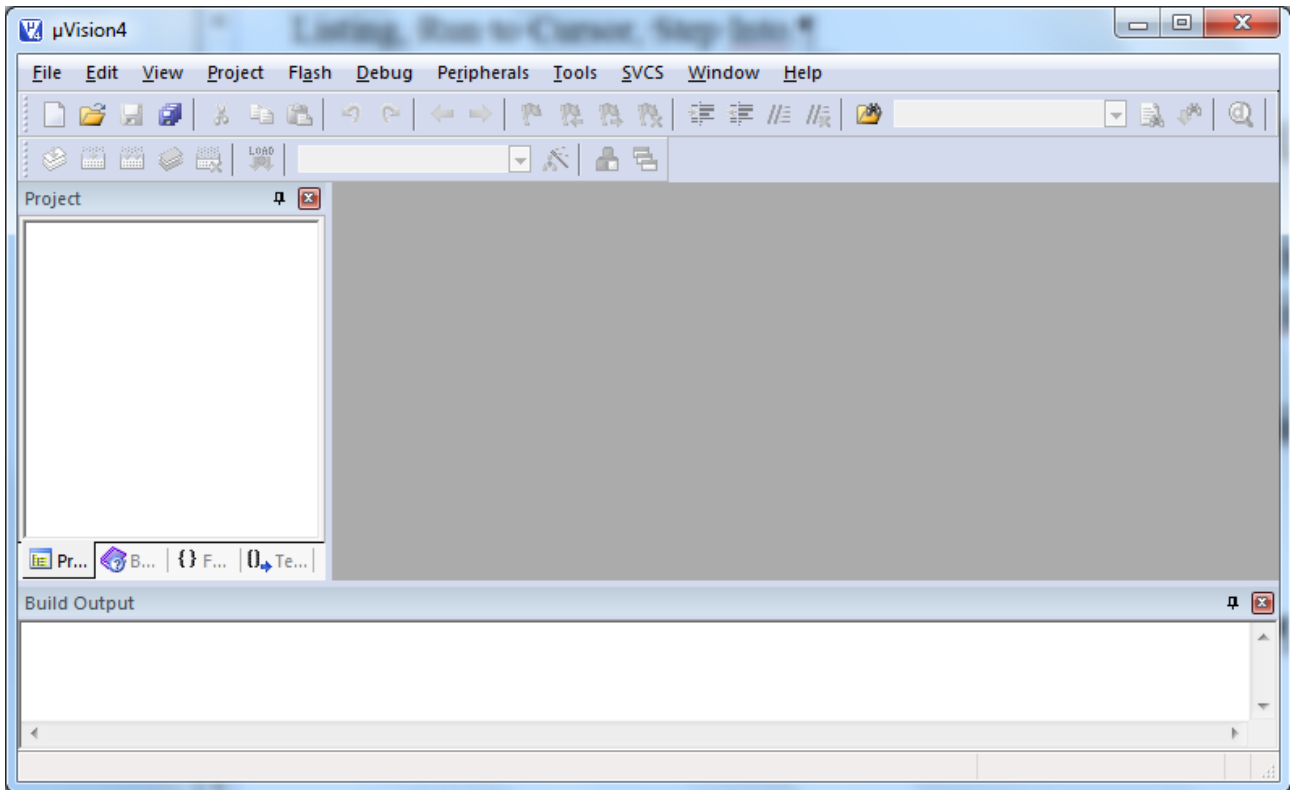


Додаток А

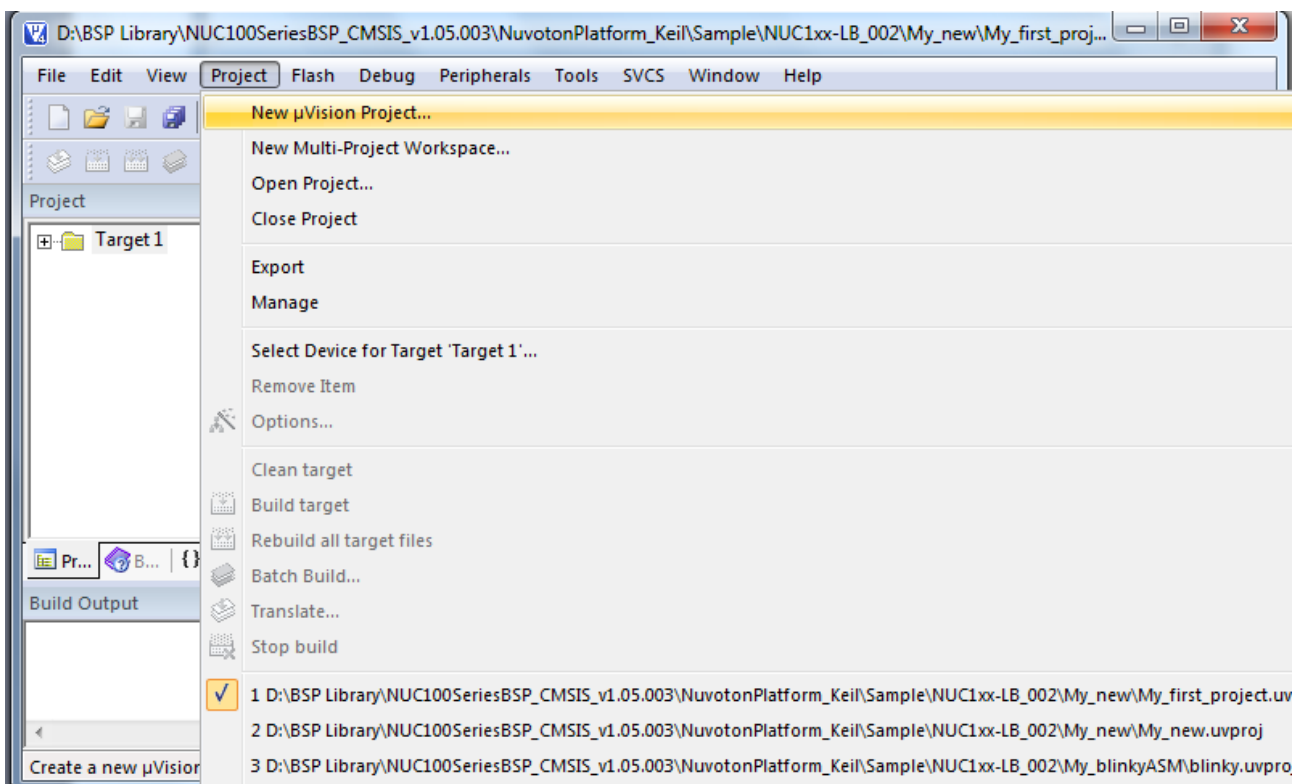
Створення проекту у середовищі μ Vision4 Keil MDK

