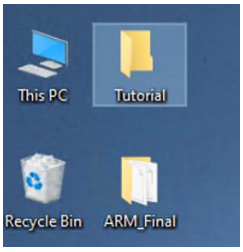
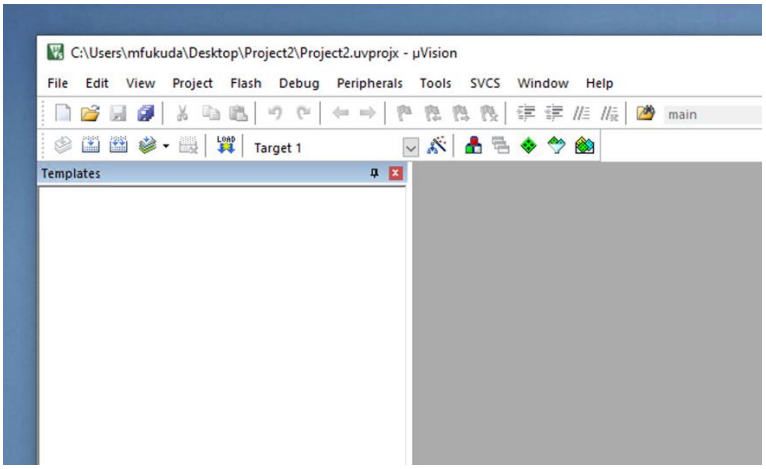
Keil uVersion Programming

1. How to Set up Your Project

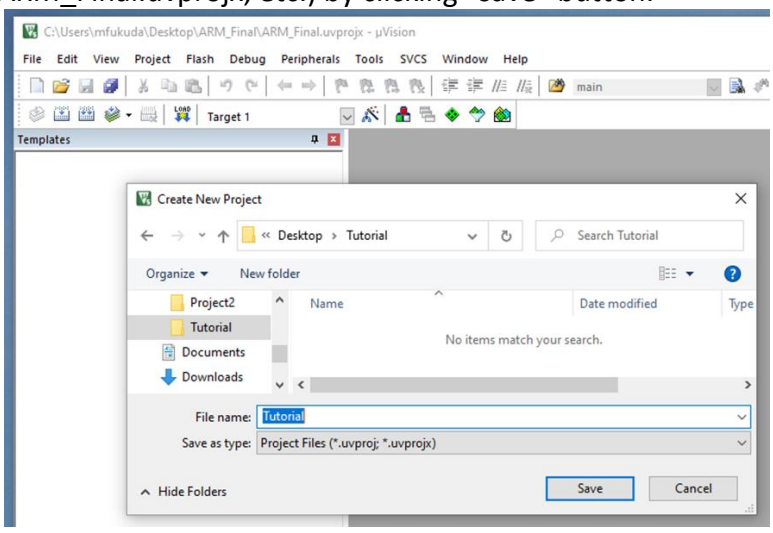
1.1. Create a new folder on the desktop, (e.g. Tutorial or ARM_Final)



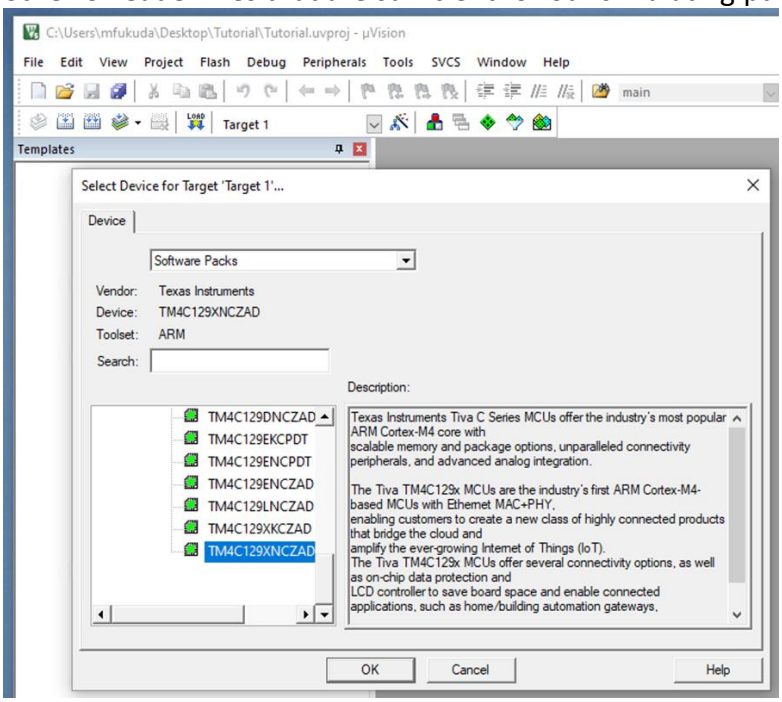
1.2. Start Keil uVersion.



