

How to write Assembly programs in Keil

(Version 1.4)

Sepehr Naimi



Mazidi & Naimi tutorials

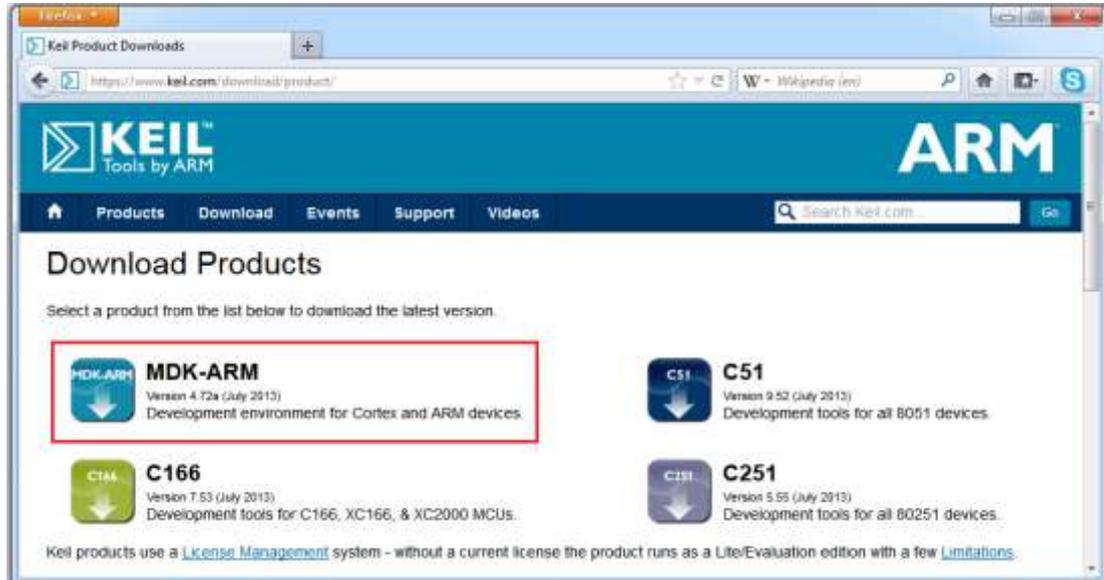
MicroDigitalEd.com and BIHE University

Contents

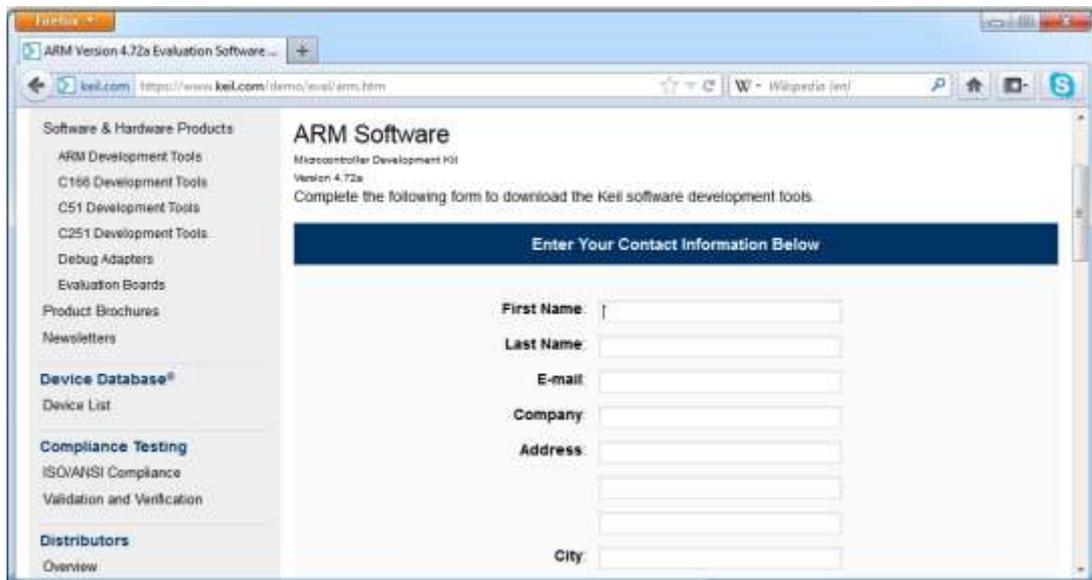
Downloading and installing the Keil IDE.....	3
Creating an Assembly project in Keil	5
Building.....	9
Debugging and Tracing.....	9

