Programming the Blue Pill Board & Debugging in Keil Using ST-Link

Step-by-step tutorial

Sepehr Naimi

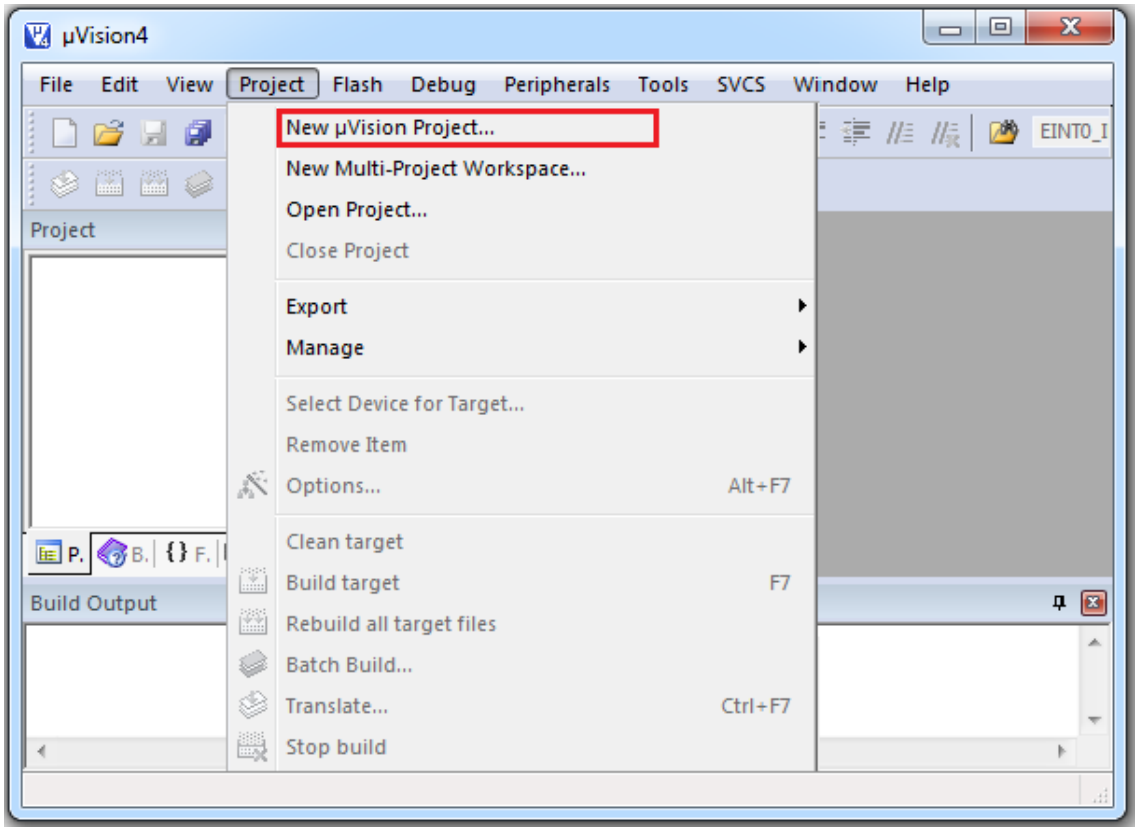
www.NicerLand.com

10/31/2018

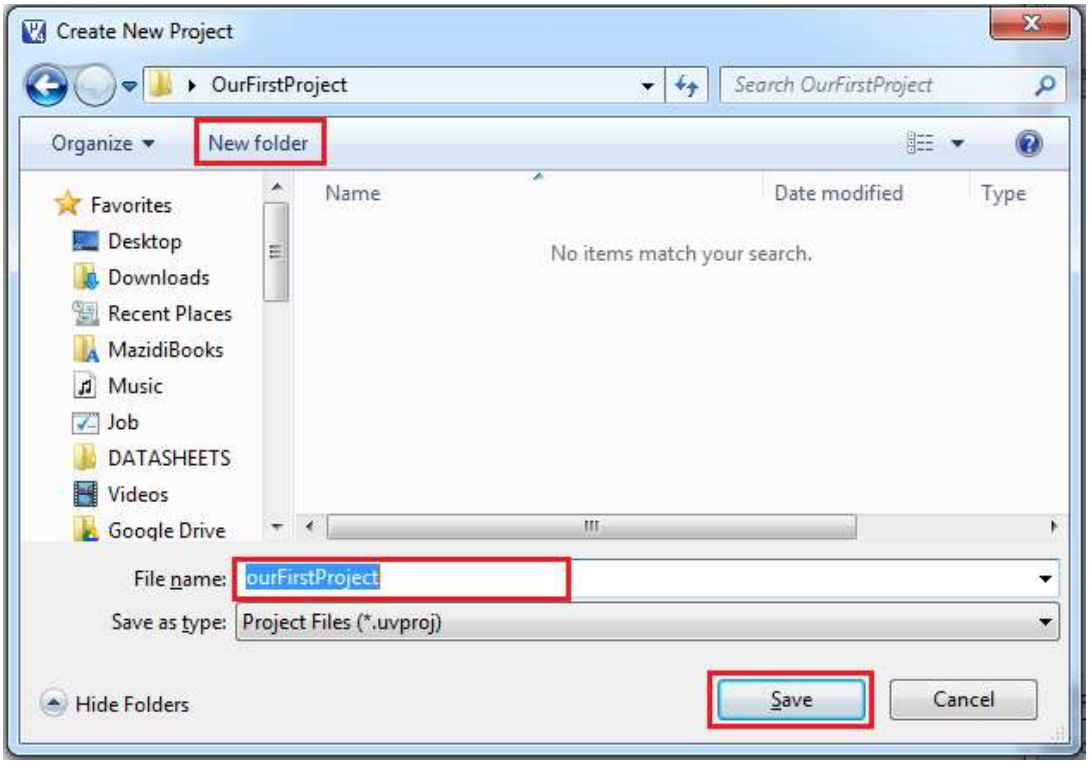
Contents

Creating a 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.

