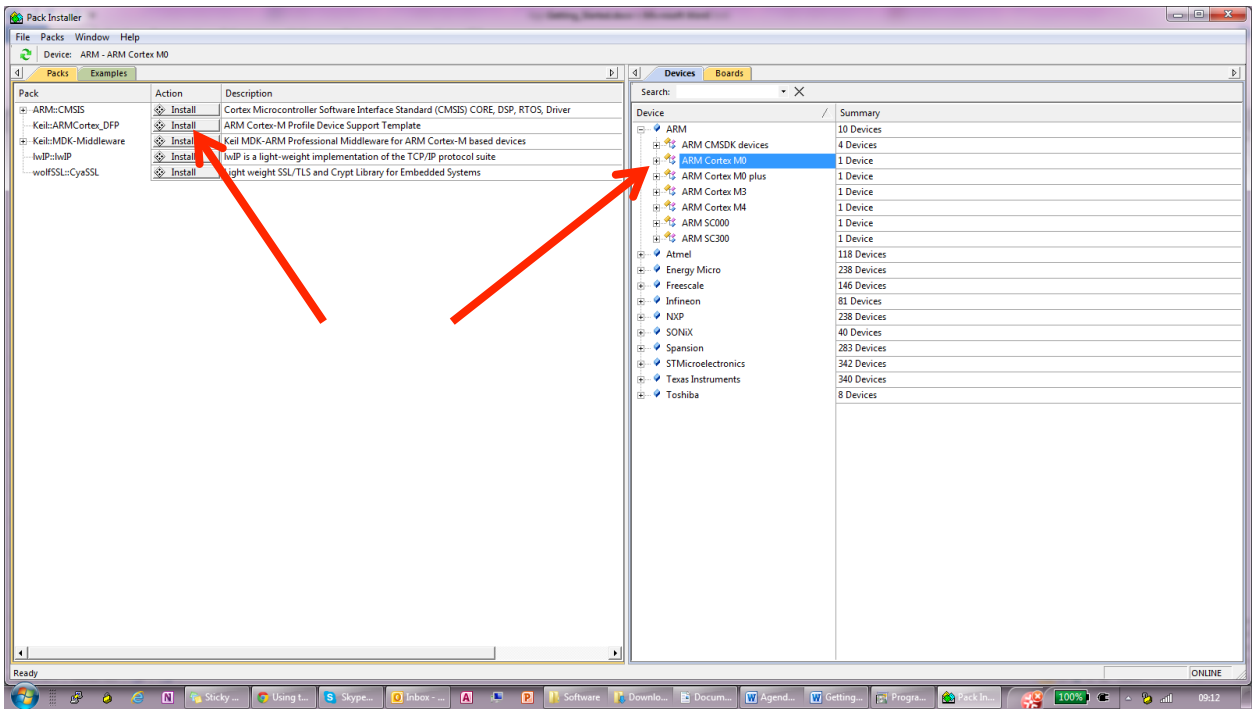
Cortex M0 Assembly Programming

SUMMARY

1. Consider you have a simple Cortex M0 based Micro-controller with the following memory map
    a. Internal SRAM – 1 KB – Starting Address 0x0000 0000
    b. A LED Peripheral – One word – Address 0x5000 0000
2. We will write assembly program to toggle the LEDs and simulate the program
3. We will analyze the resulting binary file and the disassembly file
4. In the Next Section we will see how to design AHB-Lite Compliant LED Peripheral

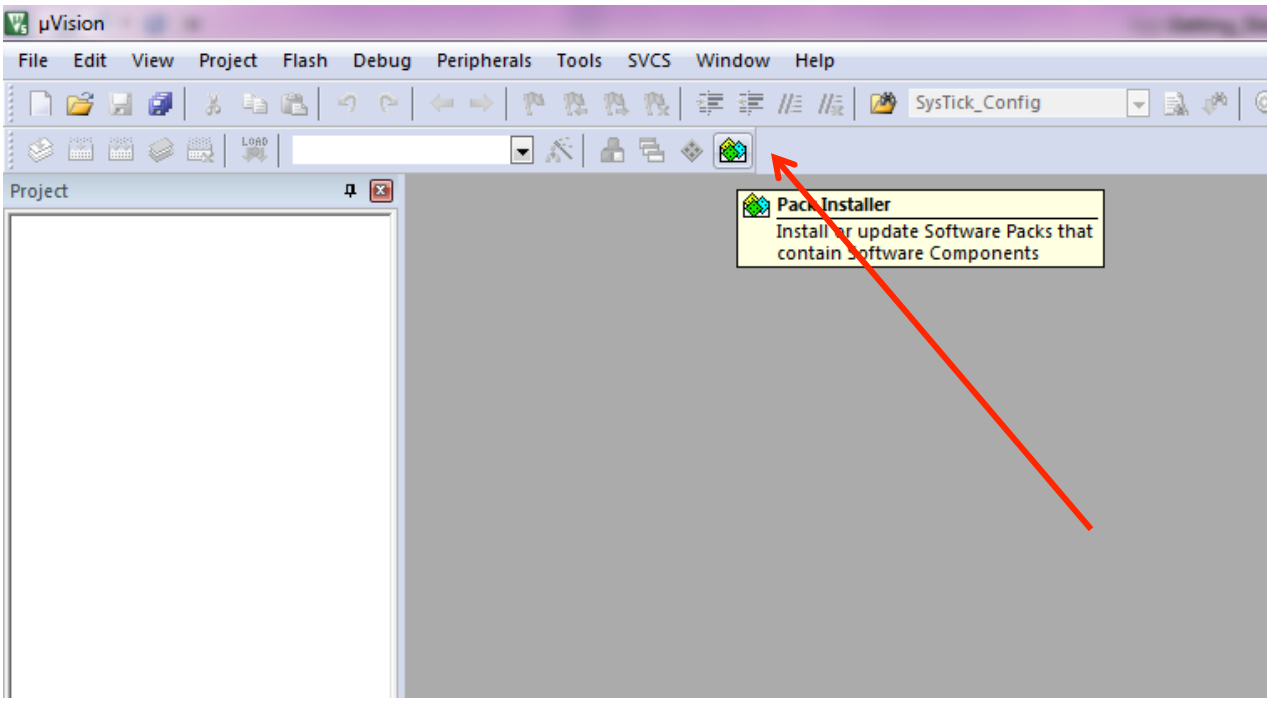
INSTALL KEIL MDK

1. Install KEIL MDK. You can download the software suite from http://www.keil.com/arm/mdk.asp
2. Install ARM Cortex M Profile Device Pack: **KEIL ARMCORTEX:DFP** as shown below,



