IIT PALAKKAD

Indian Institute of Technology Palakkad
भारतीय प्रौद्योगिकी संस्थान पालक्काड
Under Ministry of Human Resource Development, Govt. of India
मानव संसाधन विकास मंत्रालय के अधीन, भारत सरकार

EE2080 Microprocessor Systems Design and Interfacing Laboratory

Assembly programming using FRDM-KL25Z development board and Keil Uvision5.

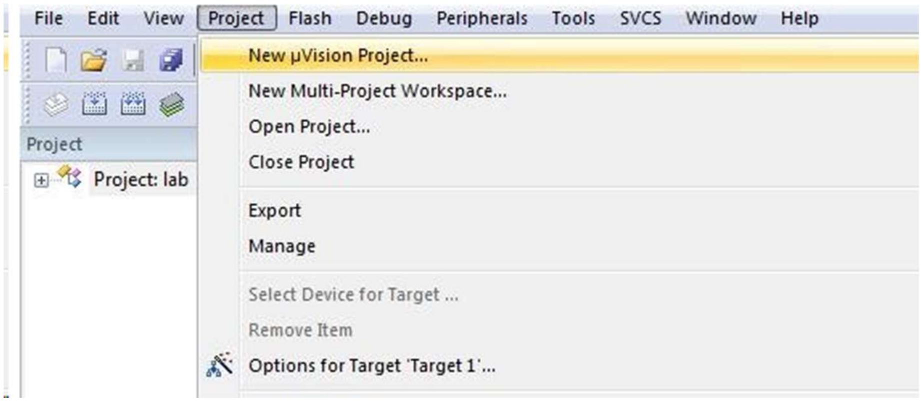
Steps:

1. Open the folder **EE2080 2019** in the desktop.

2. Create a new folder with your **roll no_expt.no** as the **folder name**. If you are doing in groups, the folder name should be **roll no1_roll no2_expt.no**. All the files for the experiment should be saved in this folder. **For each lab a separate folder in the above format should be created.**

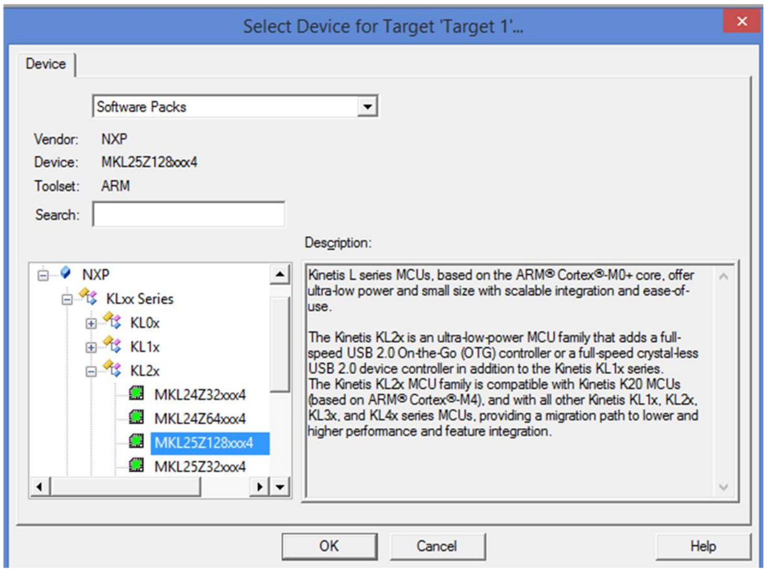
3. Open **Keil uVision5** application from the desktop by double clicking on the icon .

4. Click on Project→ New uVision Project



5. Enter the project name and save as Project files with extension *.uvproj or *.uvprojx in the folder you have created (roll no_expt.no).

6. A new window will be opened where we have to select the device that is going to be used. Select **MKL25Z128xxx4** as device (NXP→KL2x→ MKL25Z128xxx4). The description of the selected device will be displayed in the description box. Click OK.



Software Packs

Vendor: NXP
Device: MKL25Z128xxx4
Toolset: ARM
Search:

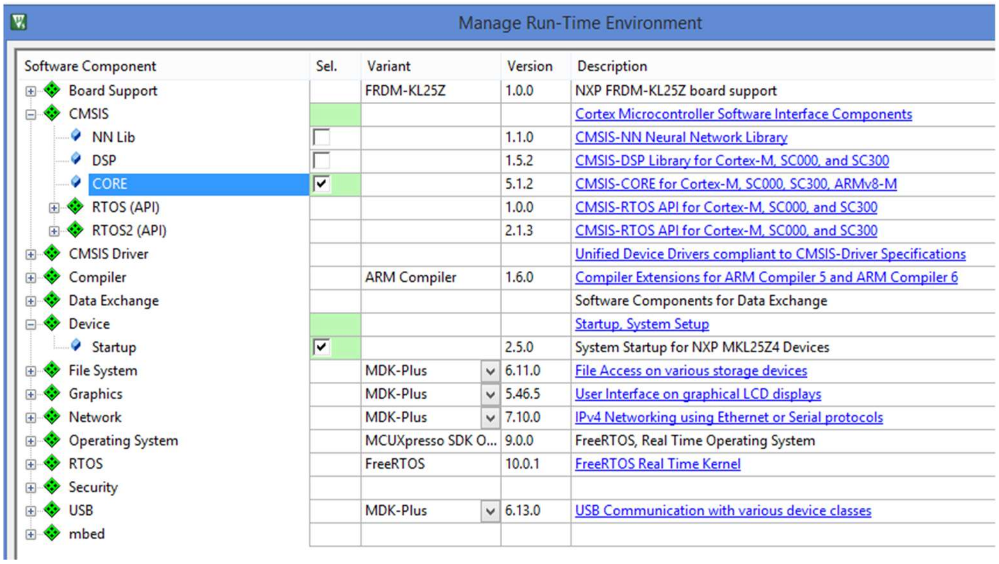
- NXP
  - KLxx Series
    - KL0x
    - KL1x
    - KL2x
      - MKL24Z32xxx4
      - MKL24Z64xxx4
      - MKL25Z128xxx4
      - MKL25Z32xxx4

Description:

Kinetis L series MCUs, based on the ARM® Cortex®-M0+ core, offer ultra-low power and small size with scalable integration and ease-of-use.

