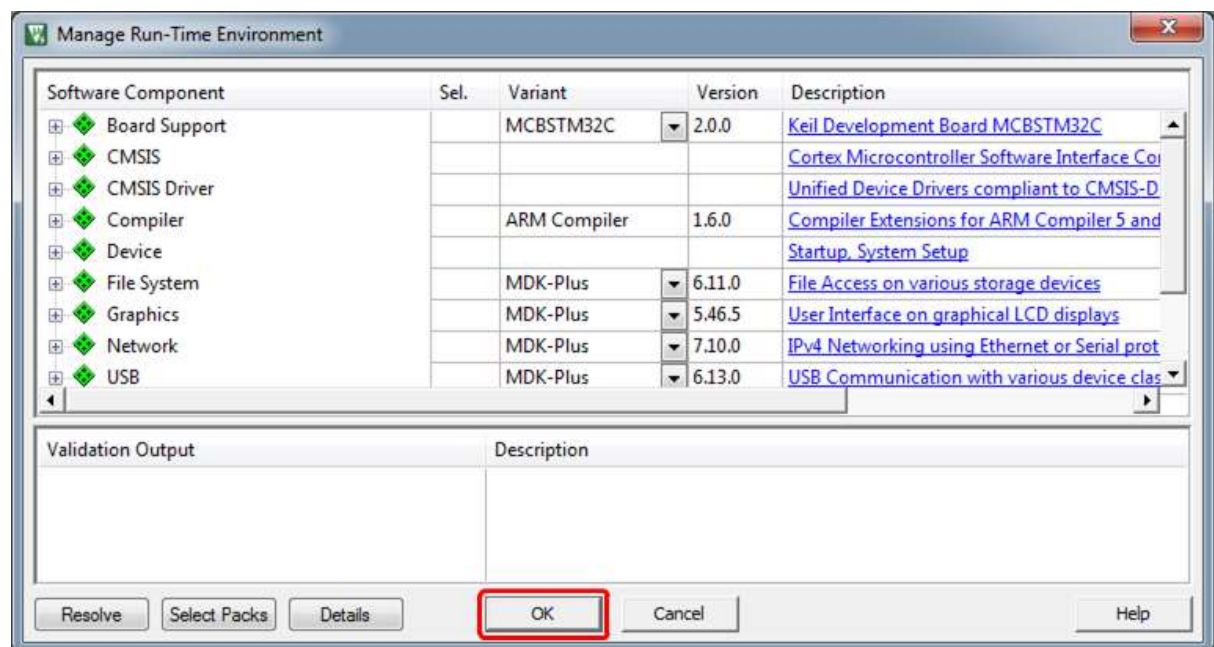
4. In the tree expand **STMicroelectronics**. (If STMicroelectronics is not in the tree, read “installing Keil and STM32F103” step-by-step tutorial from our website.) Click on **STM32F103** and choose **STM32F103C8**. Then press **OK**.