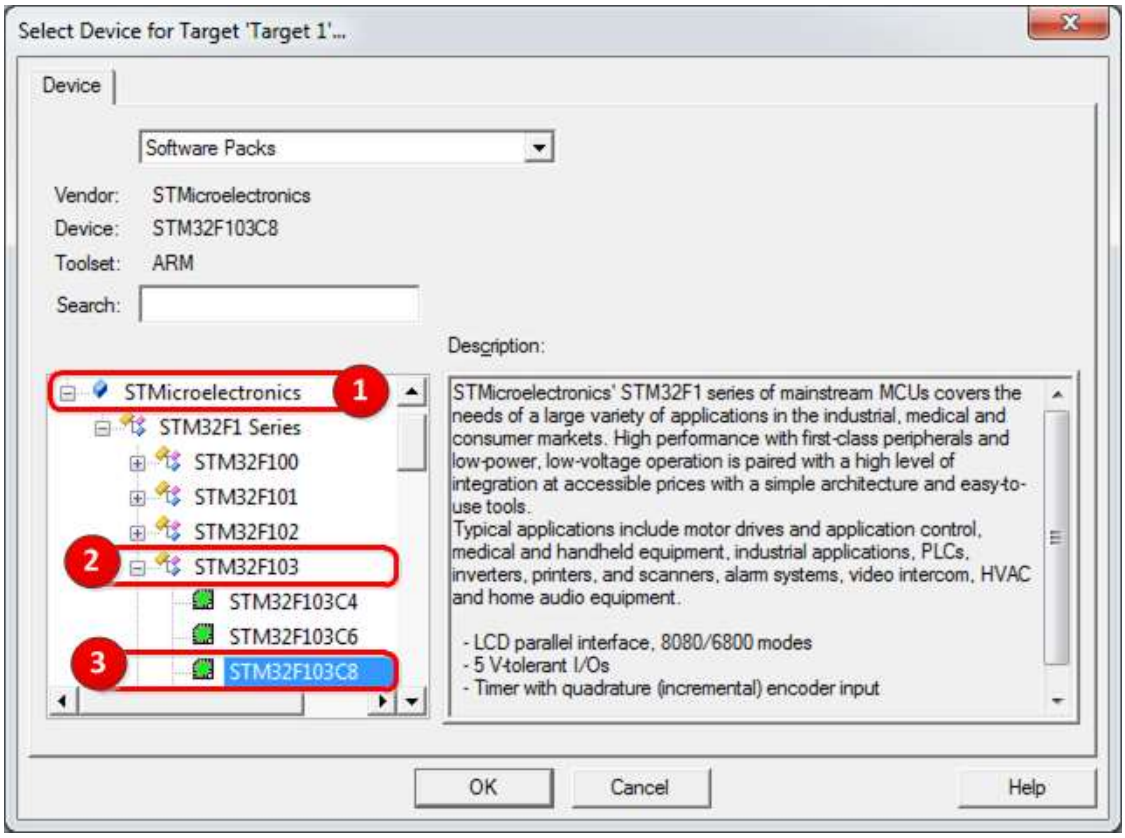


3. Create a new folder and Name it ***OurFirstProject.*** Type the name ***ourFirstProject*** for the project name and click ***Save***.

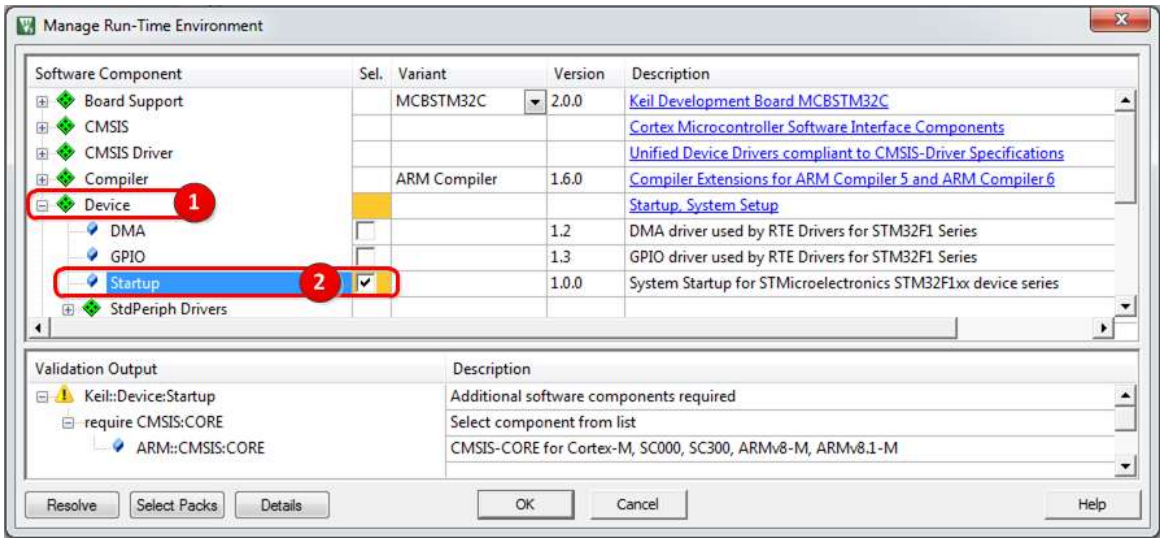


4. In the tree expand ***STMicroelectronics***. (If STMicroelectronics is not in the tree, read "installing Keil and STM32F103" step-by-step tutorial from our website.) Click on ***STM32F103*** and choose ***STM32F103C8***. Then press ***OK***.