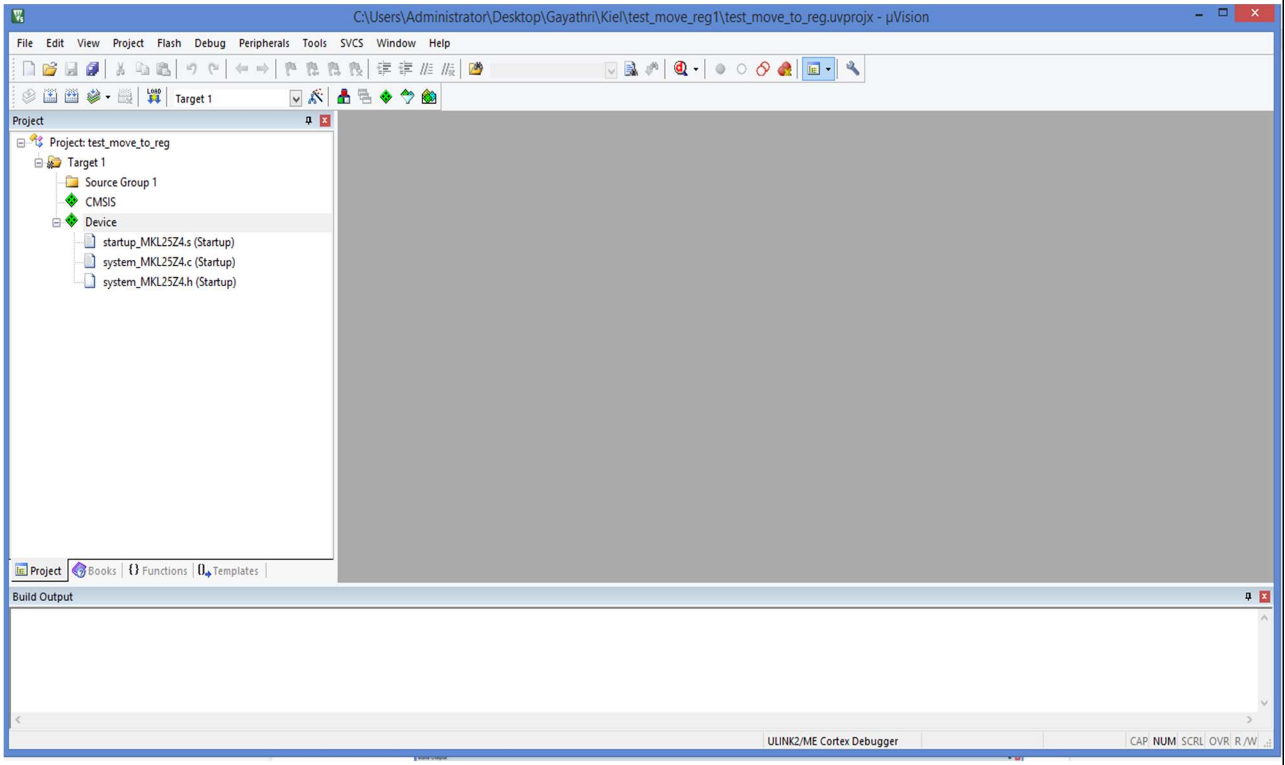
The Kinetis KL2x is an ultra-low-power MCU family that adds a full-speed USB 2.0 On-the-Go (OTG) controller or a full-speed crystal-less USB 2.0 device controller in addition to the Kinetis KL1x series.
The Kinetis KL2x MCU family is compatible with Kinetis K20 MCUs (based on ARM® Cortex®-M4), and with all other Kinetis KL1x, KL2x, KL3x, and KL4x series MCUs, providing a migration path to lower and higher performance and feature integration.

7. In the '**Manage Run-Time Environment**' window that appears, make the selection as shown below and click Ok. This is for providing the startup settings for your board.

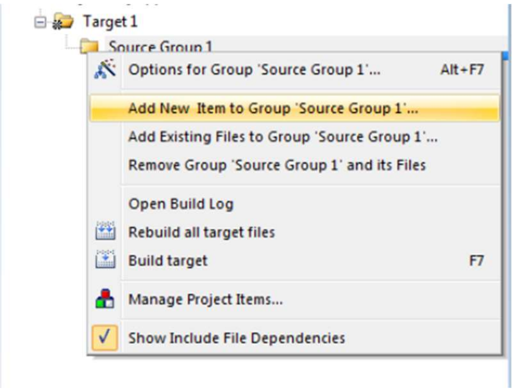


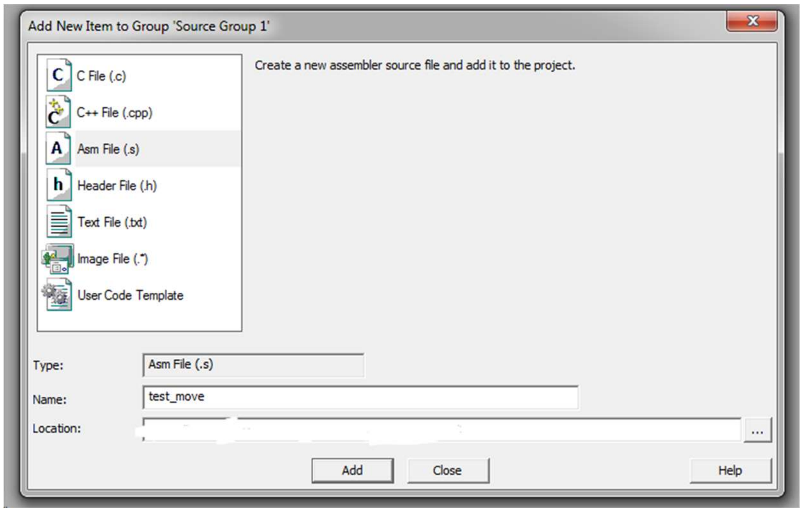
Software Component	Sel.	Variant	Version	Description
Board Support		FRDM-KL25Z	1.0.0	NXP FRDM-KL25Z board support
CMSIS				Cortex Microcontroller Software Interface Components
NN Lib	□		1.1.0	CMSIS-NN Neural Network Library
DSP	□		1.5.2	CMSIS-DSP Library for Cortex-M, SC000, and SC300
CORE	☑		5.1.2	CMSIS-CORE 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
Data Exchange				Software Components for Data Exchange
Device				Startup, System Setup
Startup	☑		2.5.0	System Startup for NXP MKL25Z4 Devices
File System		MDK-Plus	6.11.0	File Access on various storage devices
Graphics		MDK-Plus	5.46.5	User Interface on graphical LCD displays
Network		MDK-Plus	7.10.0	IPv4 Networking using Ethernet or Serial protocols
Operating System		MCUXpresso SDK O...	9.0.0	FreeRTOS, Real Time Operating System
RTOS		FreeRTOS	10.0.1	FreeRTOS Real Time Kernel
Security				
USB		MDK-Plus	6.13.0	USB Communication with various device classes
mbed				

8. On clicking Ok you will get a window as shown below:


C:\Users\Administrator\Desktop\Gayathri\Kiel\test_move_reg1\test_move_to_reg.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: test_move_to_reg
Target 1
Source Group 1
CMSIS
Device
startup_MKL25Z4.s (Startup)
system_MKL25Z4.c (Startup)
system_MKL25Z4.h (Startup)
Project
Books
{} Functions
0+ Templates
Build Output
ULINK2/ME Cortex Debugger
CAP NUM SCRL OVR R/W

9. To create the assembly code file, right click on your target folder → Select ‘Add new item to Group’ → Select the type of file as Asm file (.s) → Give a name for the file → Click ‘Add’ → Ensure that the file is added to the project.