1. Загальний вигляд та інтерфейс μ Vision4

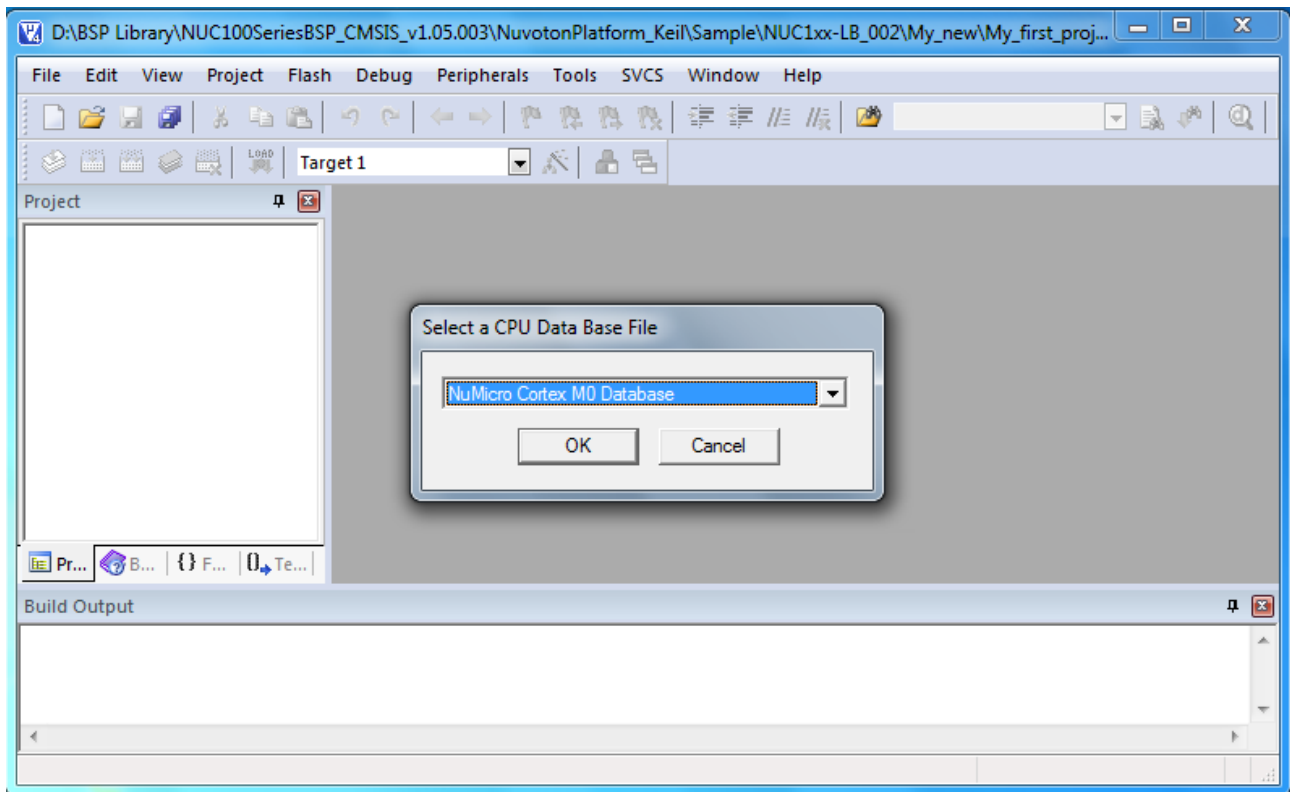


2. Створення нового проекту.

