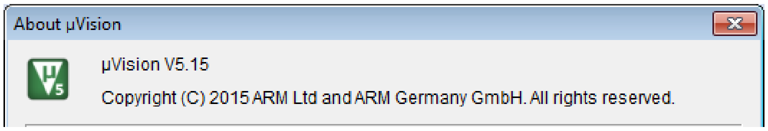
Getting Started in Assembly Programming with Keil uVision and MSP432

This tutorial is written on uVision v5.15 and Texas Instruments MSP432 LaunchPad.



About µVision

µVision V5.15

Copyright (C) 2015 ARM Ltd and ARM Germany GmbH. All rights reserved.

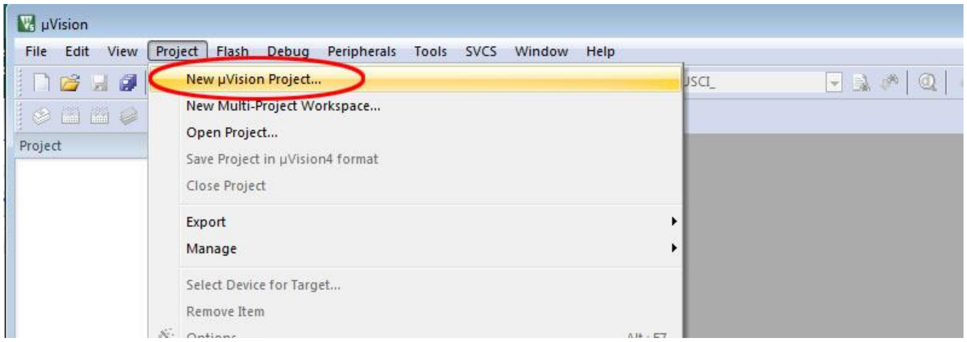
Assembly Programming with MSP432

MSP432 has an ARM Cortex-M4F core. It supports Thumb-2 code only. As described in Chapter 8 of the text, with Unified Assembly Language, you may write assembly instructions the same way you write them for ARM processor. Even though the binary code generated for Thumb-2 is different than the ARM, they perform the same function. The only minor differences are in the ranges of the immediate value.

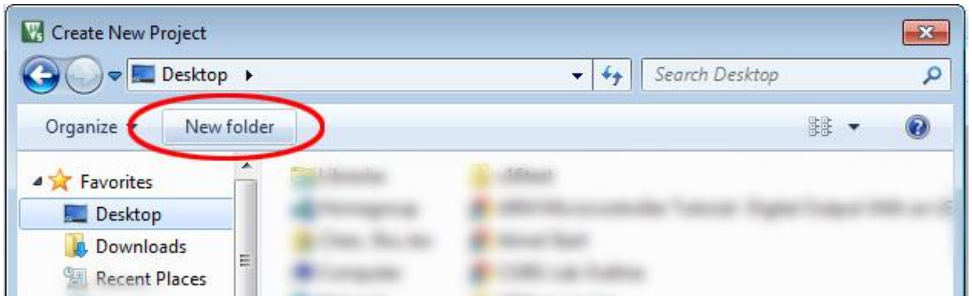
1. Launch Keil uVision.

Create a Project with Project Wizard

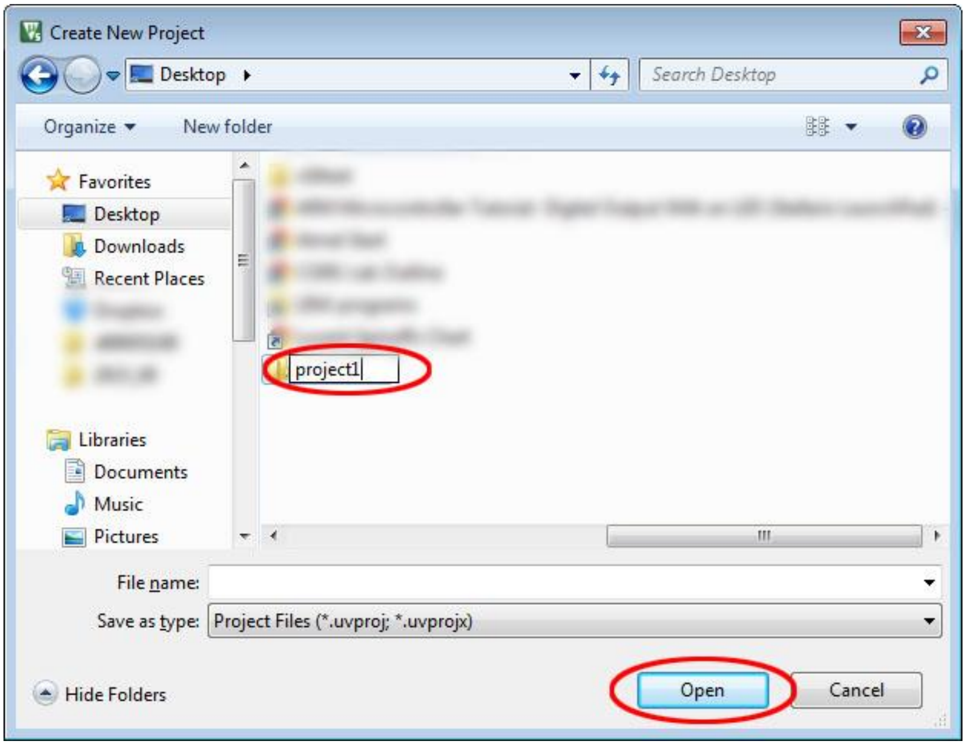
2. From the menu, select Project > New uVision Project…



3. In the New uVision Project window, browse to the **Desktop**.
4. If you did not create a folder for the project before launching uVision, you may create a folder using the **New folder** menu item in the **Create New Project** window.



5. Enter a name for the project folder. We will call it **project1** and click **Open**. This will bring us into the newly created folder **project1**.



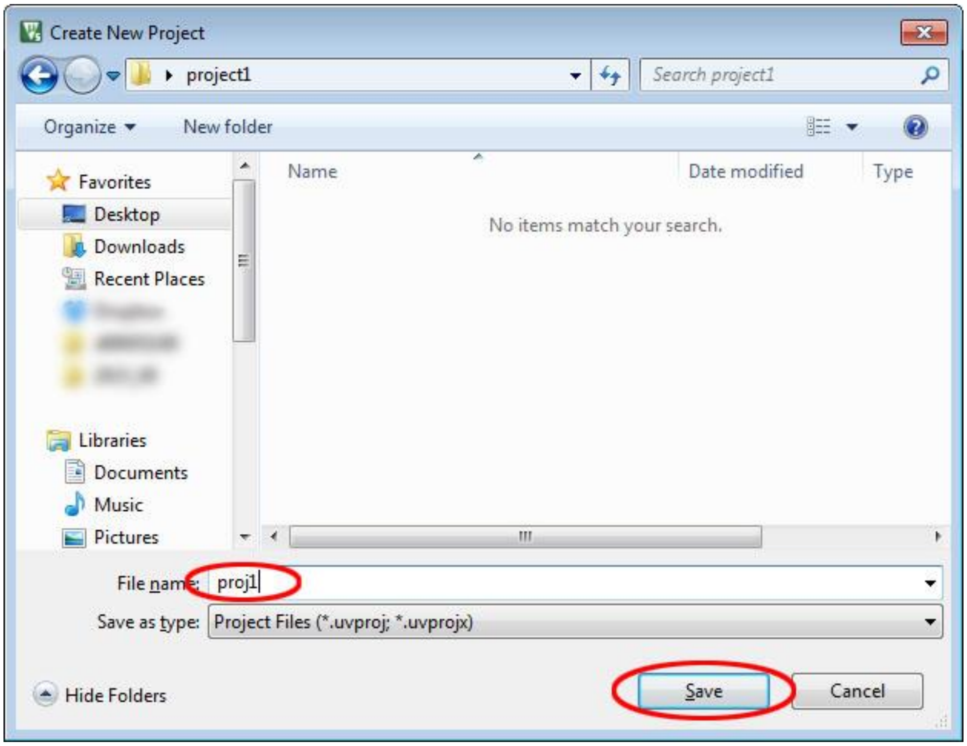
Create New Project

project1

File name:

