

# An Introduction to Keil uVersion

## 0. Objectives

This lab will introduce you to Keil uVersion – a free software as a subset of Keil MDK, an ARM software development environment that emulates and runs your program on a real ARM processor. Keil uVersion is used in Lab1b, and the final project.

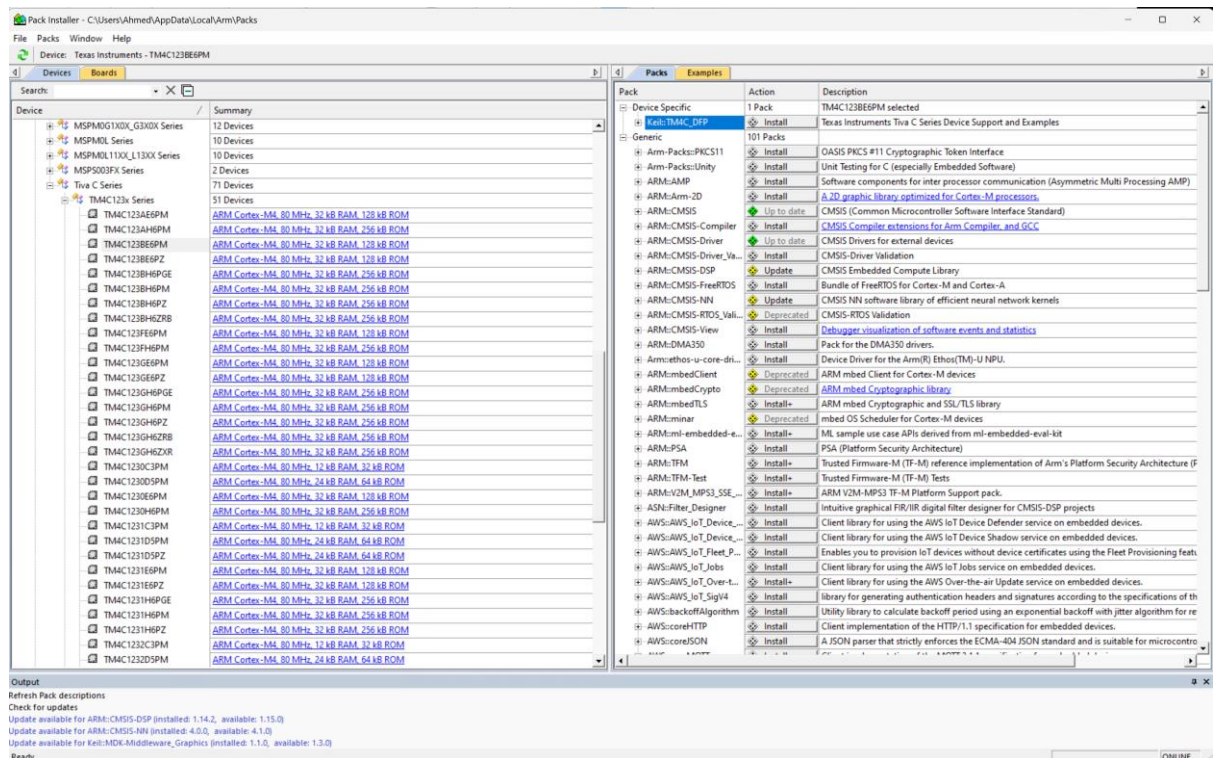
## 1. Download and Install Keil Version

- UW1-320 windows: already installed.
- Your own windows: visit <https://www.keil.com/download/>, choose “Product Downloads” and “MDK-ARM”, and submit your request to download MDK536.EXE.



1-a

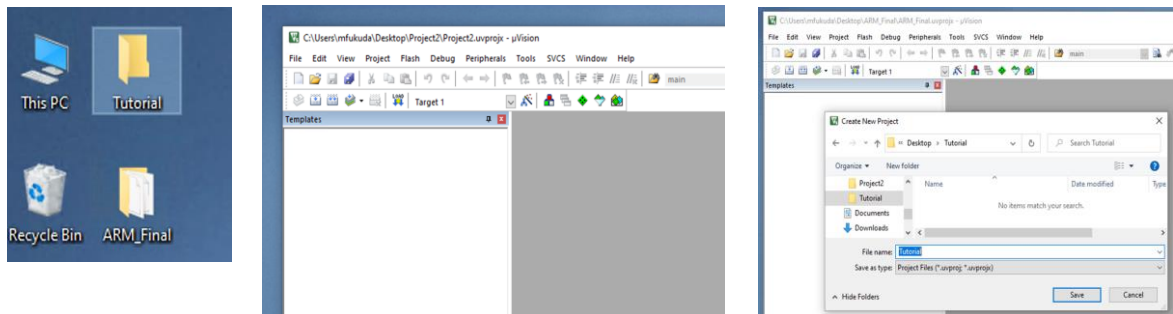
After installation is completed, install the TM4C pack from Pack Installer app.



