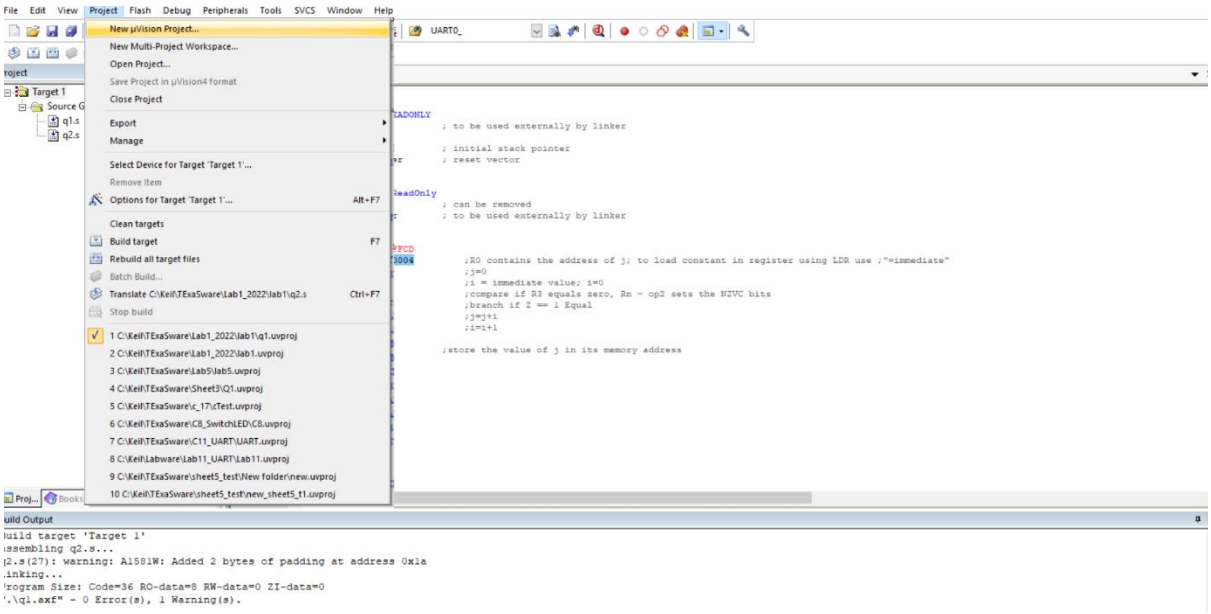
Lab 1

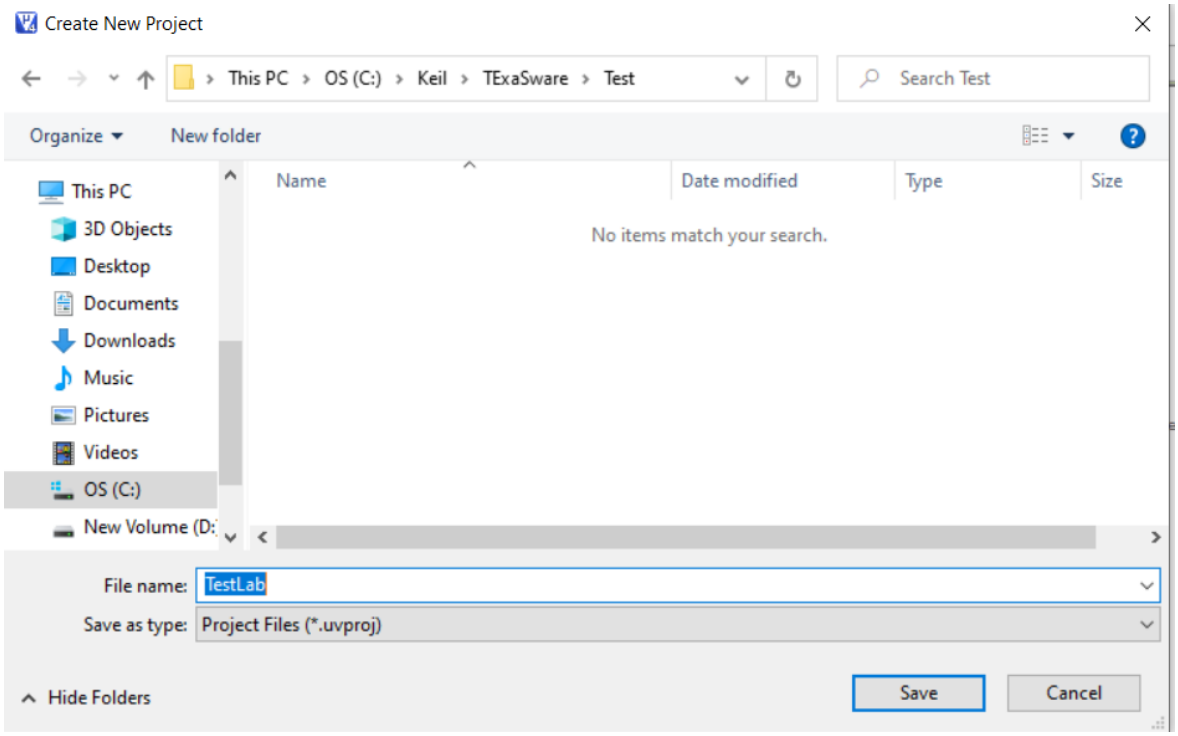
Introduction

To use the simulated behavior of Keil:

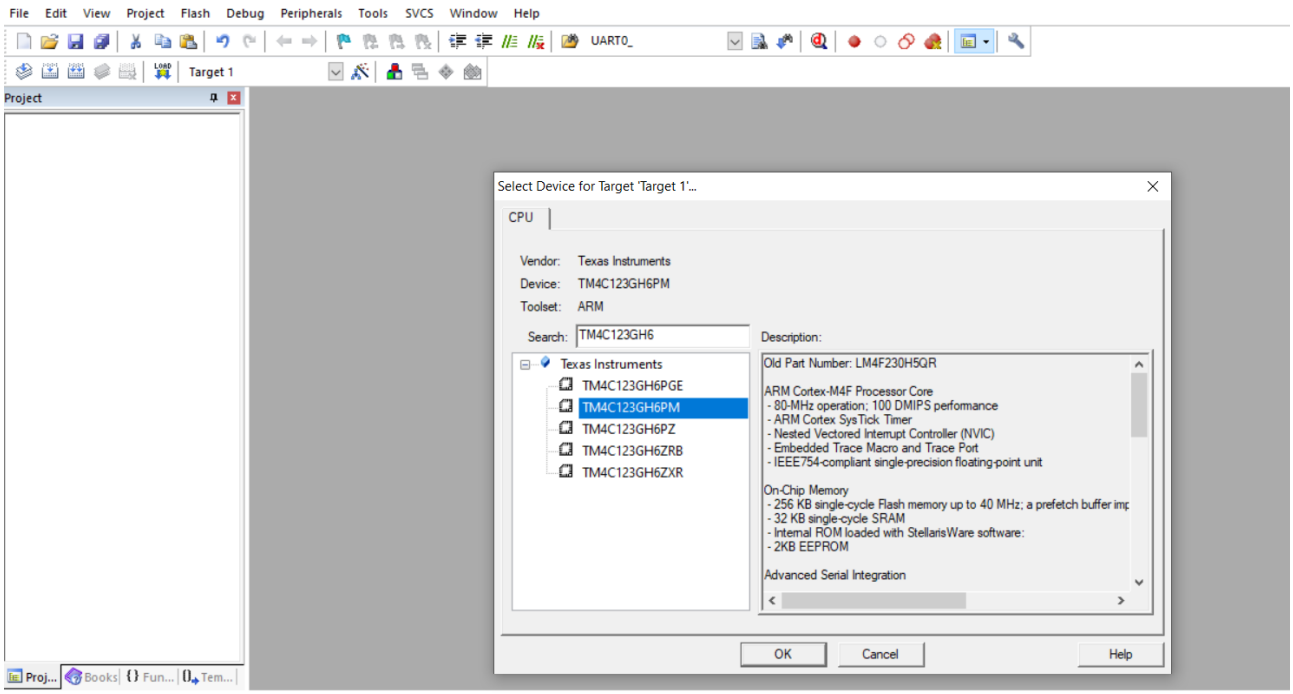
1. Create new project.
    a. By selecting “New uVision project” from project tab in the tool bar of the IDE.