Downloading and installing the Keil IDE

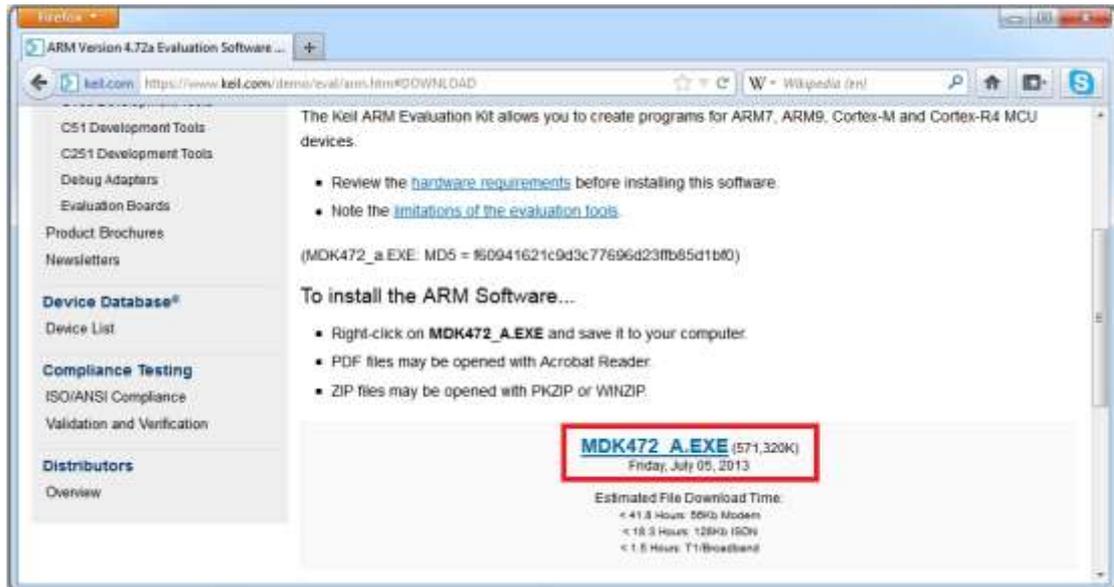
1. Click on the following link:
<https://www.keil.com/download/product/>
2. Choose MDK-ARM.



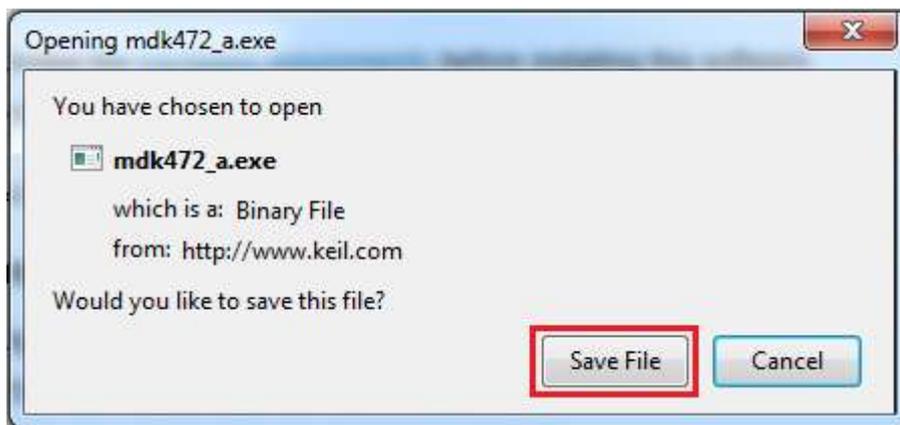
3. Fill out the form and then click on the **submit** button, at the bottom of the page.



- Click on the **MDKxxx.EXE** link.



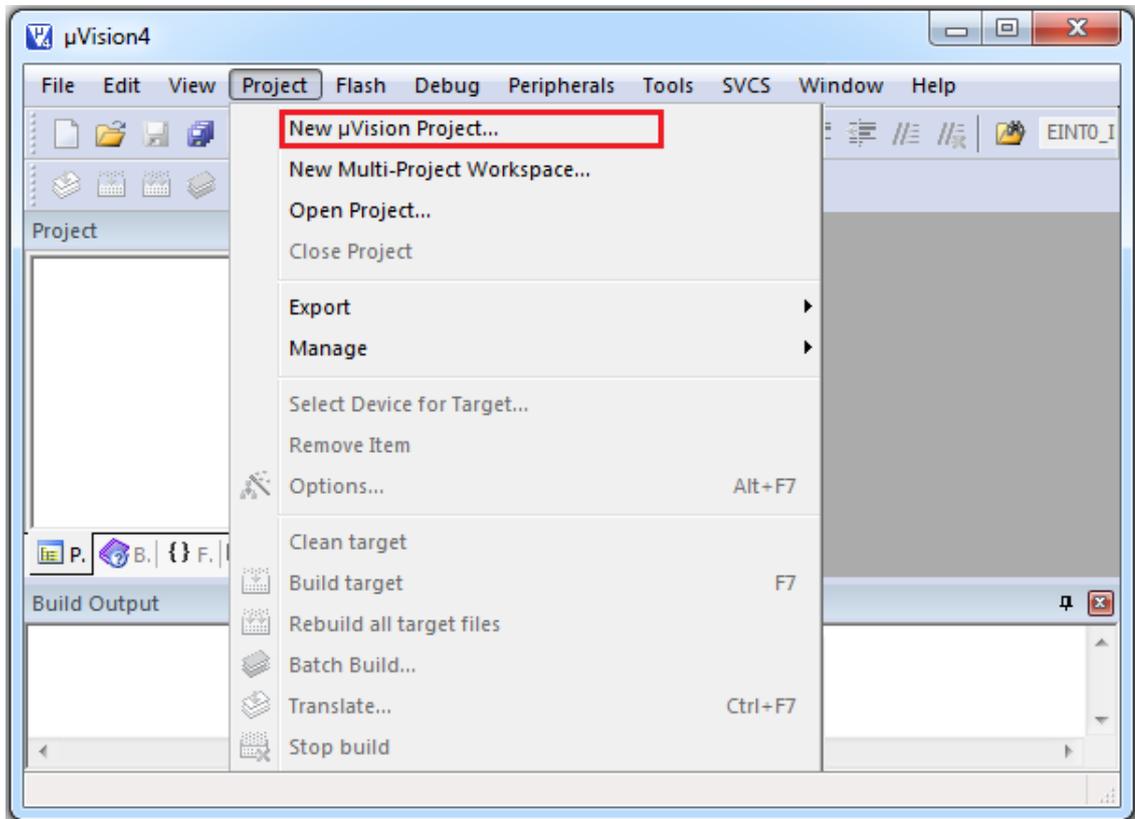
- Choose **Save**.