Target 1
Source Group 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
Open Build Log
Rebuild all target files
Build target F7
Manage Project Items...
Show Include File Dependencies

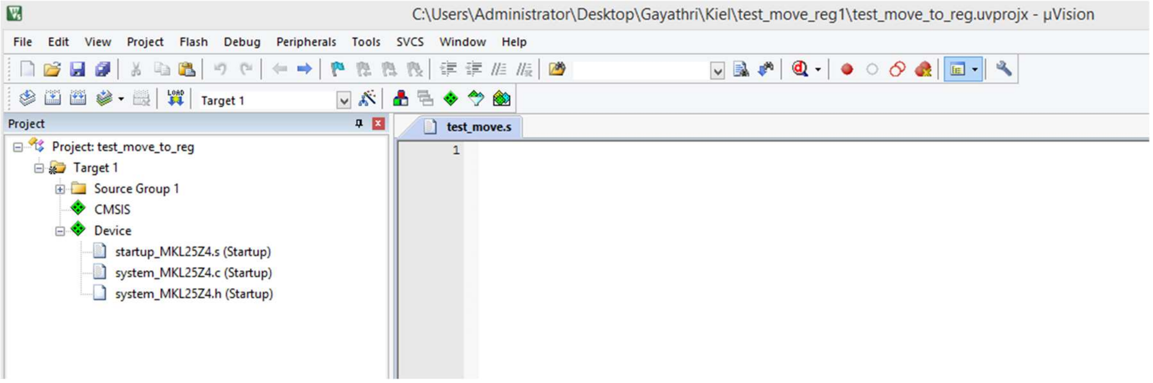


Create a new assembler source file and add it to the project.

Type: Asm File (.s)

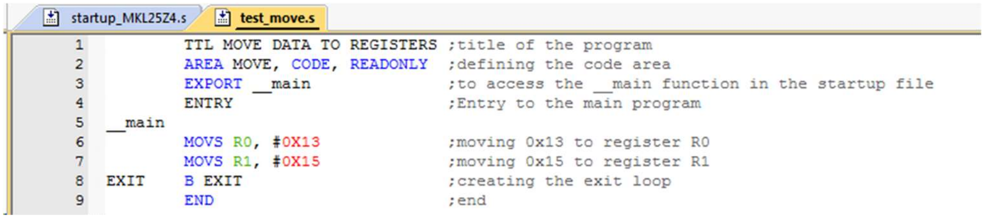
Name: test_move

Location:


C:\Users\Administrator\Desktop\Gayathri\Kiel\test_move_reg1\test_move_to_reg.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: test_move_to_reg
Target 1
Source Group 1
CMSIS
Device
startup_MKL25Z4.s (Startup)
system_MKL25Z4.c (Startup)
system_MKL25Z4.h (Startup)
test_move.s
1

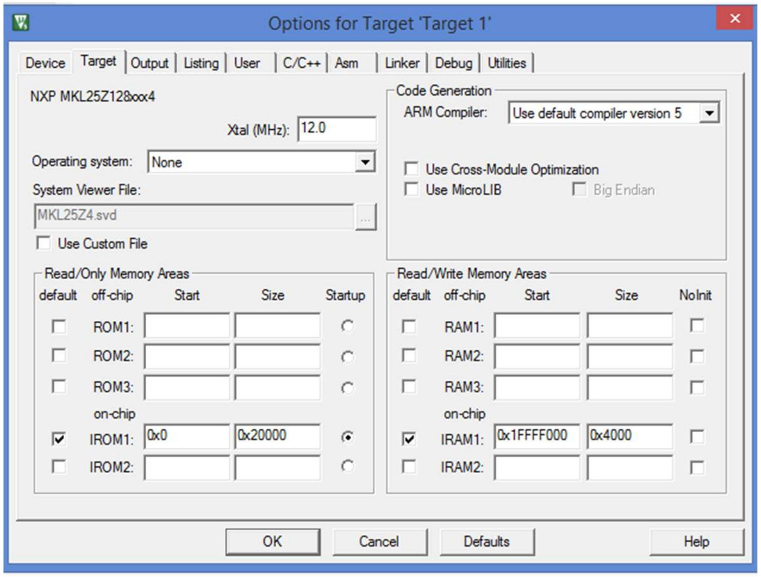
10. Writing an assembly program: Open your asm file.

    Here is a sample program to move values 0x13 and 0x15 into registers R0 and R1 respectively.



```
        TTL MOVE DATA TO REGISTERS ;title of the program
        AREA MOVE, CODE, READONLY  ;defining the code area
        EXPORT __main              ;to access the __main function in the startup file
        ENTRY                      ;Entry to the main program
__main
        MOVS R0, #0X13             ;moving 0x13 to register R0
        MOVS R1, #0X15             ;moving 0x15 to register R1
EXIT    B EXIT                     ;creating the exit loop
        END                        ;end
```

11. Click on the icon (Options for target) on the build toolbar. You will get a window as below:



NXP MKL25Z128xxx4

Xtal (MHz): 12.0

Operating system: None

System Viewer File: MKL25Z4.svd

□ Use Custom File

Code Generation

ARM Compiler: Use default compiler version 5

□ Use Cross-Module Optimization

□ Use MicroLIB □ Big Endian

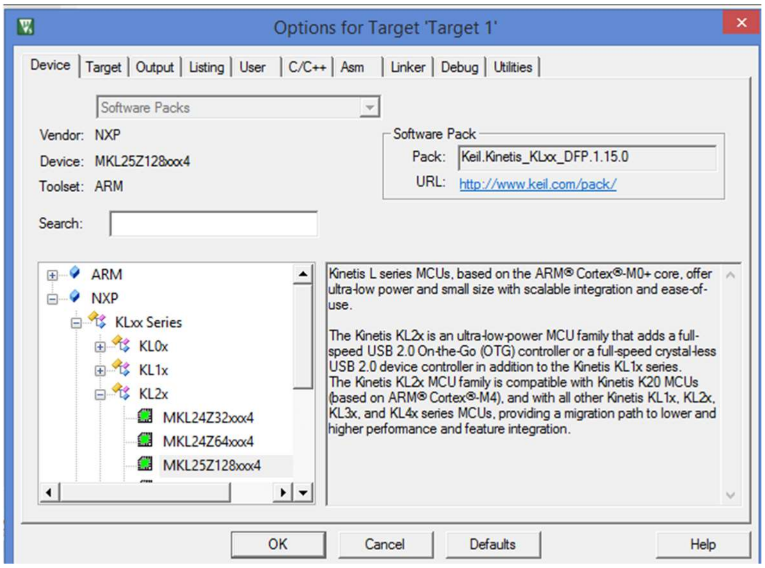
Read/Only Memory Areas

default	off-chip	Start	Size	Startup
□	ROM1:			○
□	ROM2:			○
□	ROM3:			○
	on-chip			
☑	IROM1:	0x0	0x20000	●
□	IROM2:			○

Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
□	RAM1:			□
□	RAM2:			□
□	RAM3:			□
	on-chip			
☑	IRAM1:	0x1FFFF000	0x4000	□
□	IRAM2:			□

12. Click on the 'Device' tab. You will see a list of devices from various companies. Select NXP → MKL25Z128xxx4 (the board that is going to be used). Once you select it the description of board will be shown on the side of the window.