Copyright © ARM Ltd 2013

3. If you have already installed KEIL MDK-ARM without any packs, you can install the same by clicking pack installer as shown below,

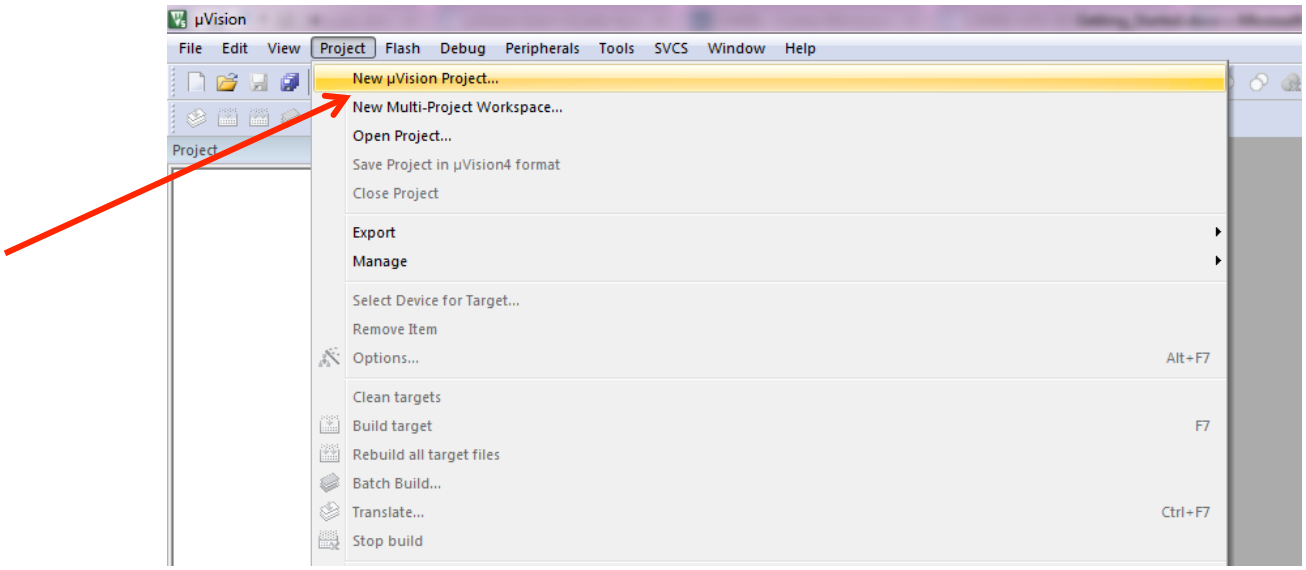


4. Install the license key provided. Steps to install are given in KEIL MDK License Installation Guide.

