어셈블리프로그램 설계 및 실습

Lab #1

MDK-ARM Setup & Basic Example

Kwangwoon University
Embedded System Architecture Lab

Course Outline

주차	강의내용	실습내용
1	Introduction to assembly programming (Reading - Binary number system from 디지털논리회로 I) MU0 processor (1.3)	
2	ARM processors and programmer's model (2.3 & 부교재 Ch.3) Introduction to ARM assembly program (3.4 & 부교재 4.4)	ARM software development tool 설치 Hello world example (tool 사용법 포함)
3	ARM instruction set – introduction (5.1-5.3 & 부교재 4.1) ARM instruction set – Single data transfer (3.2 & 5.10- 5.11)	Conditional execution Data transfer from/to memories
4	추석	
5	ARM instruction set – control flow & data processing (3.3 & 5.4-5.5)	Loop examples Brach vs conditional exec.
6	ARM instruction set – second operand & multiplication (3.1 & 5.7-5.8 & 부교재 4.4)	Factorial
7	ARM instruction set – Block data transfer & stacks (5.12 & 부교재 4.23)	Subroutine call
8	중간고사	
9	Floating point number & addition	Floating point addition/multiplication
10	ARM instruction set – Pseudo instructions	Character array processing
11	Multiply by a constant (부교재 5.3) Booth multiplication (참고) ARM Assembly Programming Performance Issues (부교재 5.9)	프로젝트 소개 & release
12	프로젝트	프로젝트 시작
13	프로젝트	프로젝트 진행
14	프로젝트	프로젝트 결과발표
15	기말고사	

Grade Information

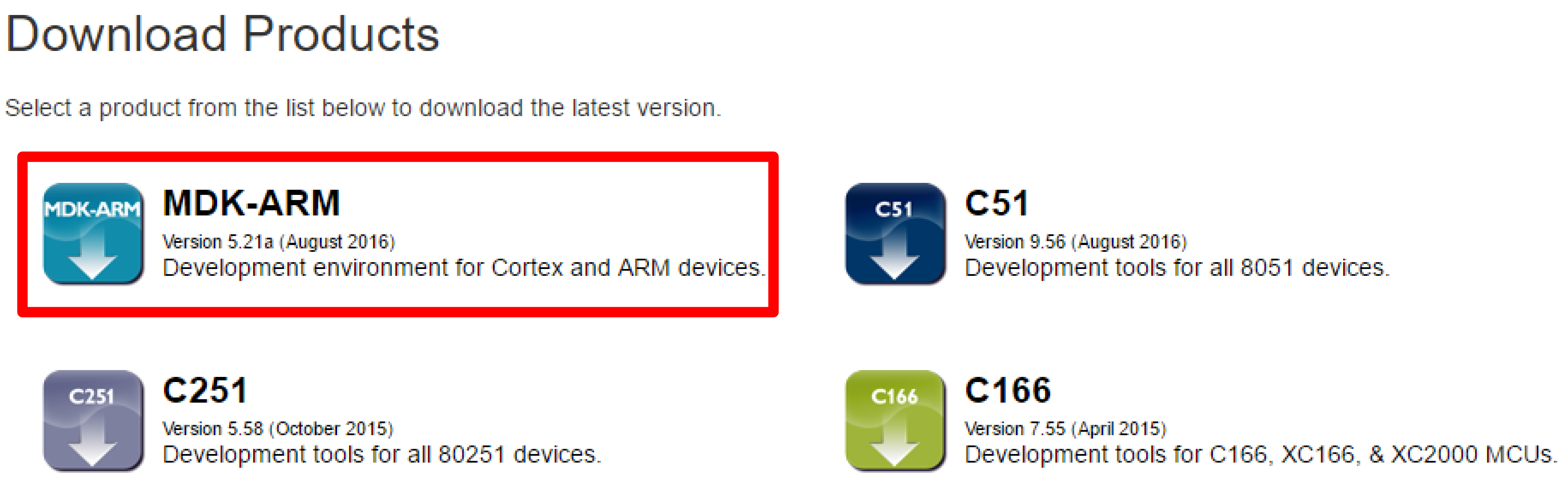
- 중간고사: 20%
- 실습과제: 25%
  - 과제 제출기간 다음 수업전까지 (ex) 9/6 과제는 9/13 16:29까지 제출
  - 딜레이 x
- 프로젝트: 45%
  - 설계과제는 제안서, 결과보고서를 통하여 채점을 수행
  - 기말고사 유무는 추후 공지
- 수업태도: 10%
- 조교 문의사항
  - 조원희 조교(비 201호) - e-mail : jwh6896@kw.ac.kr
  - 조수익 조교(비 201호) - e-mail : azx1593@naver.com

Outline

- 인터넷 강의
  - 이론에 대한 강의
- Q&A
  - 인터넷 강의 중 질문할 사항에 대해서 Q&A시간을 가짐
- 실습
  - 인터넷 강의와 Q&A시간에 배운 점을 활용하여 실습 진행
  - 결과값과 Performance를 확인

Keil User Registration (1/3)

- Connect to MDK homepage
  - https://www.keil.com/download/product/
- Click the link


Download Products

Select a product from the list below to download the latest version.

MDK-ARM
Version 5.21a (August 2016)
Development environment for Cortex and ARM devices.