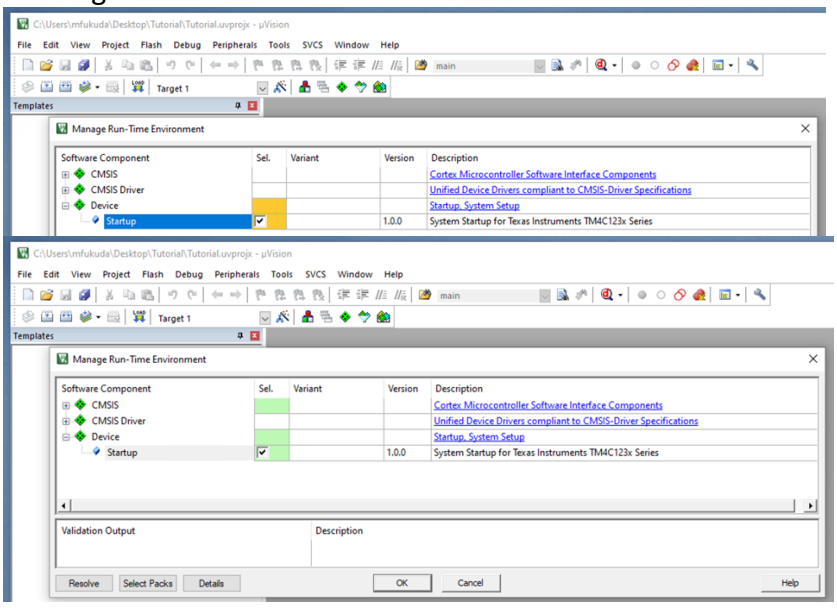
1.3. Click “Project” and choose “new uVersion project”. Choose the new folder, (e.g., Tutorial or ARM_Final) Thereafter, create ProjectName.uvprojx file, (e.g., Tutorial.uvprojx or ARM_Final.uvprojx, etc.) by clicking “save” button.



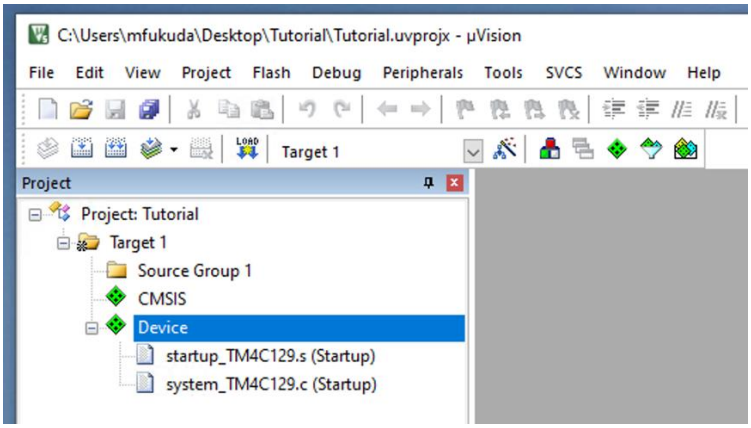
1.4. The “Select Device” menu will pop out. Choose Texas Instruments and TM4C129XNCZAD. Note that UWB EE uses TM4C123GH6PM as their real device. But CSS422 runs ARM assembly code only in simulation and TM4C129XNCZAD gives a simple SystemInit C function and the other C header files that are sufficient for our simulating purposes.



1.5. The “Manage Run-Time Environment” menu will pop out. If not, click “Project” and choose “Manage” → “Run-Time Environment”. There check out Device “Startup” and click “resolve”.



1.6. All programming environments are set up now.



2. How Your Program Is Invoked

2.1. The program execution starts with startup_TM4C129.s' line 64 that is the 1st entry of the interrupt vector table. This entry points to the address of Reset_Handler. The control jumps to Reset_handler (on line 204) and calls SystemInit( ) on line 209.

```
 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  __Vectors       DCD     __initial_sp              ; Top of Stack
 64                  DCD     Reset_Handler             ; Reset Handler
202  ; Reset Handler
203
204  Reset_Handler   PROC
205                  EXPORT  Reset_Handler             [WEAK]
206                  IMPORT  SystemInit
207                  IMPORT  __main
208                  LDR     R0, =SystemInit
209                  BLX     R0
210                  LDR     R0, =__main
211                  BX      R0
212                  ENDP
213
```

