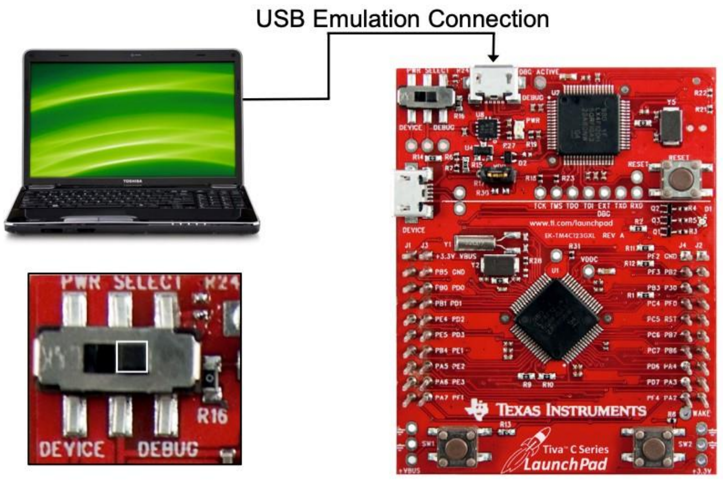
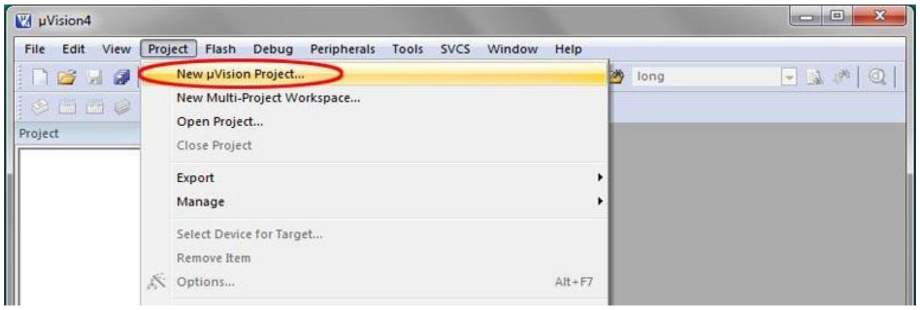
Getting Started in C programming with Keil uVision

Board Connection and Power Switch Setting

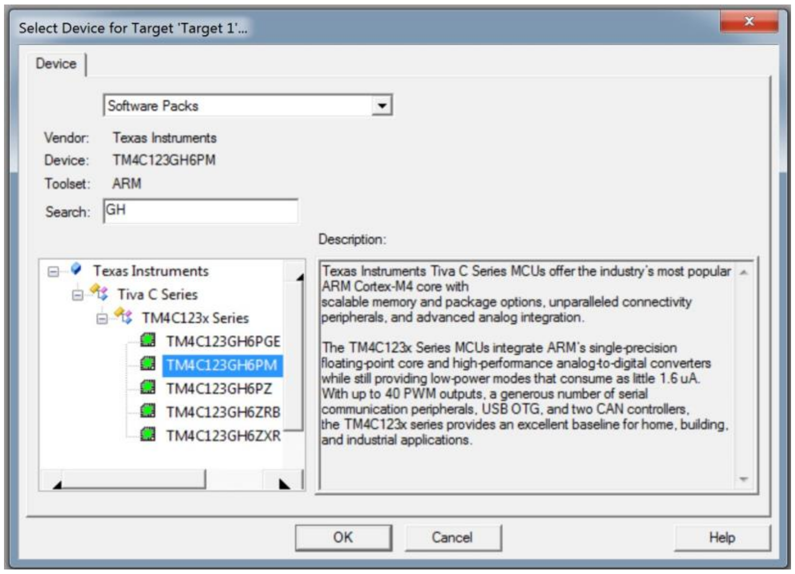


Create a Project with the Guidance of Project Wizard

1. From the menu, select Project > New uVision Project…


µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
New µVision Project...
New Multi-Project Workspace...
Open Project...
Close Project
Export
Manage
Select Device for Target...
Remove Item
Options... Alt+F7
Project

2. A **Select Device for Target 'Target 1'…** window will pop up. The device used on the Tiva LaunchPad is Texas Instruments TM4C123GH6PM. To select this device, first you need to click on the **+** sign to the left of Texas Instruments to expand the selections.


Select Device for Target 'Target 1'...

Device

Software Packs

Vendor: Texas Instruments
Device: TM4C123GH6PM
Toolset: ARM
Search: GH

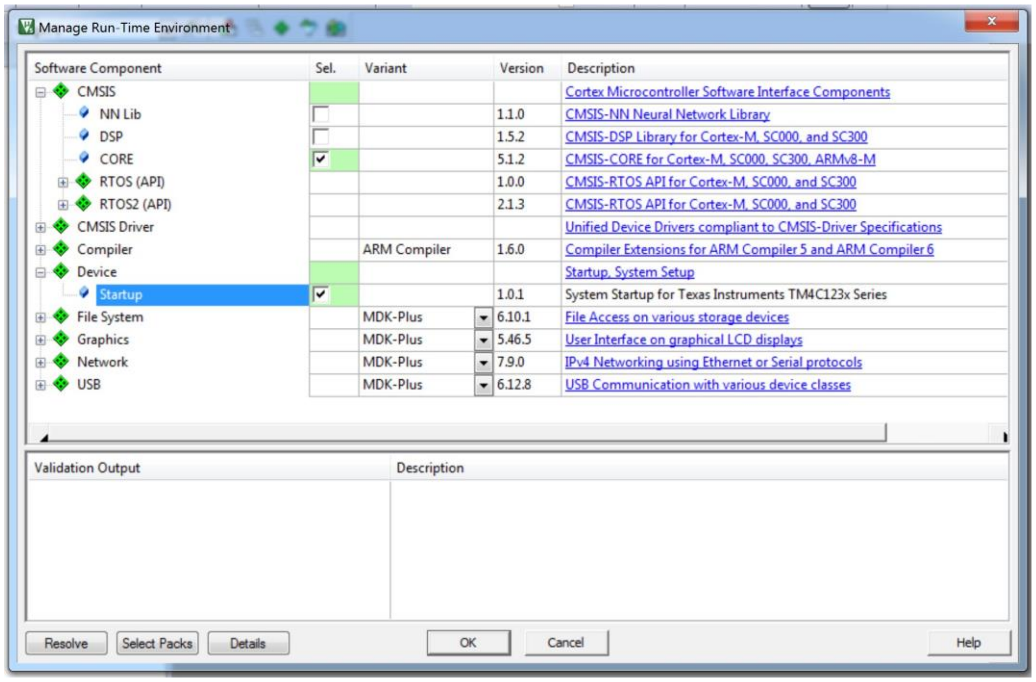
- Texas Instruments
  - Tiva C Series
    - TM4C123x Series
      - TM4C123GH6PGE
      - TM4C123GH6PM
      - TM4C123GH6PZ
      - TM4C123GH6ZRB
      - TM4C123GH6ZXR