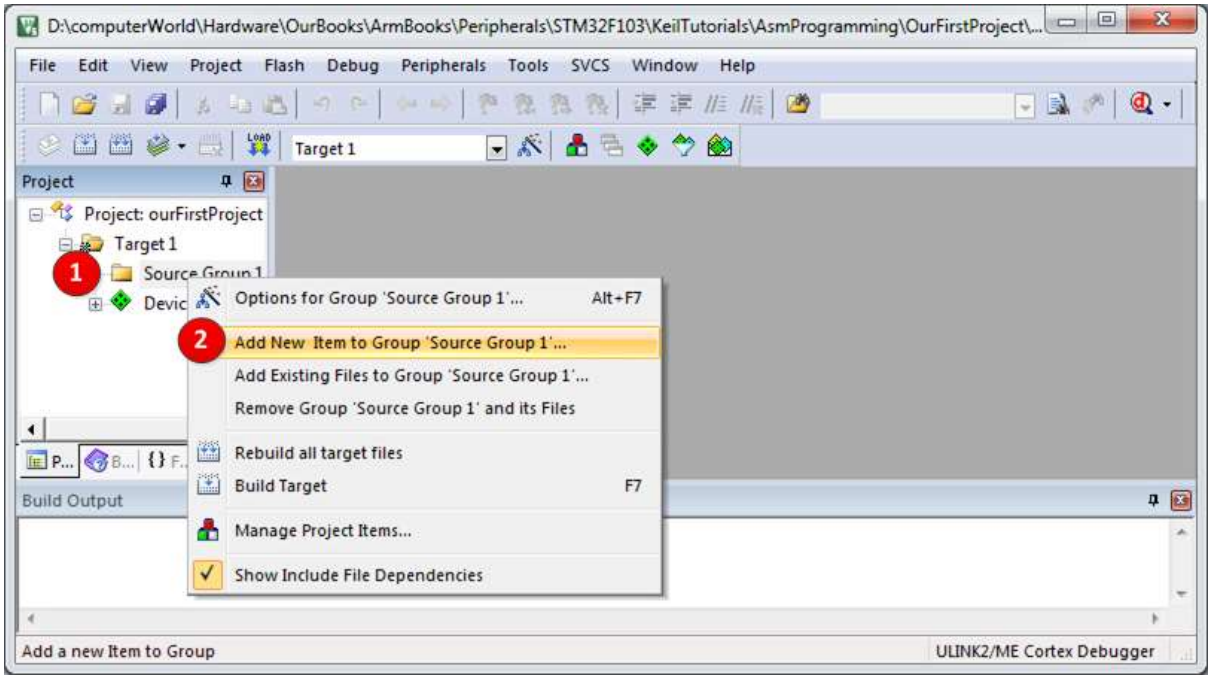
Note
When you choose a chip some general information of the chip is shown in the ***Description*** box.



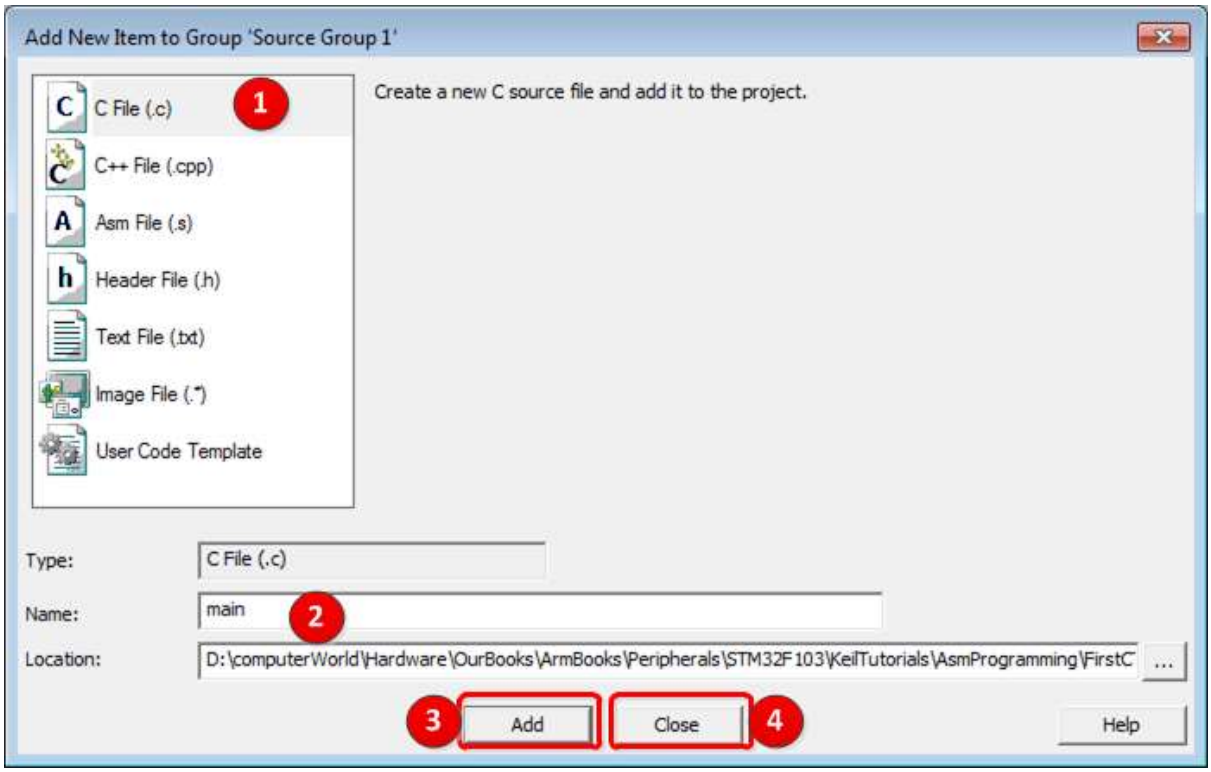
5. From the software component tree click on Device and add the Startup file by clicking the checkbox next to Startup. Then, click on the OK button.