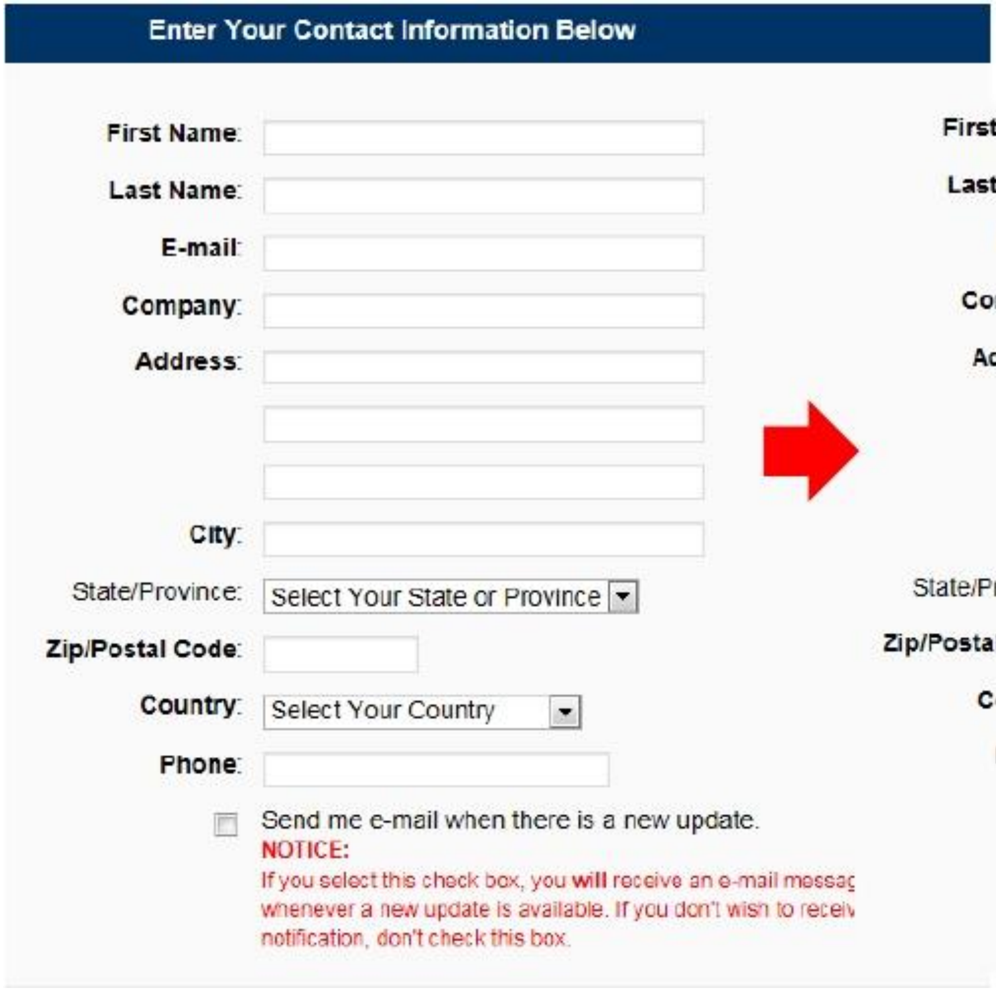
C51
Version 9.56 (August 2016)
Development tools for all 8051 devices.

C251
Version 5.58 (October 2015)
Development tools for all 80251 devices.

C166
Version 7.55 (April 2015)
Development tools for C166, XC166, & XC2000 MCUs.

Keil User Registration (2/3)

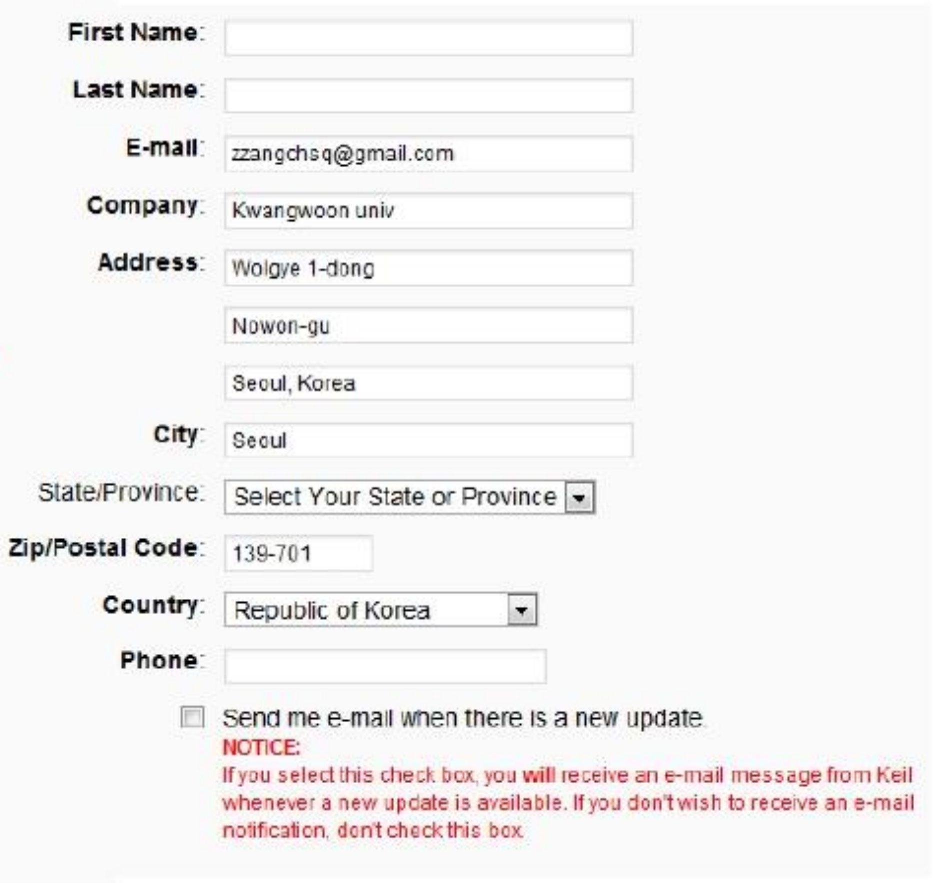
- Fill the form



Enter Your Contact Information Below

First Name:	
Last Name:	
E-mail:	
Company:	
Address:	
City:	
State/Province:	Select Your State or Province
Zip/Postal Code:	
Country:	Select Your Country
Phone:	

☐ Send me e-mail when there is a new update.
NOTICE:
If you select this check box, you **will** receive an e-mail messag
whenever a new update is available. If you don't wish to receiv
notification, don't check this box.