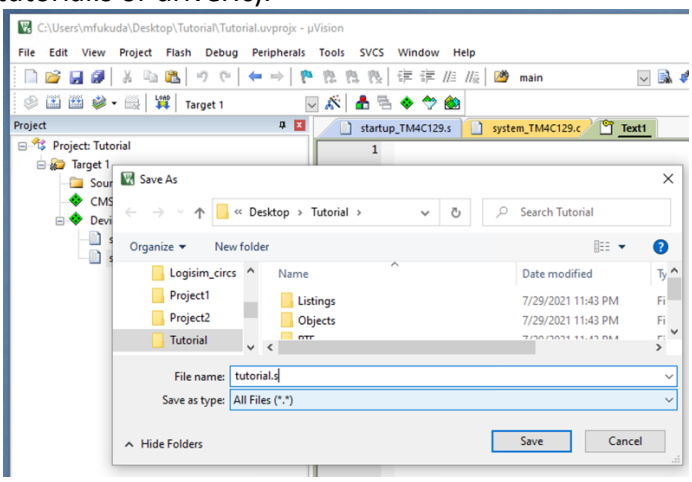
2.2. SystemInit( ) is coded in system_TMC4C129.c in lines 47-61 as follows. It sets up the system clock and returns back to startup_TM4C123.s' assembly routine on line210. The execution finally calls __main on line 211, which is your program, e.g. tutorial.s' __main or driver.c's main( ).

```
47  void SystemInit (void)
48  {
49    /* TODO: Updates required to fully work with TM4C129 series devices */
50    #if (__FPU_USED == 1)
51      SCB->CPACR |= ((3UL << 10*2) |                 /* set CP10 Full Access */
52                     (3UL << 11*2)  );               /* set CP11 Full Access */
53    #endif
54
55  #ifdef UNALIGNED_SUPPORT_DISABLE
56    SCB->CCR |= SCB_CCR_UNALIGN_TRP_Msk;
57  #endif
58
59    SystemCoreClock = __SYSTEM_CLOCK;
60
61  }
62
```

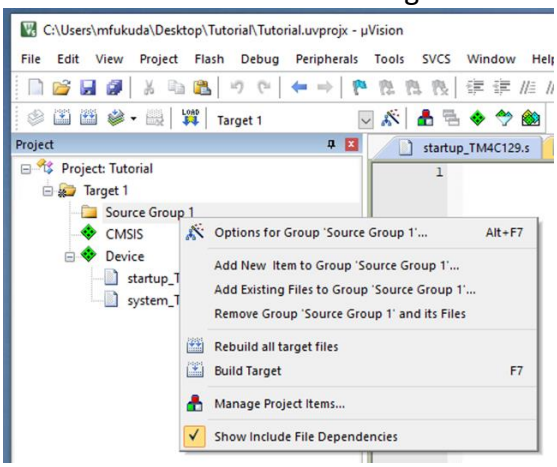
3. How to Write and Compile a New Program

3. 1. To write a new program, click "File" and choose "New" that creates a new file initially named "Text". Before coding any program, rename, save, and add this file under Target 1/Source Group1.

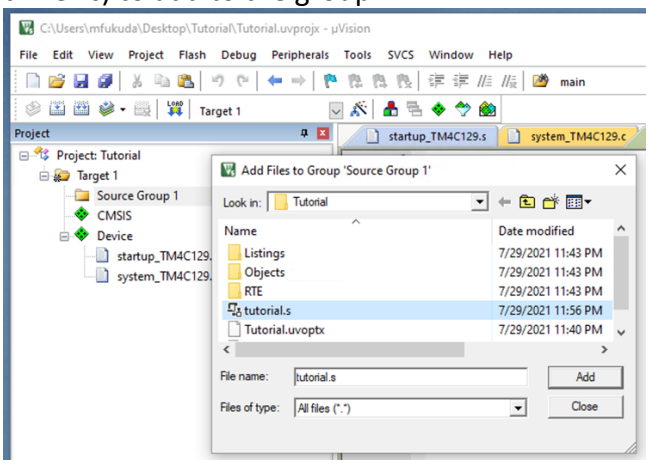
(a) Choose "File" and "Save As", and thereafter specify a new file name to save, (e.g., tutorial.s or driver.c).