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



7. Choose the type of file as ***C File (.c)*** and name it as ***main***. Click on the ***Add*** button and then click on ***Close***.



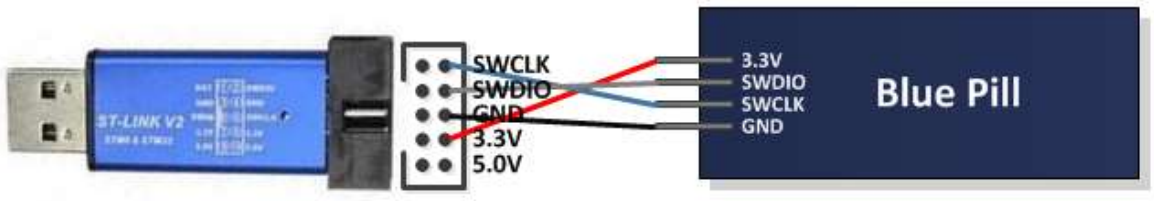
8. Type the following sample program in the ***main.c*** file.

```
#include <stm32f10x.h>

void delay_ms(uint16_t t);

int main()
{
	RCC->APB2ENR |= 0xFC; //Enable GPIO ports clocks

	GPIOC->CRH = 0x44344444; //PC13 as output

	while(1)
	{
		GPIOC->ODR ^= (1<<13); //toggle PC13
		delay_ms(1000);
	}
}


//The following delay is tested with Keil and 72MHz
void delay_ms(uint16_t t)
{
	volatile unsigned long l = 0;
	for(uint16_t i = 0; i < t; i++)
		for(l = 0; l < 6000; l++)
		{
		}
}
```

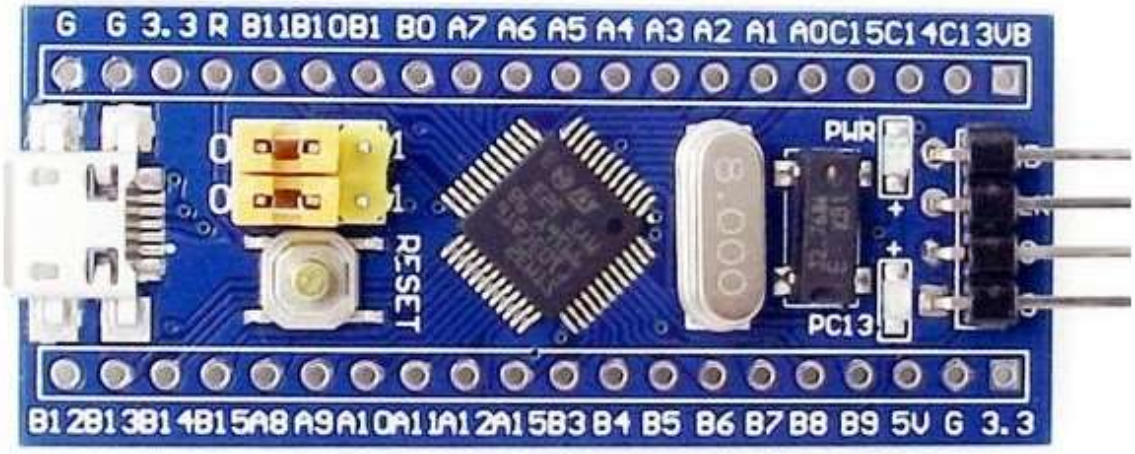
9. Press ***Ctrl+S*** to save the file.