**Note**

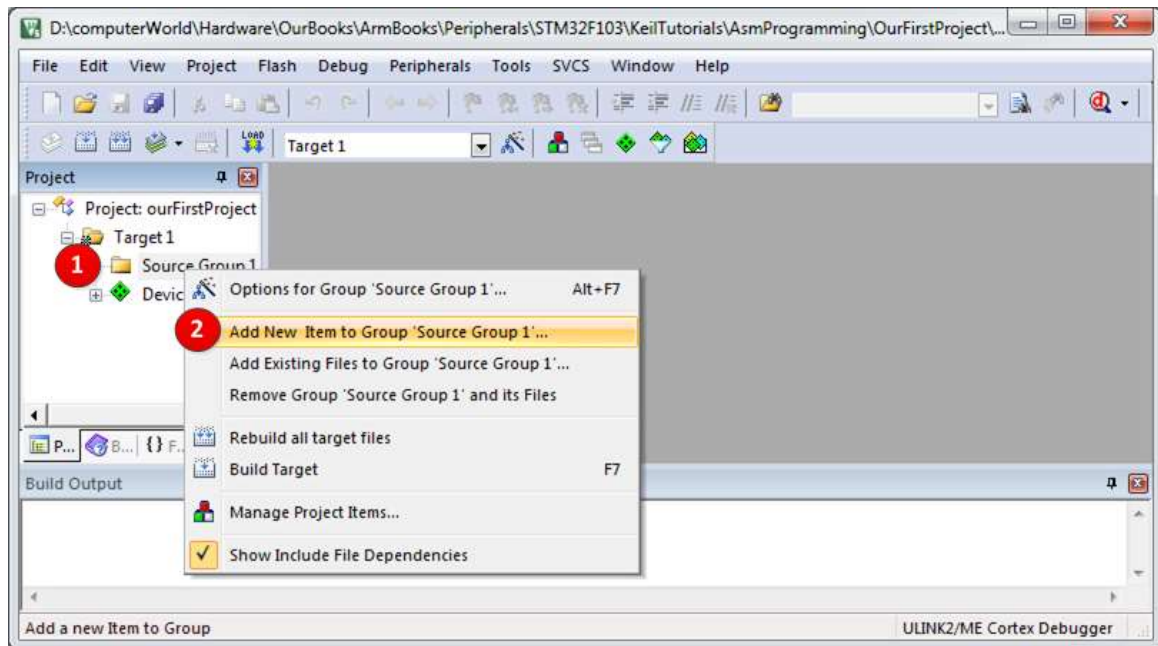
When you choose the chip some general information of the chip is shown in the **Description** box.



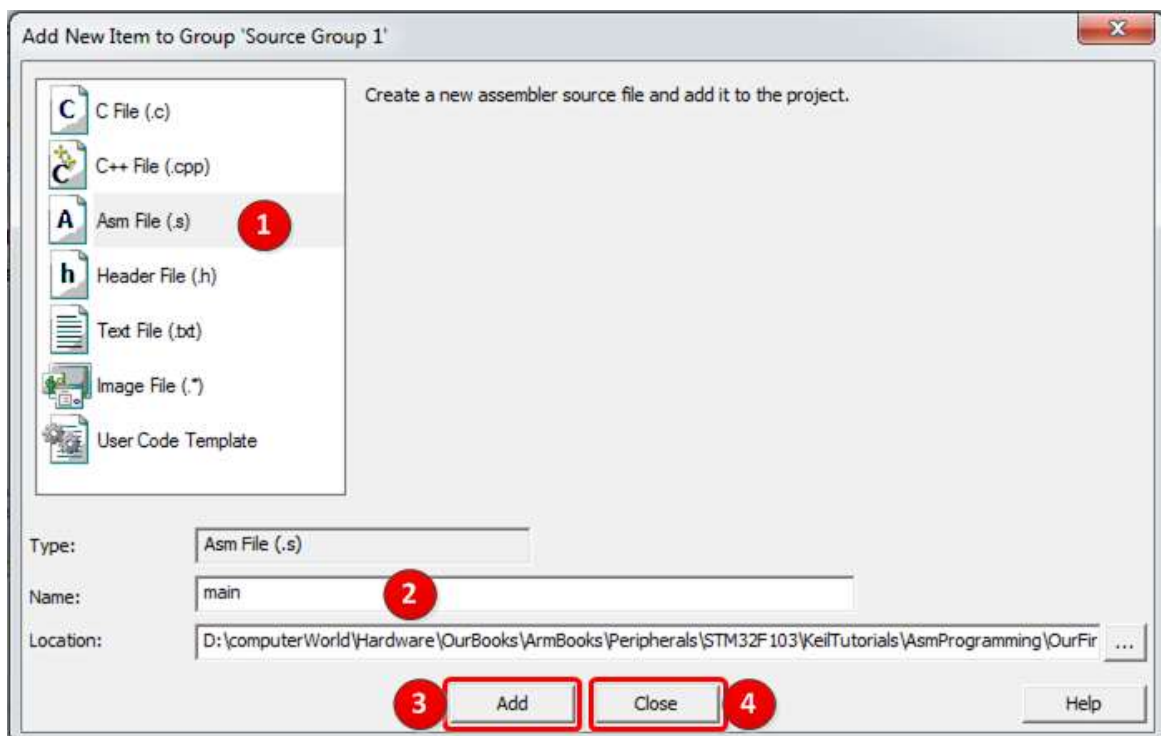
5. Do not add any components to the project. Just click **OK**.