Vendor: NXP

Device: MKL25Z128xxx4

Toolset: ARM

Search:

Software Pack

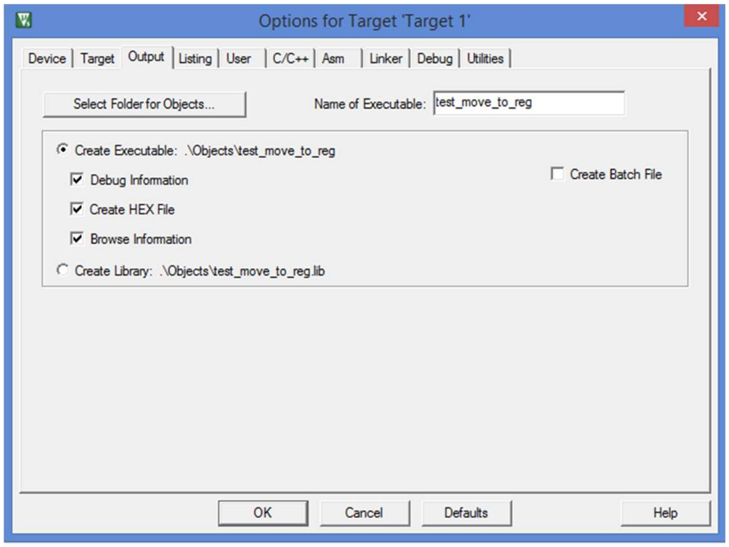
Pack: Keil.Kinetis_KLxx_DFP.1.15.0

URL: http://www.keil.com/pack/

Kinetis L series MCUs, based on the ARM® Cortex®-M0+ core, offer ultra-low power and small size with scalable integration and ease-of-use.

The Kinetis KL2x is an ultra-low-power MCU family that adds a full-speed USB 2.0 On-the-Go (OTG) controller or a full-speed crystal-less USB 2.0 device controller in addition to the Kinetis KL1x series.
The Kinetis KL2x MCU family is compatible with Kinetis K20 MCUs (based on ARM® Cortex®-M4), and with all other Kinetis KL1x, KL2x, KL3x, and KL4x series MCUs, providing a migration path to lower and higher performance and feature integration.

13. Click 'Output' tab → Tick 'Create HEX file'.



Options for Target 'Target 1'
Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities
Select Folder for Objects...
OK
Cancel
Defaults
Help

Name of Executable: test_move_to_reg

● Create Executable: .\Objects\test_move_to_reg

☑ Debug Information

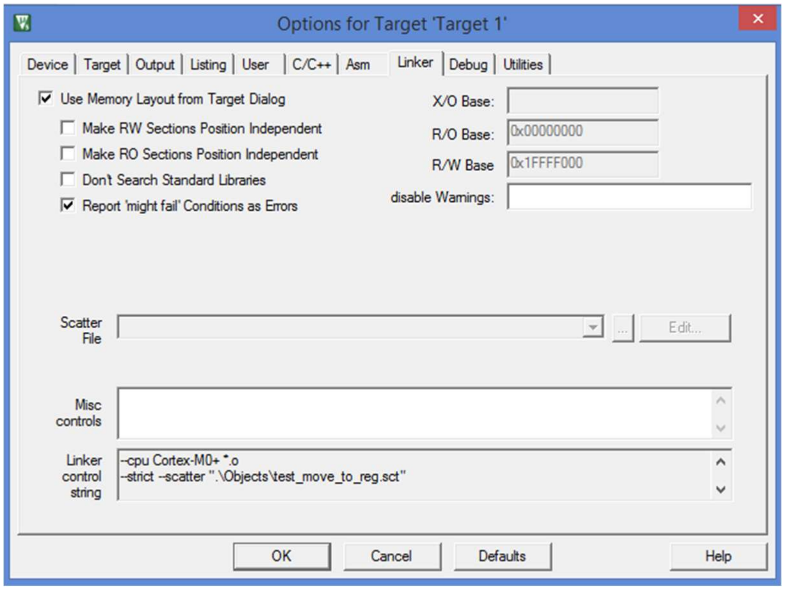
☑ Create HEX File

☑ Browse Information

○ Create Library: .\Objects\test_move_to_reg.lib

□ Create Batch File

14. Do not change anything on the ‘Listing’, ‘User’, ‘C/C++’, ‘Asm’ tabs. Take the ‘Linker’ tab → Tick ‘Use Memory Layout from Target Dialog’.



Options for Target 'Target 1'
Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities
Edit...
OK
Cancel
Defaults
Help

☑ Use Memory Layout from Target Dialog

□ Make RW Sections Position Independent

□ Make RO Sections Position Independent

□ Don't Search Standard Libraries

☑ Report 'might fail' Conditions as Errors

X/O Base:

R/O Base: 0x00000000

R/W Base 0x1FFFF000

disable Warnings:

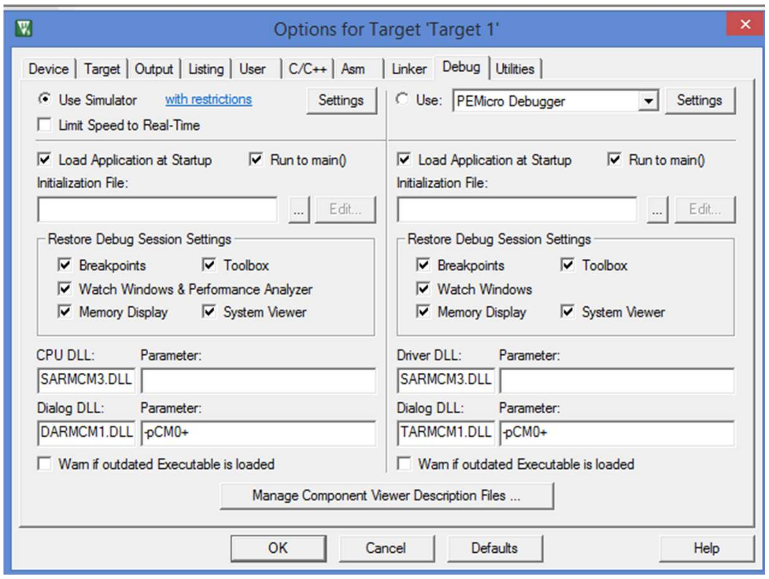
Scatter File

Misc controls

Linker control string

```
--cpu Cortex-M0+ *.o
--strict --scatter ".\Objects\test_move_to_reg.sct"
```

15. Click ‘Debug’ tab and select ‘Use simulator’ as we are trying out simulation first without using the board and click Ok.



Options for Target 'Target 1'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

● Use Simulator with restrictions [Settings] | ○ Use: PEMicro Debugger [Settings]

□ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main() | ☑ Load Application at Startup ☑ Run to main()

Initialization File: [...] [Edit...] | Initialization File: [...] [Edit...]