Save as type: Project Files (*.uvproj; *.uvprojx)

6. Enter a name for the project file. We will call it **proj1** and click **Save**.



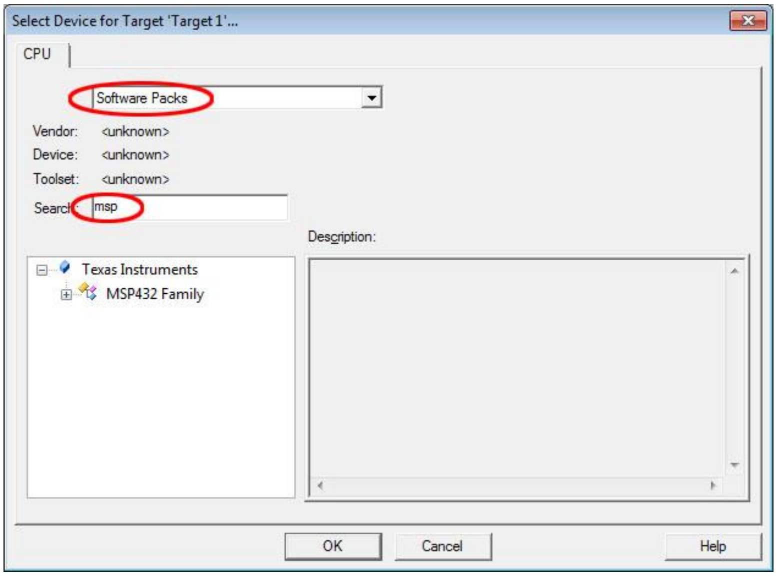
Create New Project

File name: proj1

Save as type: Project Files (*.uvproj; *.uvprojx)

No items match your search.

7. The **Select Device for Target 'Target 1'…** window will pop up. A list of devices supported will show up in the window at the lower left corner. These are the devices with the Device Family Software Pack installed. You may start drilling down the selections to find the device or type a substring of the device code in the Search window and the display in the window will narrow down to only the ones with match.



Select Device for Target 'Target 1'...

Software Packs

Vendor: <unknown>

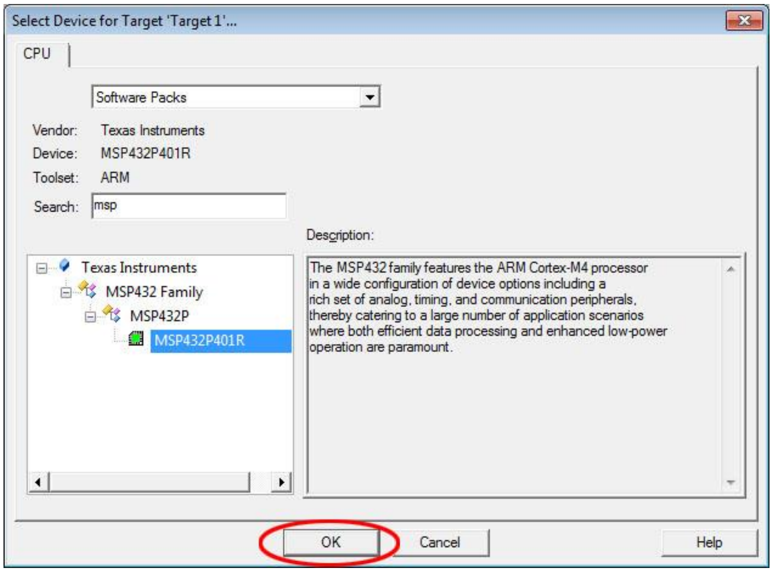
Device: <unknown>

Toolset: <unknown>

Search: msp

Description:

8. We will be using Texas Instruments MSP432P401R. To select this device, click on the **+** sign to expand the selections until you find the device. Click to highlight the device then click **OK** button.



Select Device for Target 'Target 1'...

Software Packs

Vendor: Texas Instruments

Device: MSP432P401R

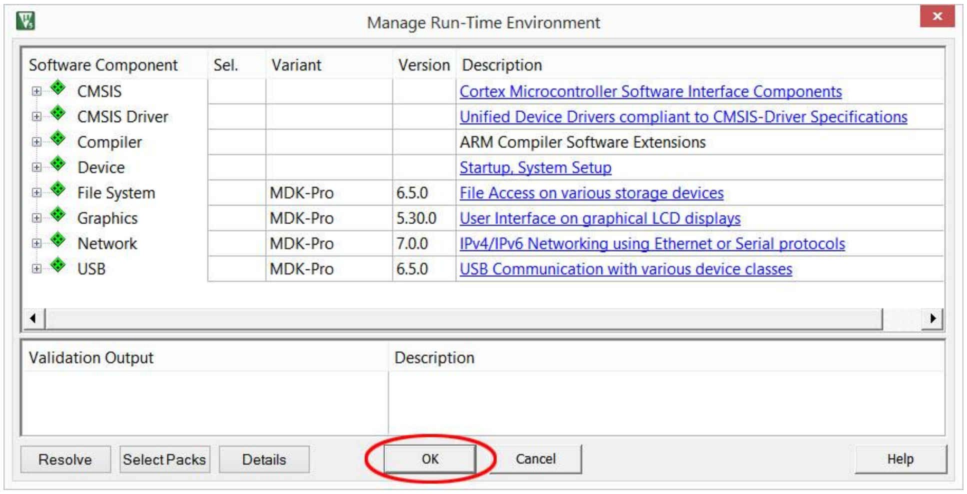
Toolset: ARM

Search: msp

Description:

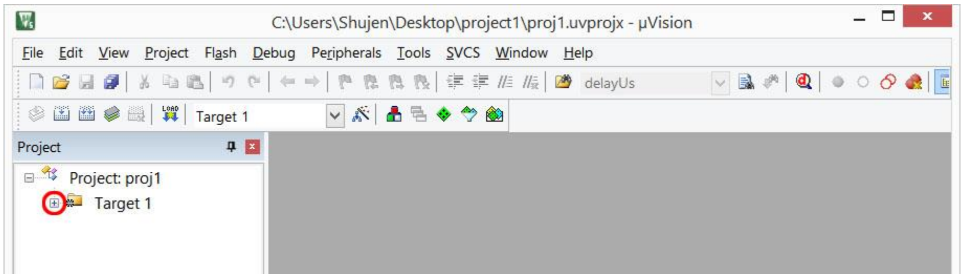
The MSP432 family features the ARM Cortex-M4 processor in a wide configuration of device options including a rich set of analog, timing, and communication peripherals, thereby catering to a large number of application scenarios where both efficient data processing and enhanced low-power operation are paramount.

9. A dialog box to **Manage Run-Time Environment** pops up. Click **OK** to close it.

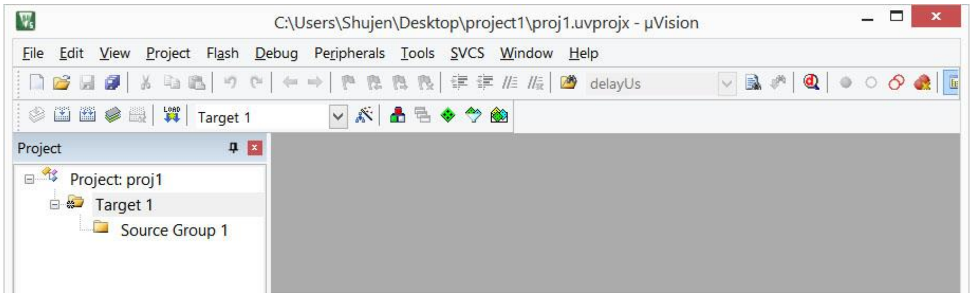


Software Component	Sel.	Variant	Version	Description
CMSIS				Cortex Microcontroller Software Interface Components
CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications
Compiler				ARM Compiler Software Extensions
Device				Startup, System Setup
File System		MDK-Pro	6.5.0	File Access on various storage devices
Graphics		MDK-Pro	5.30.0	User Interface on graphical LCD displays
Network		MDK-Pro	7.0.0	IPv4/IPv6 Networking using Ethernet or Serial protocols
USB		MDK-Pro	6.5.0	USB Communication with various device classes

10. In the Project window, a target was created with the default name **Target 1**. Click on the **+** sign to the left of Target 1 to expand the folder.