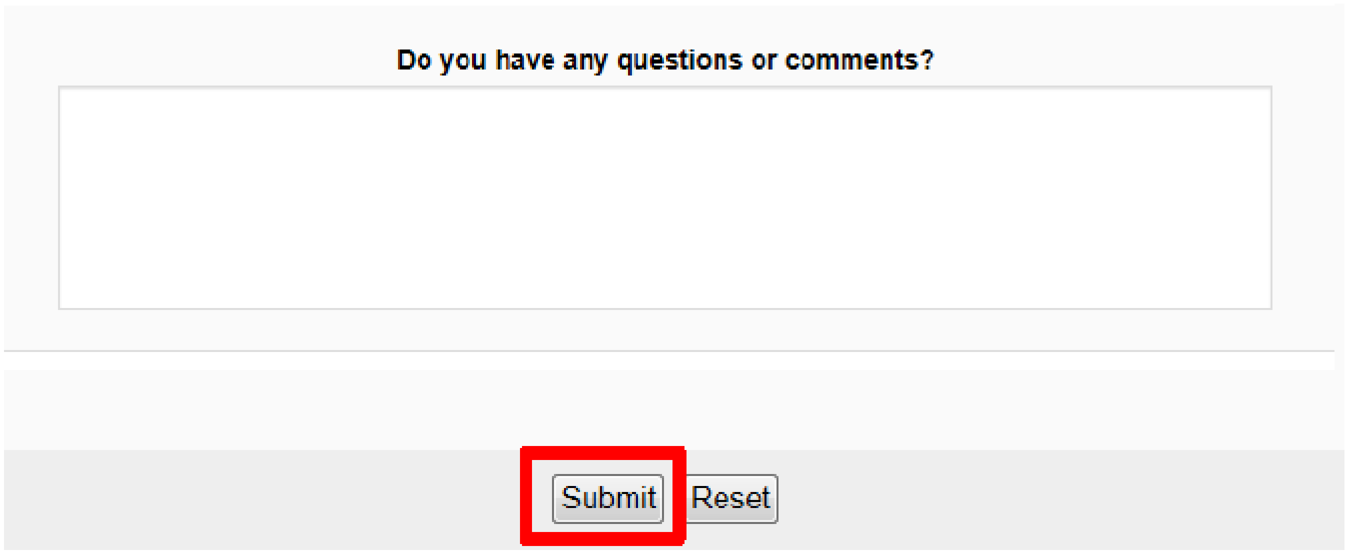


First Name:	
Last Name:	
E-mail:	zzangchsq@gmail.com
Company:	Kwangwoon univ
Address:	Wolgye 1-dong
	Nowon-gu
	Seoul, Korea
City:	Seoul
State/Province:	Select Your State or Province
Zip/Postal Code:	139-701
Country:	Republic of Korea
Phone:	

☐ Send me e-mail when there is a new update.
NOTICE:
If you select this check box, you **will** receive an e-mail message from Keil whenever a new update is available. If you don't wish to receive an e-mail notification, don't check this box.

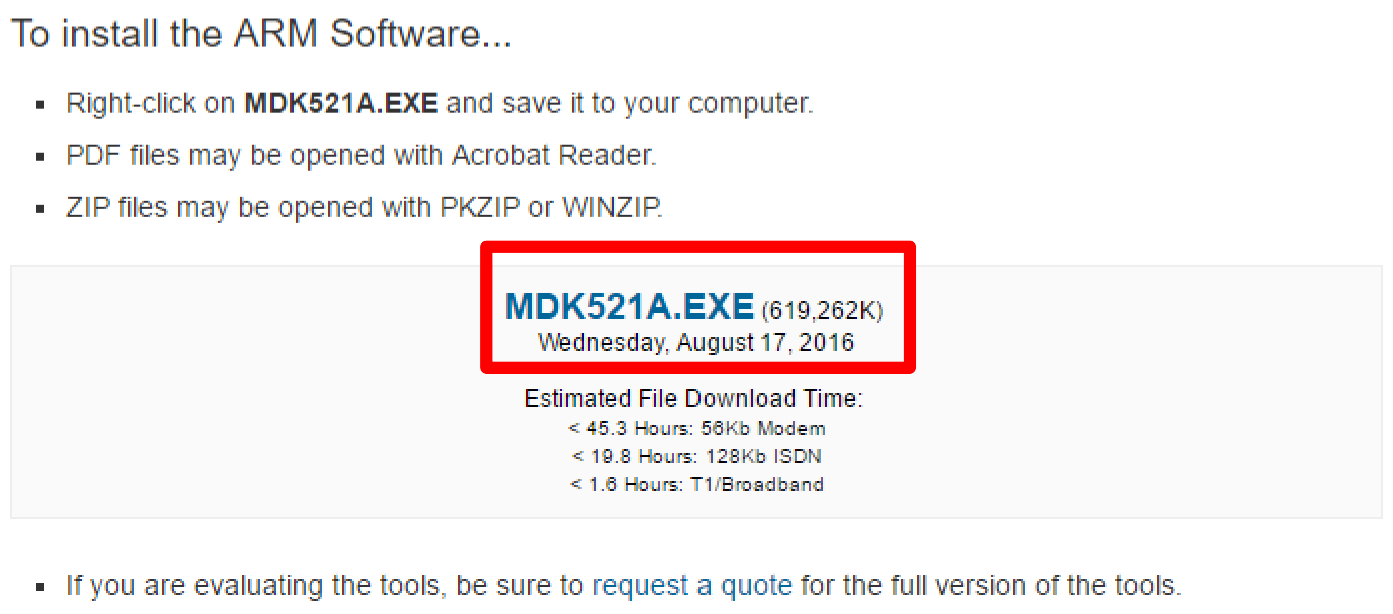
Keil User Registration (3/3)