Restore Debug Session Settings | Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox | ☑ Breakpoints ☑ Toolbox

☑ Watch Windows & Performance Analyzer | ☑ Watch Windows

☑ Memory Display ☑ System Viewer | ☑ Memory Display ☑ System Viewer

CPU DLL: SARMCM3.DLL Parameter: | Driver DLL: SARMCM3.DLL Parameter:

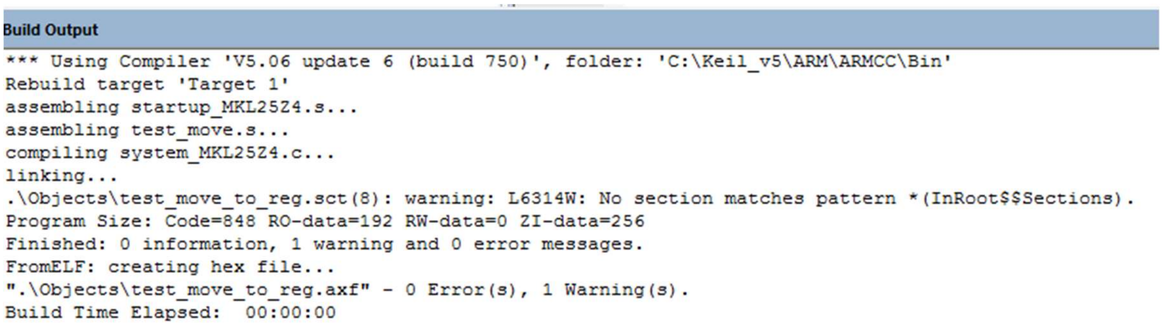
Dialog DLL: DARMCM1.DLL Parameter: -pCM0+ | Dialog DLL: TARMCM1.DLL Parameter: -pCM0+

□ Warn if outdated Executable is loaded | □ Warn if outdated Executable is loaded

16. Building your program: Right click on the Source group → Rebuild all target files or click on the icon in the toolbar.

    Important: You have to rebuild your program after every modification.

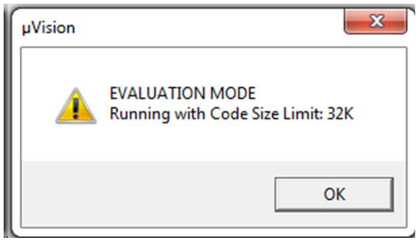
    Check if build is completed successfully:



Build Output

```
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Rebuild target 'Target 1'
assembling startup_MKL25Z4.s...
assembling test_move.s...
compiling system_MKL25Z4.c...
linking...
.\Objects\test_move_to_reg.sct(8): warning: L6314W: No section matches pattern *(InRoot$$Sections).
Program Size: Code=848 RO-data=192 RW-data=0 ZI-data=256
Finished: 0 information, 1 warning and 0 error messages.
FromELF: creating hex file...
".\Objects\test_move_to_reg.axf" - 0 Error(s), 1 Warning(s).
Build Time Elapsed:  00:00:00
```

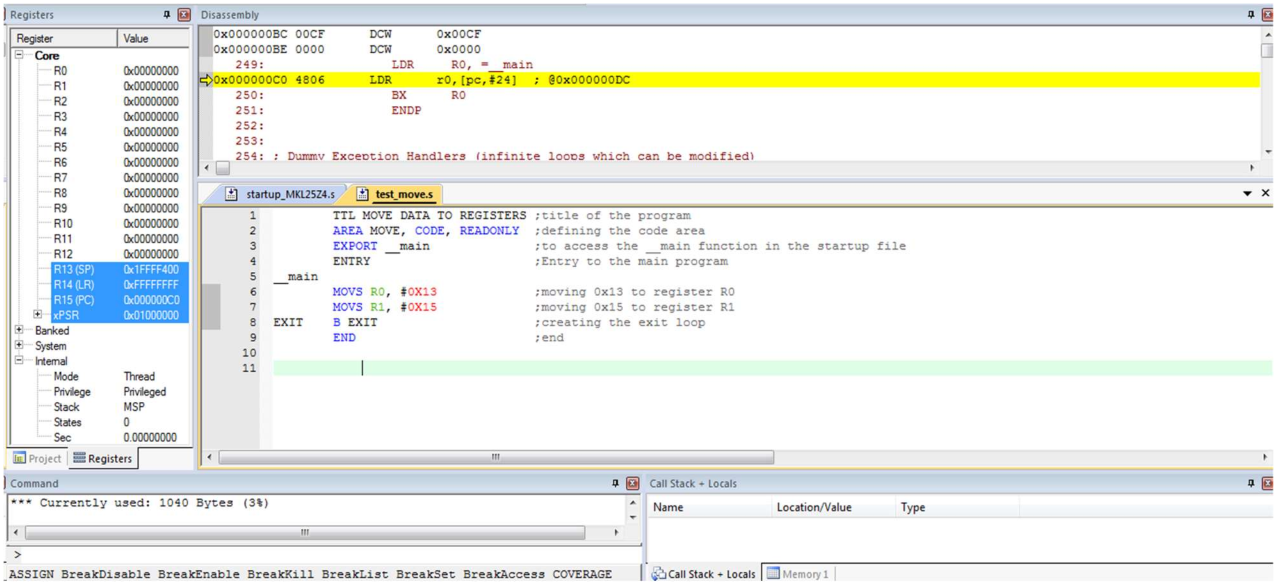
17. Debugging the code: Click 'Debug' → Start/Stop debug session → Click Ok in the dialog box that appears.



µVision

EVALUATION MODE
Running with Code Size Limit: 32K

18. You get the debug windows as shown below:



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)	0x1FFFF400
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x000000C0
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	0
Sec	0.00000000

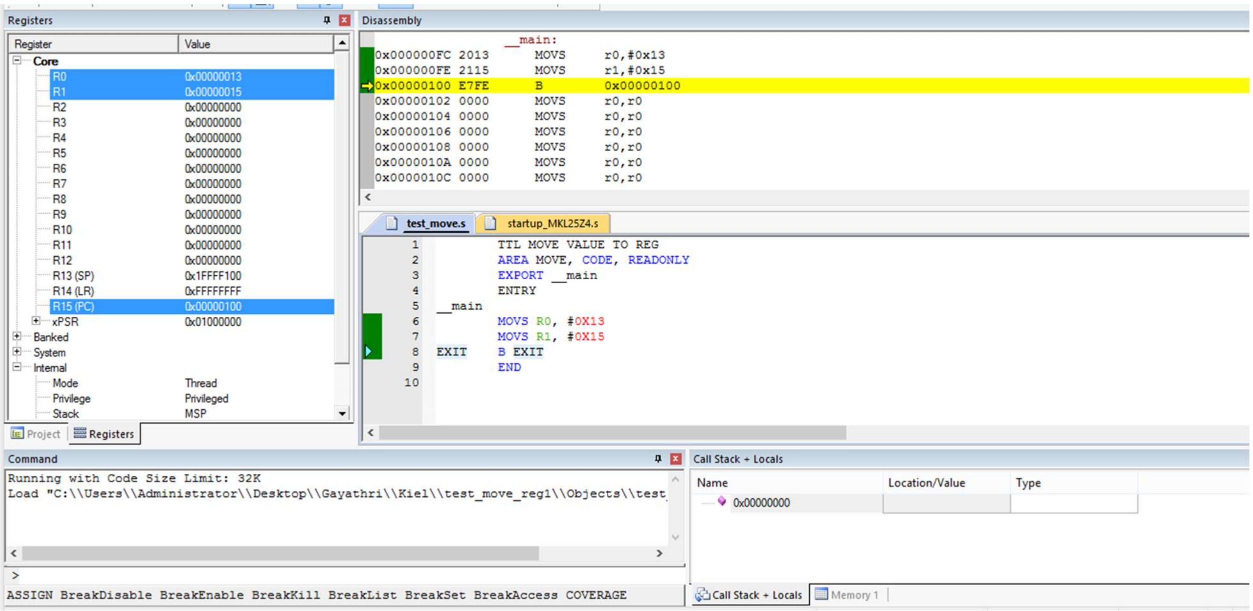
```
0x000000BC 00CF      DCW      0x00CF
0x000000BE 0000      DCW      0x0000
   249:                 LDR      R0, =__main
0x000000C0 4806      LDR      r0,[pc,#24]  ; @0x000000DC
   250:                 BX       R0
   251:                 ENDP
   252:
   253:
   254: ; Dummy Exception Handlers (infinite loops which can be modified)
```

```
        TTL MOVE DATA TO REGISTERS ;title of the program
        AREA MOVE, CODE, READONLY  ;defining the code area
        EXPORT __main              ;to access the __main function in the startup file
        ENTRY                      ;Entry to the main program
__main
        MOVS R0, #0X13             ;moving 0x13 to register R0
        MOVS R1, #0X15             ;moving 0x15 to register R1
EXIT    B EXIT                     ;creating the exit loop
        END                        ;end
```

```
*** Currently used: 1040 Bytes (3%)
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
```