1-b

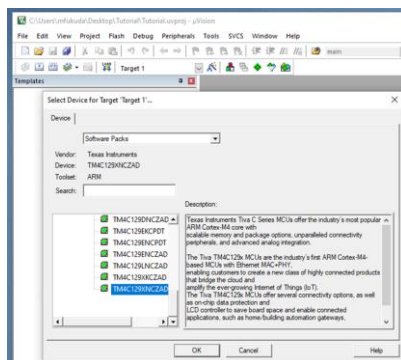
## 2. Project Set-up

- a. Create a new folder, start Keil uVersion, click “Project” to choose “new uVersion project”, choose the new folder you created, type in a project name in “File name” field, and finally click the “save” button.



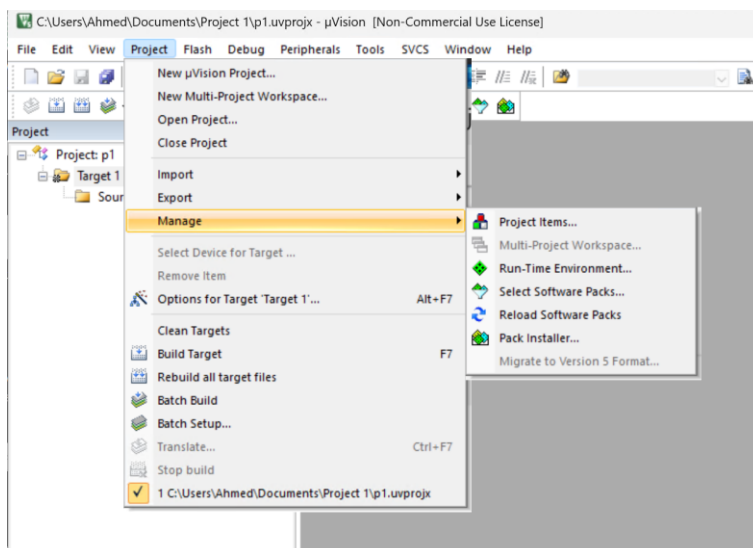
2-a

- b. In the “Select Device” menu, choose “TM4C1233H6PM” for Lab1b or “TM4C129XNCZAD” for the final project.

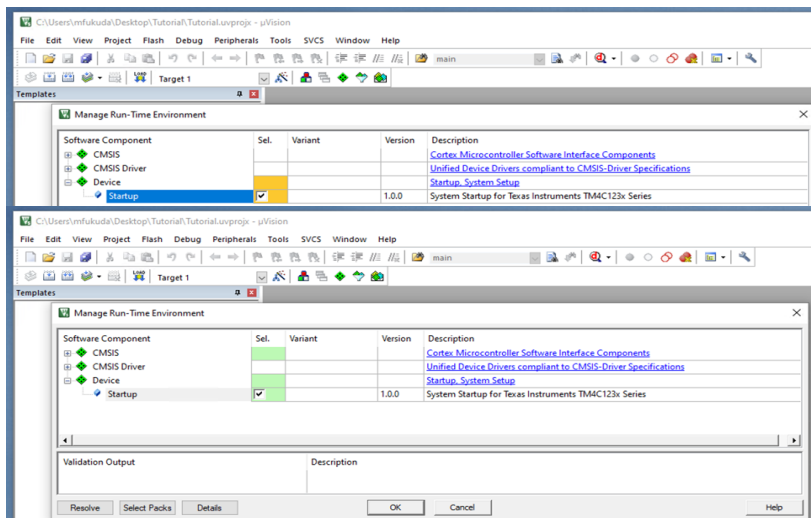


2-b

- c. In the “Manage Run-Time Environment” menu, check out Device “Startup” and click “resolve”.