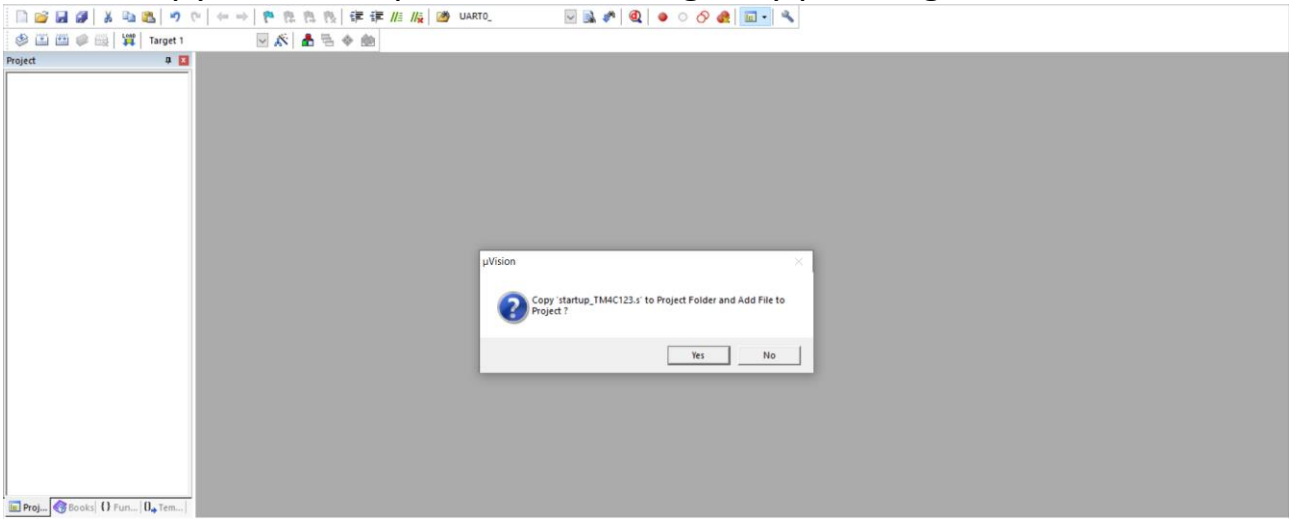
b. Choose the location of the project and rename it.



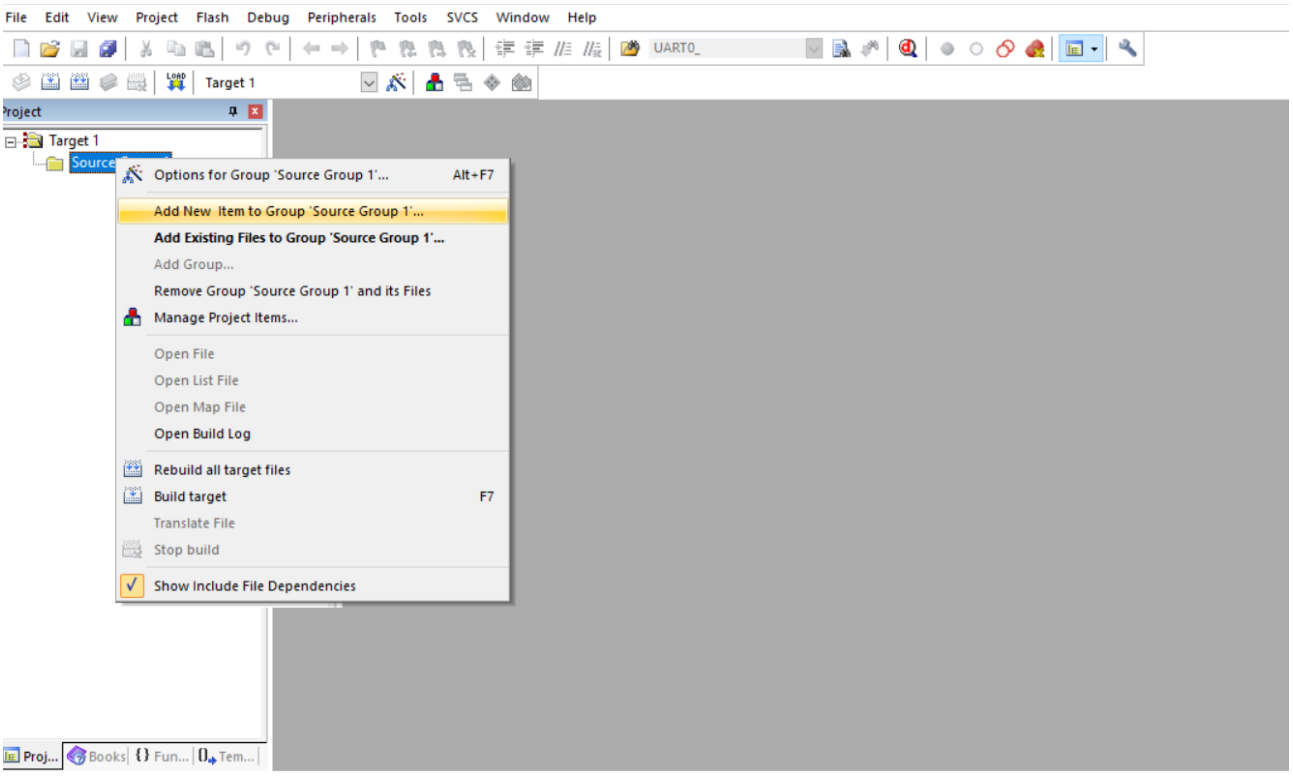
2. Choose the target TM4C123GH6PM device.