Name	Location/Value	Type

19. Use these icons as needed to run your code. Running using single step at a time is illustrated below:



Register	Value
Core	
R0	0x00000013
R1	0x00000015
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x1FFFF100
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000100
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP

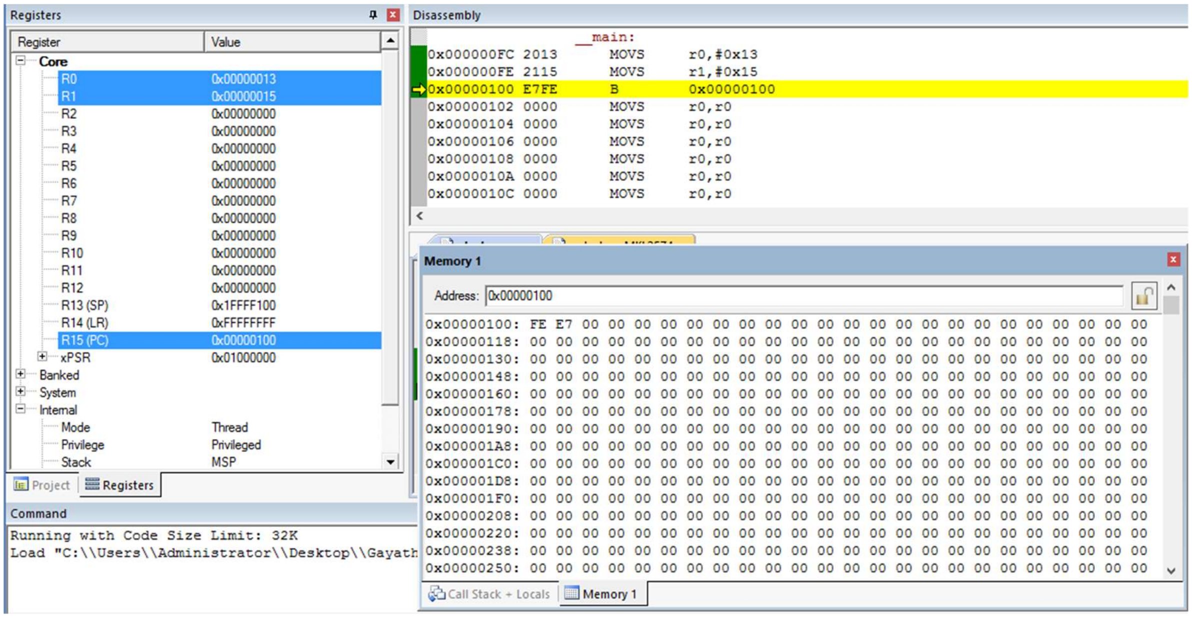
```
                     __main:
0x000000FC 2013         MOVS     r0,#0x13
0x000000FE 2115         MOVS     r1,#0x15
0x00000100 E7FE         B        0x00000100
0x00000102 0000         MOVS     r0,r0
0x00000104 0000         MOVS     r0,r0
0x00000106 0000         MOVS     r0,r0
0x00000108 0000         MOVS     r0,r0
0x0000010A 0000         MOVS     r0,r0
0x0000010C 0000         MOVS     r0,r0
```

```
        TTL MOVE VALUE TO REG
        AREA MOVE, CODE, READONLY
        EXPORT __main
        ENTRY
__main
        MOVS R0, #0X13
        MOVS R1, #0X15
EXIT    B EXIT
        END
```

```
Running with Code Size Limit: 32K
Load "C:\\Users\\Administrator\\Desktop\\Gayathri\\Kiel\\test_move_reg1\\Objects\\test
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
```

Name	Location/Value	Type
0x00000000		

You can see the contents of the memory by clicking the memory tab as shown below:



Registers

Register	Value
Core	
R0	0x00000013
R1	0x00000015
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x1FFFF100
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000100
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP

Disassembly

```
                   __main:
0x000000FC 2013      MOVS     r0,#0x13
0x000000FE 2115      MOVS     r1,#0x15
0x00000100 E7FE      B        0x00000100
0x00000102 0000      MOVS     r0,r0
0x00000104 0000      MOVS     r0,r0
0x00000106 0000      MOVS     r0,r0
0x00000108 0000      MOVS     r0,r0
0x0000010A 0000      MOVS     r0,r0
0x0000010C 0000      MOVS     r0,r0
```