Connecting the ST-Link

10. Connect the ST-Link debugger to the Blue pill using 4 female wires as shown in the following figure.

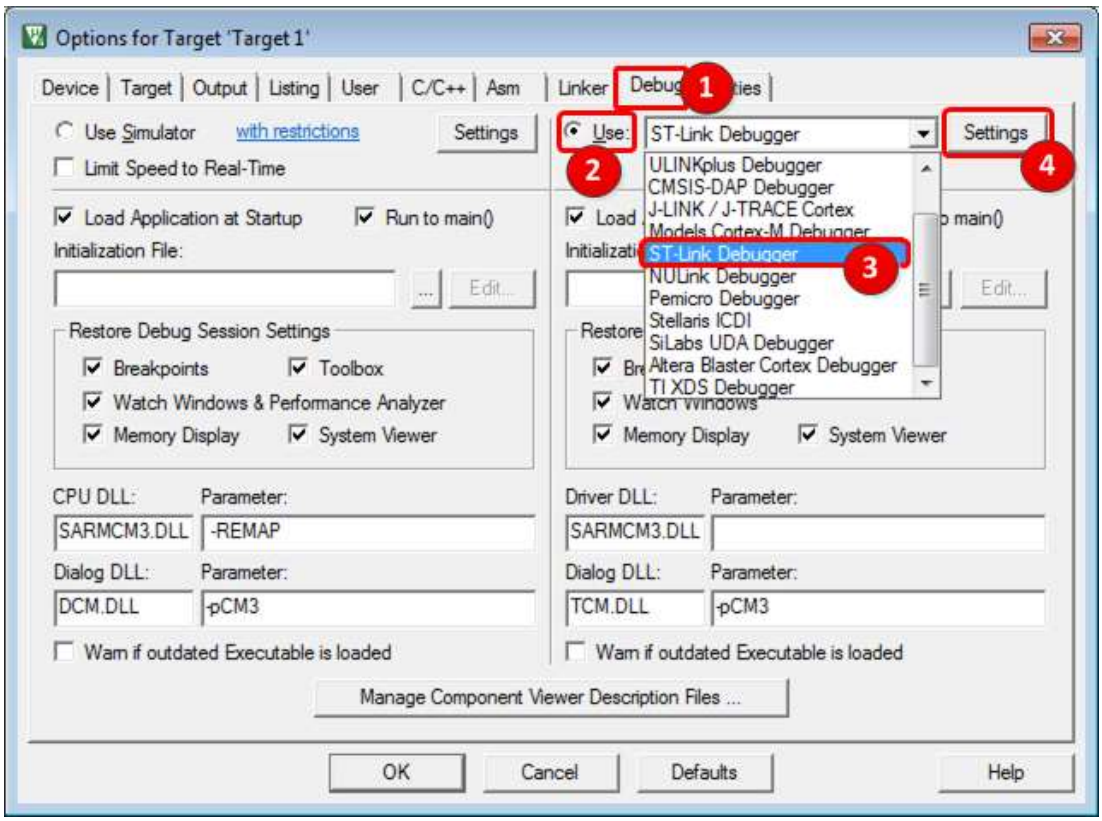


11. Connect the USB socket of the ST-Link debugger to your computer.
12. Configure the yellow jumpers of the board as shown in the following figure. The jumpers should be set to 0.