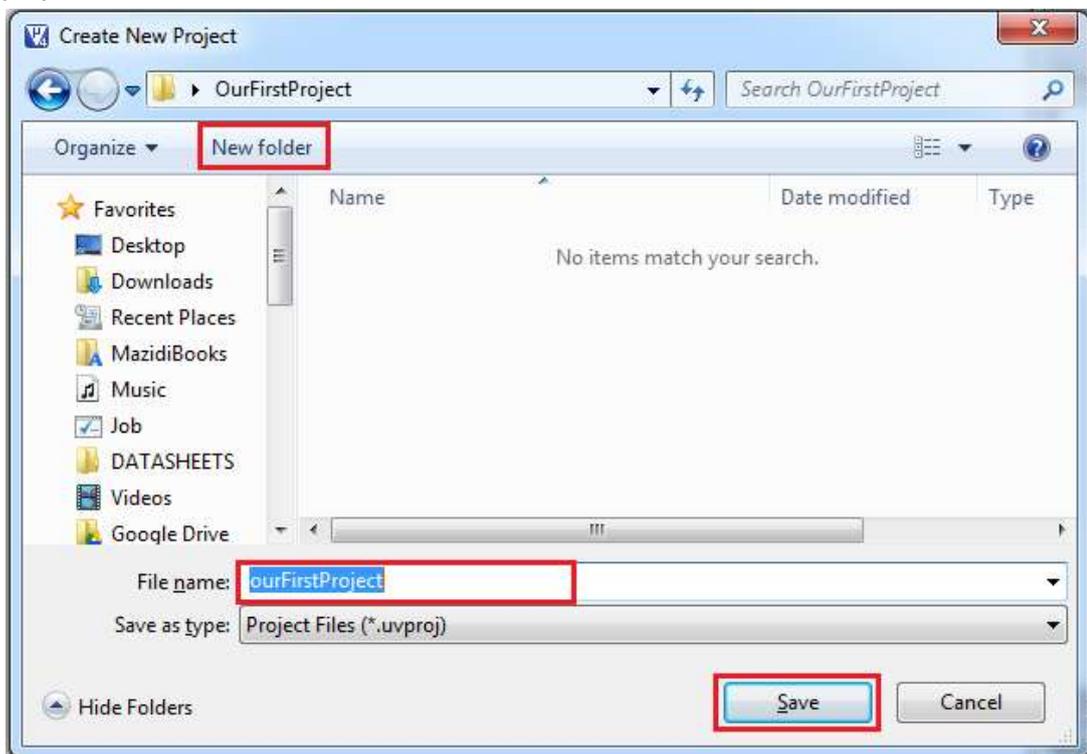
- Execute the downloaded file. Installing the software is straight forward; choose the **Next** button in each step, until the setup finishes.