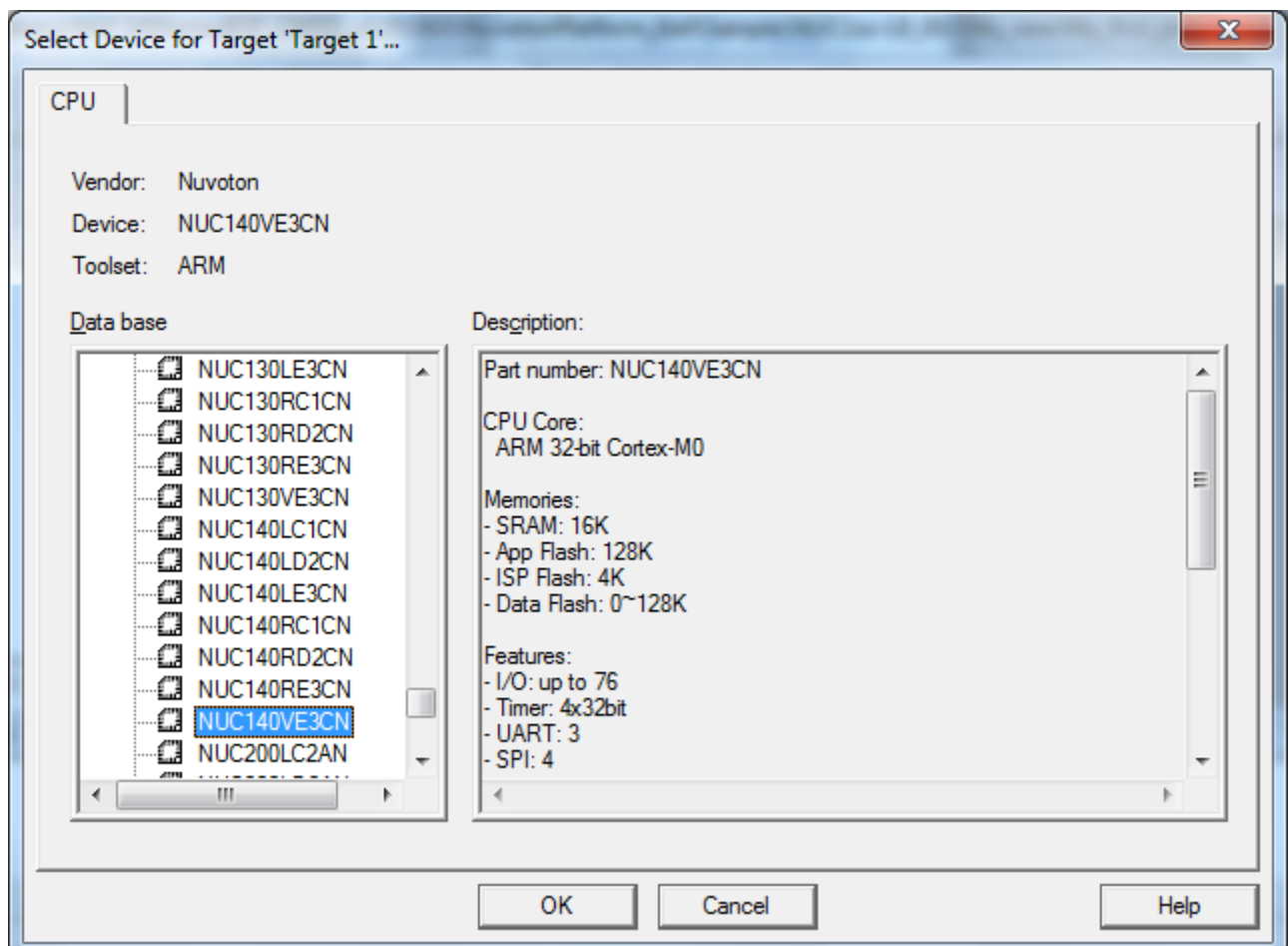
У меню Project слід натиснути “New μ Vision Project”