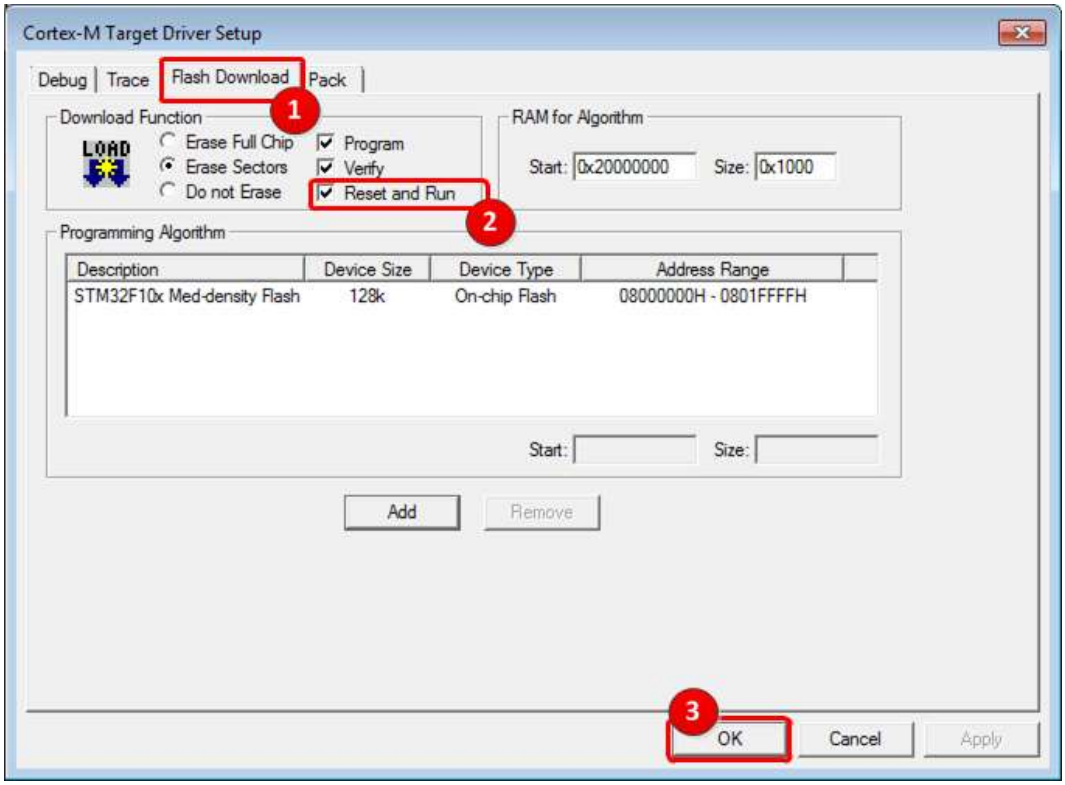


Choosing the ST-Link Debugger

13. Open the ***Projects*** menu and click on ***Options for Target Target1*** or press ***Alt+F7***.
    1. Click on the ***Debug*** tab.
    2. Click on the ***Use*** debugger radio.
    3. Choose ***ST-Link Debugger*** in the Combo box.
    4. Click on the ***Settings*** button.



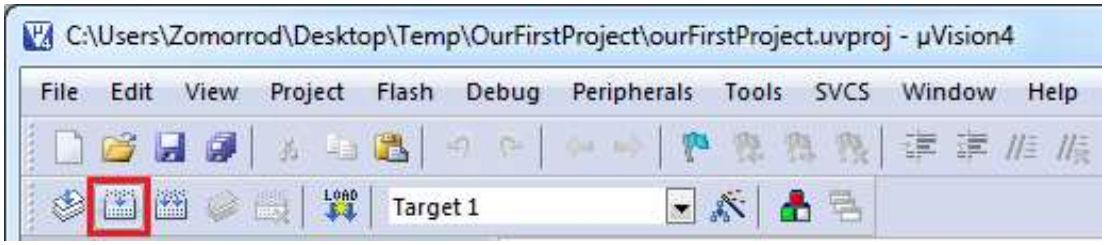
14. In the Target Driver Setup window:
    1. Click on the Flash Download tab.
    2. Enable “Reset and Run”.
    3. Click on the OK button.



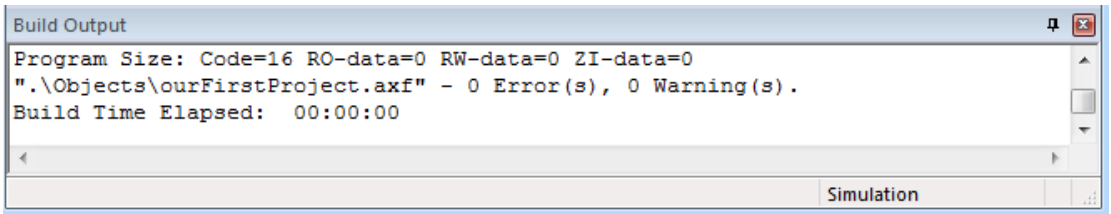
15. Close the Options window by clicking on the ***OK*** button.

Building

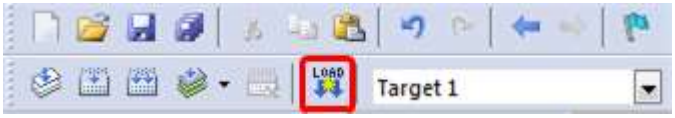
16. To compile click on the ***Build*** icon or choose ***build target*** from the ***Project*** menu.