CREATE A NEW PROJECT

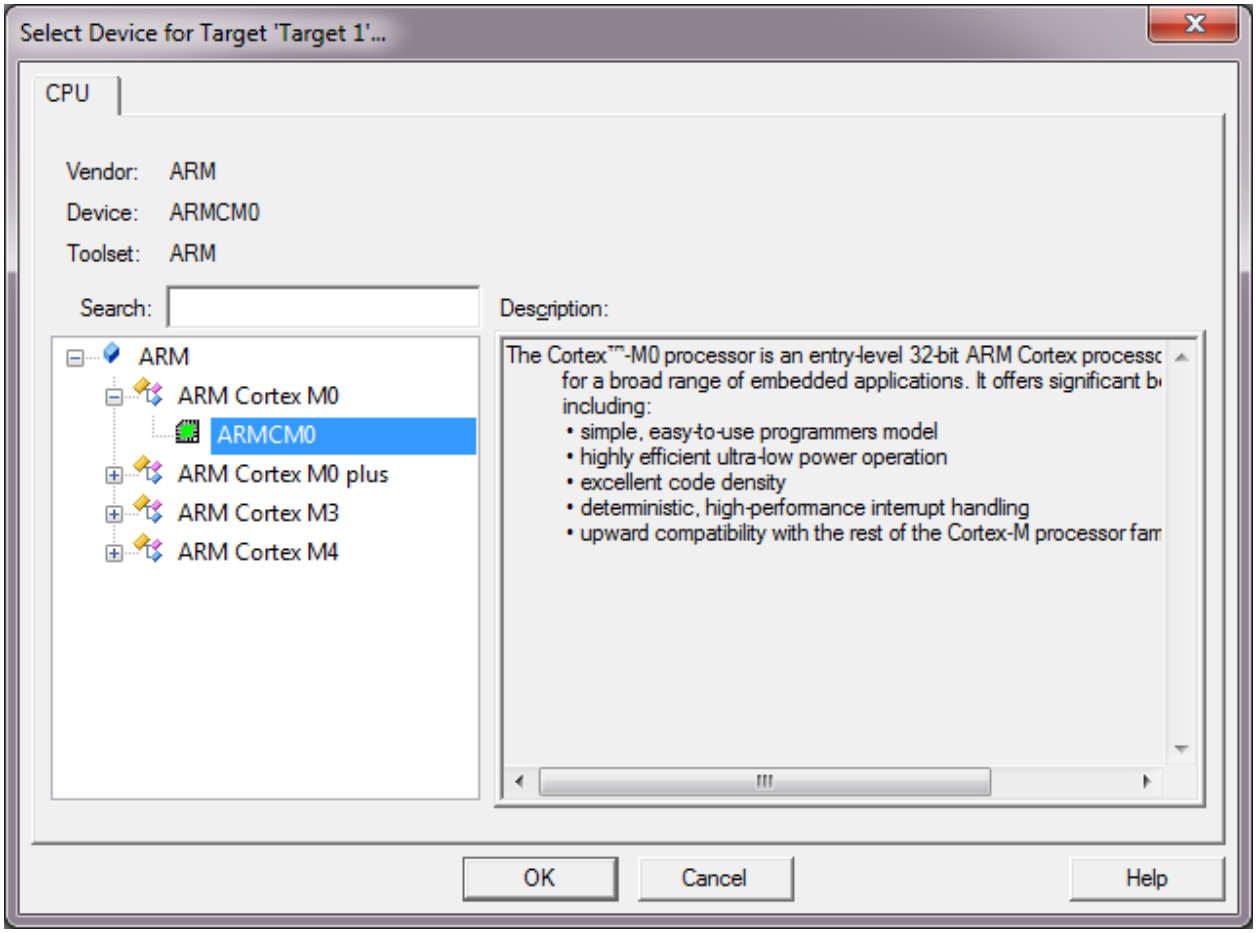
1. Create a new project

Copyright © ARM Ltd 2013



2. Give the project name as “lab”

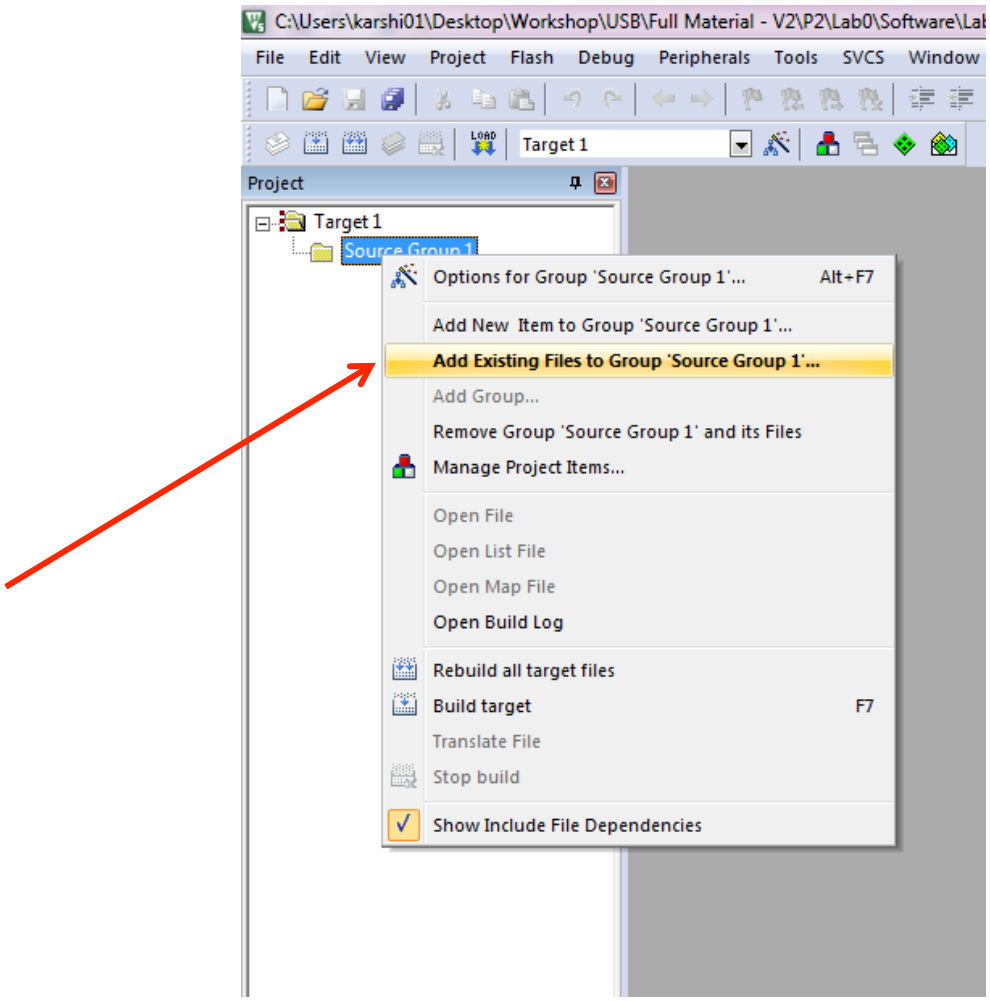
3. In select Device Target, choose ARM → Cortex M0