- Click "submit" button



Download

- Click the link and download program



To install the ARM Software...

- Right-click on **MDK521A.EXE** and save it to your computer.
- PDF files may be opened with Acrobat Reader.
- ZIP files may be opened with PKZIP or WINZIP.

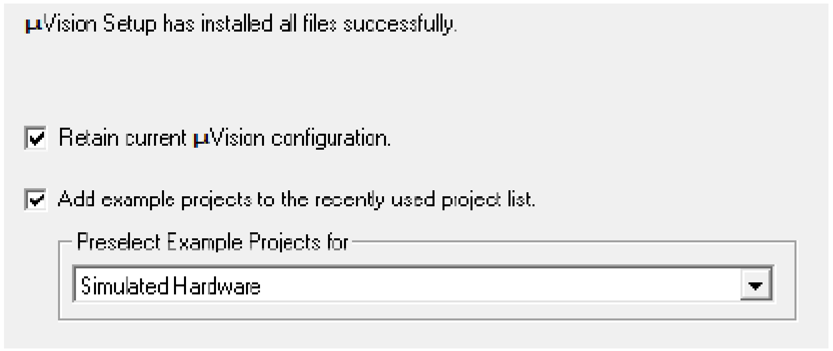
MDK521A.EXE (619,262K)
Wednesday, August 17, 2016

Estimated File Download Time:
< 45.3 Hours: 56Kb Modem
< 19.8 Hours: 128Kb ISDN
< 1.6 Hours: T1/Broadband

- If you are evaluating the tools, be sure to request a quote for the full version of the tools.

Setup

- Execute “MDK521a”
  - Click “Next” button until starting setup
- When installation is completed, follow below picture



µVision Setup has installed all files successfully.

[✓] Retain current µVision configuration.

[✓] Add example projects to the recently used project list.