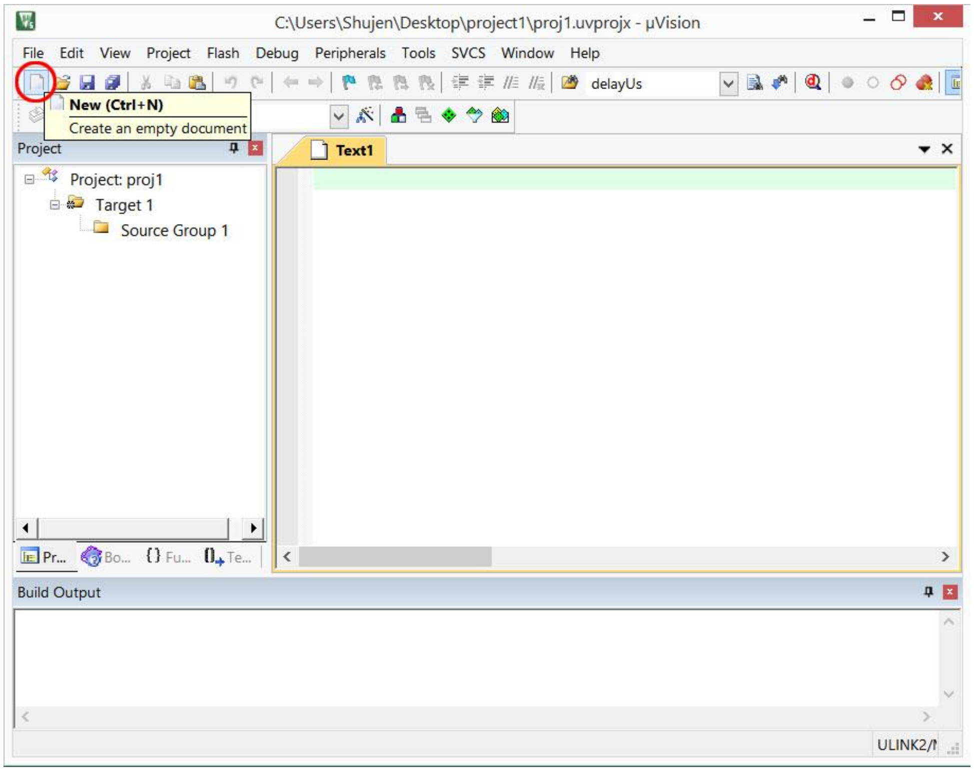

C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Target 1
Project
Project: proj1
Target 1

11. A default folder for source code files was created with the name **Source Group 1**.

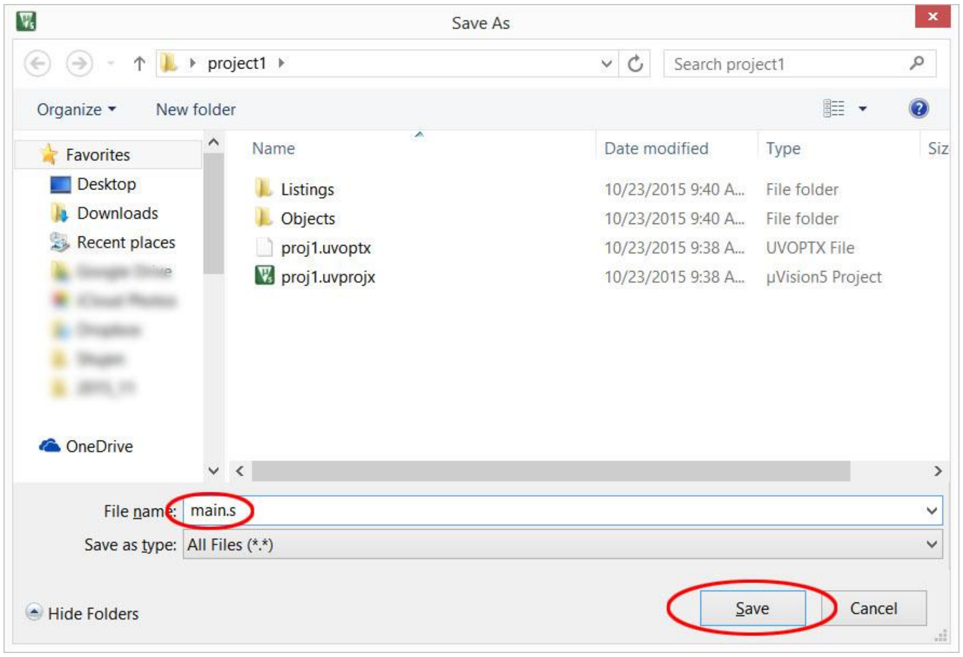

C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Target 1
Project
Project: proj1
Target 1
Source Group 1

Add a Source File to the Project

12. Click the **New** button to add a new text file to the display with the default name **Text1**.



13. 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.s** and click **Save**.

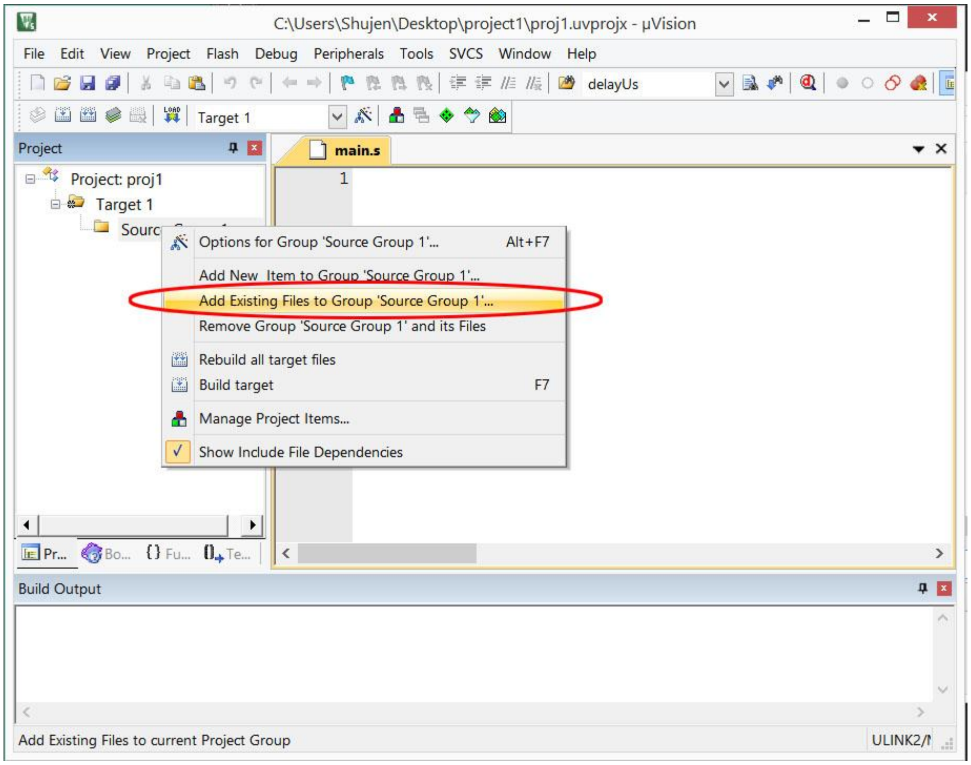


Name	Date modified	Type	Siz
Listings	10/23/2015 9:40 A...	File folder	
Objects	10/23/2015 9:40 A...	File folder	
proj1.uvoptx	10/23/2015 9:38 A...	UVOPTX File	
proj1.uvprojx	10/23/2015 9:38 A...	µVision5 Project	

File name: main.s

Save as type: All Files (*.*)