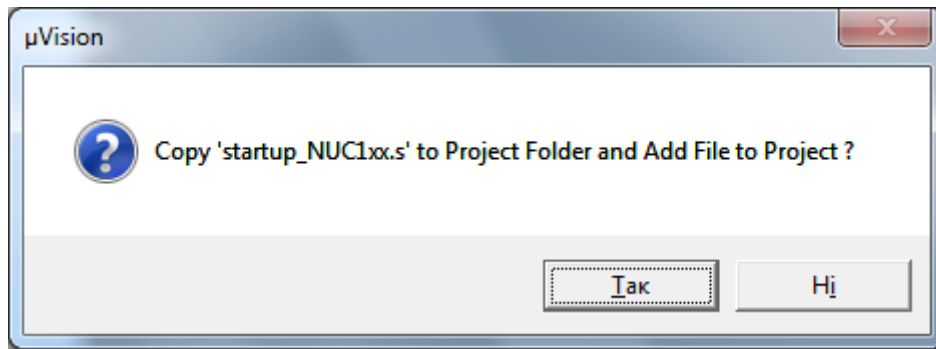
3. У вікні списку процесорів слід обрати фільтр NuMicro Cortex M0 Database.



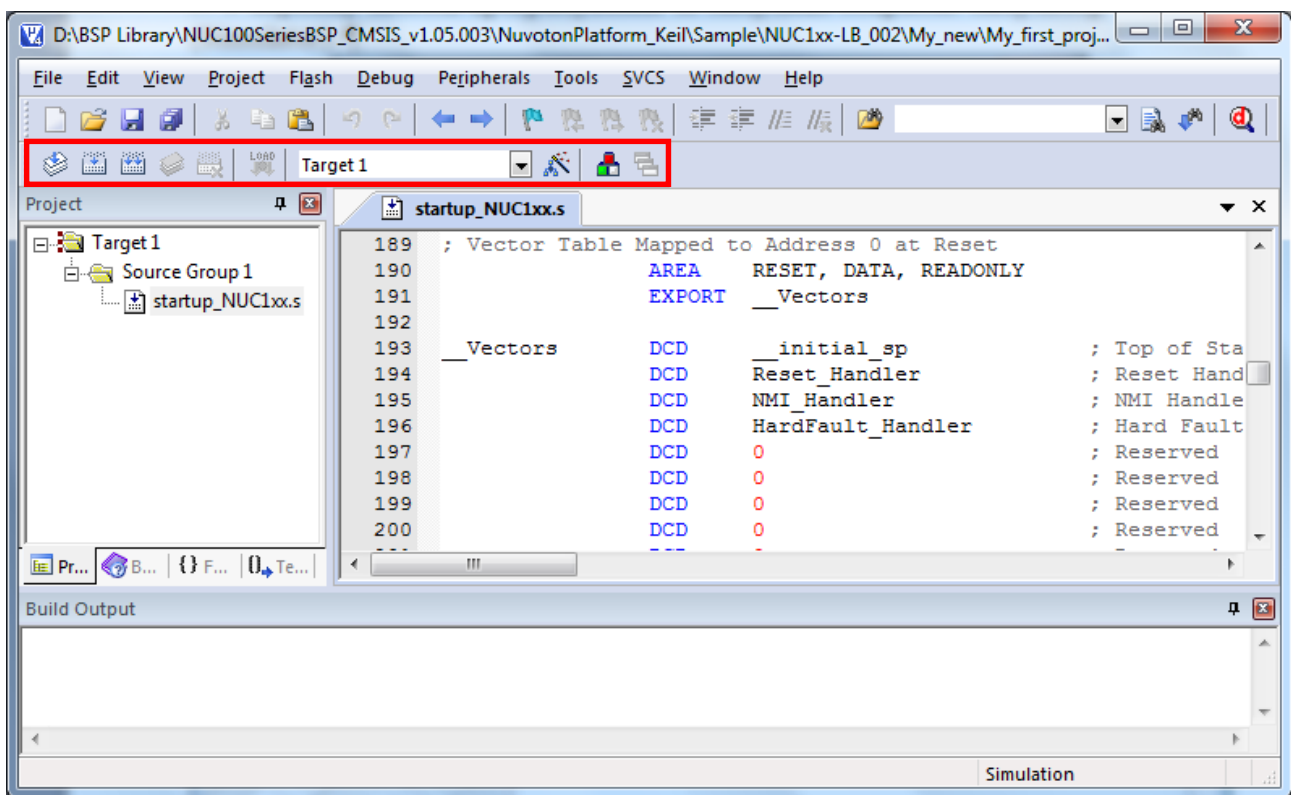
4. Обираємо назву мікроконтролера NUC140VE4CN, що використовується на навчальній платі NU LB NUC140.