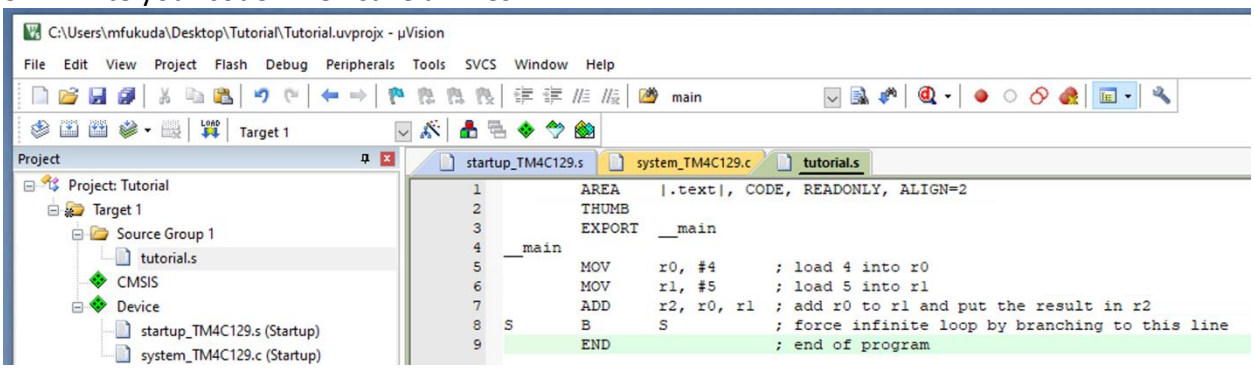
(b) Move the mouse cursor to Target1/Source Group 1 and click the right button to open the menu. Choose “Add Existing Files to Group”.



(c) The “Add Files to Group” menu pops out. There, specify the file name, (e.g., tutorial.s or driver.c) to add to the group.



3.2. Write your code. Then save all files.

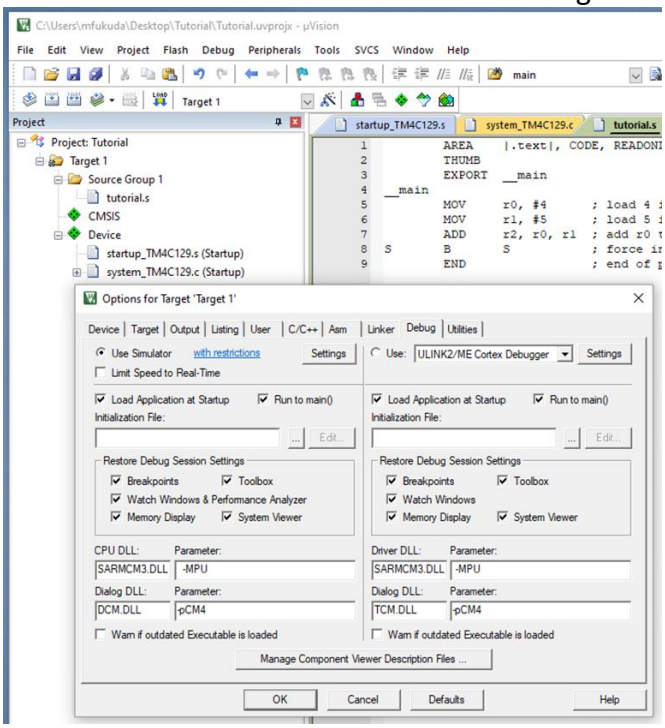


3.3. To build an executable, choose “Project” and “Build Target”. If you didn’t receive any error messages, the compilation went in success.