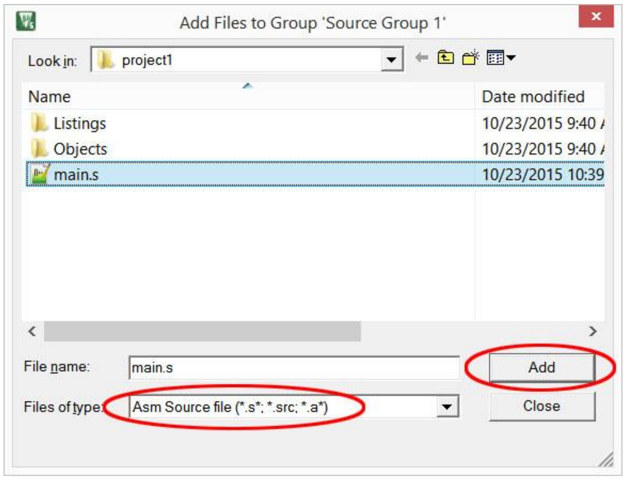
14. You will notice the file name in the tab changed to main.c
15. The new file needs be added to the project. Right click on the folder **Source Group 1** in the Project window and select **Add Existing Files to Group 'Source Group 1'…**



C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Target 1
Project
Project: proj1
Target 1
Sourc
main.s
1
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
Rebuild all target files
Build target F7
Manage Project Items...
Show Include File Dependencies
Pr... Bo... Fu... Te...
Build Output
ULINK2/I

Add Existing Files to current Project Group

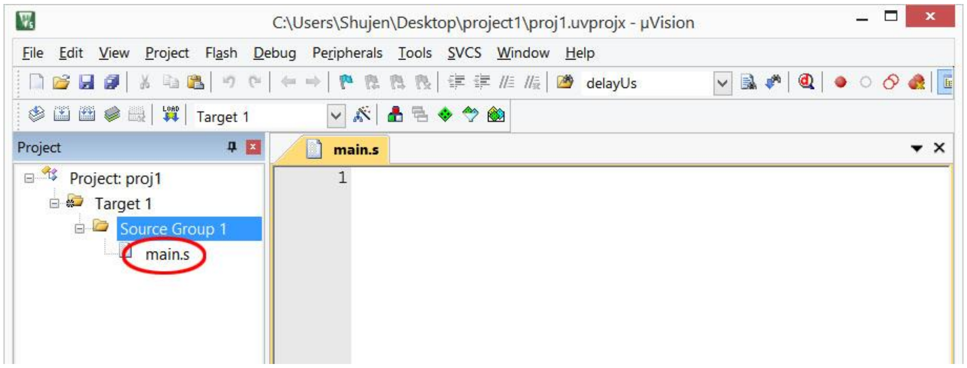
16. In the dialog box, browse to the project folder if it is not already there. Change Files of type: to Asm Source file, Click select **main.s** then click **Add**.



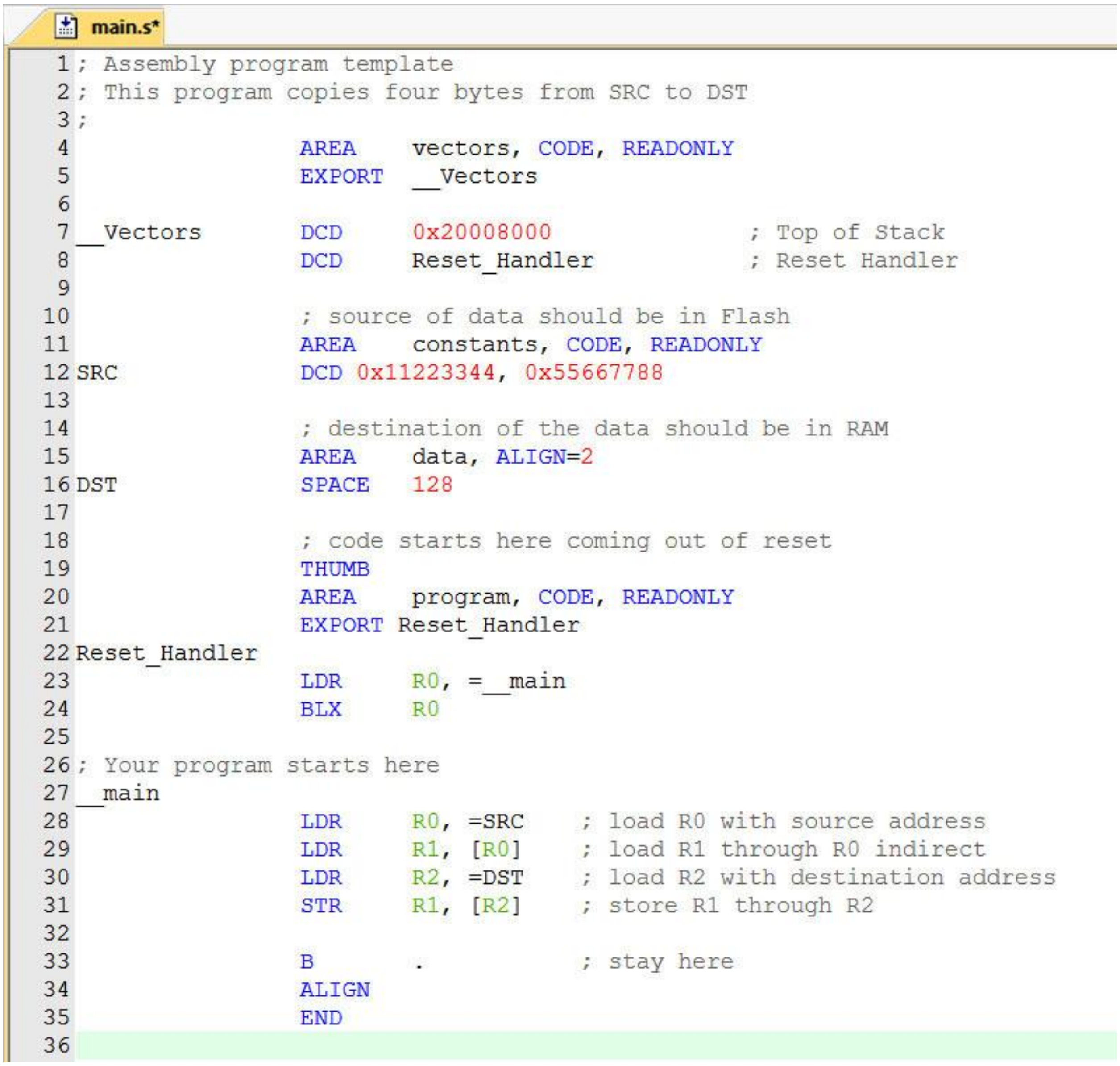
File name: main.s

Files of type: Asm Source file (*.s*; *.src; *.a*)

17. Click Close to close the dialog box. The file should appear in the project window under Source Group 1 folder.

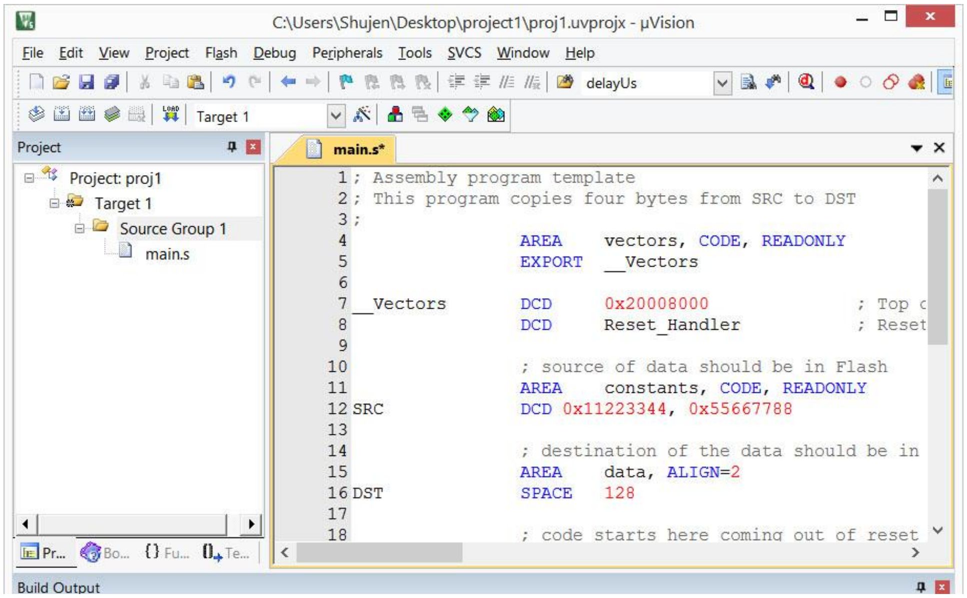

C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Target 1
Project
main.s
Project: proj1
Target 1
Source Group 1
main.s
1