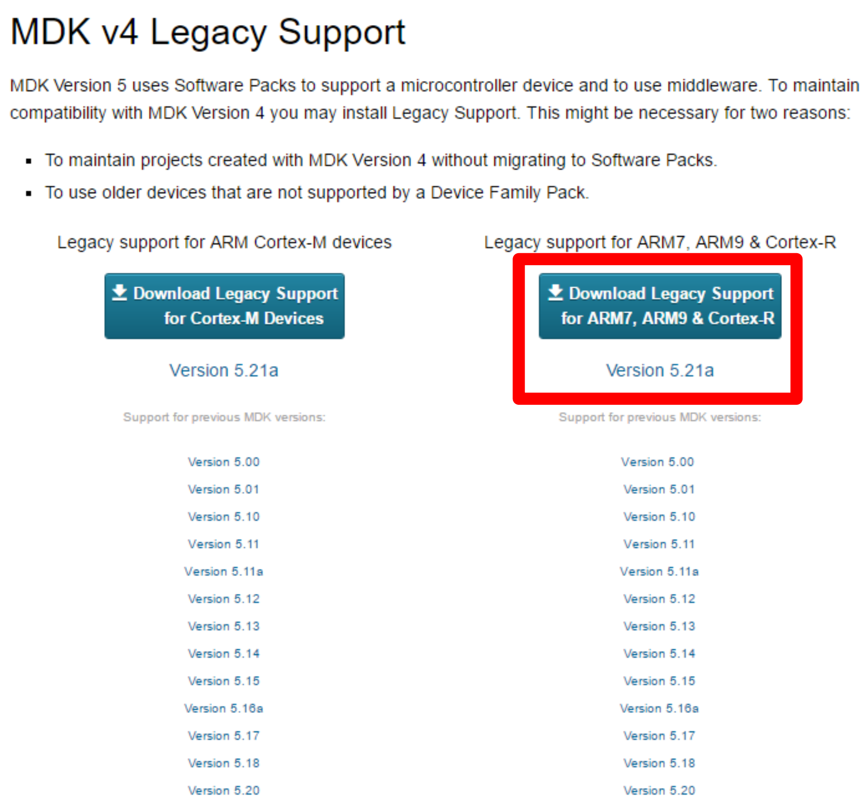
Preselect Example Projects for

Simulated Hardware

- And Then, Click “Next” button
- Click Finish button

MDK v4 Legacy Support

- Connect link
  - http://www2.keil.com/mdk5/legacy/
- Download



MDK v4 Legacy Support

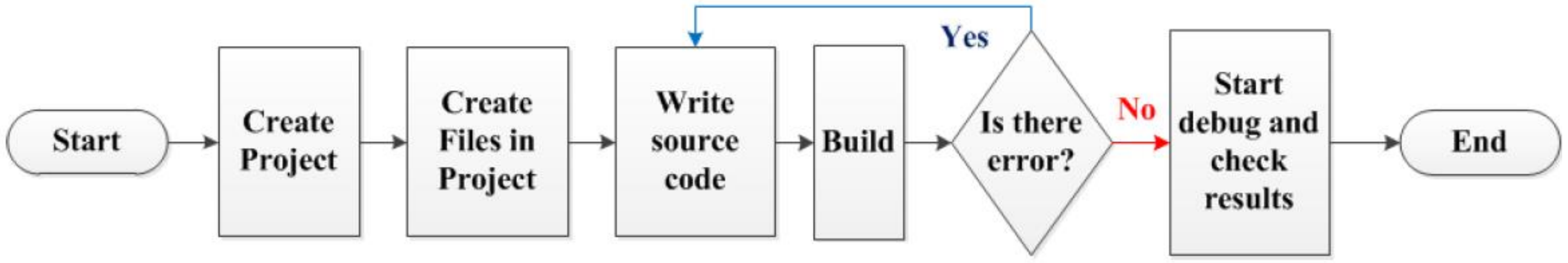
MDK Version 5 uses Software Packs to support a microcontroller device and to use middleware. To maintain compatibility with MDK Version 4 you may install Legacy Support. This might be necessary for two reasons:

- To maintain projects created with MDK Version 4 without migrating to Software Packs.
- To use older devices that are not supported by a Device Family Pack.

Legacy support for ARM Cortex-M devices	Legacy support for ARM7, ARM9 & Cortex-R
Download Legacy Support for Cortex-M Devices	Download Legacy Support for ARM7, ARM9 & Cortex-R
Version 5.21a	Version 5.21a
Support for previous MDK versions:	Support for previous MDK versions:
Version 5.00	Version 5.00
Version 5.01	Version 5.01
Version 5.10	Version 5.10
Version 5.11	Version 5.11
Version 5.11a	Version 5.11a
Version 5.12	Version 5.12
Version 5.13	Version 5.13
Version 5.14	Version 5.14
Version 5.15	Version 5.15
Version 5.16a	Version 5.16a
Version 5.17	Version 5.17
Version 5.18	Version 5.18
Version 5.20	Version 5.20

Assembly Programming using Keil uVision5

- Flow

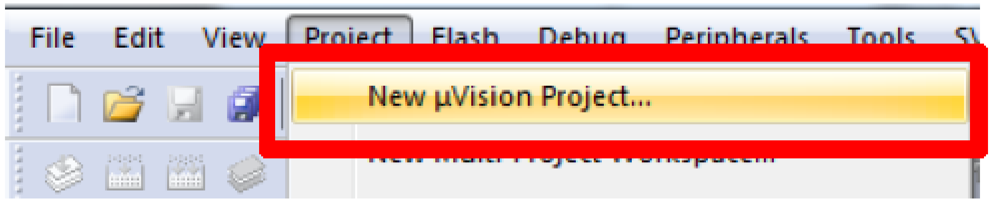


- Create Project
- Create Files
- Write source code
- Build
- Debug

Project Creation (1/5)