Copyright © ARM Ltd 2013

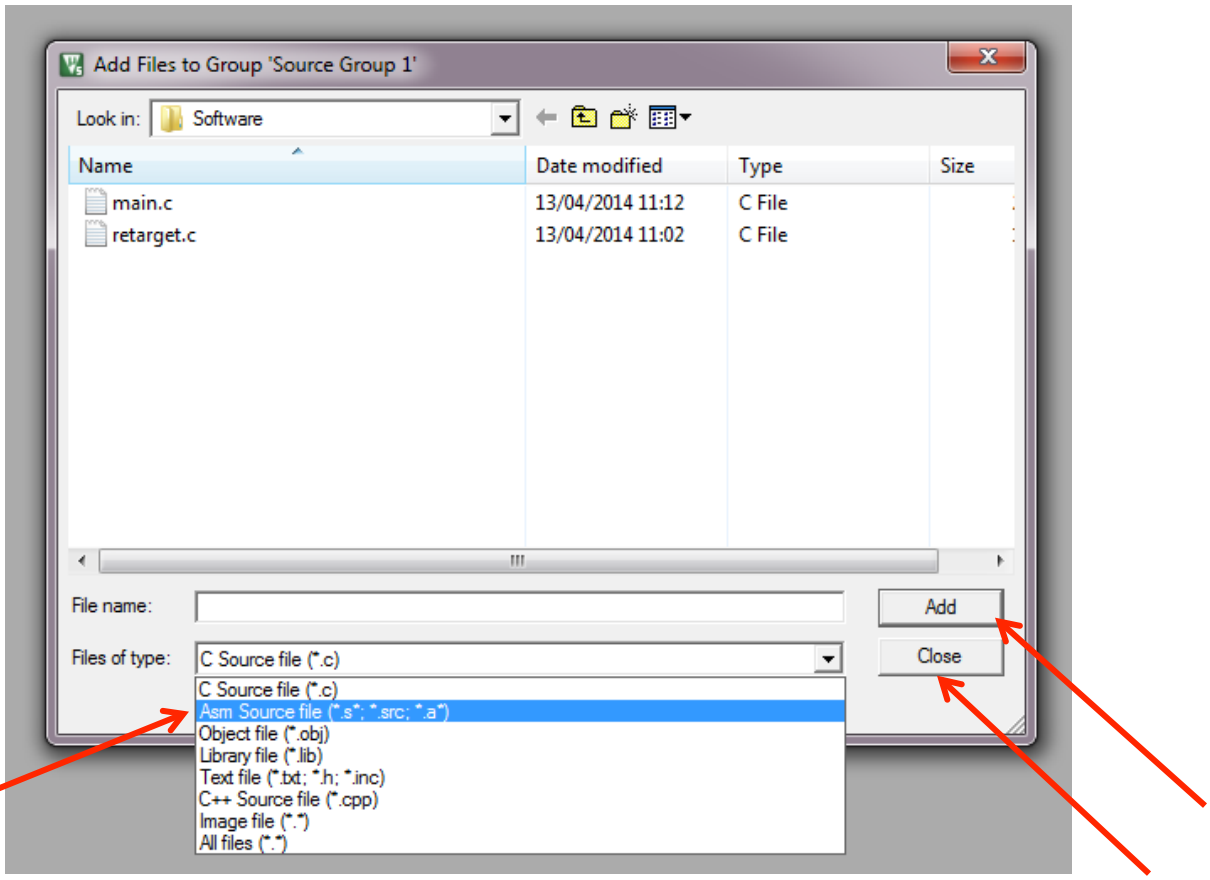
4. Skip the next step “Manage Run Tim environment” by clicking cancel.

5. Add file cm0dsasm.s into the project,



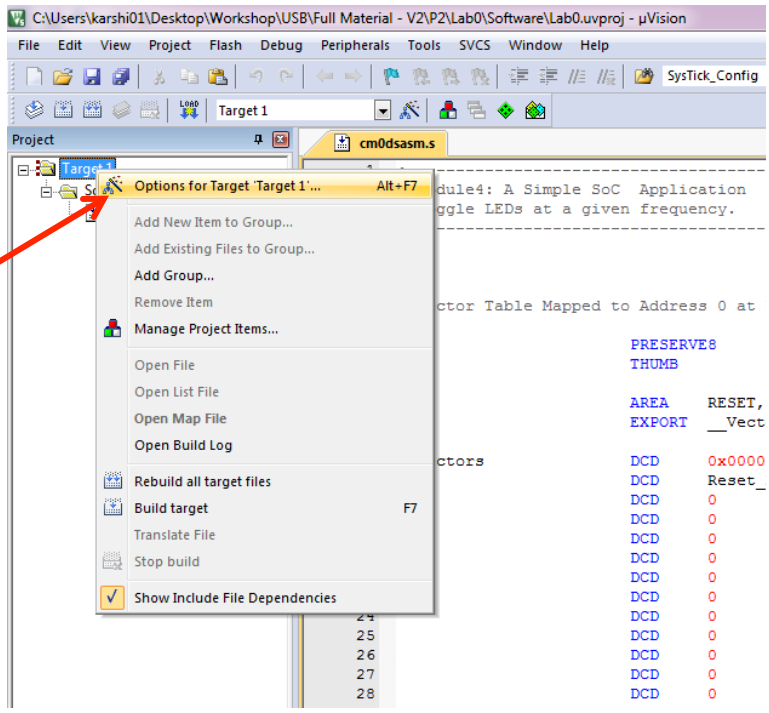
6. Choose “ARM Source Files” in the drop down “File Types” and then select cm0dsasm.s

Copyright © ARM Ltd 2013



7. Once you select the file, click “Add” and then click “Close”
8. The will be added into “Source Group1” under “Target1”
9. Right click on Target1 in project navigator and click “Options for Target1”