18. Copy the following code into the main.s editor window.



```
; Assembly program template
; This program copies four bytes from SRC to DST
;
                AREA    vectors, CODE, READONLY
                EXPORT  __Vectors

__Vectors       DCD     0x20008000              ; Top of Stack
                DCD     Reset_Handler           ; Reset Handler

                ; source of data should be in Flash
                AREA    constants, CODE, READONLY
SRC             DCD 0x11223344, 0x55667788

                ; destination of the data should be in RAM
                AREA    data, ALIGN=2
DST             SPACE   128

                ; code starts here coming out of reset
                THUMB
                AREA    program, CODE, READONLY
                EXPORT Reset_Handler
Reset_Handler
                LDR     R0, =__main
                BLX     R0

; Your program starts here
__main
                LDR     R0, =SRC    ; load R0 with source address
                LDR     R1, [R0]    ; load R1 through R0 indirect
                LDR     R2, =DST    ; load R2 with destination address
                STR     R1, [R2]    ; store R1 through R2

                B       .           ; stay here
                ALIGN
                END

```

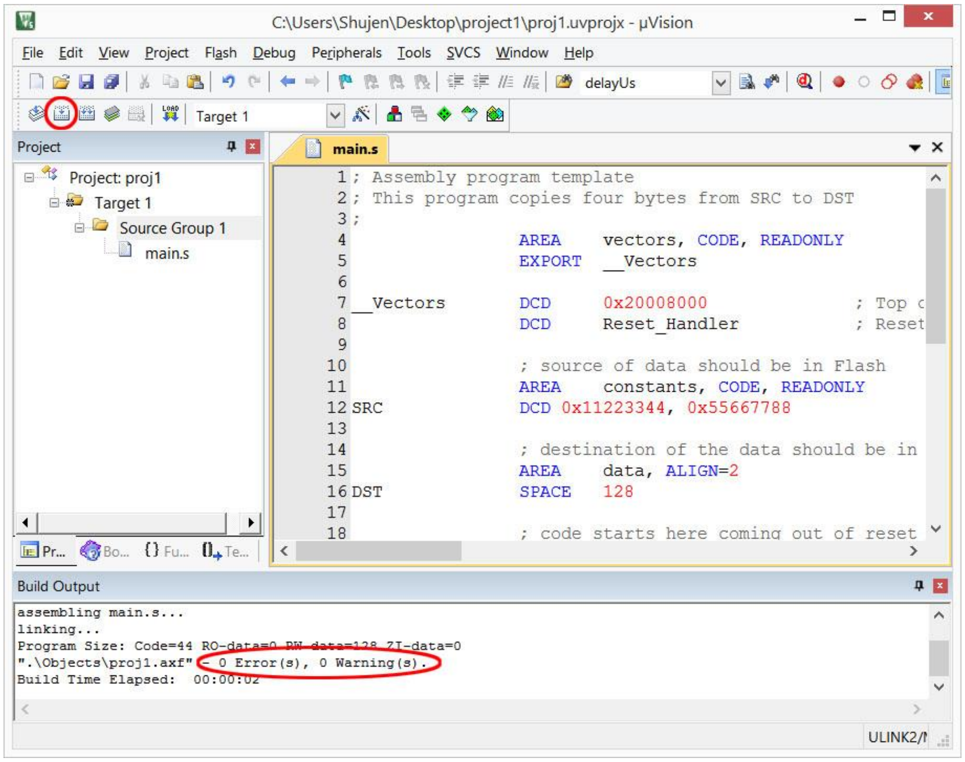
19. The file name in the tab will have an '*' next to it. It symbolizes that the file has been changed without saving. You may click the save button to save the file or proceed to build the project. The file is automatically saved before the build.


C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Target 1
Project
Project: proj1
Target 1
Source Group 1
main.s
main.s*
Pr... Bo... {} Fu... 0+ Te...
Build Output

```
; Assembly program template
; This program copies four bytes from SRC to DST
;
                AREA    vectors, CODE, READONLY
                EXPORT  __Vectors

__Vectors       DCD     0x20008000              ; Top c
                DCD     Reset_Handler           ; Reset

                ; source of data should be in Flash
                AREA    constants, CODE, READONLY
SRC             DCD 0x11223344, 0x55667788

                ; destination of the data should be in
                AREA    data, ALIGN=2
DST             SPACE   128

                ; code starts here coming out of reset
```

Build the Project and Test with Simulator

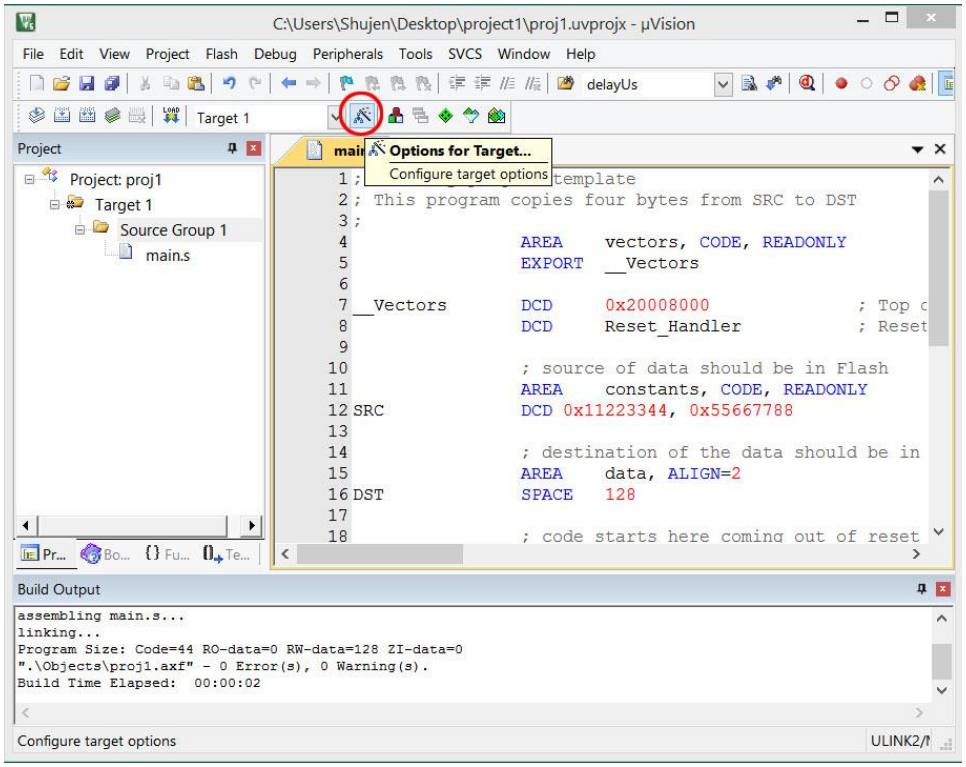
20. Click on the **Build** button and wait for the build to complete. Make sure there are no error messages or warning messages.