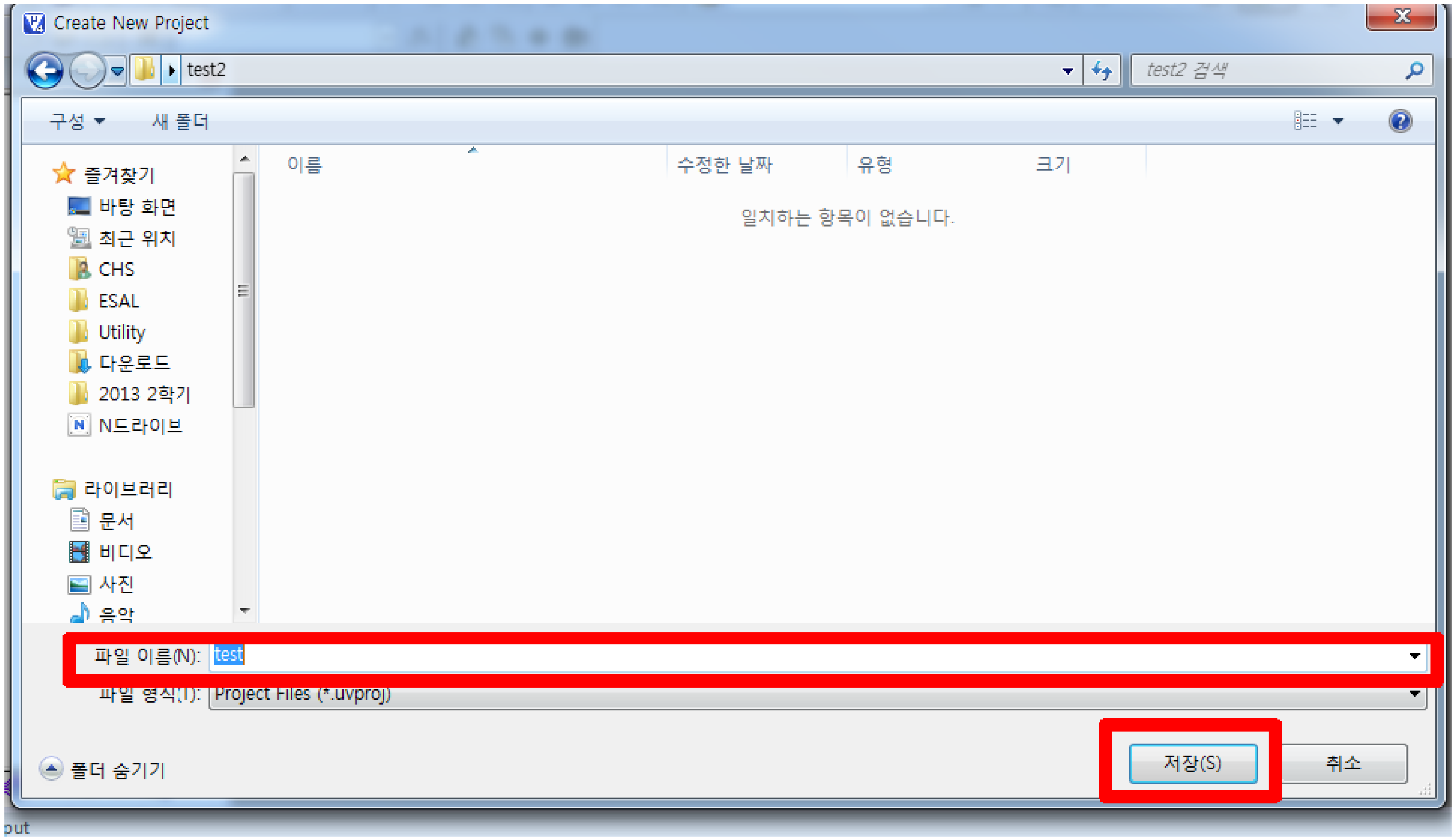
- Execute Keil uVision5

- Click 'Project' tab and click 'New μVision Project'



Project Creation (2/5)

- Write project name



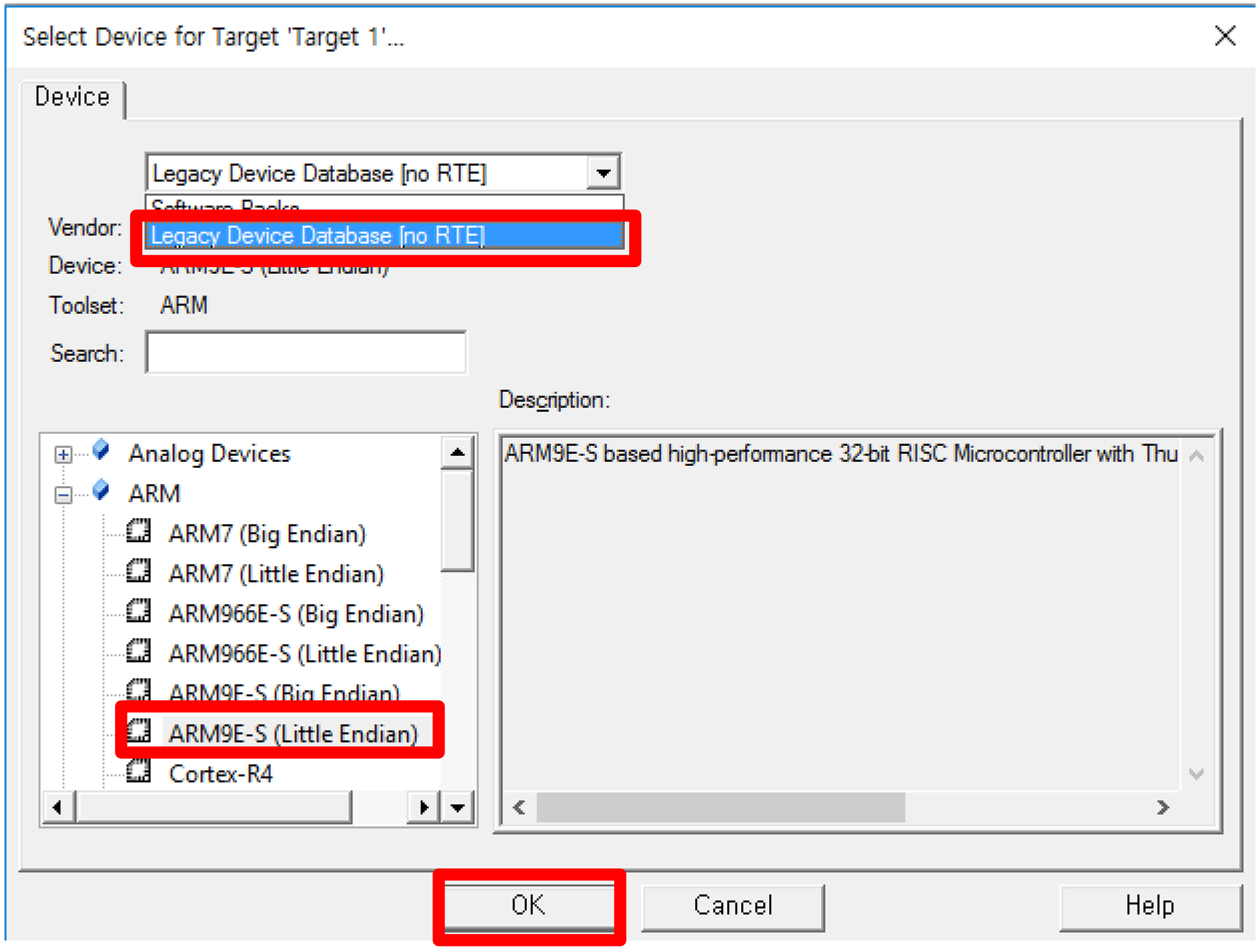
일치하는 항목이 없습니다.

파일 이름(N): test

파일 형식(T): Project Files (*.uvproj)

Project Creation (3/5)

- Select device for target
  - Legacy Device Database → ARM → ARM9E-S (Little Endian)



Select Device for Target 'Target 1'...