Memory 1

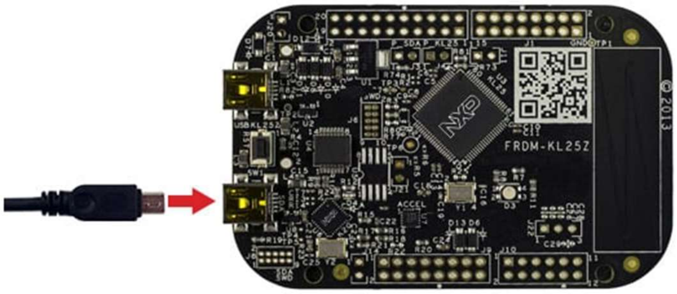
Address: 0x00000100

```
0x00000100: FE E7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000118: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000148: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000178: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000001A8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000001C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000001D8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000001F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000208: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000238: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000250: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Command

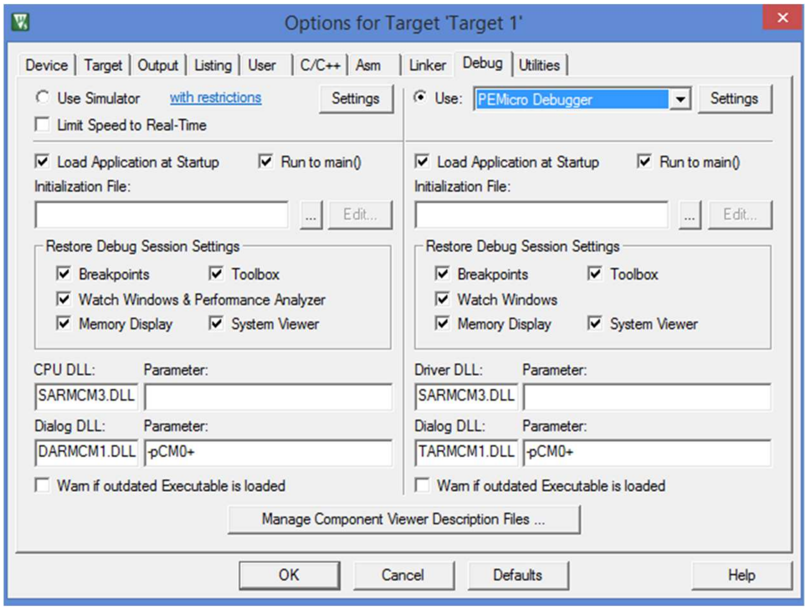
```
Running with Code Size Limit: 32K
Load "C:\\Users\\Administrator\\Desktop\\Gayath
```

20. Connect the USB cable to the ‘Open SDA’ port of the FRDM KL25Z board.



(Figure Courtesy: https://www.nxp.com)

21. Selecting the hardware debugging mode: Click on the icon (Options for target) on the build toolbar. Go to ‘Debug’ tab and select ‘PEMicro Debugger’ (this is the debugger that we are using).



Options for Target 'Target 1'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

○ Use Simulator with restrictions Settings
□ 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:
Dialog DLL: DARMCM1.DLL Parameter: -pCM0+

□ Warn if outdated Executable is loaded

● Use: PEMicro Debugger Settings

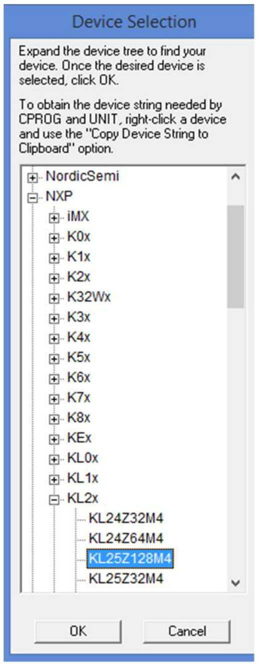
☑ Load Application at Startup ☑ Run to main()
Initialization File:

Restore Debug Session Settings
☑ Breakpoints ☑ Toolbox
☑ Watch Windows
☑ Memory Display ☑ System Viewer

Driver DLL: SARMCM3.DLL Parameter:
Dialog DLL: TARMCM1.DLL Parameter: -pCM0+

□ Warn if outdated Executable is loaded

Click on 'Settings' beside the 'PEMicro Debugger'. In the 'Interface', select 'OpenSDA Embedded Tower Debug-USB Port' as the interface type. Click Refresh list. If your device is properly connected, it will detect the Open SDA port of your device and will update the connection port part. In the device selection area click 'Select New Device' and select your device and Click Ok.



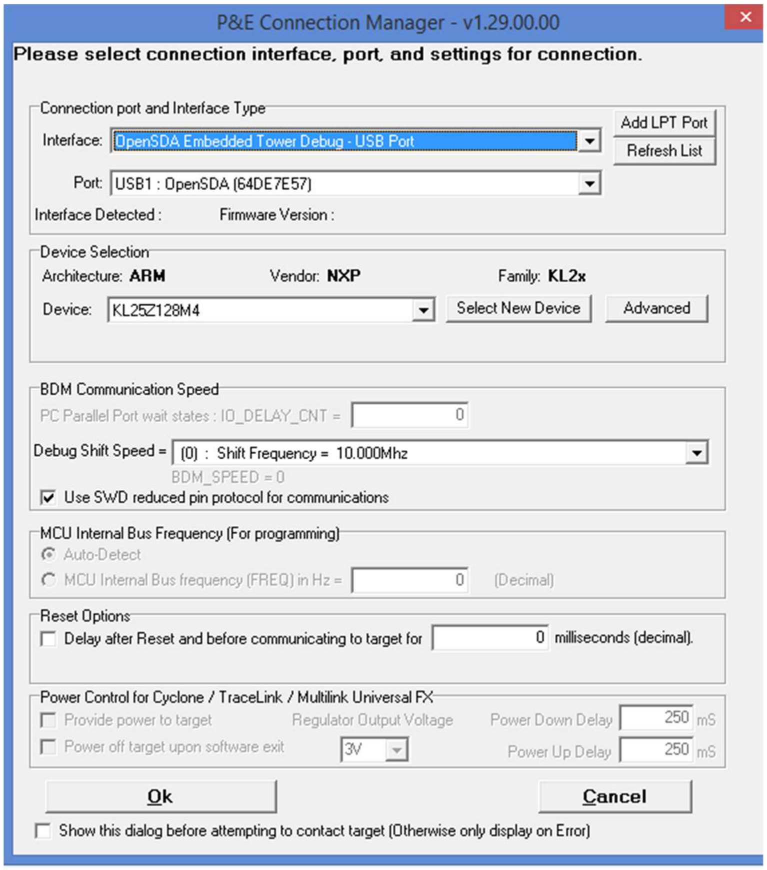
Device Selection

Expand the device tree to find your device. Once the desired device is selected, click OK.

To obtain the device string needed by CPROG and UNIT, right-click a device and use the "Copy Device String to Clipboard" option.