C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Target 1
Project
Project: proj1
Target 1
Source Group 1
main.s
main.s
Pr... Bo... {} Fu... 0+ Te...
Build Output
ULINK2/M

```
; Assembly program template
; This program copies four bytes from SRC to DST
;
                AREA    vectors, CODE, READONLY
                EXPORT  __Vectors

__Vectors       DCD     0x20008000              ; Top c
                DCD     Reset_Handler           ; Reset

                ; source of data should be in Flash
                AREA    constants, CODE, READONLY
SRC             DCD 0x11223344, 0x55667788

                ; destination of the data should be in
                AREA    data, ALIGN=2
DST             SPACE   128

                ; code starts here coming out of reset
```

```
assembling main.s...
linking...
Program Size: Code=44 RO-data=0 RW-data=128 ZI-data=0
".\Objects\proj1.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:02
```

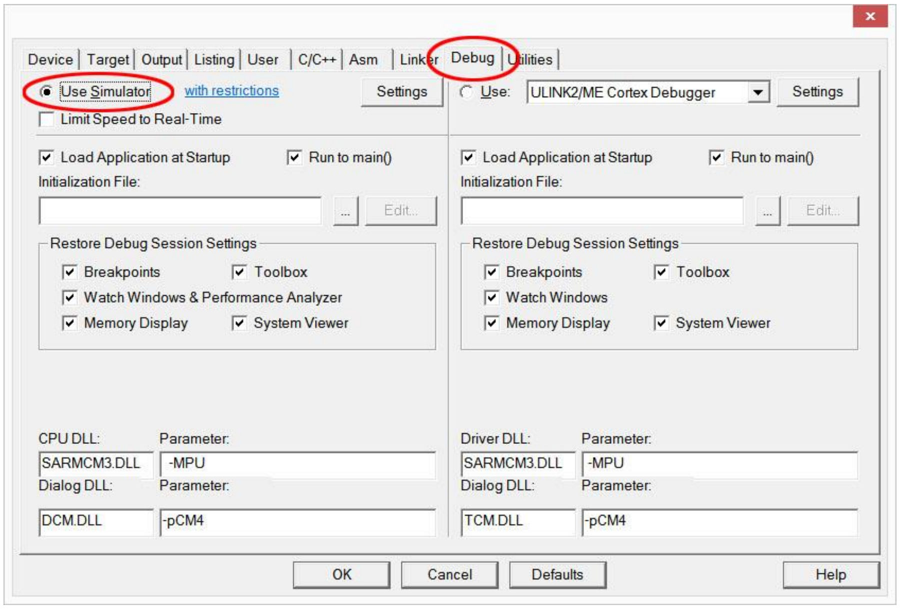
21. Now we are ready to test the program with the simulator. Click on the **Options for Target** button.



```
;              template
; This program copies four bytes from SRC to DST
;
                AREA    vectors, CODE, READONLY
                EXPORT  __Vectors

__Vectors       DCD     0x20008000              ; Top c
                DCD     Reset_Handler           ; Reset

                ; source of data should be in Flash
                AREA    constants, CODE, READONLY
SRC             DCD 0x11223344, 0x55667788

                ; destination of the data should be in
                AREA    data, ALIGN=2
DST             SPACE   128

                ; code starts here coming out of reset
```

```
assembling main.s...
linking...
Program Size: Code=44 RO-data=0 RW-data=128 ZI-data=0
".\Objects\proj1.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:02
```

22. Select **Debug** tab and click the radio button and select **Use Simulator** on the left to test the code with the simulator.


Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities
with restrictions
Settings
Settings
Edit...
Edit...
OK
Cancel
Defaults
Help

● 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 ☑ System Viewer

CPU DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: DCM.DLL Parameter: -pCM4

○ Use: ULINK2/ME Cortex Debugger

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows

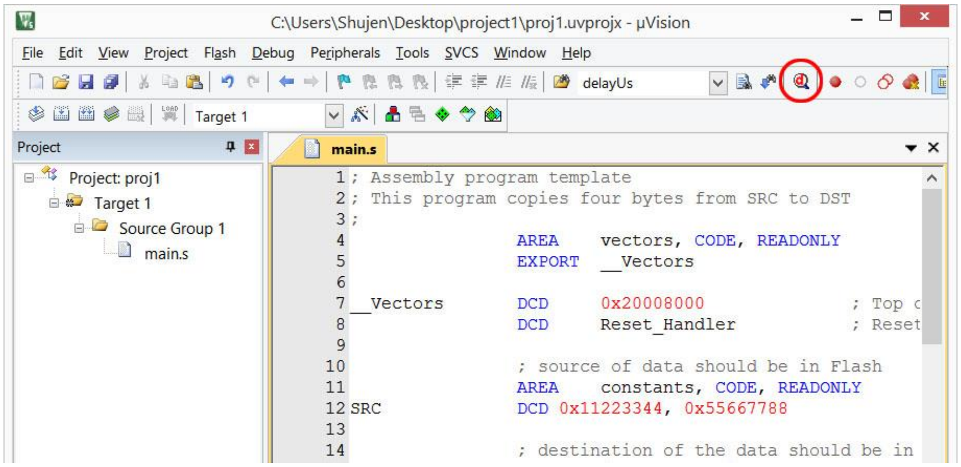
☑ Memory Display ☑ System Viewer

Driver DLL: SARMCM3.DLL Parameter: -MPU

Dialog DLL: TCM.DLL Parameter: -pCM4

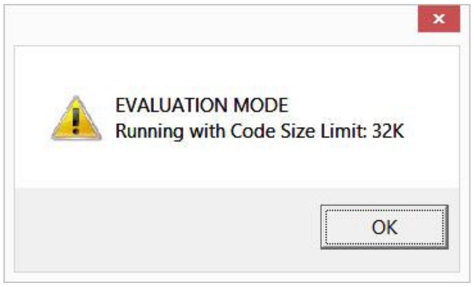
23. Click **OK** button to close the Options window

24. To simulate the program, you need to go into the debugger. Click the **Debug** button on the right.


C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Target 1
Project
Project: proj1
Target 1
Source Group 1
main.s
main.s

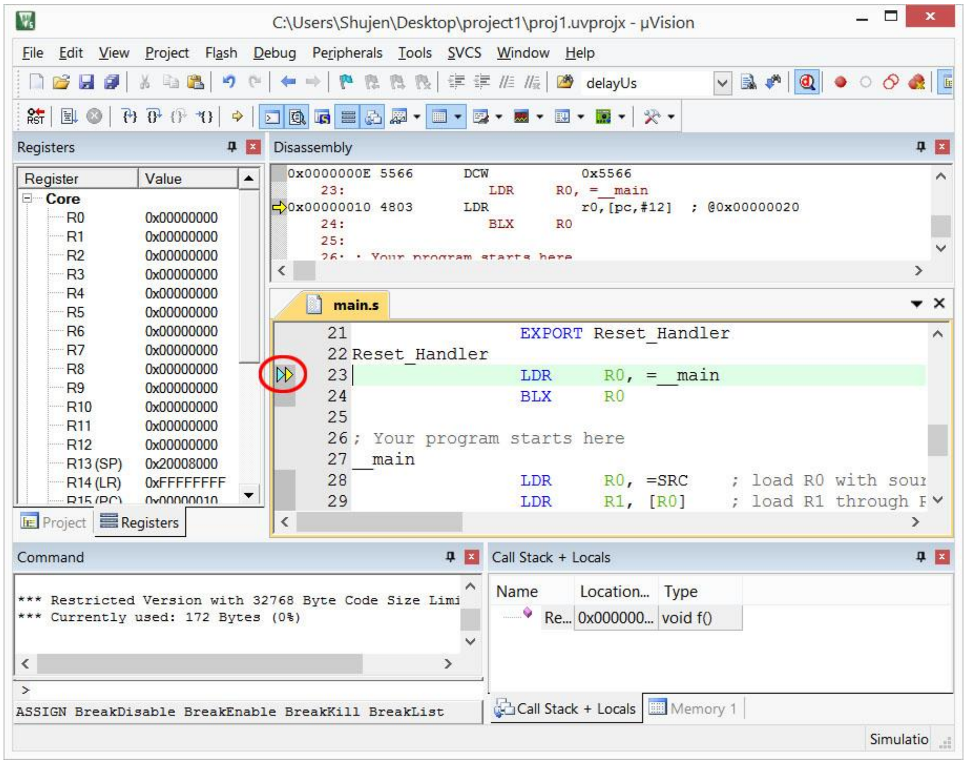
```
; Assembly program template
; This program copies four bytes from SRC to DST
;
                AREA    vectors, CODE, READONLY
                EXPORT  __Vectors

__Vectors       DCD     0x20008000              ; Top c
                DCD     Reset_Handler           ; Reset

                ; source of data should be in Flash
                AREA    constants, CODE, READONLY
SRC             DCD 0x11223344, 0x55667788

                ; destination of the data should be in
```

25. If you are running MDK-ARM Lite Edition, a message window will pop up to warn the code size limitation. Click **OK** to proceed.



EVALUATION MODE
Running with Code Size Limit: 32K

26. When entering the debugger, the IDE changes the perspective to the debug view. The program counter (which is indicated by a yellow triangle) is set at the beginning of the Reset_Handler.