- Right click on **Source Group 1** and choose Add New Item to Group. This makes a new file and adds it to the project.



- Choose the type of file as **Asm File(.s)** and name it as **main**. Click on the **Add** button and then click on **Close**.



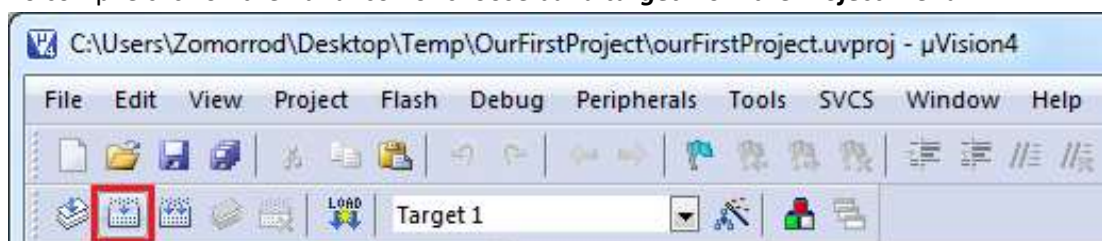
8. Type the following sample program in the *main.s* file.

```
1 EXPORT Reset_Handler
2 EXPORT __Vectors
3
4 ;-----
5 ;The vector table
6 ;-----
7 AREA RESET, DATA, READONLY
8 __Vectors
9 DCD stack_init ;init loc. 0 to 3 with stack bottom
10 DCD Reset_Handler ;store Reset_Handler into loc. 4 to 7
11
12 ; reserving 400 bytes of RAM for stack
13 AREA STACK, DATA, READWRITE
14 SPACE 400
15 stack_init
16
17 AREA MY_STARTUP, CODE, READONLY
18 Reset_Handler PROC
19 LDR R5,=__main
20 BX R5 ; go to __main
21 ENDP
22 ;-----
23 ; our main program
24 ;-----
25 AREA MY_PROG, CODE, READONLY
26 __main PROC
27 MOV R1,#6 ; R1 = 6
28 MOV R2,#4 ; R2 = 4
29 ADD R3,R1,R2 ; R3 = R1 + R2
30 H B H ;stay here forever
31 ENDP
32
33 END
34
```

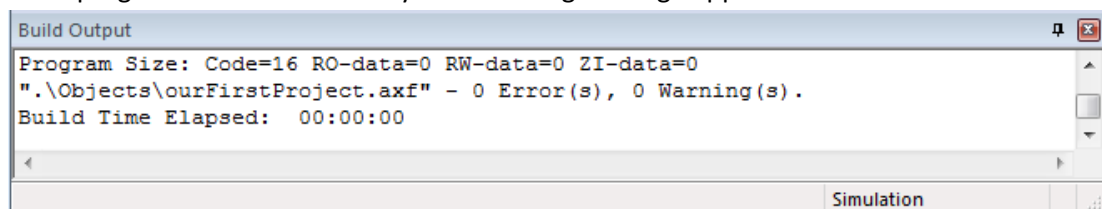
9. Press **Ctrl+S** to save the file.