2-c



2-d

### 3. How Your Program is Invoked

1. startup\_TM4C123.s or startup\_TM4C129 will be executed to **jump to Reset\_Handler**
2. BLX (branch indirect with link) will be executed to **call SystemInit** (which is in system\_TMC4123.c or system\_TMC4129.c) to initialize the system clock.
3. BX (branch indirect) will be executed to **call main** (which should be implemented either **\_\_main in your\_cortex\_prog.s** or **main() in your\_c\_prog.c**).
4. Note: in your final project, you'll write main() in driver.c that intends to verify your Cortex assembly code.

```

33 Stack_Size      EQU      0x00000200
34
35                  AREA     STACK, NOINIT, READWRITE, ALIGN=3
36 Stack_Mem        SPACE    Stack_Size
37 __initial_sp
38
39
40 ; <h> Heap Configuration
41 ; <o> Heap Size (in Bytes) <0x0-0xFFFFFFFF:8>
42 ; </h>
43
44 Heap_Size        EQU      0x00000000
45
46                  AREA     HEAP, NOINIT, READWRITE, ALIGN=3
47 __heap_base
48 Heap_Mem         SPACE    Heap_Size
49 __heap_limit
50
51
52 PRESERVE8
53 THUMB
54
55
56 ; Vector Table Mapped to Address 0 at Reset
57
58                  AREA     RESET, DATA, READONLY
59 EXPORT __Vectors
60 EXPORT __Vectors_End
61 EXPORT __Vectors_Size
62
63 ; Reset Handler
64
65 Reset_Handler    PROC
66                  EXPORT Reset_Handler [WEAK]
67                  IMPORT SystemInit
68                  IMPORT __main
69                  LDR     R0, =SystemInit
70                  BLX     R0
71                  LDR     R0, =__main
72                  BX      R0
73                  ENDP

```

### 4. How to Write and Compile

**Lab1b:** Take either option a) or b). For your exercise, I recommend you to take option a).

- a. Overwrite startup\_TM4C123.s:
  - Delete all statements.
  - Write the 1<sup>st</sup> assembly program on 4.ARMAsembler.pptx' slide p9
- b. Or, download First\_Prog.zip and extract it. Thereafter, open the First\_Prog folder and click Firt\_Prog.uvprojx to have Keil uVersion open this program.

```

StackSize      THUMB                ; Marks the THUMB mode of operation
               EQU      0x00000100  ; Define stack size of 256 bytes
               AREA     STACK, NOINIT, READWRITE, ALIGN=3
MyStackMem     SPACE      StackSize
               AREA     RESET, READONLY
               EXPORT   __Vectors
__Vectors
               DCD      MyStackMem + StackSize ; stack pointer for empty stack: 0x2000.0100
               DCD      Reset_Handler ; reset vector 0x0000.0008-0009
               AREA     MYCODE, CODE, READONLY
               ENTRY
               EXPORT   Reset_Handler

Reset_Handler

               MOV      R0, #0        ; initialize value of sum
               MOV      R1, #2        ; First even number
               MOV      R2, #5        ; Counter for the loop iterations

lbegin
               CBZ      R2, lend      ; Terminate loop if counter is zero
               ADD      R0, R1        ; Build the sum
               ADD      R1, #2        ; Generate next even number
               SUB      R2, #1        ; Decrement the number
               B         lbegin

lend
               B         lend
               END

```

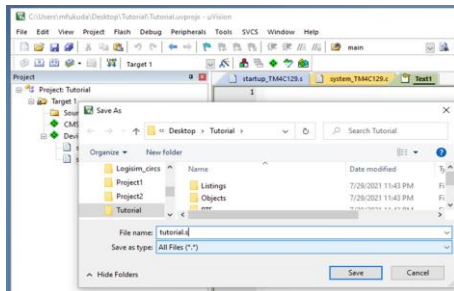
**Final Project:** you need to create multiple files with .s or .c attributes.

1. (a) Click “File” and choose “New” that creates a new file initially named “Text”. Immediately save the file by renaming it with .s or .c and adding it under Target 1/Source Group 1. (b) Right click on Target 1/Source Group 1, choose “Add existing files to Group”, and (c) specify the file name to add it to the group.