C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Registers
Disassembly
main.s
Project
Registers
Command
Call Stack + Locals
Memory 1
Simulatio

Register	Value
Core	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20008000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000010

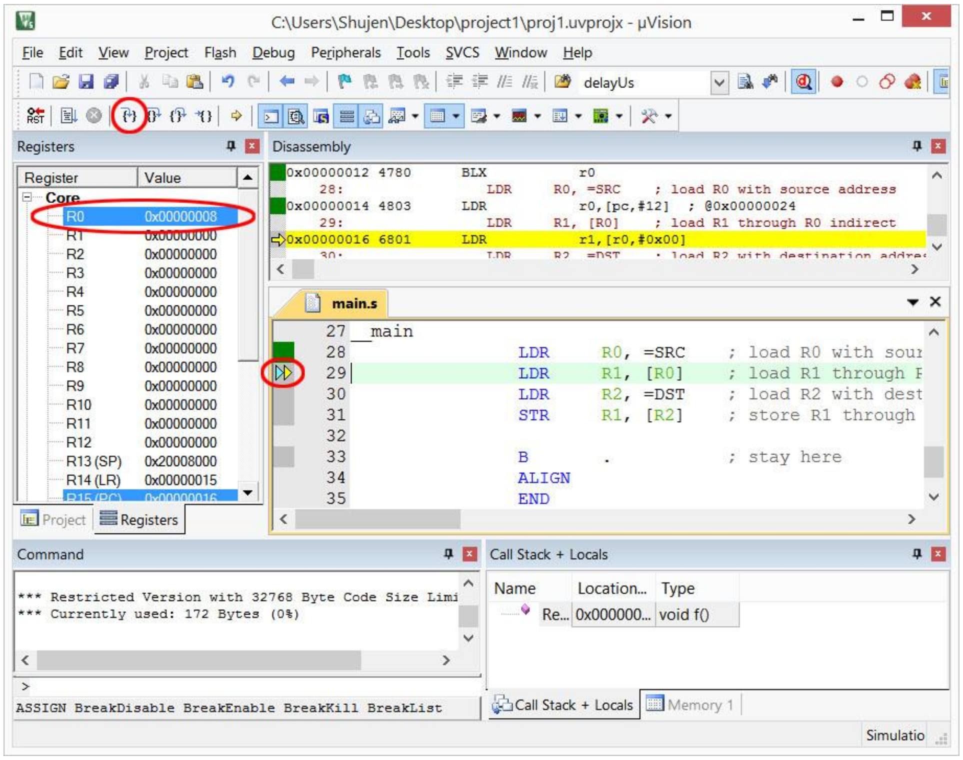
```
0x0000000E 5566      DCW        0x5566
    23:                 LDR     R0, =__main
0x00000010 4803      LDR            r0,[pc,#12]  ; @0x00000020
    24:                 BLX     R0
    25:
    26: ; Your program starts here
```

```
                EXPORT Reset_Handler
Reset_Handler
                LDR     R0, =__main
                BLX     R0

; Your program starts here
__main
                LDR     R0, =SRC     ; load R0 with sour
                LDR     R1, [R0]     ; load R1 through R
```

```
*** Restricted Version with 32768 Byte Code Size Limi
*** Currently used: 172 Bytes (0%)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList
```

Name	Location...	Type
Re...	0x000000...	void f()

27. Click Step button until the execution passes instruction “LDR R0, =SRC” (The pointer is pointing to the address of the PC, which is the next instruction to be executed). This instruction loads R0 with the address of SRC. When the content of the R0 changed, it is highlighted as seen below.


C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Registers
Disassembly
main.s
Project
Registers
Command
Call Stack + Locals
Memory 1
Simulatio

Register	Value
Core	
R0	0x00000008
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20008000
R14 (LR)	0x00000015
R15 (PC)	0x00000016

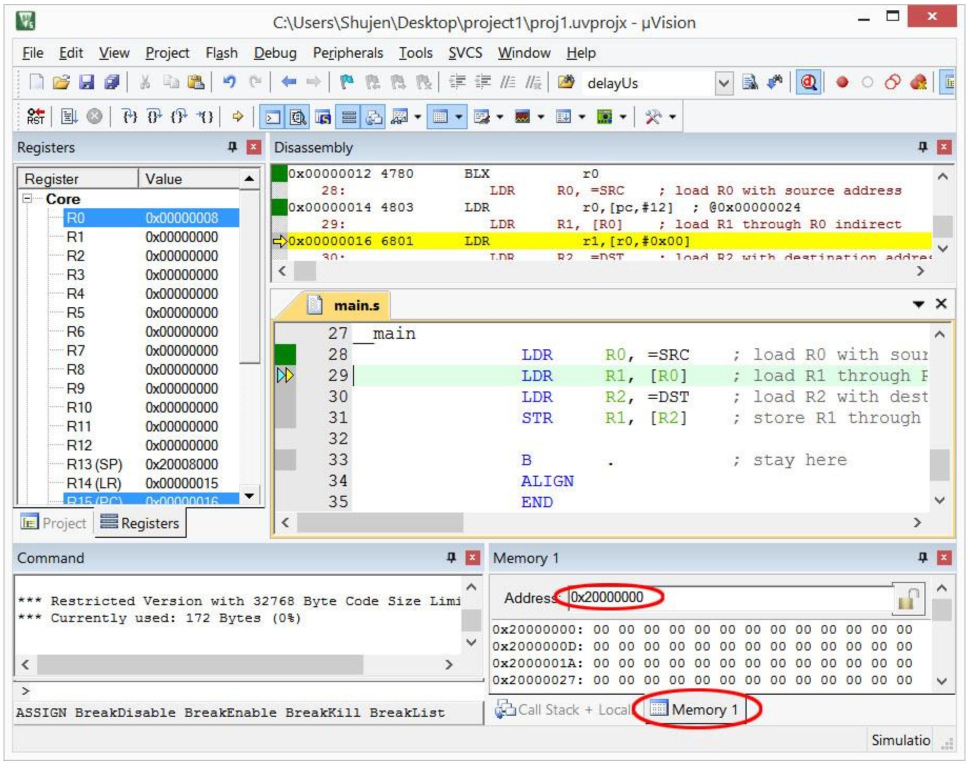
```
0x00000012 4780      BLX            r0
    28:                 LDR     R0, =SRC     ; load R0 with source address
0x00000014 4803      LDR            r0,[pc,#12]  ; @0x00000024
    29:                 LDR     R1, [R0]     ; load R1 through R0 indirect
0x00000016 6801      LDR            r1,[r0,#0x00]
    30:                 LDR     R2, =DST     ; load R2 with destination addre
```

```
__main
                LDR     R0, =SRC     ; load R0 with sour
                LDR     R1, [R0]     ; load R1 through R
                LDR     R2, =DST     ; load R2 with dest
                STR     R1, [R2]     ; store R1 through

                B       .            ; stay here
                ALIGN
                END
```

```
*** Restricted Version with 32768 Byte Code Size Limi
*** Currently used: 172 Bytes (0%)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList
```

Name	Location...	Type
Re...	0x000000...	void f()

28. At the lower right window, click **Memory 1** tab and type 0x20000000 in the **Address** box. You should see that the memory content is all 0. We will see the memory content changes next.