Description:

Texas Instruments Tiva C Series MCUs offer the industry's most popular ARM Cortex-M4 core with scalable memory and package options, unparalleled connectivity peripherals, and advanced analog integration.

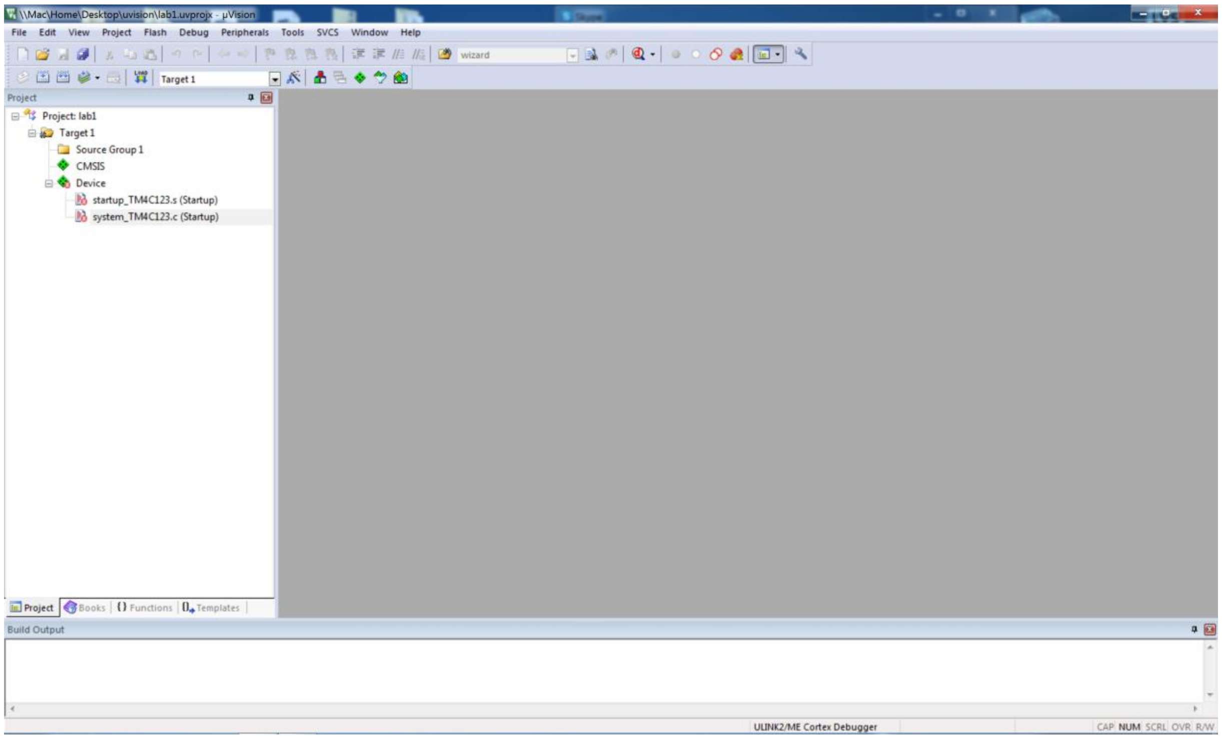
The TM4C123x Series MCUs integrate ARM's single-precision floating-point core and high-performance analog-to-digital converters while still providing low-power modes that consume as little 1.6 uA. With up to 40 PWM outputs, a generous number of serial communication peripherals, USB OTG, and two CAN controllers, the TM4C123x series provides an excellent baseline for home, building, and industrial applications.

3. Scroll down the device selections to find **TM4C123GH6PM** (you may type "GH" in the search field to filter the devices), click on the device name then click **OK**.
4. A dialog pops up to manage the run-time environment. Select CMSIS ➔ CORE and Device ➔ Startup.