5. Далі, майстер створення проекту запропонує скопіювати файл шаблону startup_NUC1xx.s програми. На пропозицію слід погодитися.



6. Опісля, попереднього кроку проект вже має бути створений.



Далі можна побудувати проект. Для цього у меню Project слід натиснути “Build Project”, або скористатися швидким викликом функції «F7» або відповідною піктограмою  на панелі керування .

Проте під час першої компіляції виникне помилка:

```
Error: L6411E: No compatible library exists with a definition of
startup symbol __main.
Target not created
```

Ця помилка свідчить про те, що з startup_NUC1xx.s викликається підпрограма __main , яка у цьому файлі відсутня. Тому створимо її в окремому файлі main.s:

```

INCLUDE reg_def.s
AREA    |.text|, CODE, READONLY

__main
        PROC
        EXPORT  __main                [WEAK]

        BX LR
        ENDP

END

```

Окрім підпрограми `__main` також викликається підпрограма `SystemInit`. Створимо також окремий файл `SystemInit.s`

```

INCLUDE reg_def.s
AREA    |.text|, CODE, READONLY
SystemInit  PROC
        EXPORT  SystemInit          [WEAK]

        BX LR
        ENDP


END

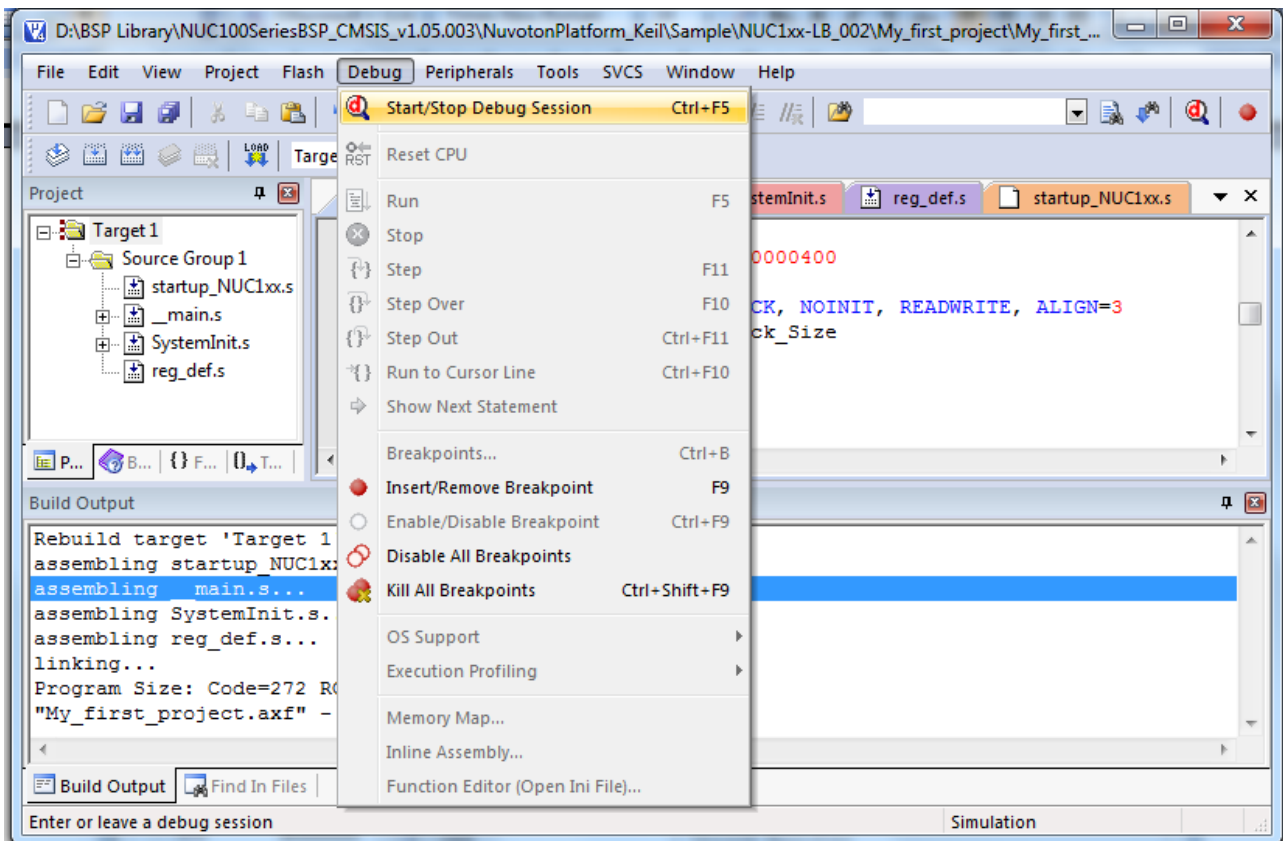
```

Опісля створення цих файлів і включення їх до проекту, побудова проекту має бути успішною.