Creating an Assembly project in Keil

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



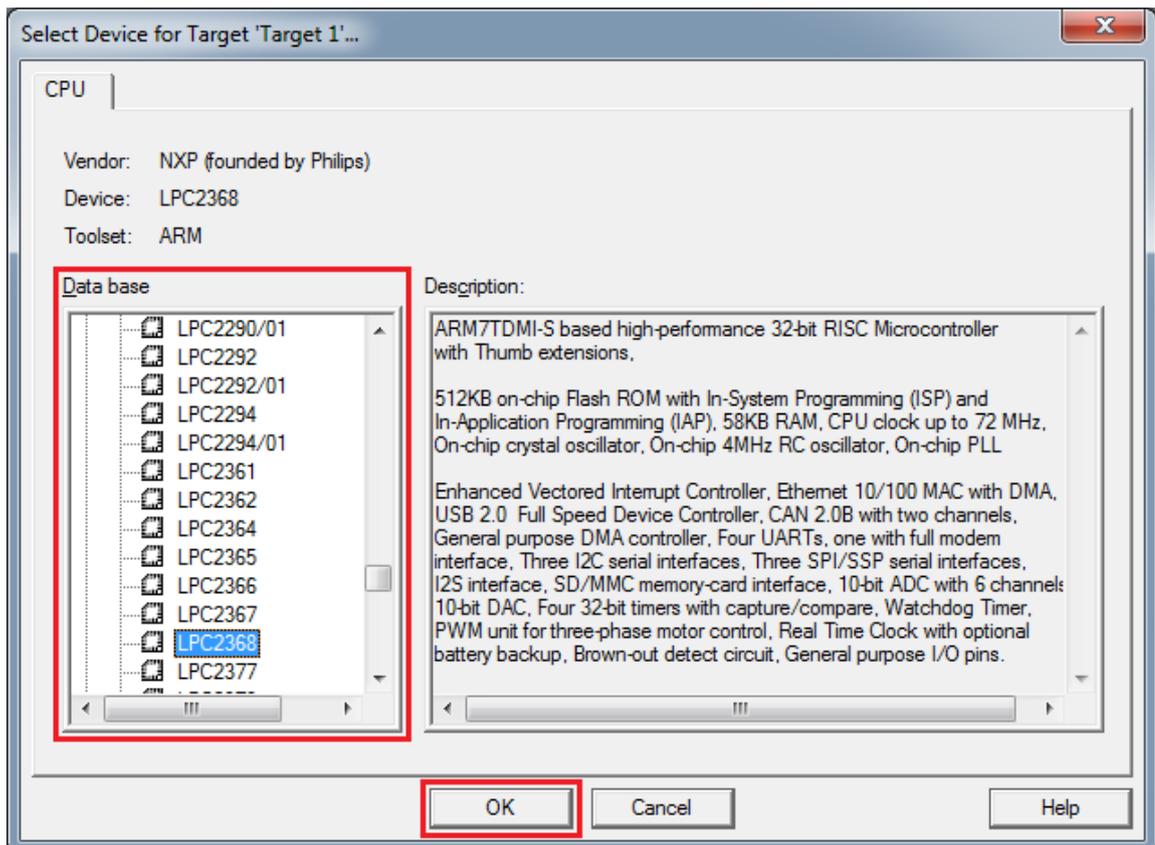
9. Create a new folder and Name it **OurFirstProject**. Type the name **ourFirstProject** for the project and click **Save**.



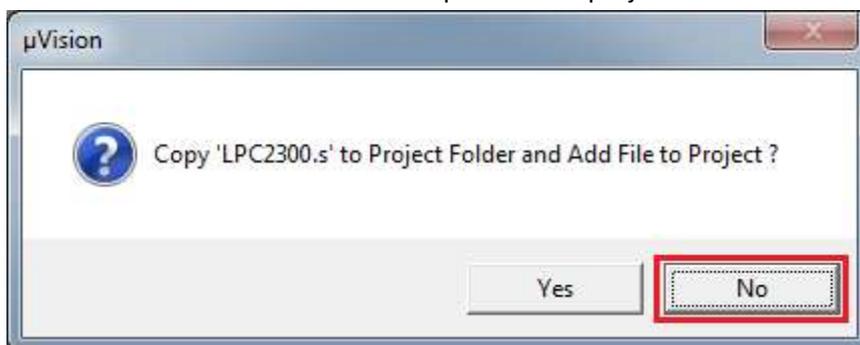
10. In the **Data base** tree, choose the vendor and then the chip you want to use and then click **OK**. For example, if you want to use the **LPC2368**, click on the **NXP** and then on the **LPC2368** and then press **OK**.

Caution

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



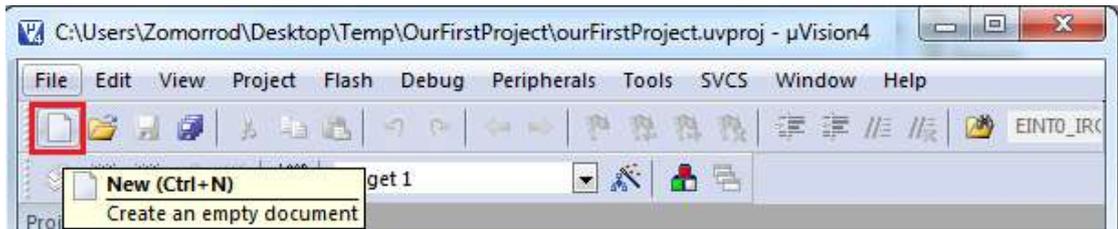
11. Click the **No** button to add the startup file to the project.