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



18. Click on the Download icon or press F8 to program the board.

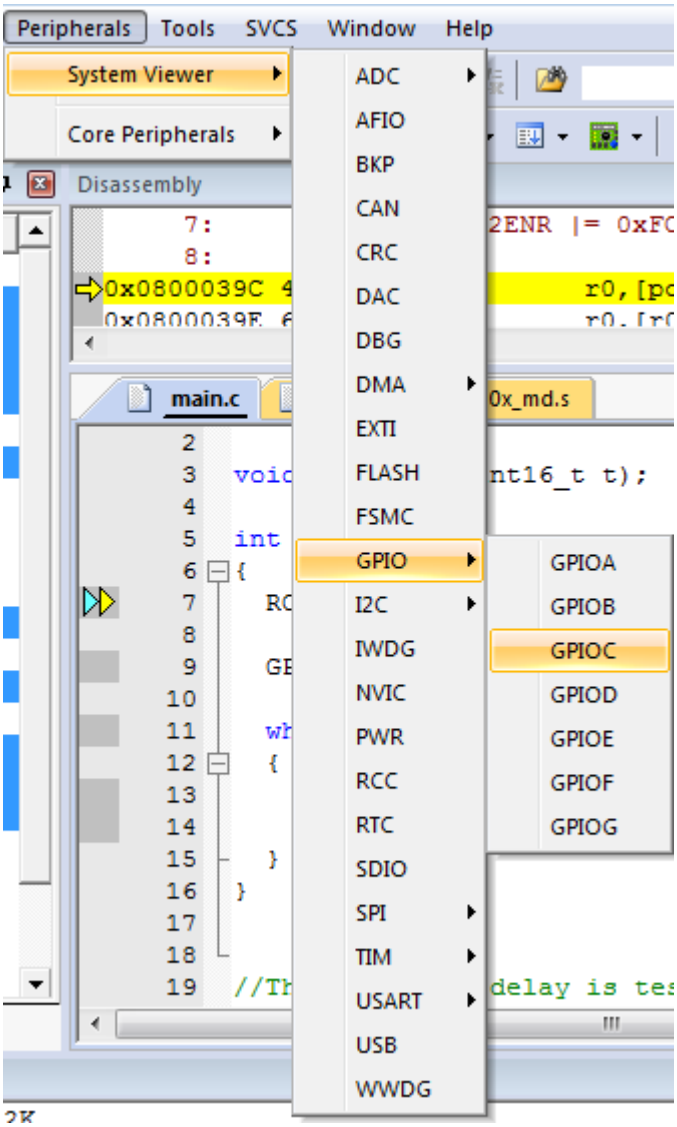


19. Press the ***Reset*** button of the blue pill board. If the board program successfully, the green LED of the board should start blinking.

Debugging and Tracing

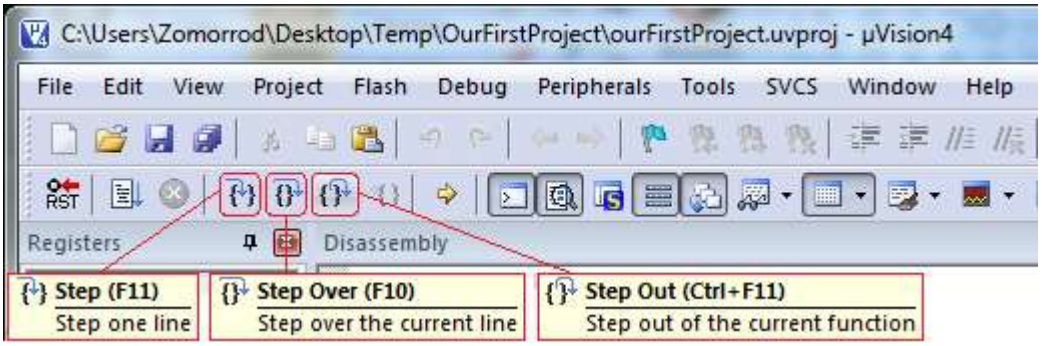
20. To start debugging click on ***Start/Stop Debug Session*** icon or choose ***Start/Stop Debug Session*** from the ***Debug*** menu. (or simply press ***Ctrl+F5***)

21. Go to the ***Peripherals*** menu and then ***System Viewer***. It has tools for monitoring different peripherals. For now, choose ***GPIOC*** from ***GPIO***. It shows the registers of ***GPIOC***; 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***.