## Building

10. To compile click on the **Build** icon or choose **build target** from the **Project** menu.

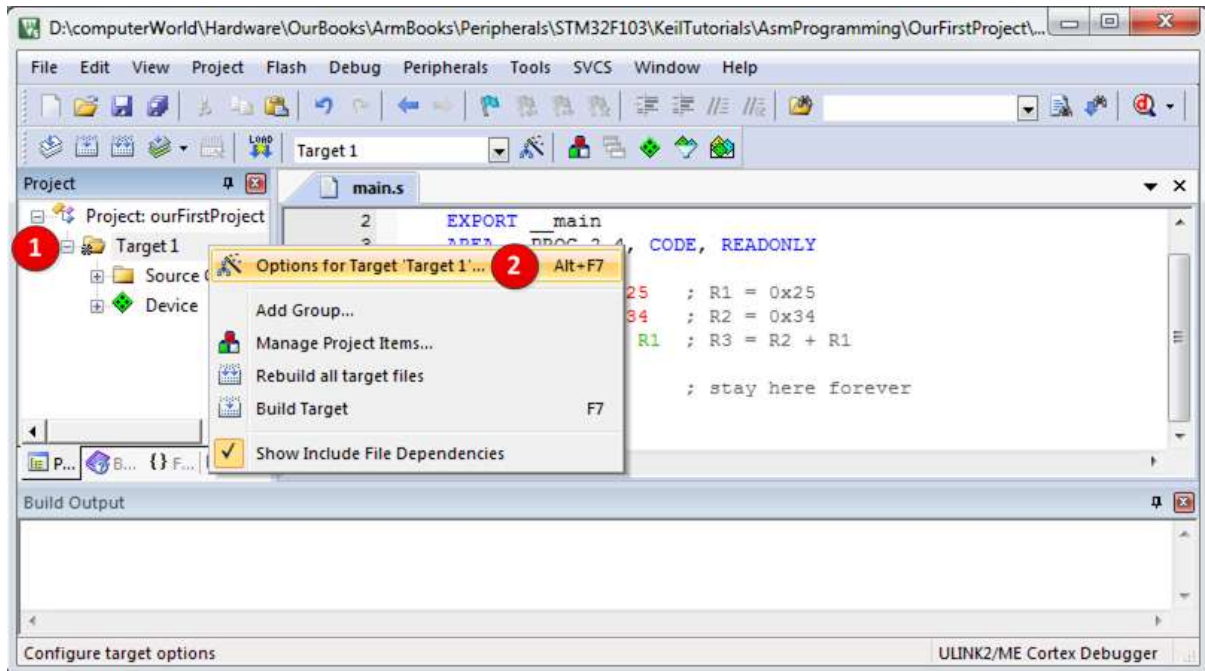


11. If the program is built successfully the following message appears:

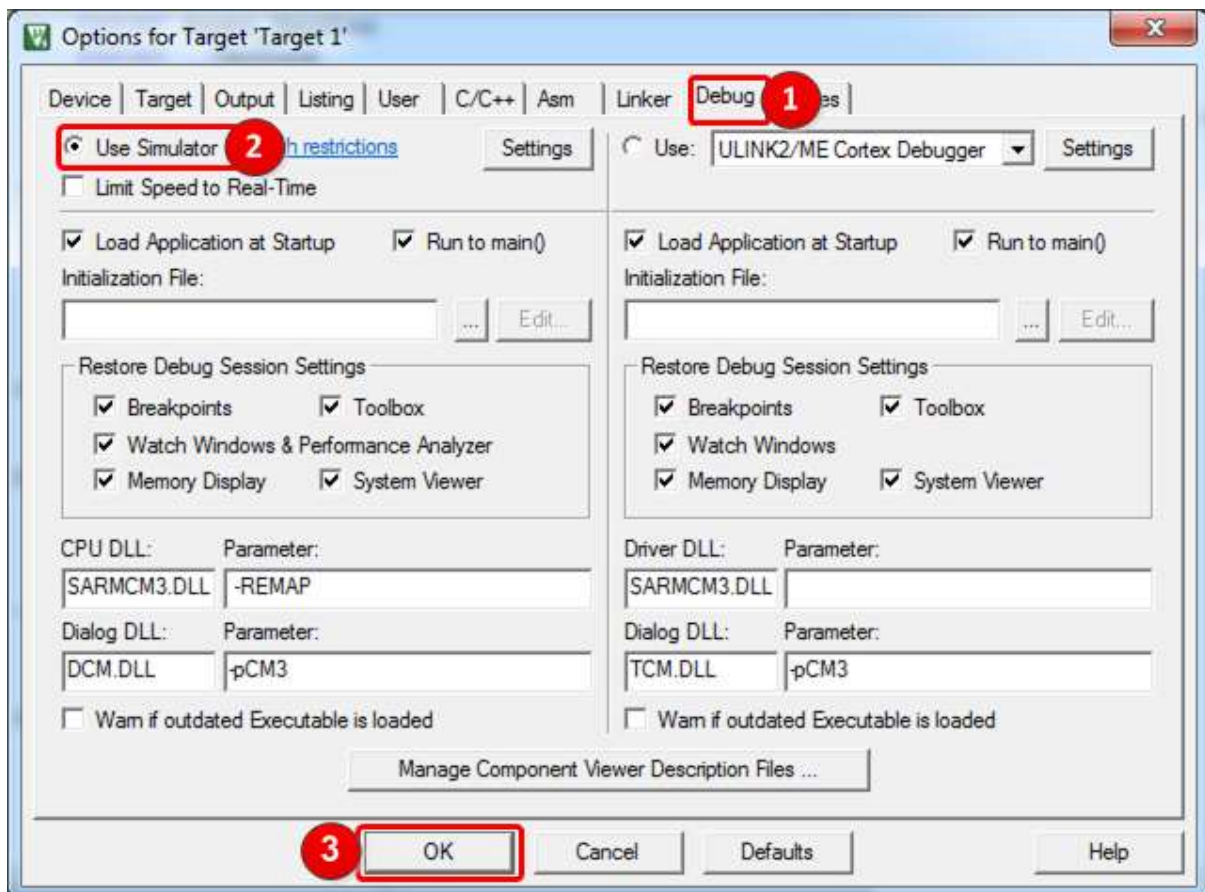


## Choosing Simulator as the Debugger

12. Right click on Target 1 and choose Options.

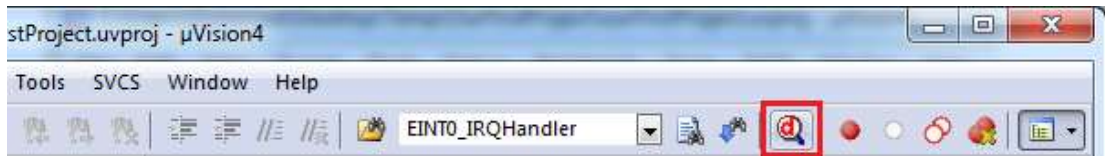


13. Click on the Debug tab and choose Simulator. Then, press OK.



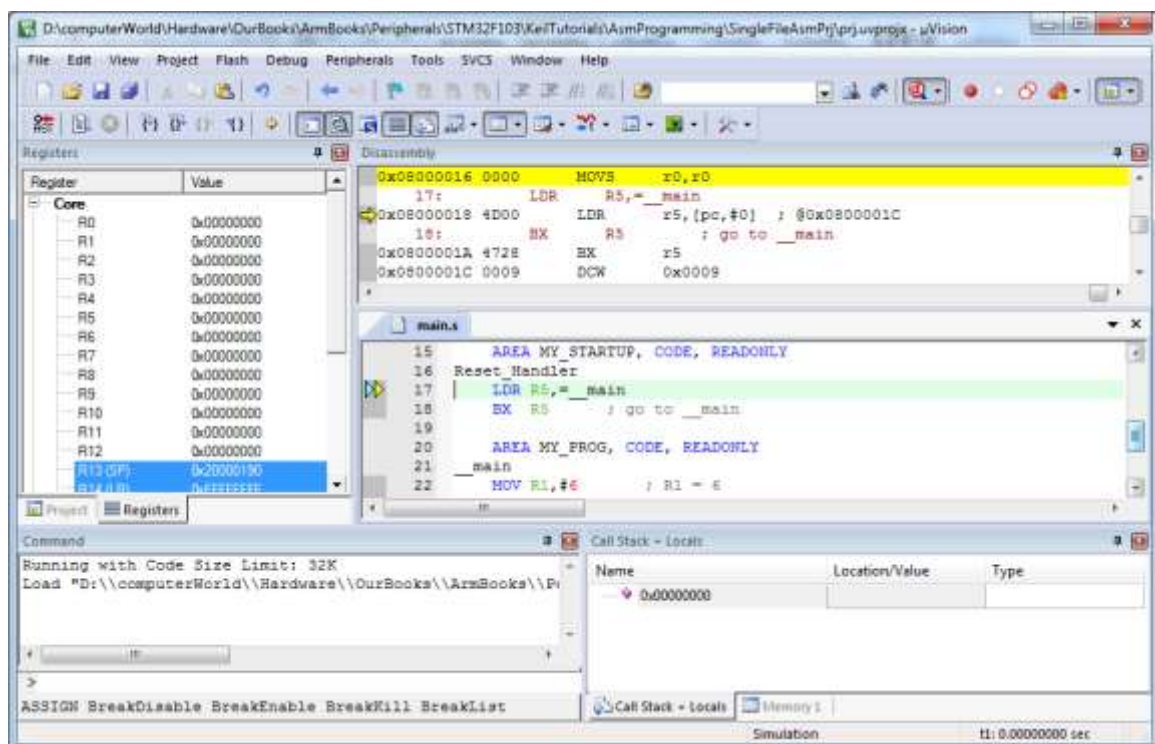
## Debugging and Tracing

14. To start debugging click on **Start/Stop Debug Session** icon or choose **Start/Stop Debug Session** from the **Debug** menu. (or simply press **Ctrl+F5**)



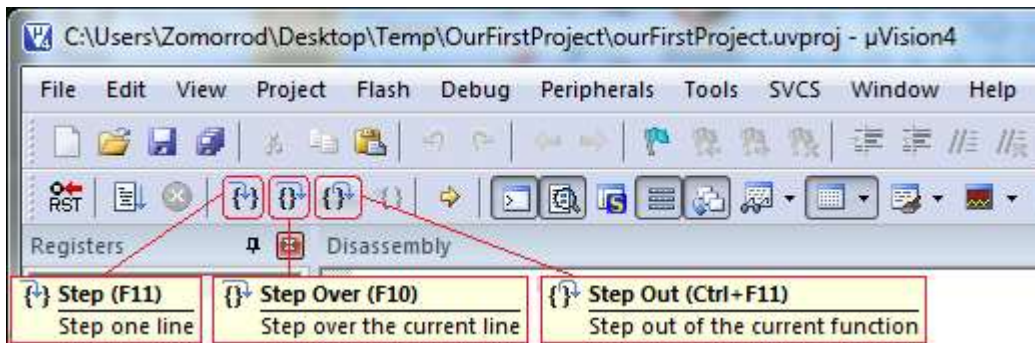
15. If it starts tracing successfully, a yellow arrow appears in front of the next instruction to be executed. See the following picture.

- In the disassembly window, you see the machine code for each instruction. For example, the “LDR R5,=\_\_main” pseudo-instruction is converted to instruction “LDR r5,[pc,#0]” whose machine code is 4D00 and it is located in memory address 0x08000018.
- In the Registers window, you see the current values for the CPU registers.



- To trace the program, use the **Step Over** button or click on **Step Over** from the **Debug** menu. It executes the instructions of the program one after another. To trace the program, you can use the **Step** button, as well. The difference between the **Step Over** and **Step** is in executing functions. While **Step** goes into the function and executes its instructions one by one, **Step Over** executes the function completely and goes to the instruction next to the function. When you are in the function and you want the function to be executed completely you can use **Step Out**. In the case, the instructions of the function will be executed, it returns from the function, and goes to the instruction which is next to the function call. Press the Step Over a few times and see the registers values.





- d. To exit from the debugging mode press **Start/Stop Debug Session**.
16. Lines 25 to 31 of the sample program, make the main program. Now, you can change the main program or substitute it with the Assembly programs of the book and debug the programs using the Keil IDE.