Caution

In your practical projects you should choose to add the startup file, as will be discussed in a separate tutorial.

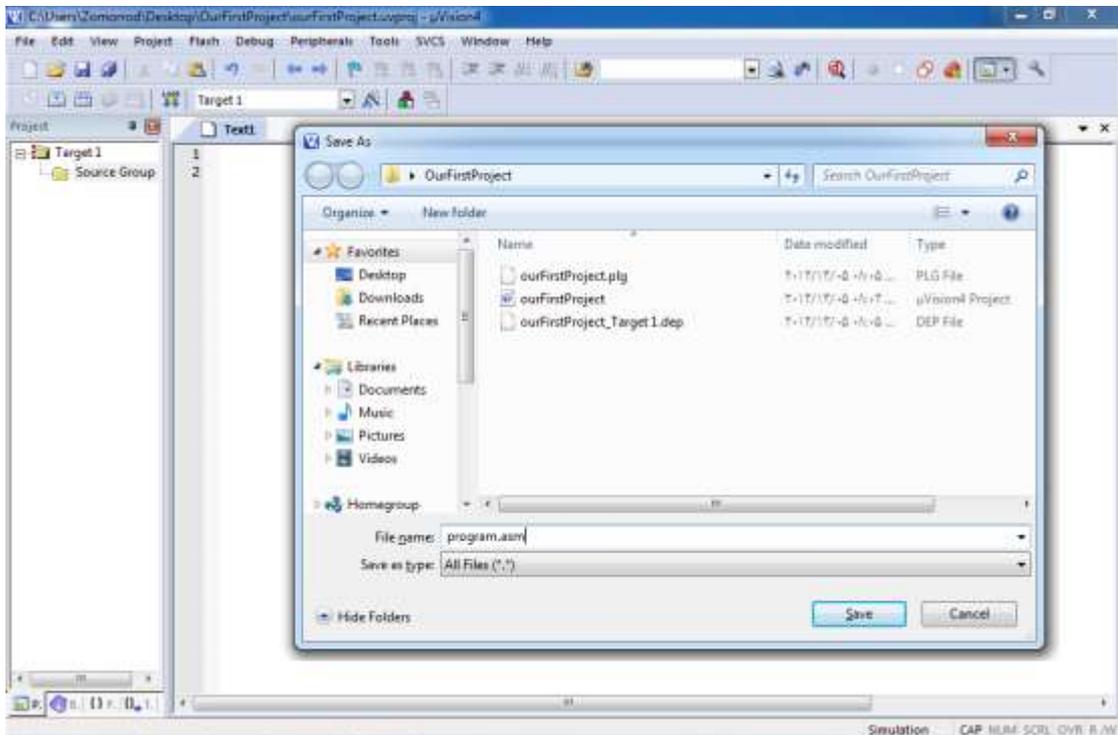
- Make a new file by clicking on the New Icon (you can make a new file by pressing **Ctrl+N** or choosing **New** from the **File** menu, as well.)



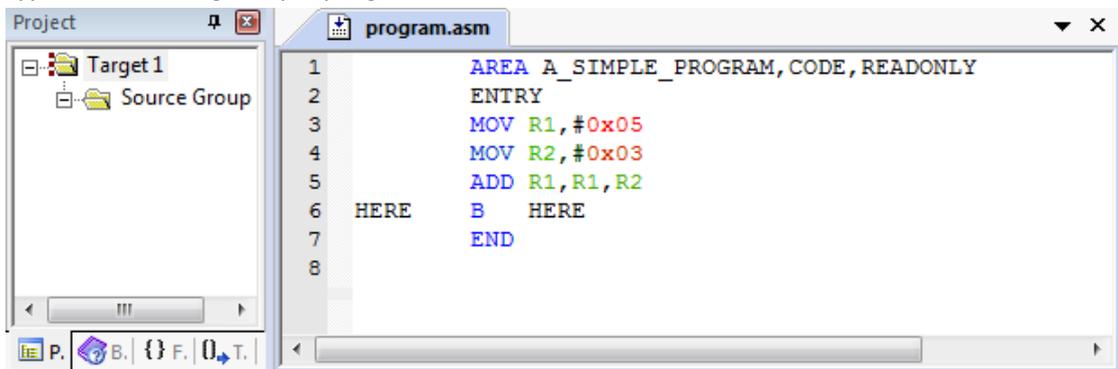
- Press **Ctrl+S** to save the new file. (You can also save the file by choosing **Save** from the **File** menu.)