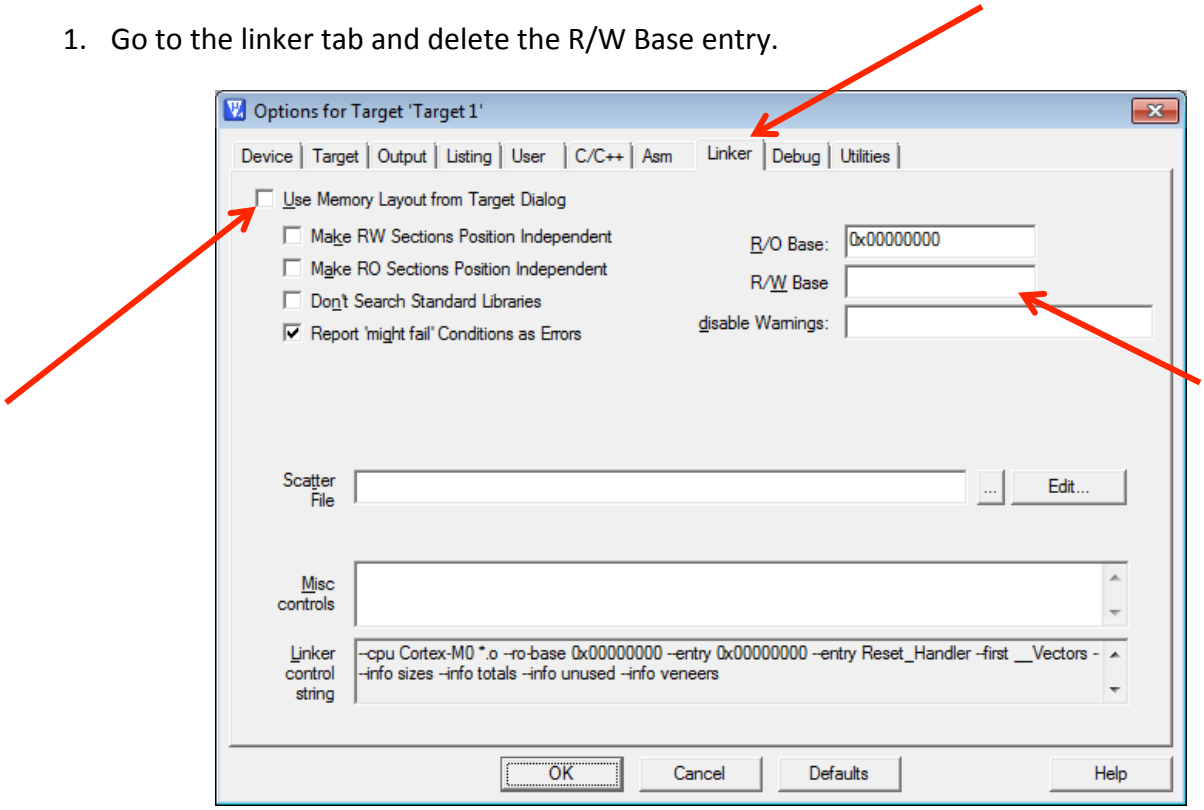
Copyright © ARM Ltd 2013



10. This opens up the configuration window for your project

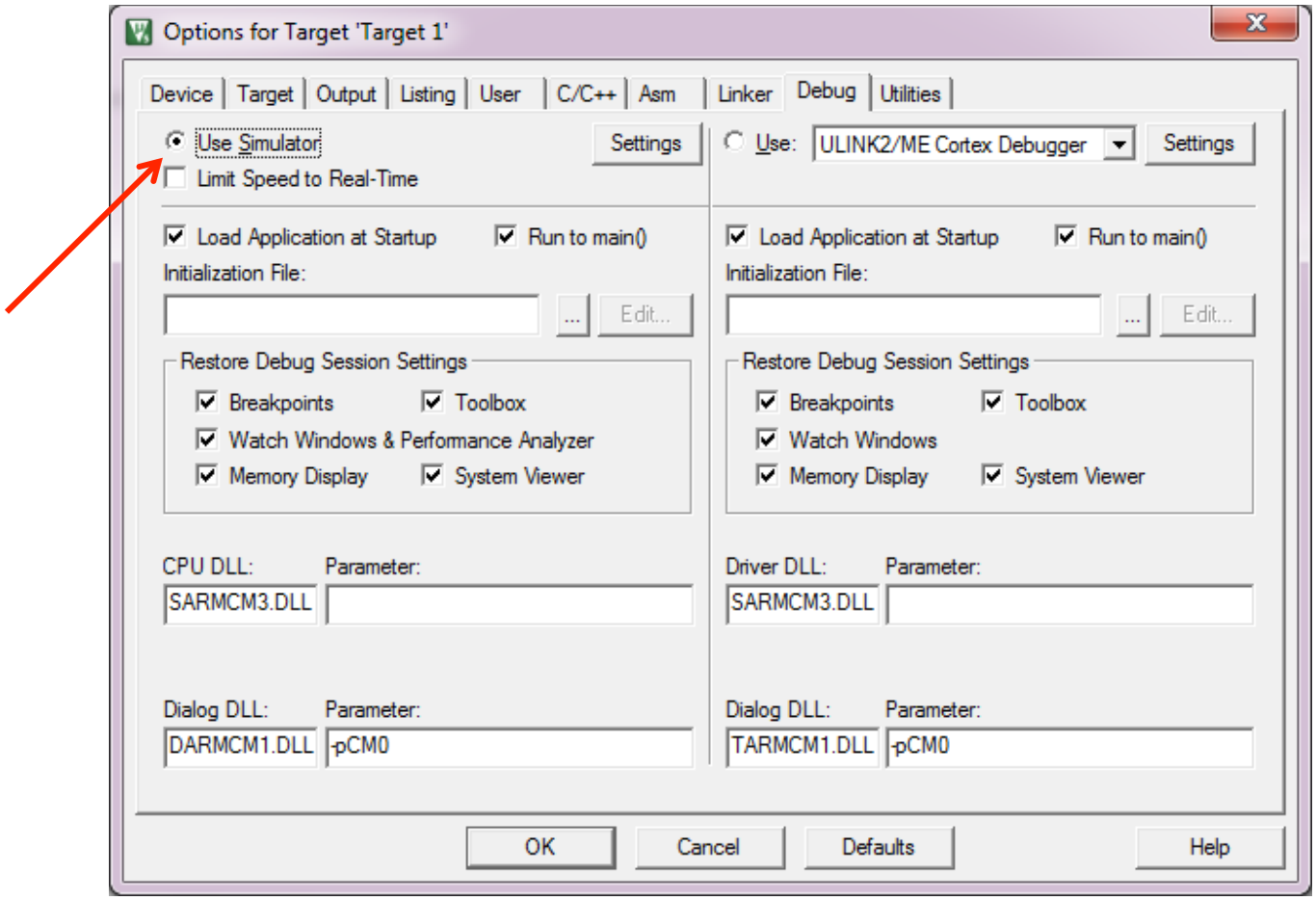
CHANGING THE CONFIGURATION OPTIONS

1. Go to the linker tab and delete the R/W Base entry.



Configuration in Linker Tab

2. Go to the debug tab and change the choice to “USE SIMULATOR”