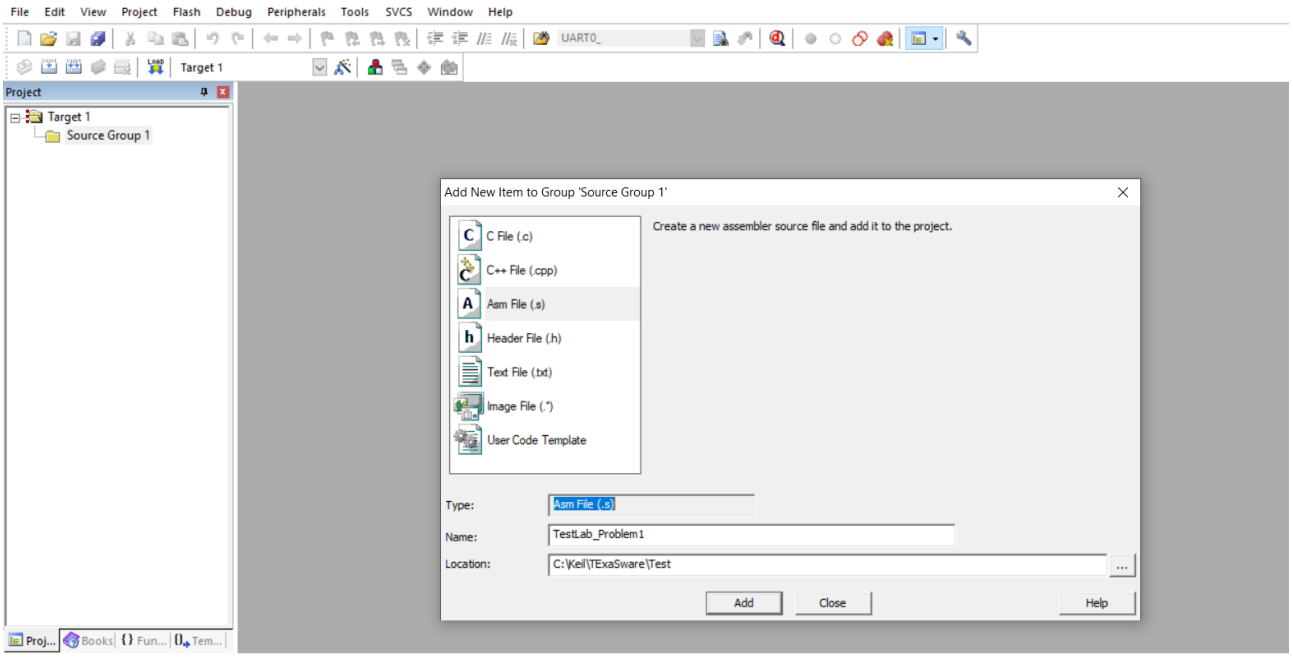
3. Do not copy the startup code of the target by pressing “No”.



4. Create a new assembly file by right click on “Source Group 1” and choose “Add New Item to Group…. ”.



5. Choose assembly file and rename your assembly file.



6. Paste the below startup code in the created assembly file.

```
                AREA RESET, DATA, READONLY
                EXPORT  __Vectors

__Vectors

                DCD     0x20008000
                DCD     Reset_Handler
                ALIGN

                AREA myCode, CODE, ReadOnly
                ENTRY
                EXPORT Reset_Handler

        Reset_Handler
```

Assembler Directives

AREA directive tells the assembler to define a new section of memory (SRAM or ROM).

CODE: contains machine instructions/const. data (R)

DATA: no instructions allowed here

READONLY: placed in ROM, for CODE by default

READWRITE: placed in SRAM for variables

ENTRY indicates to the assembler the beginning of the executable code

END indicates to the assembler the end of the source (asm) file

ALIGN {2} ensures the next instruction is 32-bit {16-bit} aligned

EQU defines a constant value or a fixed address. It does not set aside storage for a data item, but associates a constant number with a data or an address label

DCB, DCW, DCD allocate **aligned** byte, half-word (16-bit), word (32-bit) memory locations

SPACE is used for uninitialized data

Lab Exercise

Q1. Write an assembly code that performs the same functionality as the following C code:

```
int sum;
int n=20;
sum = 0;
while (n>0) {
   sum = sum + n;
   n = n -1;
 }
```

where address of n begins at 0x20000000, and address of sum begins at 0x20000004.

Lab Submission

Q2. Write assembly code that performs the below C code:

```
int j=0;
int i;
for ( i = 0 ; i < 15 ; i++){

        j = j + i;

}
```

where address of j begins at 0x20000004.

For the lab submission, you should submit a pdf document contains the following.

1. Cover page that contains
    a. your name,
    b. your ID,
    c. your department.
2. Place snapshots to show general register values and the special register values from the debugging window.
3. Place your code in the document.
4. Your document will be submitted on LMS.