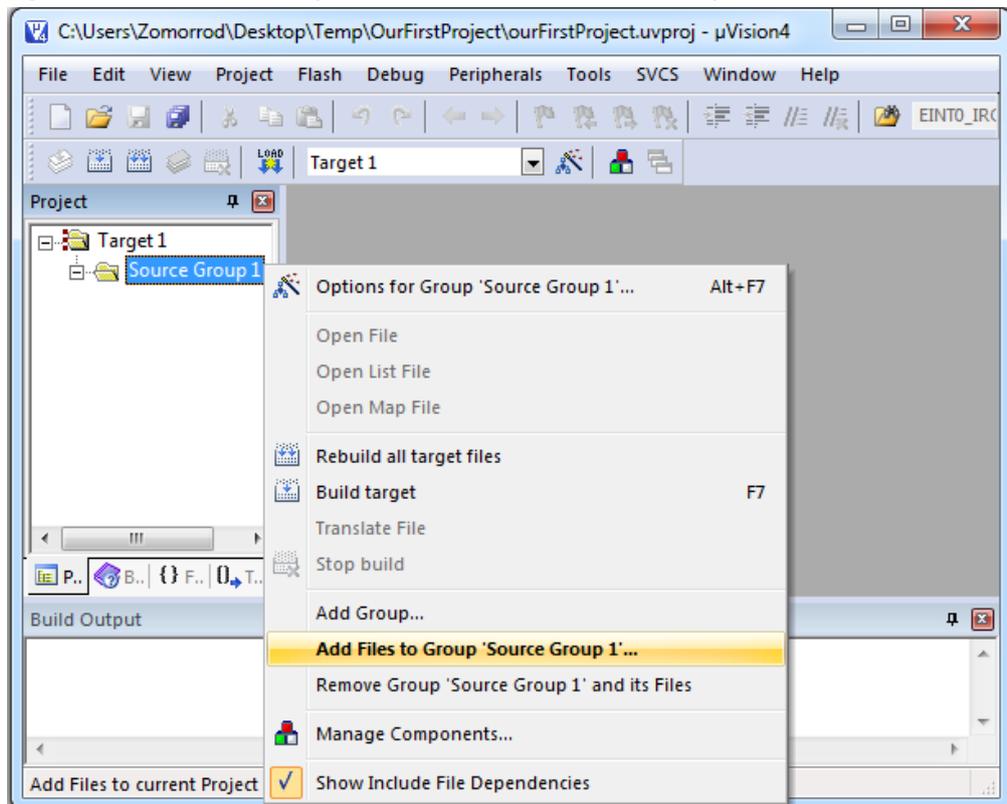
- Name the file as **program.asm** and save it in the **OurFirstProject** directory.



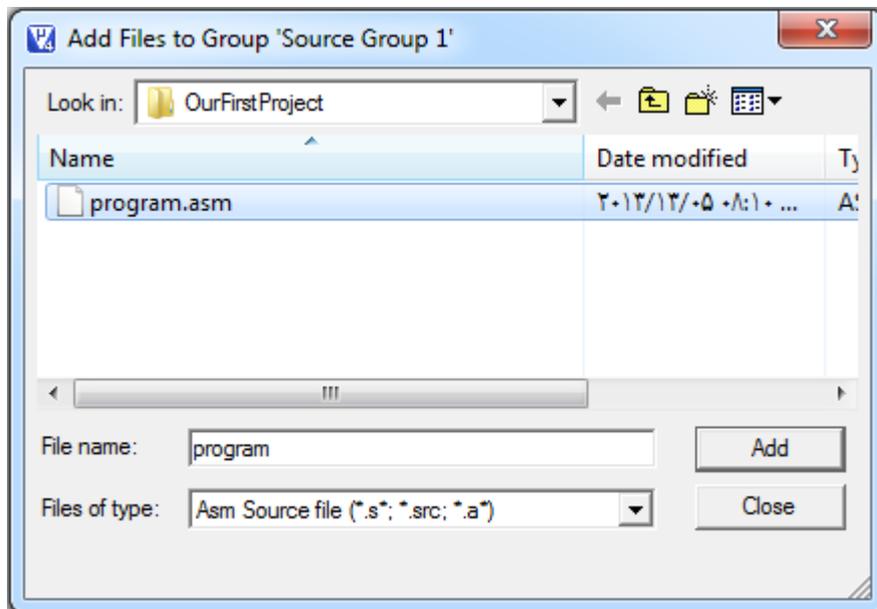
- Type the following sample program in the file



16. Add the **program.asm** file to the project. To do so:
- Right click on **Source Group 1** and choose **Add Files to Group**.

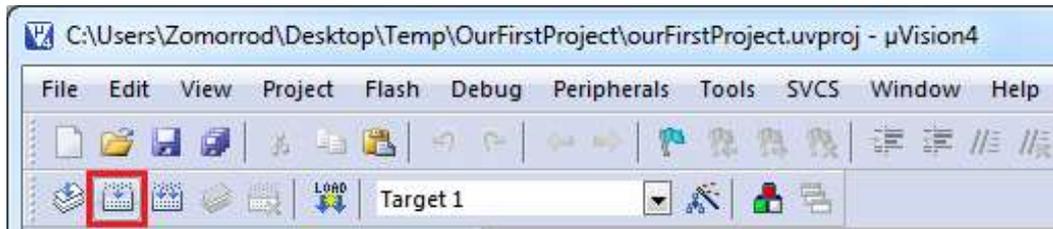


- Then go to the **OurFirstProject** directory and choose **Program.asm**, press **Add** and then **Close**.