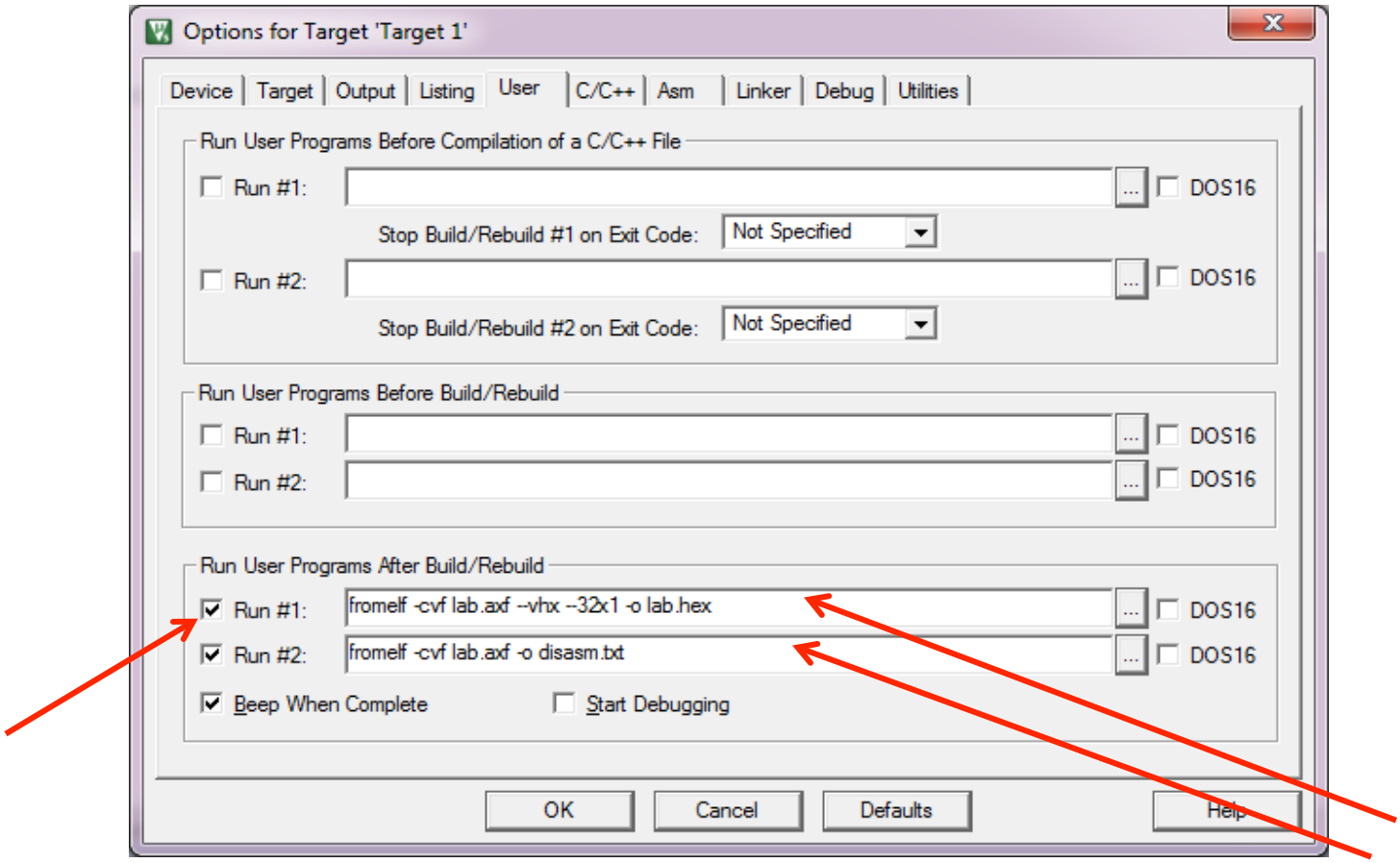
3. Go to USER tab and place these fromelf commands in “RUN THESE COMMANDS AFTER BUILD” section.

    a. **fromelf -cvf lab.axf --vhx --32x1 -o lab.hex**

    b. **fromelf -cvf lab.axf -o disasm.txt**

More information about fromelf utility can be found here:
http://www.keil.com/support/man/docs/armutil/armutil_caccdhia.htm