Manage Run-Time Environment

Software Component	Sel.	Variant	Version	Description
CMSIS				Cortex Microcontroller Software Interface Components
NN Lib	☐		1.1.0	CMSIS-NN Neural Network Library
DSP	☐		1.5.2	CMSIS-DSP Library for Cortex-M, SC000, and SC300
CORE	✓		5.1.2	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M
RTOS (API)			1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
RTOS2 (API)			2.1.3	CMSIS-RTOS API for Cortex-M, SC000, and SC300
CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications
Compiler		ARM Compiler	1.6.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
Device				Startup, System Setup
Startup	✓		1.0.1	System Startup for Texas Instruments TM4C123x Series
File System		MDK-Plus	6.10.1	File Access on various storage devices
Graphics		MDK-Plus	5.46.5	User Interface on graphical LCD displays
Network		MDK-Plus	7.9.0	IPv4 Networking using Ethernet or Serial protocols
USB		MDK-Plus	6.12.8	USB Communication with various device classes

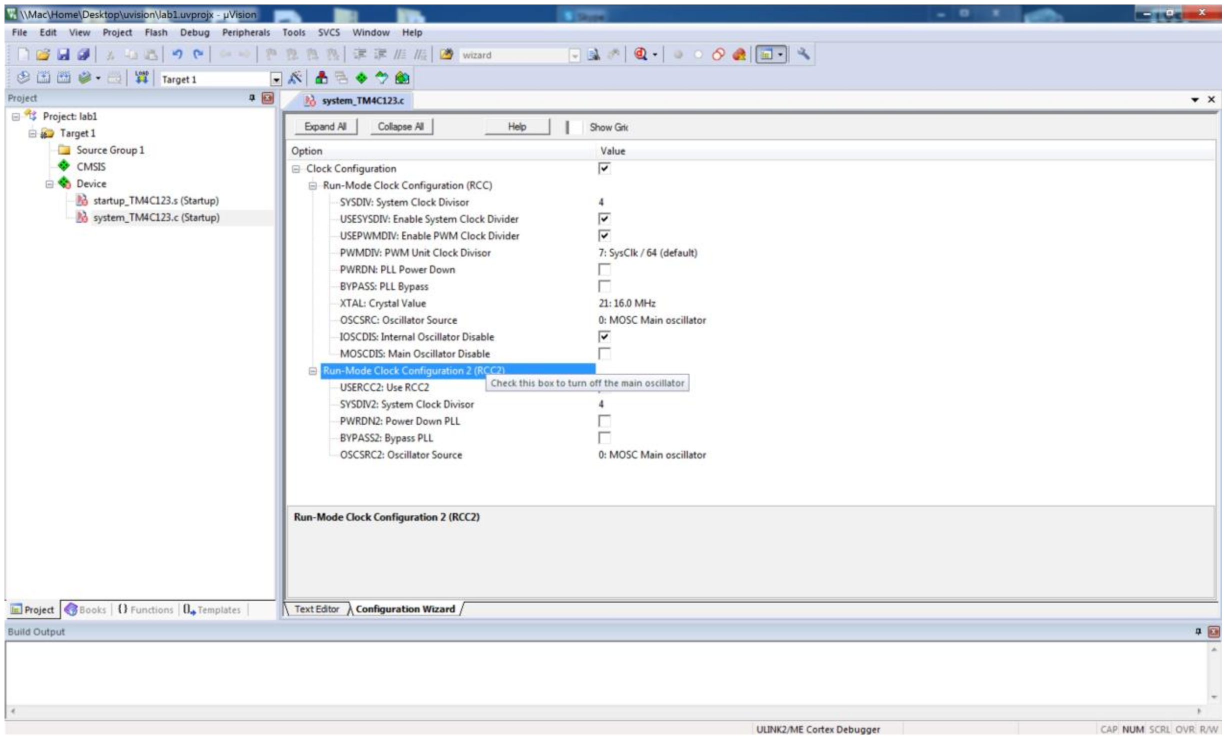
5. In the Project window, a target is created with the default name **Target1**. Click on the **+** sign to the left of Target1 to expand the folder.


\\Mac\Home\Desktop\uvision\lab1.uvprojx - µVision

Project: lab1
- Target 1
  - Source Group 1
  - CMSIS
  - Device
    - startup_TM4C123.s (Startup)
    - system_TM4C123.c (Startup)

Add a Source File to the Project

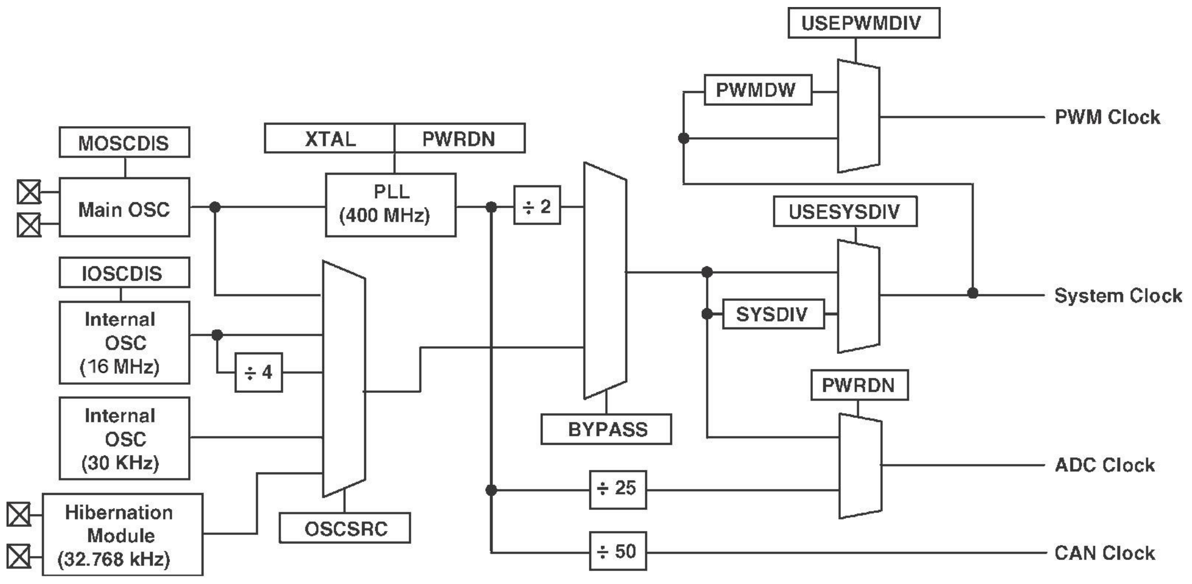
6. To configure the clock, double click on the "system_TM4C.c" file, then select the "Configuration Wizard" tab