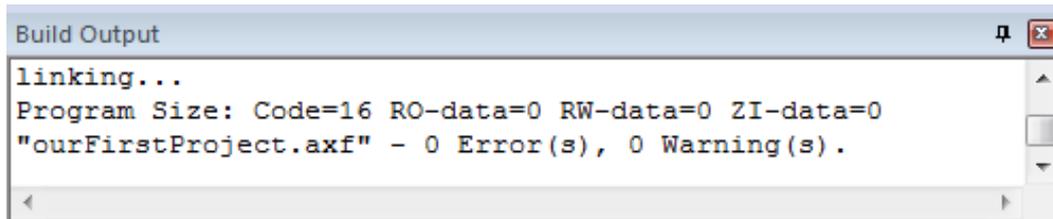


Building

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

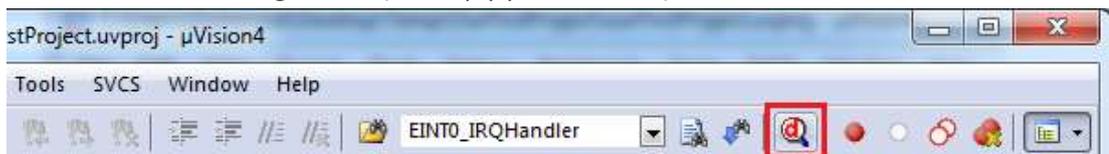


18. If the program is built successfully the following message appears:

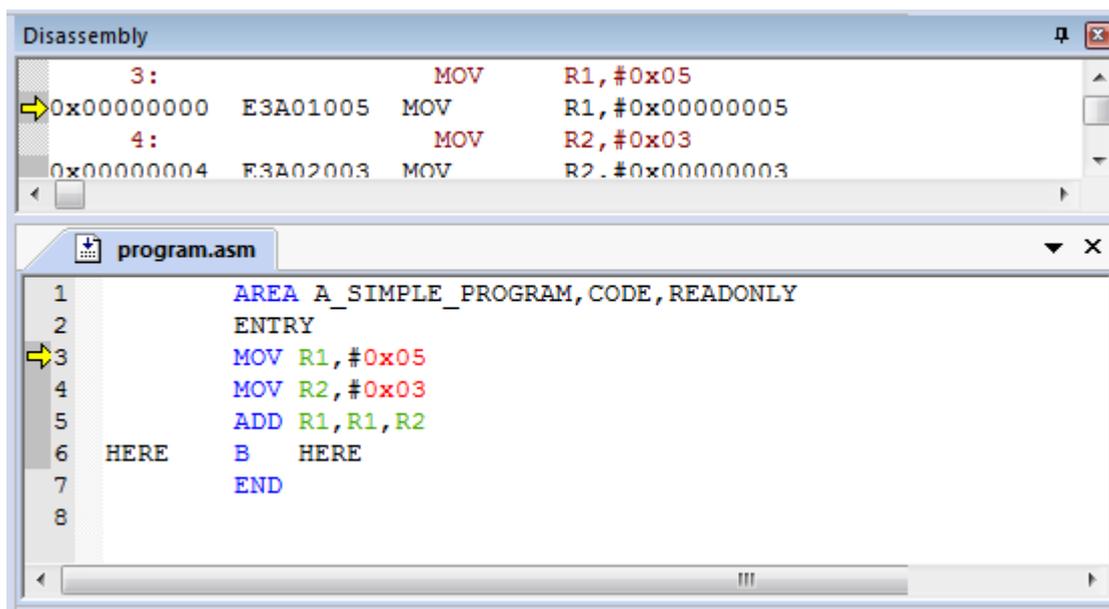


Debugging and Tracing

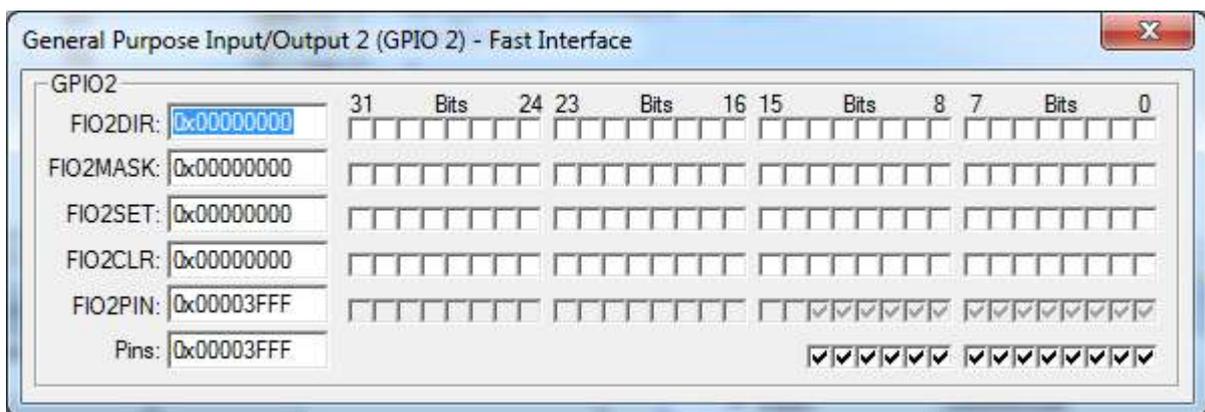
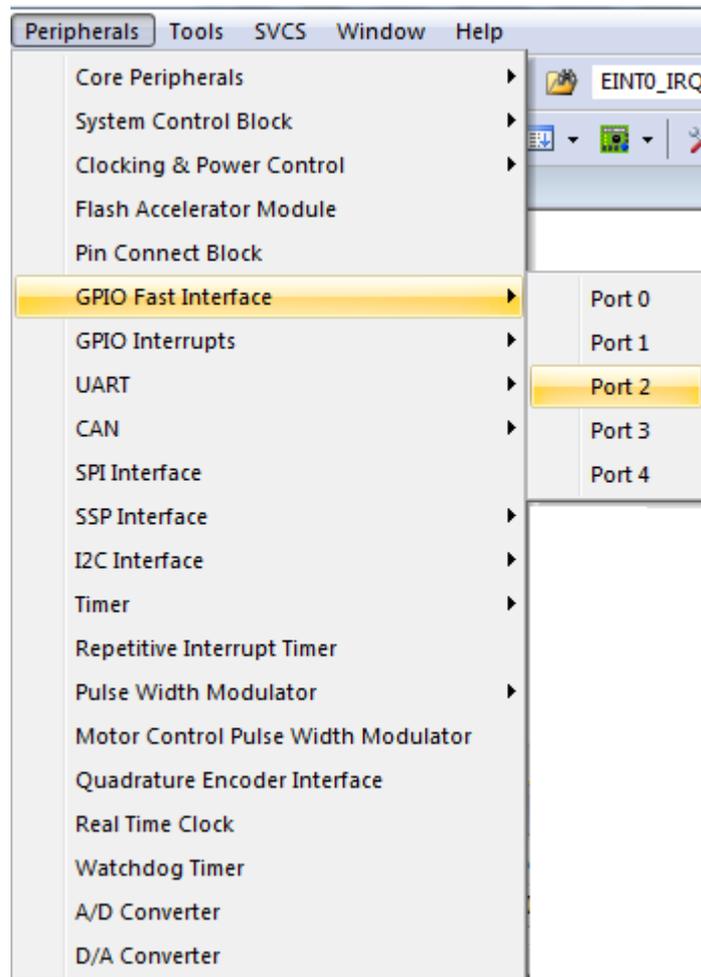
19. To start debugging click on **Start/Stop Debug Session** icon or choose **Start/Stop Debug Session** from the **Debug** menu. (or simply press **Ctrl+F5**)



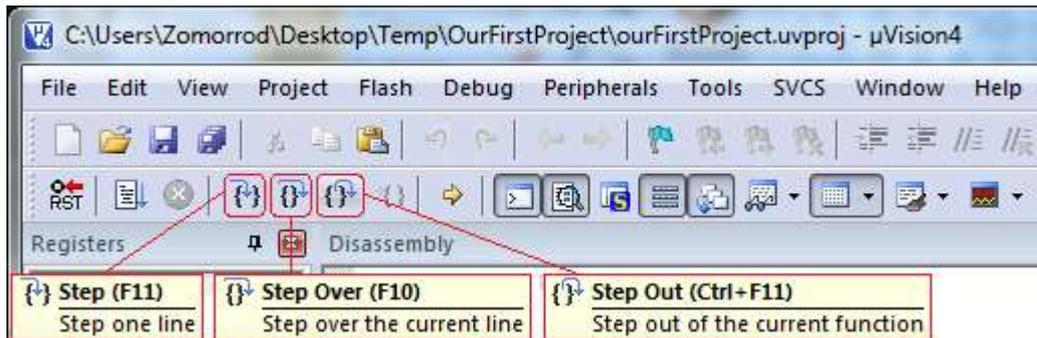
20. If it starts tracing successfully, a cursor appears in front of the next instruction to be executed.



- Go to the **Peripherals** menu. It has tools for monitoring different peripherals. For now, choose **Port 2** from **GPIO Fast Interface**. It shows the registers of **Port 2**; you can see values of registers while tracing the program or change their values by clicking on each bit.



22. 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. To see the difference between them, trace the program once with **Step Over** and then with **Step**. 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.



23. To exit from the debugging mode press **Start/Stop Debug Session**.