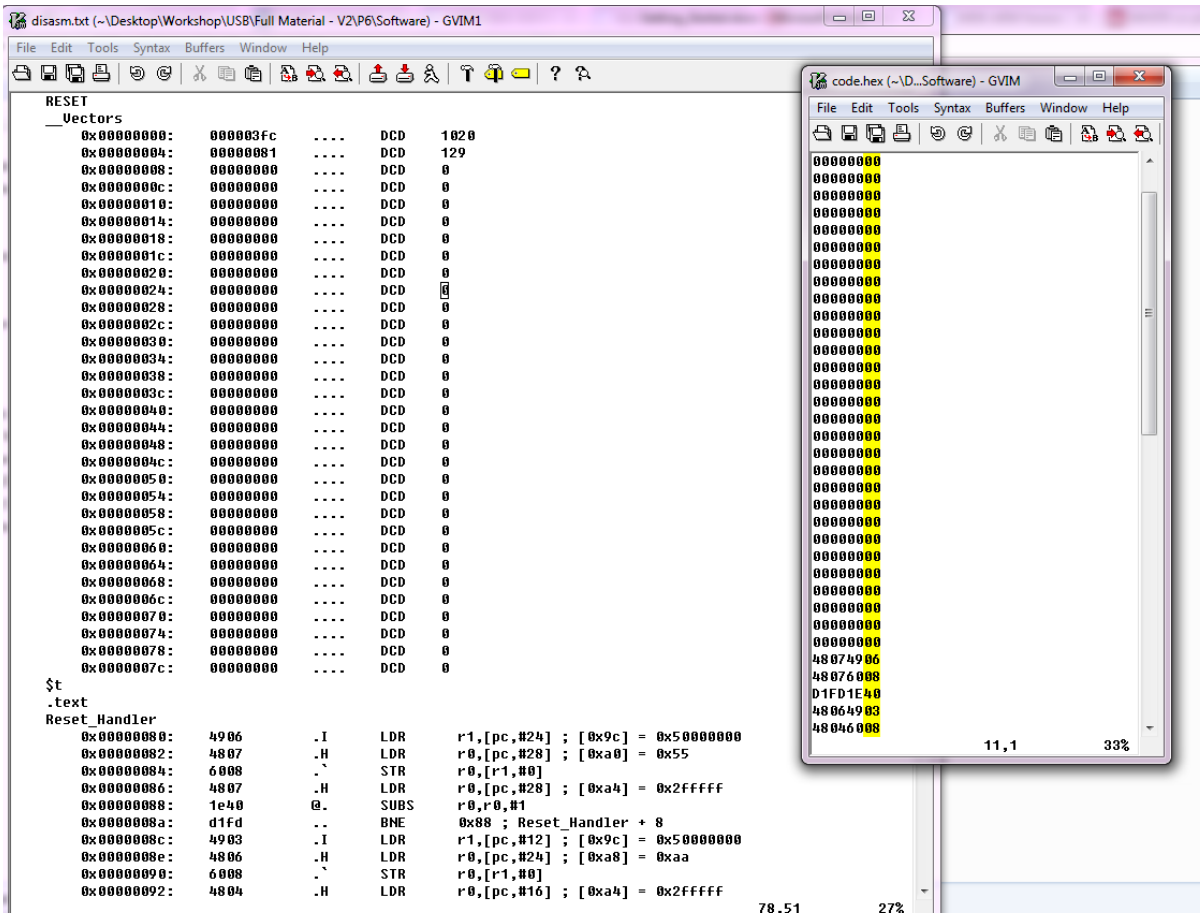
Copyright © ARM Ltd 2013



BUILD THE PROJECT

1. Build Target (Project → Build Target)
2. Analyze disasm.txt and lab.hex file generated in your project directory and compare with the source file cm0dsasm.s