Device

Legacy Device Database [no RTE]

Software Packs

Legacy Device Database [no RTE]

Vendor:

Device: ARM9E-S (Little Endian)

Toolset: ARM

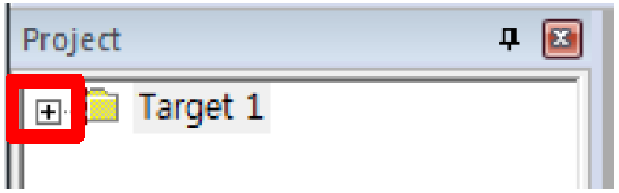
Search:

Description:

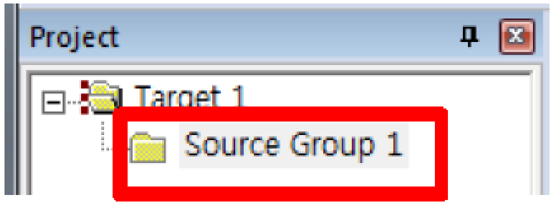
ARM9E-S based high-performance 32-bit RISC Microcontroller with Thu

Project Creation (4/5)

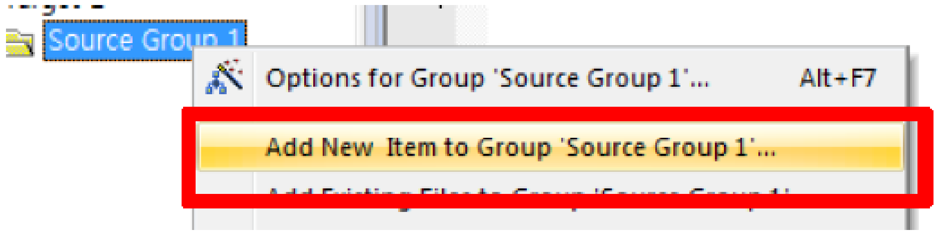
- Activate "Target 1" byclick + button



- Right click on "Source Group1"

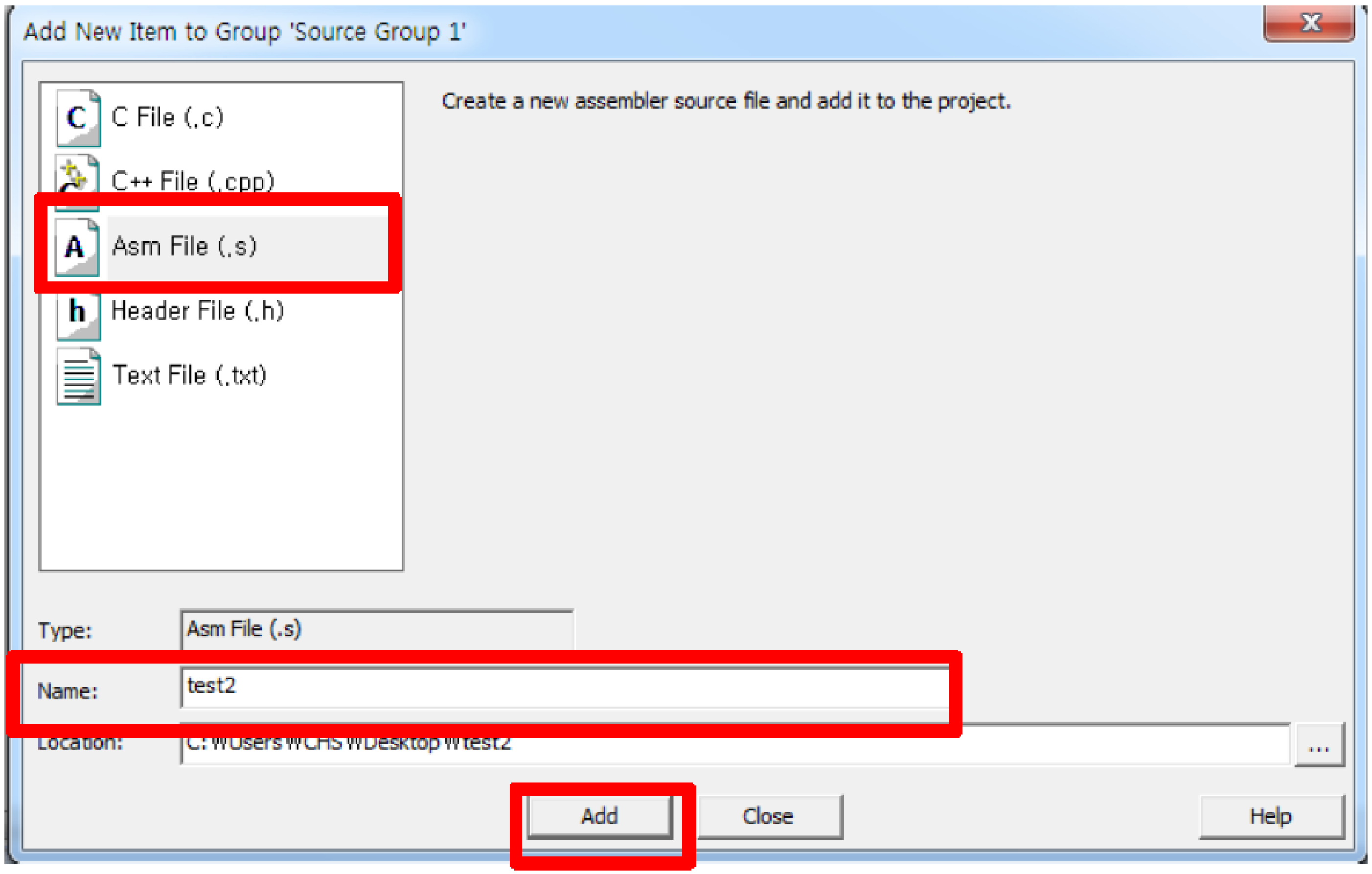


- Click "Add New Item to Group …"


Source Group 1
Options for Group 'Source Group 1'... Alt+F7
Add New Item to Group 'Source Group 1'...