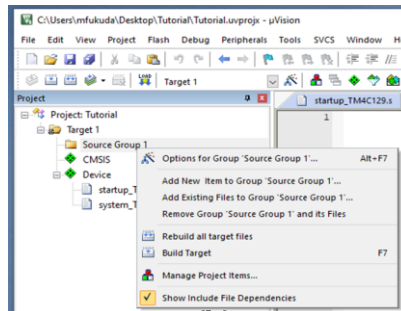
Or

2. Right click on Target 1/Source Group 1, click on “Add New Item to Group” and elect the .s or .c, write its name in given space, and click “Add”.

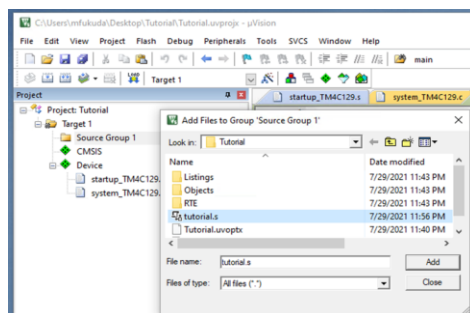
Write your code. Then save all files.



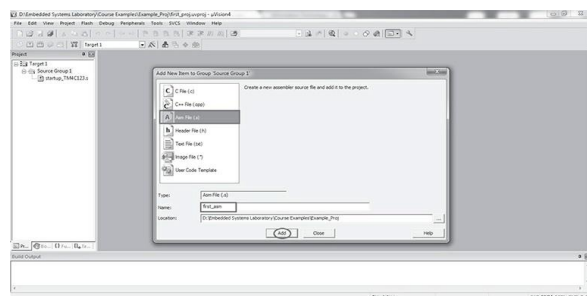
1-(a)



1-(b)



1-(c)



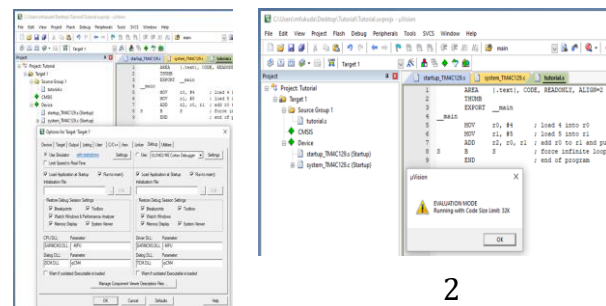
2

In both Lab1b and the final project, you need to build an executable at the end: Choose “Project” and “Build Target”.

```
Build Output
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Program Files\Keil\ARM\ARMCC\Bin'
Build target 'Target 1'
assembling tutorial.s...
assembling startup_TM4C129.s...
compiling system_TM4C129.c...
linking...
Program Size: Code=356 RO-data=512 RW-data=4 ZI-data=516
".\Objects\tutorial.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:01
```

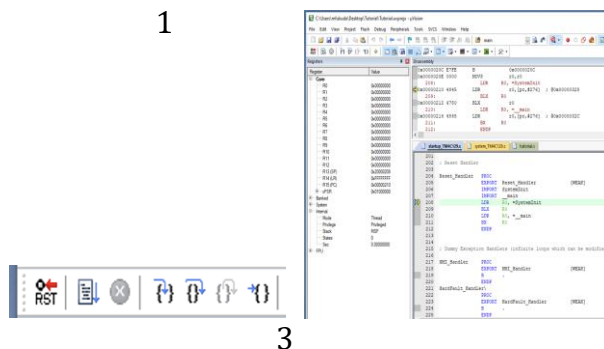
## 5. How to Run Your Program

1. Choose “Project” and “Option for Target”.  
Click the “Debug” menu and “Use Simulator”
2. Choose “Debug” and “Start/Stop Debugging Session”, which then shows a warning message. Simply, click “OK” to start a debugging session.
3. Use the simulation control buttons.
  - Reset
  - Run
  - Stop
  - Step one line
  - Step over the current line
  - Step out of the current function
  - Run to the current cursor line
4. You can use “insert/remove breakout points”.  
Simply click the left side of the assembly code line#.

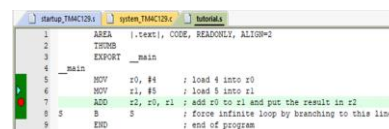


1

2



3



4

### What to submit

1. Your source code named startup\_TM4C123.s
2. A snapshot of Keil uVersion’s registers, disassembly, and startup\_TM4C123.s windows when your program reached “B lend”.