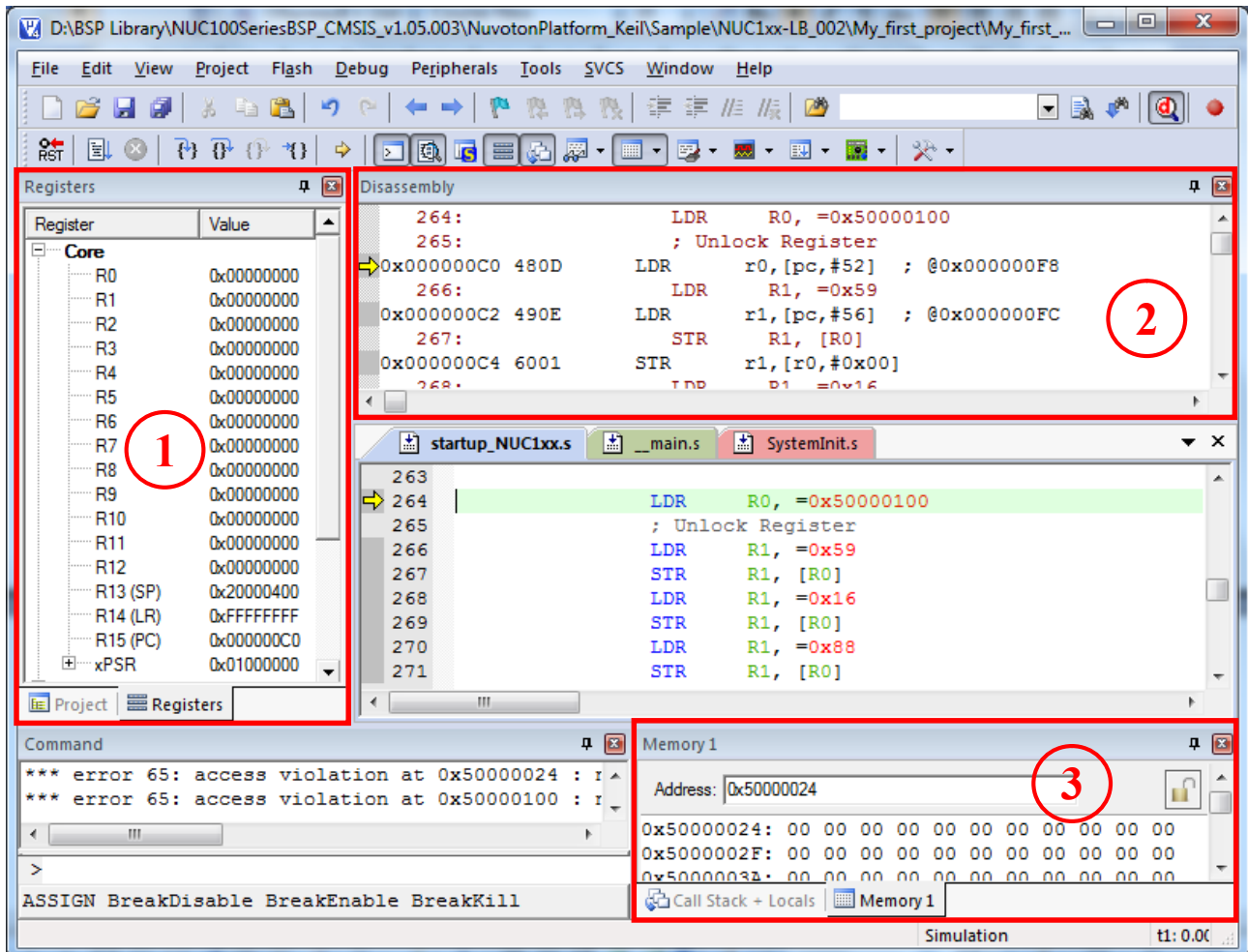
Program Size: Code=100 RO-data=192 RW-data=0 ZI-data=1024

"My_first_project.axf" - 0 Error(s), 0 Warning(s).


7. Перейти до симуляції проекту можливо через меню `Debug->Start/Stop Debug Session`, швидким викликом функції «`Cntr+F5`» або відповідною піктограмою  на панелі керування.





8. У режимі ладнання програми доступні нові вікна, зокрема вікно Registers (область 1), у якому можливо переглядати поточні значення в робочих регістрах R0-R15 та xPSR ; вікно дизасемльованого варіанту програми Disassembly (область 2) ; вікно Memory (область 3), у якому можливо переглядати вміст регістрів пам'яті за адресами.