Option	Value
Clock Configuration	✓
Run-Mode Clock Configuration (RCC)	
SYSDIV: System Clock Divisor	4
USESYSDIV: Enable System Clock Divider	✓
USEPWMDIV: Enable PWM Clock Divider	✓
PWMDIV: PWM Unit Clock Divisor	7: SysClk / 64 (default)
PWRDN: PLL Power Down	
BYPASS: PLL Bypass	
XTAL: Crystal Value	21: 16.0 MHz
OSCSRC: Oscillator Source	0: MOSC Main oscillator
IOSCDIS: Internal Oscillator Disable	✓
MOSCDIS: Main Oscillator Disable	
Run-Mode Clock Configuration 2 (RCC2)	
USERCC2: Use RCC2	
SYSDIV2: System Clock Divisor	4
PWRDN2: Power Down PLL	
BYPASS2: Bypass PLL	
OSCSRC2: Oscillator Source	0: MOSC Main oscillator

Check this box to turn off the main oscillator

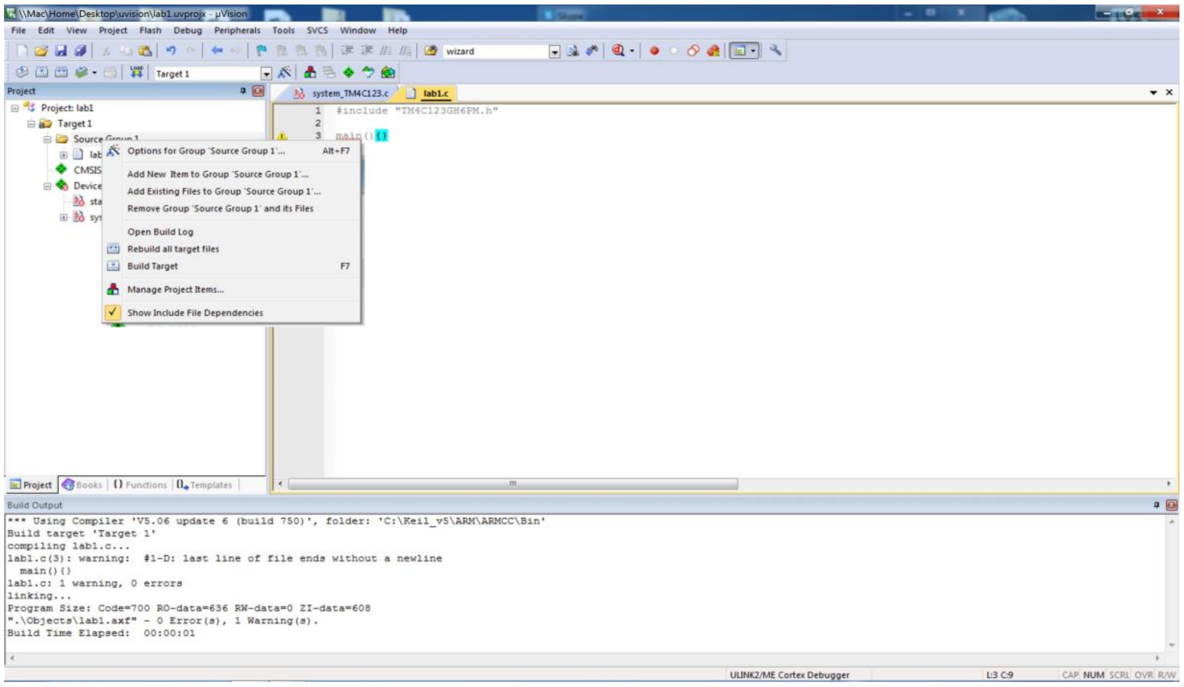
Run-Mode Clock Configuration 2 (RCC2)



7. Use the datasheet to set RCC and RCC2 different fields to configure the clock.

Add a Source File to the Project

8. Click the **New** button to add a new text file to the display with the default name **Text1**.
9. From the menu, select **File > Save As…** to open the Save As dialog box. Browse to the project folder if it is not already there. Type in the file name **main.c** and click **Save**.
10. You will notice the file name in the tab changed to **main.c**
11. The new file needs be added to the project. Right click on the folder **Source Group 1** in the Project window and select **Add Existing Files to Group 'Source Group 1'…**


\\Mac\Home\Desktop\uvision\lab1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
wizard
Target 1
Project
Project: lab1
Target 1
Source Group 1
CMSIS
Device
system_TM4C123.c
lab1.c
Options for Group 'Source Group 1'... Alt+F7
Add New Item to Group 'Source Group 1'...
Add Existing Files to Group 'Source Group 1'...
Remove Group 'Source Group 1' and its Files
Open Build Log
Rebuild all target files
Build Target F7
Manage Project Items...
Show Include File Dependencies
Project
Books
Functions
Templates
Build Output
ULINK2/ME Cortex Debugger
L:3 C:9
CAP NUM SCRL OVR R/W

```
#include "TM4C123GH6PM.h"

main(){}
```

```
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'Target 1'
compiling lab1.c...
lab1.c(3): warning:  #1-D: last line of file ends without a newline
  main(){}
lab1.c: 1 warning, 0 errors
linking...
Program Size: Code=700 RO-data=636 RW-data=0 ZI-data=608
".\Objects\lab1.axf" - 0 Error(s), 1 Warning(s).
Build Time Elapsed:  00:00:01
```