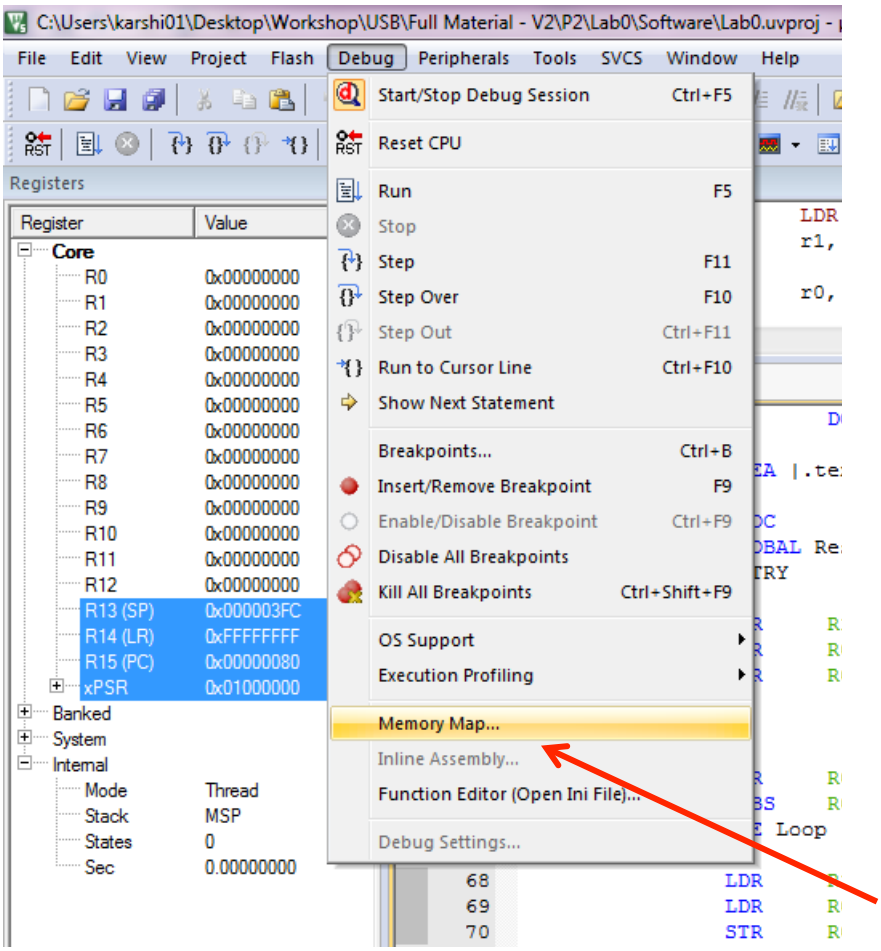
Copyright © ARM Ltd 2013

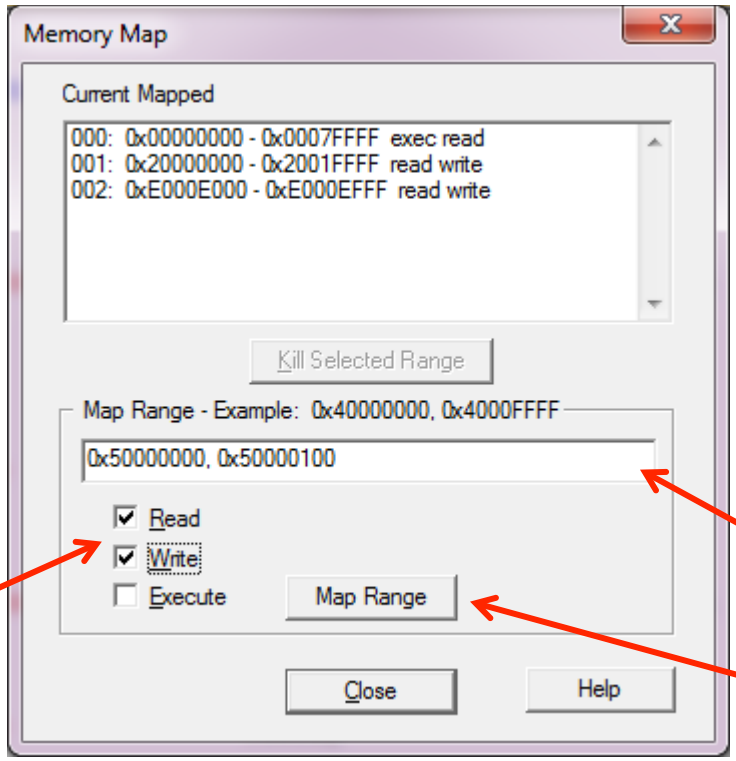


USING THE SIMULATOR

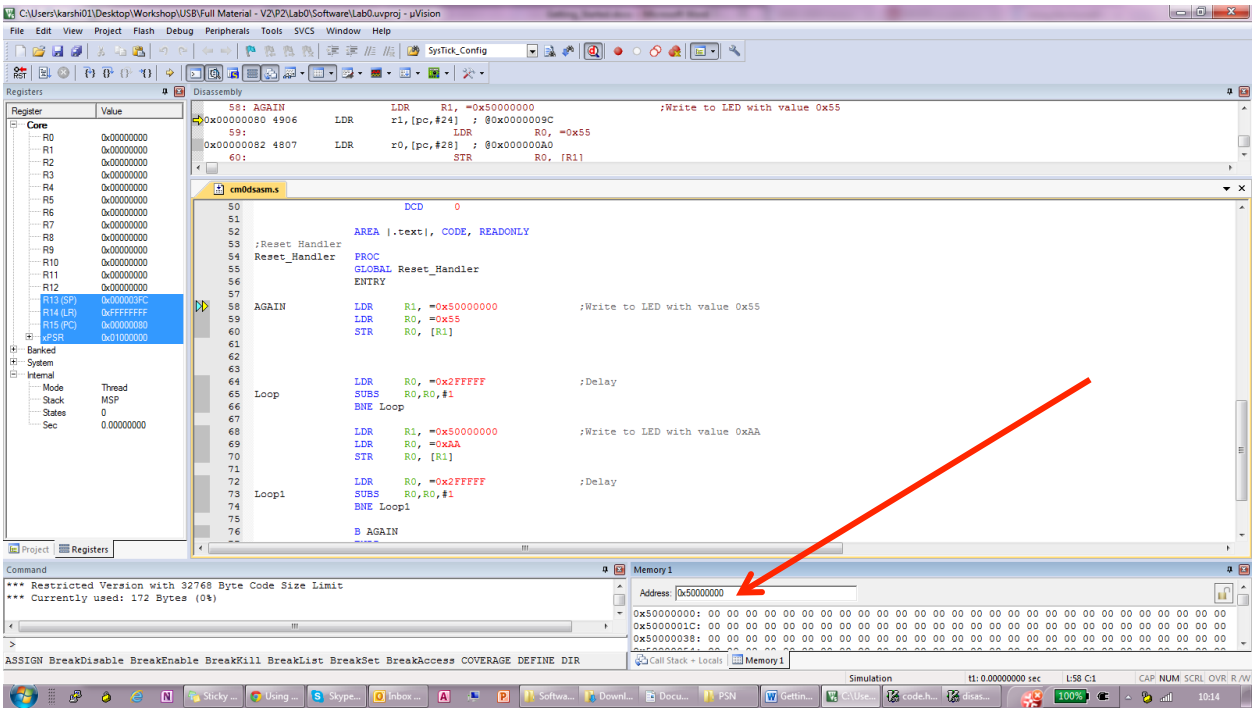
1. After you have compiled go to Debug → Start/Stop Debug Session
2. Ignore the warning message
3. Note that PC is already pointing to the Reset_Handler (This matches the entry point flag set during compilation)
4. Goto Debug → Memory Map and add the LED peripheral Memory information. Note the region should be Read and Write.

Copyright © ARM Ltd 2013


C:\Users\karshi01\Desktop\Workshop\USB\Full Material - V2\P2\Lab0\Software\Lab0.uvproj
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Start/Stop Debug Session Ctrl+F5
Reset CPU
Run F5
Stop
Step F11
Step Over F10
Step Out Ctrl+F11
Run to Cursor Line Ctrl+F10
Show Next Statement
Breakpoints... Ctrl+B
Insert/Remove Breakpoint F9
Enable/Disable Breakpoint Ctrl+F9
Disable All Breakpoints
Kill All Breakpoints Ctrl+Shift+F9
OS Support
Execution Profiling
Memory Map...
Inline Assembly...
Function Editor (Open Ini File)...
Debug Settings...
Registers
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) 0x000003FC
R14 (LR) 0xFFFFFFFF
R15 (PC) 0x00000080
xPSR 0x01000000
Banked
System
Internal
Mode Thread
Stack MSP
States 0
Sec 0.00000000
68 LDR
69 LDR
70 STR


Memory Map
Current Mapped
000: 0x00000000 - 0x0007FFFF exec read
001: 0x20000000 - 0x2001FFFF read write
002: 0xE000E000 - 0xE000EFFF read write
Kill Selected Range
Map Range - Example: 0x40000000, 0x4000FFFF
0x50000000, 0x50000100
Read
Write
Execute
Map Range
Close
Help

5. View the contents of at 0x5000_0000 in Memory1 Window.



6. Execute the image using Single Step and watch the memory contents change

7. Close the simulation using Debug → Close simuation