Виконаємо симуляцію коду в покроковому режимі, використовуючи комбінації «гарячих кнопок» або піктограми на панелі керування:


Reset (F9) – скидання  ;

Run (F5) – виконання програми  ;

Step (F11) – покрокове виконання програми  ;


Step Over (F10) – «перейти через», пройти швидко цикл програми  ;

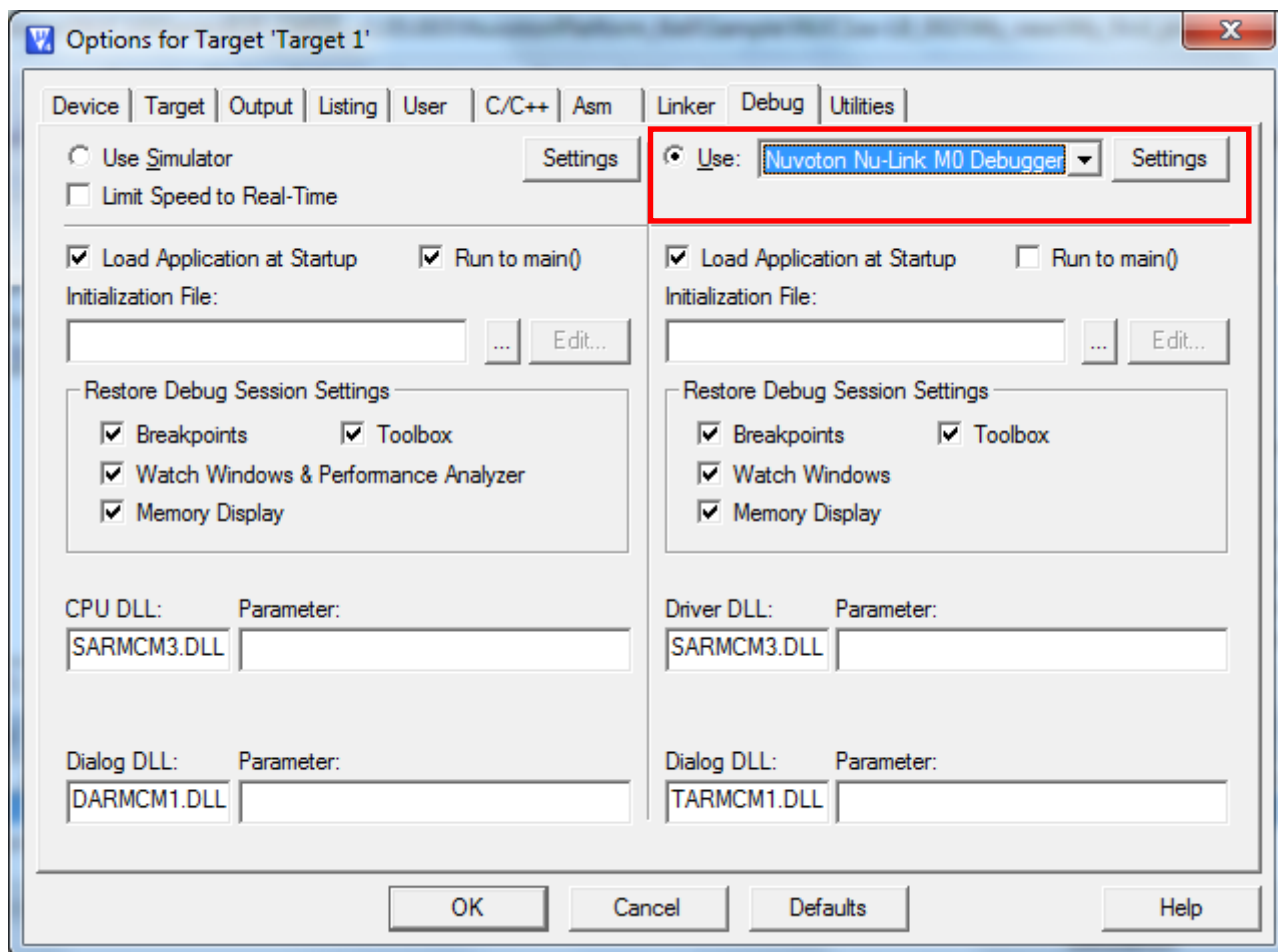
Step Out (Cntr+F11) – швидкий вихід з циклу  ;

Run to Cursor Line (Cntr+F10) – швидке виконання програми до курсору  ;