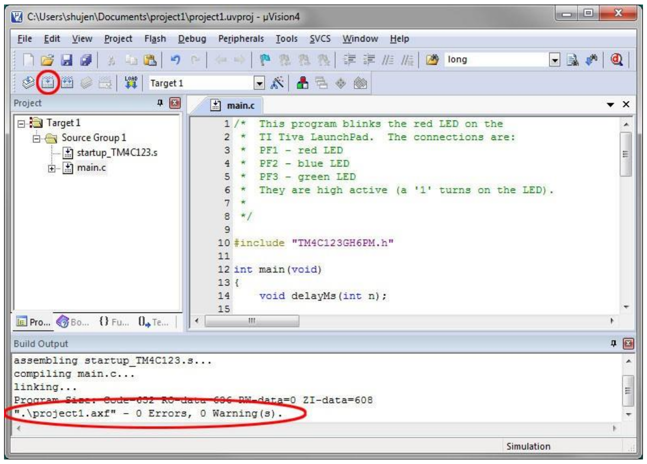
12. In the dialog box, browse to the project folder if it is not already there. Click select **main.c** then click **Add**.
13. The file should appear in the project window under Source Group 1 folder. Click Close to close the dialog box.
14. Copy and paste the following code into the main.c editor window.

```
/*  This program blinks the red LED on the
 *  TI Tiva LaunchPad.  The connections are:
 *  PF1 - red LED
 *  PF2 - blue LED
 *  PF3 - green LED
 *  They are high active (a '1' turns on the LED).
 */
#include "TM4C123GH6PM.h"
void delayMs(int n);
int main(void)
{
    // enable clock to GPIOF at clock gating control register
    SYSCTL->RCGCGPIO |= 0x20;
    // enable the GPIO pins for the LED (PF3, 2 1) as output
    GPIOF->DIR |= 0x0e;
    // enable the GPIO pins for digital function
    GPIOF->DEN |= 0x0e;
    while(1)
    {
        GPIOF->DATA = 0x02;     // turn on red LED
        delayMs(500);
        GPIOF->DATA = 0;        // turn off red LED
        delayMs(500);
    }
}
// delay in milliseconds (16 MHz CPU clock)
void delayMs(int n)
{
    int i, j;
    for(i = 0 ; i < n; i++)
         for(j = 0; j < 3180; j++)
             {}  // do nothing for 1 ms
}
```

15. The file name in the tab will have an '*****' next to it. It symbolizes that the file has been changed without saving. You may click the save button to save the file or proceed to build the project. By default the file is automatically saved before the build.

Build the Project and Download to the Target

16. Click on the "Options for target" button, then press "target" tab, then for ARM compiler choose "use default compiler version 5".
17. Click on the **Build** button and wait for the build to complete. Make sure there are no error messages and review all the warning messages.



C:\Users\shujen\Documents\project1\project1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
long
Target 1
Project
Target 1
Source Group 1
startup_TM4C123.s
main.c
main.c
Build Output
Simulation

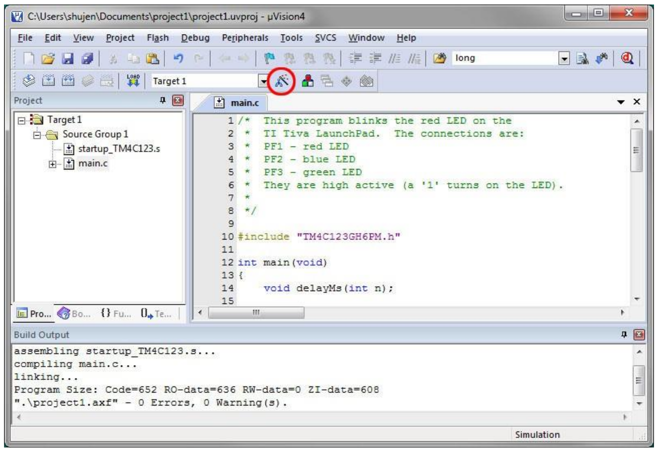
```
/*  This program blinks the red LED on the
 *  TI Tiva LaunchPad.  The connections are:
 *  PF1 - red LED
 *  PF2 - blue LED
 *  PF3 - green LED
 *  They are high active (a '1' turns on the LED).
 *
 */

#include "TM4C123GH6PM.h"

int main(void)
{
    void delayMs(int n);
```

```
assembling startup_TM4C123.s...
compiling main.c...
linking...
Program Size: Code=652 RO-data=636 RW-data=0 ZI-data=608
".\project1.axf" - 0 Errors, 0 Warning(s).
```

18. Before downloading the program to the target, we need to set up the connection between the uVision IDE and the debugger on the target circuit board. Click on the **Target Options** button.



C:\Users\shujen\Documents\project1\project1.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
long
Target 1
Project
Target 1
Source Group 1
startup_TM4C123.s
main.c
main.c
Build Output
Simulation

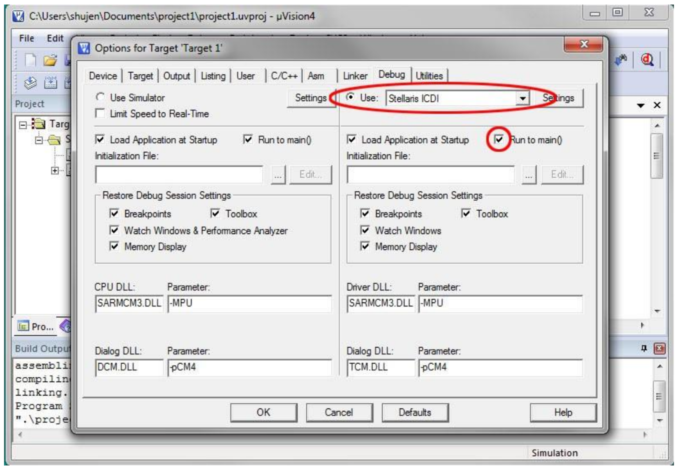
```
/*  This program blinks the red LED on the
 *  TI Tiva LaunchPad.  The connections are:
 *  PF1 - red LED
 *  PF2 - blue LED
 *  PF3 - green LED
 *  They are high active (a '1' turns on the LED).
 *
 */

#include "TM4C123GH6PM.h"

int main(void)
{
    void delayMs(int n);
```

```
assembling startup_TM4C123.s...
compiling main.c...
linking...
Program Size: Code=652 RO-data=636 RW-data=0 ZI-data=608
".\project1.axf" - 0 Errors, 0 Warning(s).
```