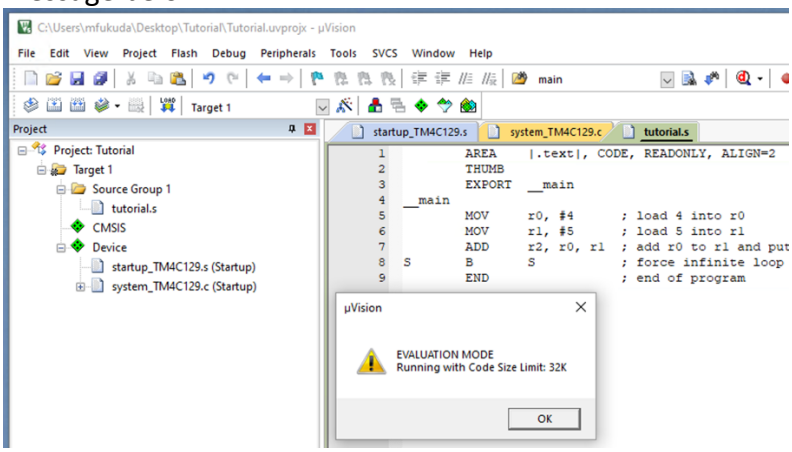
```
Build Output
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Program Files\Keil\ARM\ARMCC\Bin'
Build target 'Target 1'
assembling tutorial.s...
assembling startup_TM4C129.s...
compiling system_TM4C129.c...
linking...
Program Size: Code=356 RO-data=512 RW-data=4 ZI-data=516
".\Objects\Tutorial.axf"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

4. How to Run Your Program

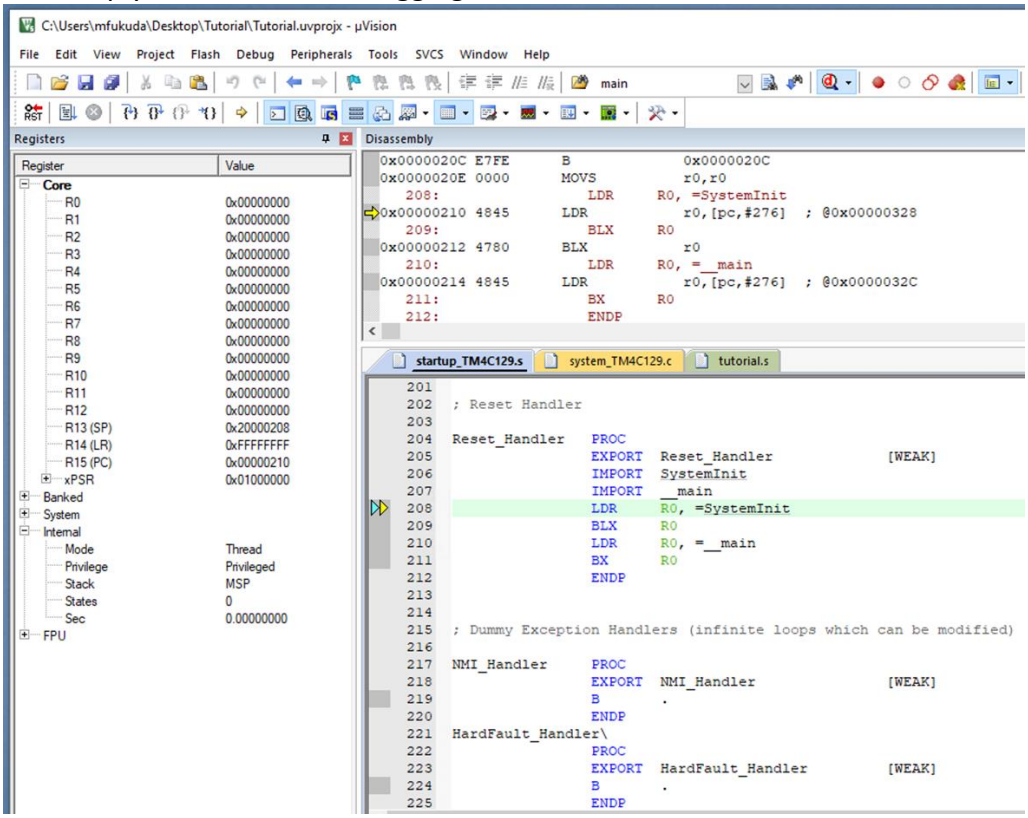
4.1. Before starting a debugging simulation, choose “Project” and “Options for Target”. You’ll see the menu shown below. Click the “Debug” menu and “Use Simulator”.



4.2. Then, choose “Debug” and “Start/Stop Debugging Session”, which then shows a warning message below:



4.3. Simply, click “OK”. A debugging session starts:

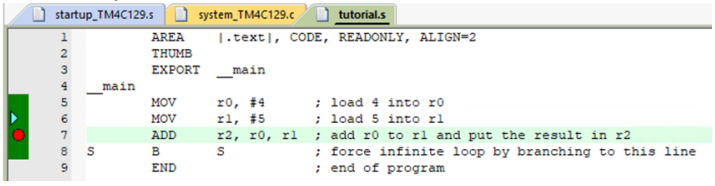


4.4. Use the following buttons to control the simulation:

These buttons from left to right intend to:

Reset
Run
Stop
Step one line
Step over the current line
Step out of the current function
Run to the current cursor line

4.5. For assembly code tracing, make sure that the current cursor line is the top of the code before clicking “step one”, “step over”, “step out”, or “run to”. Or, you may use “insert/remove breakout points”. Simply click the left side of the assembly code line# to see a red circle (a breakout). To delete it, click the red circle.



4.6. To stop the debugging mode, choose “Debug” and “Start/Stop Debugging Session”.