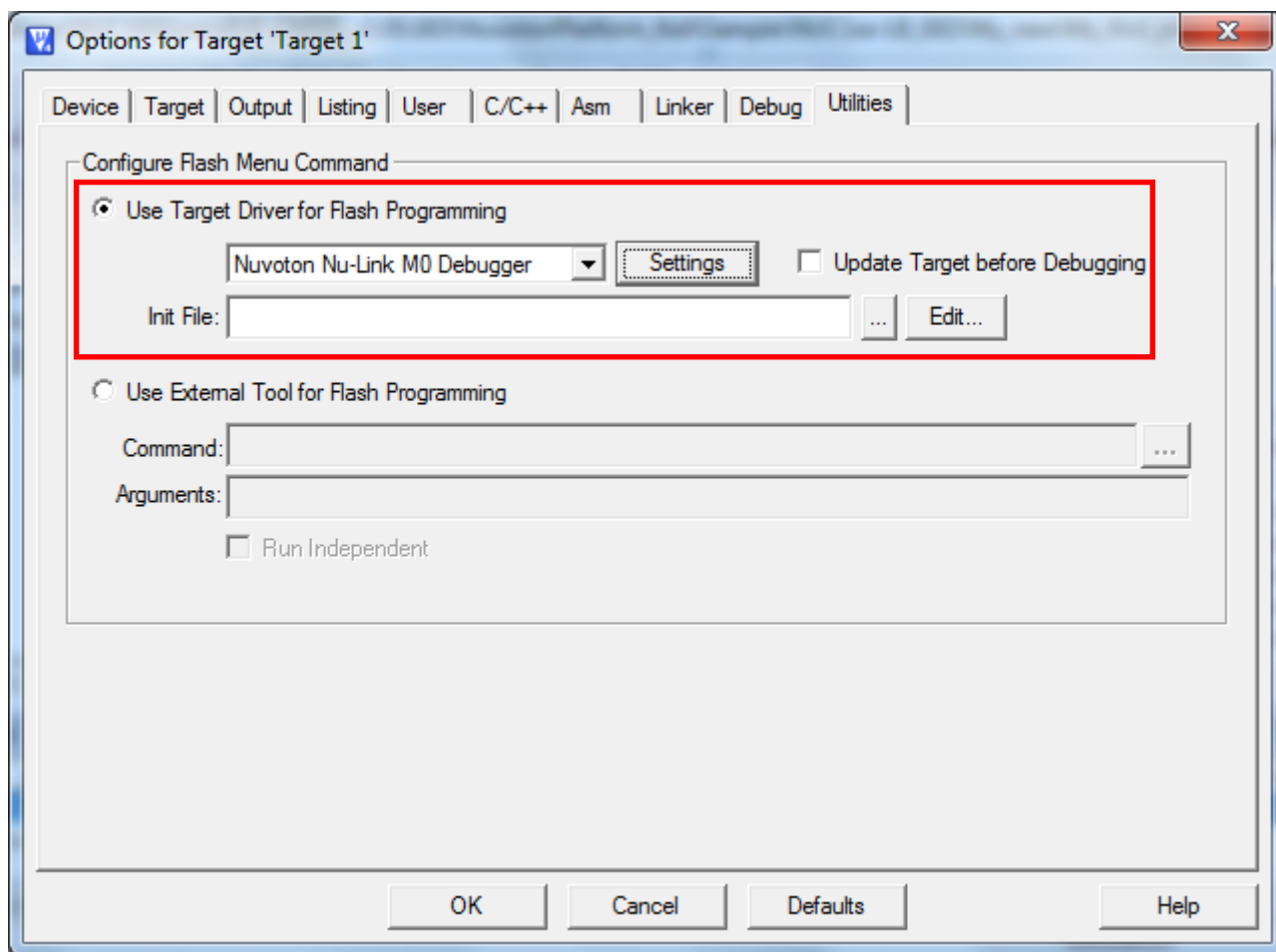
9. У полі адреса слід ввести адресу у форматі, наприклад, 0x50000100, відстежити як змінюються вміст регістрів налаштувань під час виконання програми.


10. Зверніть увагу на те, що в режимі симуляції μ Vision може відображати лише вміст регістрів процесора і не підтримує емуляцію регістрів налаштувань периферійних пристроїв. Тому для повноцінного ладнання програми слід скористатися вбудованою в процесор ARMv6-M системою ладнання, а для цього необхідно налаштувати програматор **Nuvoton Nu-Link**.

Для цього необхідно зайти в меню налаштувань проекту Project->Options for Target (Alt+F7) . В закладці Debug слід обрати зі списку Nuvoton Nu-Link M0 Debugger.



11. В закладці Utilities слід вказати «Use Target Driver for Flash Programming» та обрати зі списку Nuvoton Nu-Link M0 Debugger.



12. Виконуємо програмування мікроконтролера обравши з меню Flash->Download, або скориставшись піктограмою . Після завершення програмування можна повторити процес ладнання програми безпосередньо на мікроконтролері.