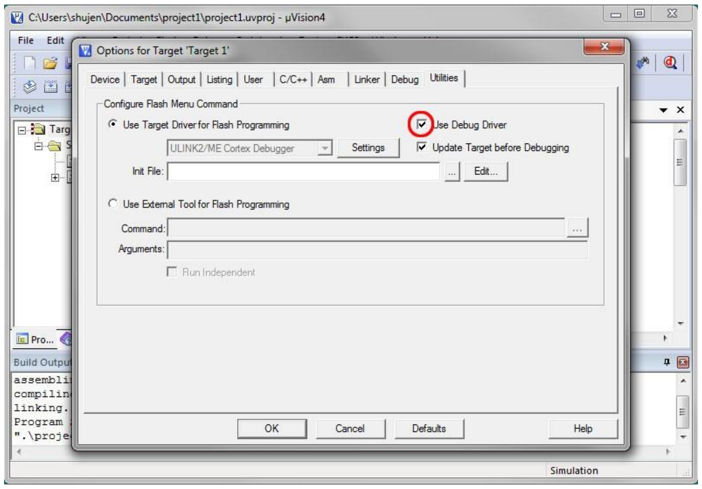
19. Select **Debug** tab and click the radio button and select Stellaris ICDI in the pull-down menu if you intend to run the program on the target circuit. Alternatively, you may select the radio button of **Use Simulator** on the left to test the code with the simulator.
20. Check the box **Run to main().**



Left: ( ) Use Simulator; ☐ Limit Speed to Real-Time; ☑ Load Application at Startup; ☑ Run to main(); Initialization File:; Restore Debug Session Settings: ☑ Breakpoints, ☑ Toolbox, ☑ Watch Windows & Performance Analyzer, ☑ Memory Display; CPU DLL: SARMCM3.DLL, Parameter: -MPU; Dialog DLL: DCM.DLL, Parameter: -pCM4

Right: (•) Use: Stellaris ICDI; ☑ Load Application at Startup; ☑ Run to main(); Initialization File:; Restore Debug Session Settings: ☑ Breakpoints, ☑ Toolbox, ☑ Watch Windows, ☑ Memory Display; Driver DLL: SARMCM3.DLL, Parameter: -MPU; Dialog DLL: TCM.DLL, Parameter: -pCM4

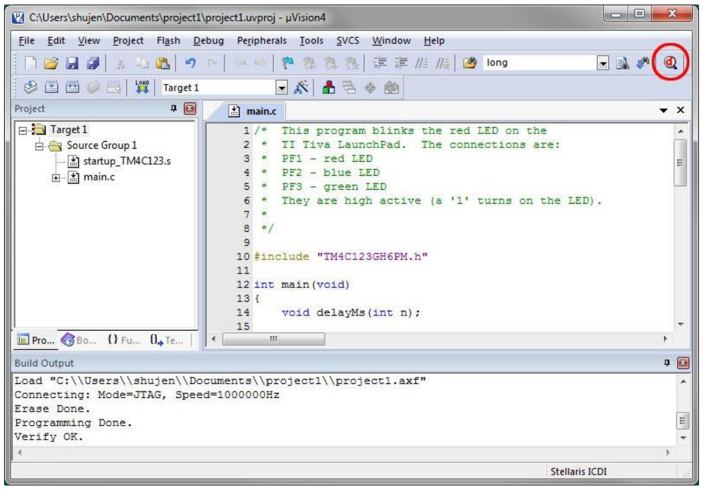
21. Select Utilities tab and check the **Use Debug Driver**.
22. Click **OK** button to close the Options window



Configure Flash Menu Command
(●) Use Target Driver for Flash Programming
[✓] Use Debug Driver
ULINK2/ME Cortex Debugger
[✓] Update Target before Debugging
Init File:
( ) Use External Tool for Flash Programming
Command:
Arguments:
[ ] Run Independent

```
assembli
compilin
linking.
Program
".\proje
```

23. You should have the ICDI device drivers installed on your computer. If you have not done so, download Stellaris ICDI device drivers and the installation guide from Texas Instrument at http://www.ti.com/tool/stellaris_icdi_drivers. Follow the instructions to install the drivers.
24. Also download Stellaris AddOn from https://developer.arm.com/documentation/ka002280/latest/
25. To download and debug the program, you need to go into the debugger. Click the **Debug** button. If you are running MDKARM Lite Edition, a message window will pop up to warn of the code size limitation. Click **OK** to proceed.