```
⊞ NordicSemi
⊟ NXP
   ⊞ iMX
   ⊞ K0x
   ⊞ K1x
   ⊞ K2x
   ⊞ K32Wx
   ⊞ K3x
   ⊞ K4x
   ⊞ K5x
   ⊞ K6x
   ⊞ K7x
   ⊞ K8x
   ⊞ KEx
   ⊞ KL0x
   ⊞ KL1x
   ⊟ KL2x
      KL24Z32M4
      KL24Z64M4
      KL25Z128M4
      KL25Z32M4
```

After updating the interface type, connection port and device, your window should look like as shown below:



P&E Connection Manager - v1.29.00.00

Please select connection interface, port, and settings for connection.

Connection port and Interface Type

Interface: OpenSDA Embedded Tower Debug - USB Port

Port: USB1 : OpenSDA (64DE7E57)

Interface Detected : Firmware Version :

Device Selection

Architecture: **ARM** Vendor: **NXP** Family: **KL2x**

Device: KL25Z128M4

BDM Communication Speed

PC Parallel Port wait states : IO_DELAY_CNT = 0

Debug Shift Speed = (0) : Shift Frequency = 10.000Mhz

BDM_SPEED = 0

☑ Use SWD reduced pin protocol for communications

MCU Internal Bus Frequency (For programming)

● Auto-Detect

○ MCU Internal Bus frequency (FREQ) in Hz = 0 (Decimal)

Reset Options

□ Delay after Reset and before communicating to target for 0 milliseconds (decimal).

Power Control for Cyclone / TraceLink / Multilink Universal FX

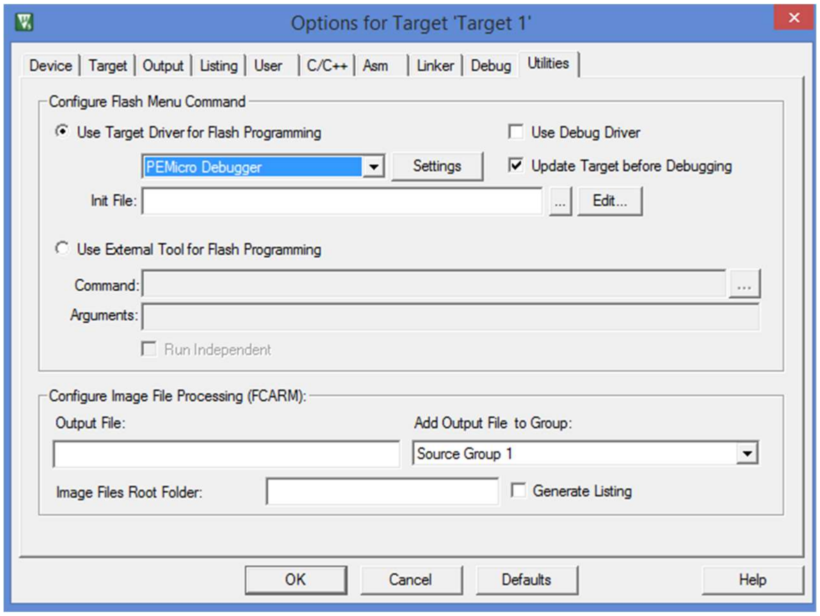
□ Provide power to target Regulator Output Voltage 3V Power Down Delay 250 mS

□ Power off target upon software exit Power Up Delay 250 mS

□ Show this dialog before attempting to contact target (Otherwise only display on Error)

Click Ok.

22. Go to ‘Debug’ tab → Uncheck the box of ‘Use Debug Driver’ and select PEmicro Debugger as the target driver and click Ok.



Options for Target 'Target 1'

Configure Flash Menu Command

● Use Target Driver for Flash Programming: PEMicro Debugger

□ Use Debug Driver

☑ Update Target before Debugging

Init File:

○ Use External Tool for Flash Programming

Command:

Arguments:

□ Run Independent

Configure Image File Processing (FCARM):

Output File:

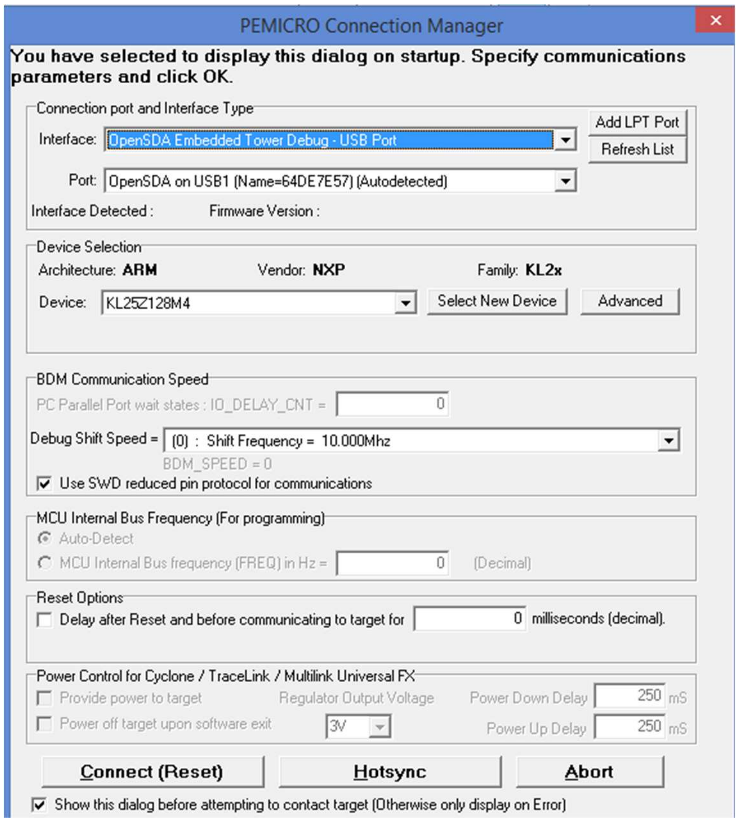
Add Output File to Group: Source Group 1

Image Files Root Folder:

□ Generate Listing

23. Rebuild your code. Check and correct the errors if any.

24. Loading the code onto the board: Click on 'Download code to Flash memory' . Select your interface and device as done in Step 25. Click Connect (Reset).



PEMICRO Connection Manager

You have selected to display this dialog on startup. Specify communications parameters and click OK.

Connection port and Interface Type

Interface: OpenSDA Embedded Tower Debug - USB Port

Port: OpenSDA on USB1 (Name=64DE7E57) (Autodetected)

Interface Detected : Firmware Version :

Device Selection

Architecture: **ARM** Vendor: **NXP** Family: **KL2x**

Device: KL25Z128M4

BDM Communication Speed

PC Parallel Port wait states : IO_DELAY_CNT = 0

Debug Shift Speed = (0) : Shift Frequency = 10.000Mhz

BDM_SPEED = 0

☑ Use SWD reduced pin protocol for communications

MCU Internal Bus Frequency (For programming)

● Auto-Detect

○ MCU Internal Bus frequency (FREQ) in Hz = 0 (Decimal)

Reset Options

□ Delay after Reset and before communicating to target for 0 milliseconds (decimal).

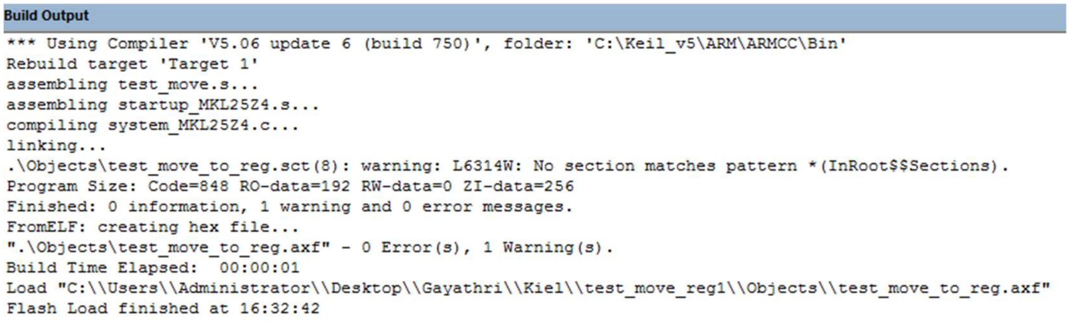
Power Control for Cyclone / TraceLink / Multilink Universal FX

□ Provide power to target Regulator Output Voltage 3V Power Down Delay 250 mS

□ Power off target upon software exit Power Up Delay 250 mS

☑ Show this dialog before attempting to contact target (Otherwise only display on Error)

25. See in the build output window if the code is loaded into the board correctly.



Build Output

```
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Rebuild target 'Target 1'
assembling test_move.s...
assembling startup_MKL25Z4.s...
compiling system_MKL25Z4.c...
linking...
.\Objects\test_move_to_reg.sct(8): warning: L6314W: No section matches pattern *(InRoot$$Sections).
Program Size: Code=848 RO-data=192 RW-data=0 ZI-data=256
Finished: 0 information, 1 warning and 0 error messages.
FromELF: creating hex file...
".\Objects\test_move_to_reg.axf" - 0 Error(s), 1 Warning(s).
Build Time Elapsed:  00:00:01
Load "C:\\Users\\Administrator\\Desktop\\Gayathri\\Kiel\\test_move_reg1\\Objects\\test_move_to_reg.axf"
Flash Load finished at 16:32:42
```

26. Perform the debugging as in Steps 17-19.