C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Registers
Disassembly
main.s
Project
Registers
Command
Memory 1
Call Stack + Locals
Memory 1
Simulatio

Register	Value
Core	
R0	0x00000008
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20008000
R14 (LR)	0x00000015
R15 (PC)	0x00000016

```
0x00000012 4780      BLX          r0
    28:                 LDR      R0, =SRC     ; load R0 with source address
0x00000014 4803      LDR          r0,[pc,#12]  ; @0x00000024
    29:                 LDR      R1, [R0]     ; load R1 through R0 indirect
0x00000016 6801      LDR          r1,[r0,#0x00]
    30:                 LDR      R2, =DST     ; load R2 with destination addre
```

```
__main
                LDR     R0, =SRC    ; load R0 with sour
                LDR     R1, [R0]    ; load R1 through R
                LDR     R2, =DST    ; load R2 with dest
                STR     R1, [R2]    ; store R1 through

                B       .           ; stay here
                ALIGN
                END
```

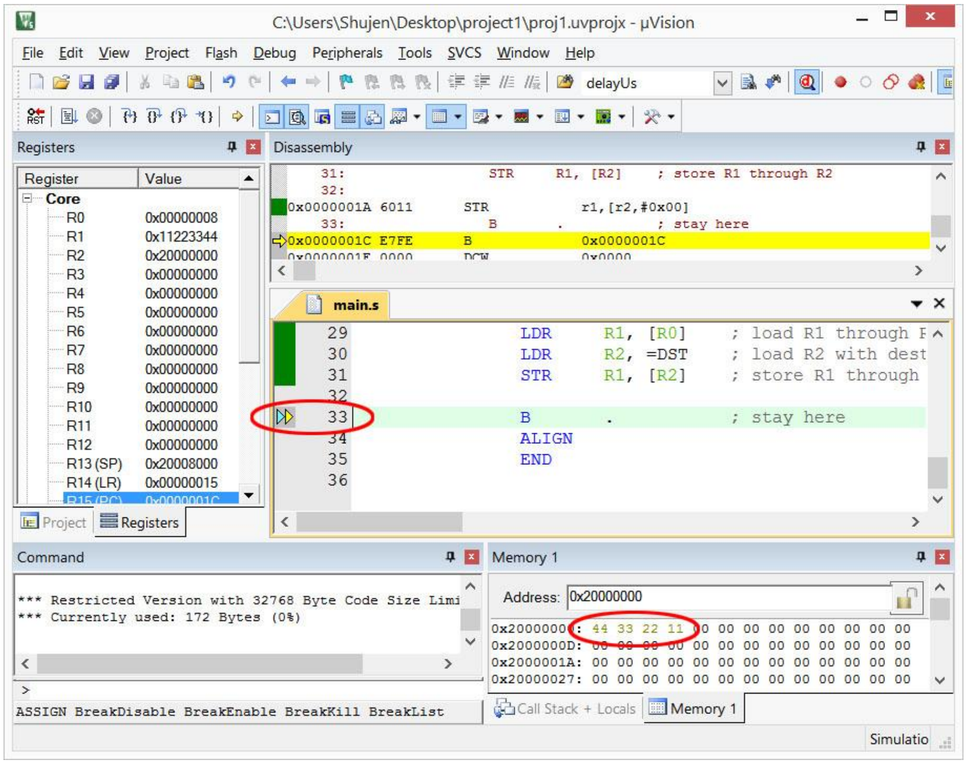
```
*** Restricted Version with 32768 Byte Code Size Limi
*** Currently used: 172 Bytes (0%)
```

```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList
```

Address: 0x20000000

```
0x20000000: 00 00 00 00 00 00 00 00 00 00 00 00 00
0x2000000D: 00 00 00 00 00 00 00 00 00 00 00 00 00
0x2000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000027: 00 00 00 00 00 00 00 00 00 00 00 00 00
```

29. Step until the execution passes the instruction “STR R1, [R2]” instruction. This instruction writes the content of R1 to memory with address in R2. Since R1 has 0x11223344 and R2 has 0x20000000, the value 0x11223344 is written to memory location 0x20000000. You should see the value in the memory window.



C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Registers
Disassembly
Project
Registers
Command
Memory 1
Call Stack + Locals
Memory 1
Simulatio

Register	Value
Core	
R0	0x00000008
R1	0x11223344
R2	0x20000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20008000
R14 (LR)	0x00000015
R15 (PC)	0x0000001C

```
    31:                     STR      R1, [R2]     ; store R1 through R2
    32:
0x0000001A 6011         STR          r1,[r2,#0x00]
    33:                     B        .            ; stay here
0x0000001C E7FE         B            0x0000001C
0x0000001E 0000         DCW          0x0000
```

main.s

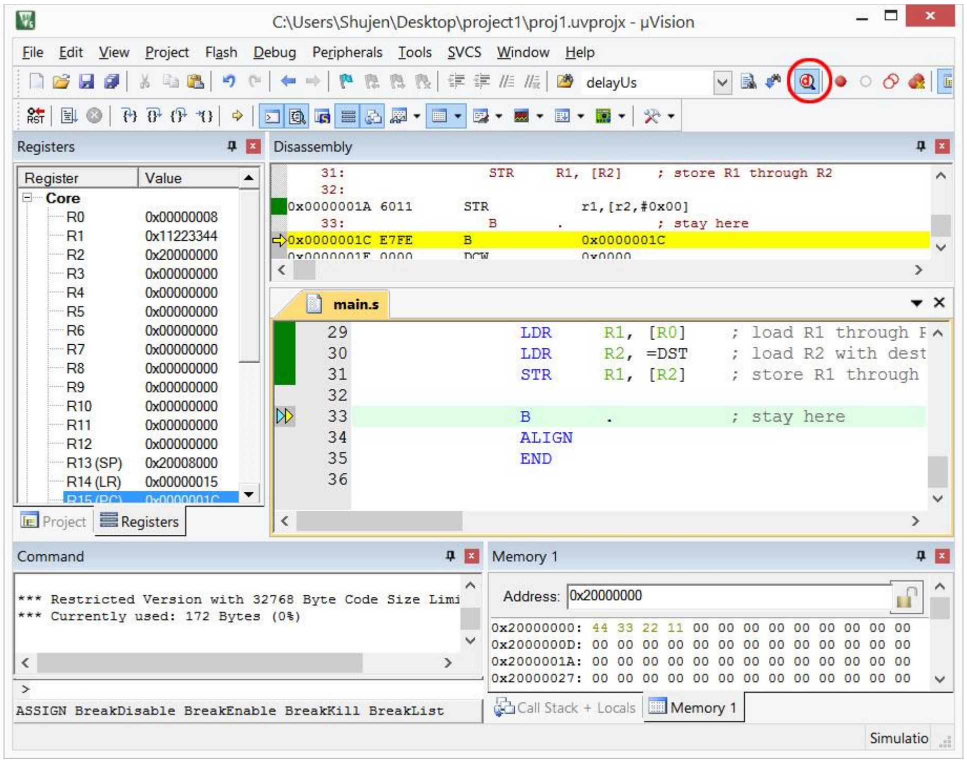
```
                LDR     R1, [R0]     ; load R1 through R
                LDR     R2, =DST     ; load R2 with dest
                STR     R1, [R2]     ; store R1 through

                B       .            ; stay here
                ALIGN
                END
```

```
*** Restricted Version with 32768 Byte Code Size Limi
*** Currently used: 172 Bytes (0%)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList
```

Address: 0x20000000

```
0x20000000: 44 33 22 11 00 00 00 00 00 00 00 00 00
0x2000000D: 00 00 00 00 00 00 00 00 00 00 00 00 00
0x2000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000027: 00 00 00 00 00 00 00 00 00 00 00 00 00
```

30. Exit debugger by clicking on the debug button again.



C:\Users\Shujen\Desktop\project1\proj1.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
delayUs
Registers
Disassembly
Project
Registers
Command
Memory 1
Call Stack + Locals
Memory 1
Simulatio

Register	Value
Core	
R0	0x00000008
R1	0x11223344
R2	0x20000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20008000
R14 (LR)	0x00000015
R15 (PC)	0x0000001C

```
    31:                     STR      R1, [R2]     ; store R1 through R2
    32:
0x0000001A 6011         STR          r1,[r2,#0x00]
    33:                     B        .            ; stay here
0x0000001C E7FE         B            0x0000001C
0x0000001E 0000         DCW          0x0000
```

main.s

```
                LDR     R1, [R0]     ; load R1 through R
                LDR     R2, =DST     ; load R2 with dest
                STR     R1, [R2]     ; store R1 through

                B       .            ; stay here
                ALIGN
                END
```

```
*** Restricted Version with 32768 Byte Code Size Limi
*** Currently used: 172 Bytes (0%)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList
```

Address: 0x20000000

```
0x20000000: 44 33 22 11 00 00 00 00 00 00 00 00 00
0x2000000D: 00 00 00 00 00 00 00 00 00 00 00 00 00
0x2000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000027: 00 00 00 00 00 00 00 00 00 00 00 00 00
```

This concludes the tutorial.