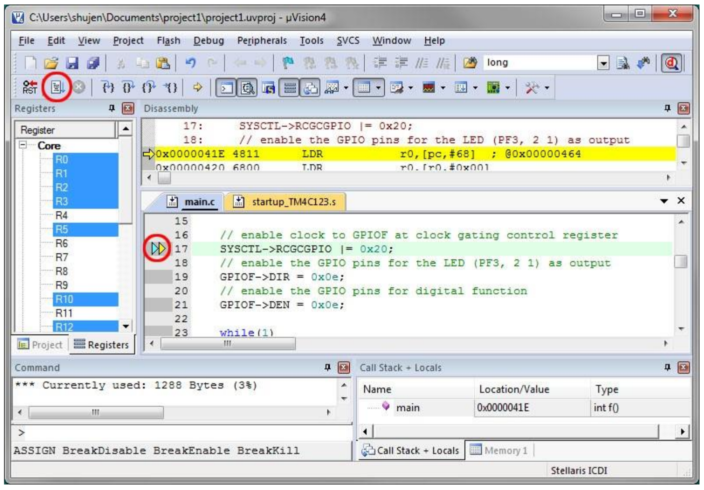
```
/*  This program blinks the red LED on the
 *  TI Tiva LaunchPad.  The connections are:
 *  PF1 - red LED
 *  PF2 - blue LED
 *  PF3 - green LED
 *  They are high active (a '1' turns on the LED).
 *
 */

#include "TM4C123GH6PM.h"

int main(void)
{
    void delayMs(int n);
```

```
Load "C:\\Users\\shujen\\Documents\\project1\\project1.axf"
Connecting: Mode=JTAG, Speed=1000000Hz
Erase Done.
Programming Done.
Verify OK.
```

26. When entering the debugger, the IDE changes the view to the debug view. The program counter is at the first executable line of the main( ) function as indicated by the arrows. Click the **Run** button and the program execution should start. If everything has been done correctly, the red LED on the LaunchPad should be blinking.



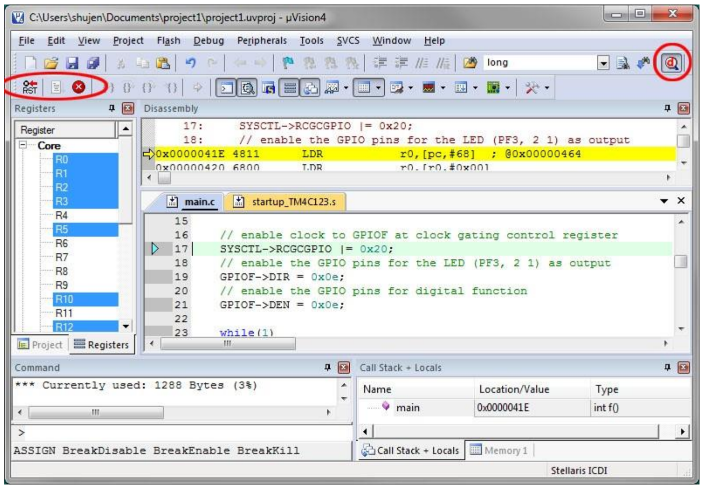
```
    17:     SYSCTL->RCGCGPIO |= 0x20;
    18:     // enable the GPIO pins for the LED (PF3, 2 1) as output
0x0000041E 4811      LDR      r0,[pc,#68]  ; @0x00000464
0x00000420 6800      LDR      r0,[r0,#0x00]
```

```
15
16      // enable clock to GPIOF at clock gating control register
17      SYSCTL->RCGCGPIO |= 0x20;
18      // enable the GPIO pins for the LED (PF3, 2 1) as output
19      GPIOF->DIR = 0x0e;
20      // enable the GPIO pins for digital function
21      GPIOF->DEN = 0x0e;
22
23      while(1)
```

```
*** Currently used: 1288 Bytes (3%)
>
ASSIGN BreakDisable BreakEnable BreakKill
```

Name	Location/Value	Type
main	0x0000041E	int f()

27. To the left, you will find the stop button (red circle with a white cross) and the reset button (with RST in it). To stop debugging, click on the debug button on the right. When you left the debugger, the program in the Tiva LaunchPad will continue running. The program will run whenever the power is applied to it.



```
    17:     SYSCTL->RCGCGPIO |= 0x20;
    18:     // enable the GPIO pins for the LED (PF3, 2 1) as output
0x0000041E 4811      LDR      r0,[pc,#68]  ; @0x00000464
0x00000420 6800      LDR      r0,[r0,#0x00]
```

```
15
16      // enable clock to GPIOF at clock gating control register
17      SYSCTL->RCGCGPIO |= 0x20;
18      // enable the GPIO pins for the LED (PF3, 2 1) as output
19      GPIOF->DIR = 0x0e;
20      // enable the GPIO pins for digital function
21      GPIOF->DEN = 0x0e;
22
23      while(1)
```

```
*** Currently used: 1288 Bytes (3%)
>
ASSIGN BreakDisable BreakEnable BreakKill
```

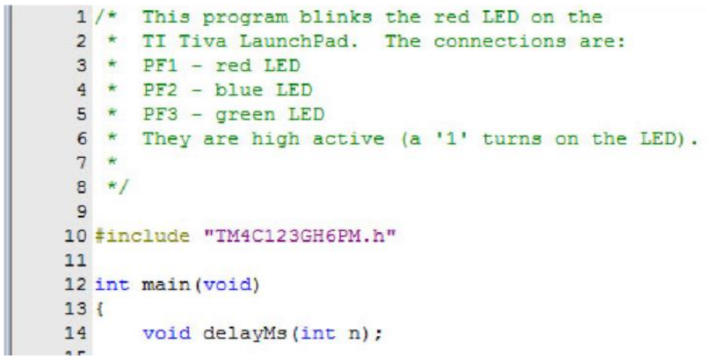
Name	Location/Value	Type
main	0x0000041E	int f()

28. Congratulations! You have finished your first Keil uVision programming for the Tiva LaunchPad.

Explanations of the Program

Line 10: Include the register definition file. Starting from uVision version 4.72, the register definitions are following the CMSIS format.

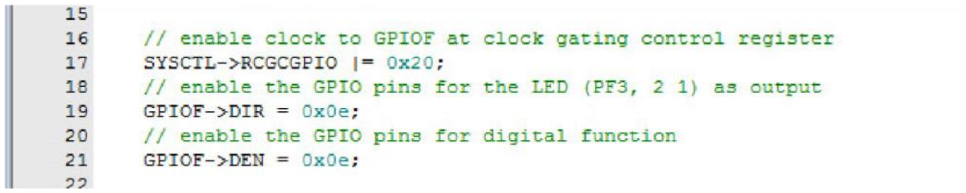
Line 14: function prototype for delayMs( ).



```
 1 /*  This program blinks the red LED on the
 2  *  TI Tiva LaunchPad.  The connections are:
 3  *  PF1 - red LED
 4  *  PF2 - blue LED
 5  *  PF3 - green LED
 6  *  They are high active (a '1' turns on the LED).
 7  *
 8  */
 9
10 #include "TM4C123GH6PM.h"
11
12 int main(void)
13 {
14     void delayMs(int n);
```

Line 17: All the peripherals are powered up without clock to conserve energy. Before configuring the port setting, the clock needs to be enabled. This line turns on the clock to Port F.

Line 19, 21: In order to drive the LEDs, the port pins need to be digital output. They are powered up with these bits set to zero. These two statements set those three bits to digital output.

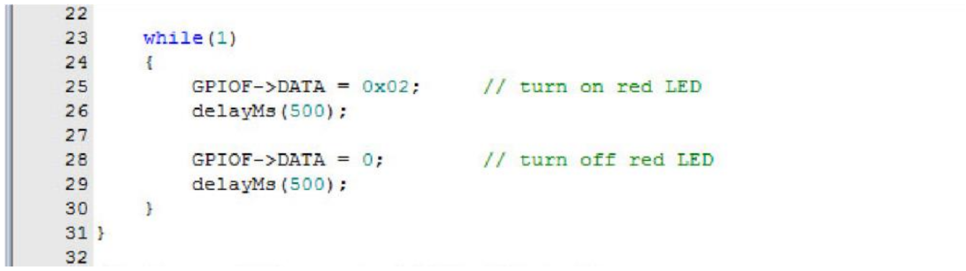


```
15
16     // enable clock to GPIOF at clock gating control register
17     SYSCTL->RCGCGPIO |= 0x20;
18     // enable the GPIO pins for the LED (PF3, 2 1) as output
19     GPIOF->DIR = 0x0e;
20     // enable the GPIO pins for digital function
21     GPIOF->DEN = 0x0e;
22
```

Line 23: The infinite loop. An embedded system program does not exit. It must have an infinite loop.

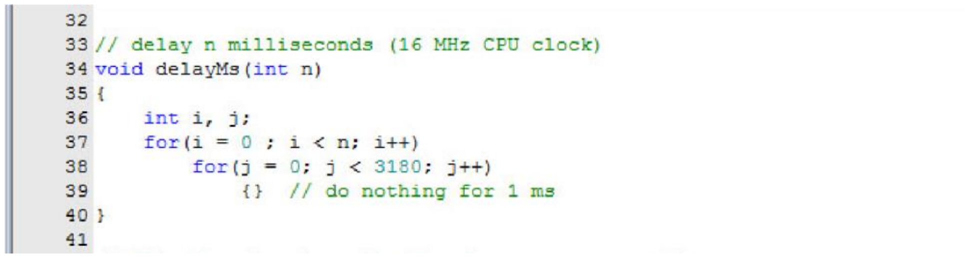
Line 25-26: Turn on the red LED for 500 ms.

Line 28-29: Turn off the red LED for 500 ms.



```
22
23     while(1)
24     {
25         GPIOF->DATA = 0x02;      // turn on red LED
26         delayMs(500);
27
28         GPIOF->DATA = 0;         // turn off red LED
29         delayMs(500);
30     }
31 }
32
```

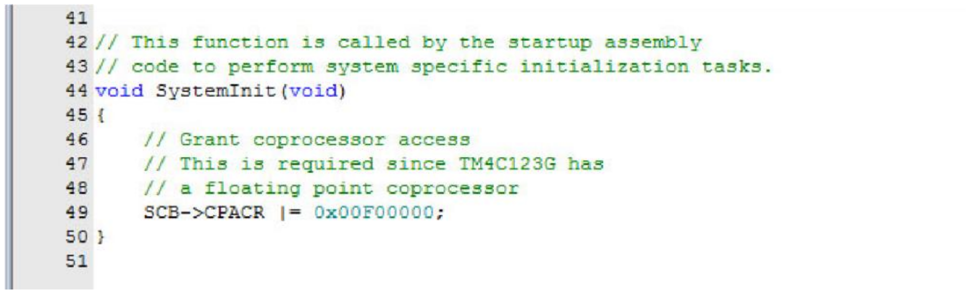
Line 34-40: This delay function takes a parameter and loops as many times as the parameter specifies. Each loop consists of four clock cycles. If the clock runs at 16 MHz so delay(500); results in 500 ms.



```
32
33 // delay n milliseconds (16 MHz CPU clock)
34 void delayMs(int n)
35 {
36     int i, j;
37     for(i = 0 ; i < n; i++)
38         for(j = 0; j < 3180; j++)
39             {}  // do nothing for 1 ms
40 }
41
```

Line 44-50: In the uVision, the initialization function is eliminated from the startup_TM4C123.s assembly file but the assembly code calls for a function SystemInit( ). The programmer needs to provide the function and in the function enable coprocessor access.

Line 51: uVision expects a newline at the end of the file, otherwise, you will receive a warning message.



```
41
42 // This function is called by the startup assembly
43 // code to perform system specific initialization tasks.
44 void SystemInit(void)
45 {
46     // Grant coprocessor access
47     // This is required since TM4C123G has
48     // a floating point coprocessor
49     SCB->CPACR |= 0x00F00000;
50 }
51
```