Project Creation (5/5)

- Click 'Asm File (.s)' and write a file name



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

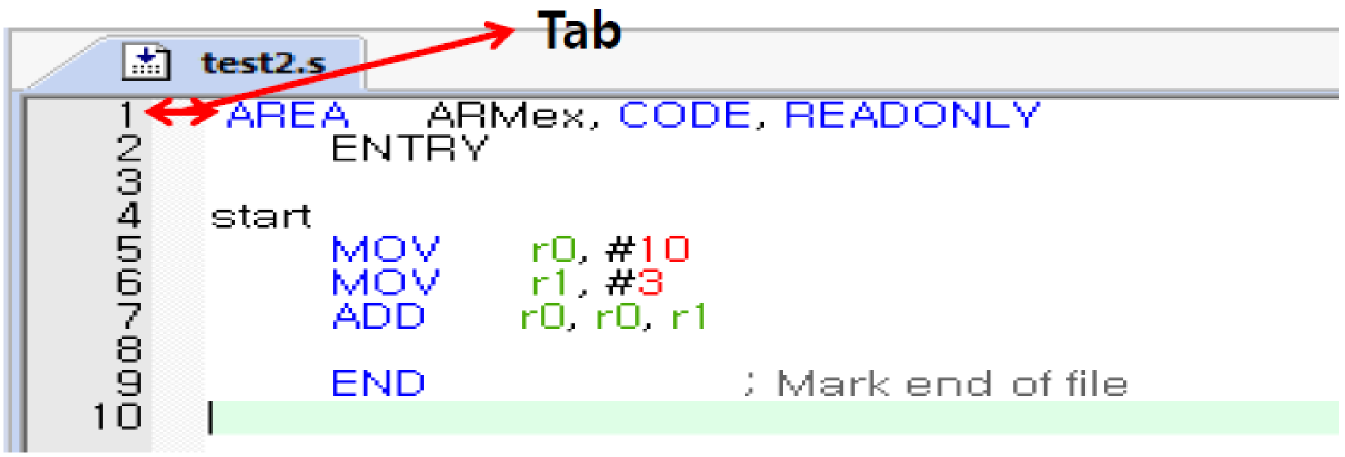
Type: Asm File (.s)

Name: test2

Location: C:\Users\CHS\Desktop\test2

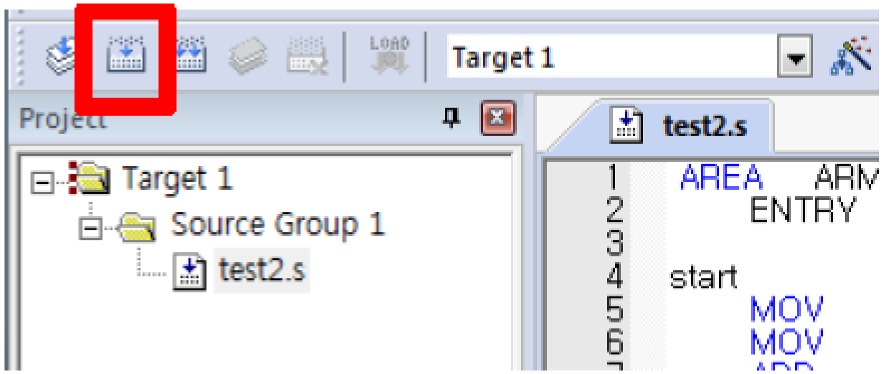
Example Execution (1/2)

- Write source like below



```
AREA	ARMex, CODE, READONLY
	ENTRY

start
	MOV	r0, #10
	MOV	r1, #3
	ADD	r0, r0, r1

	END			; Mark end of file
```

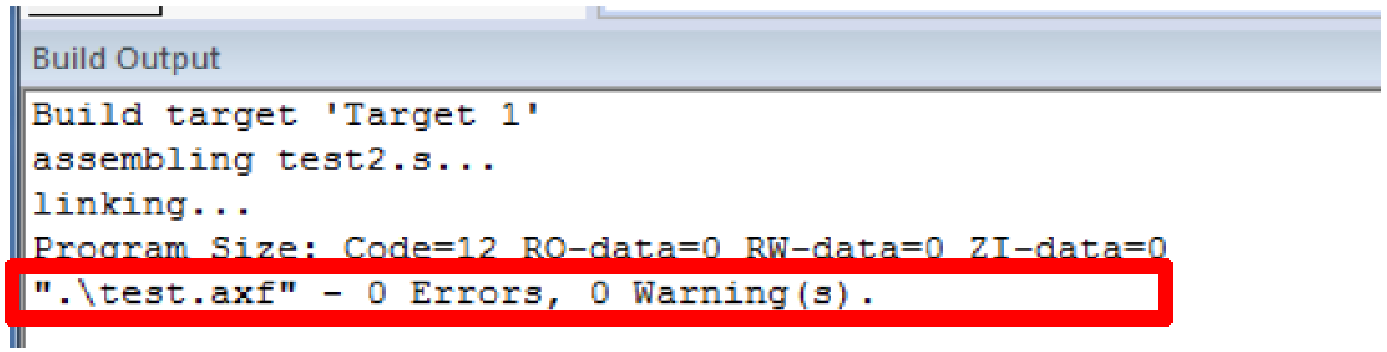
- Click on the 'build' button



```
AREA	ARM
	ENTRY

start
	MOV
	MOV
```

Example Execution (2/2)

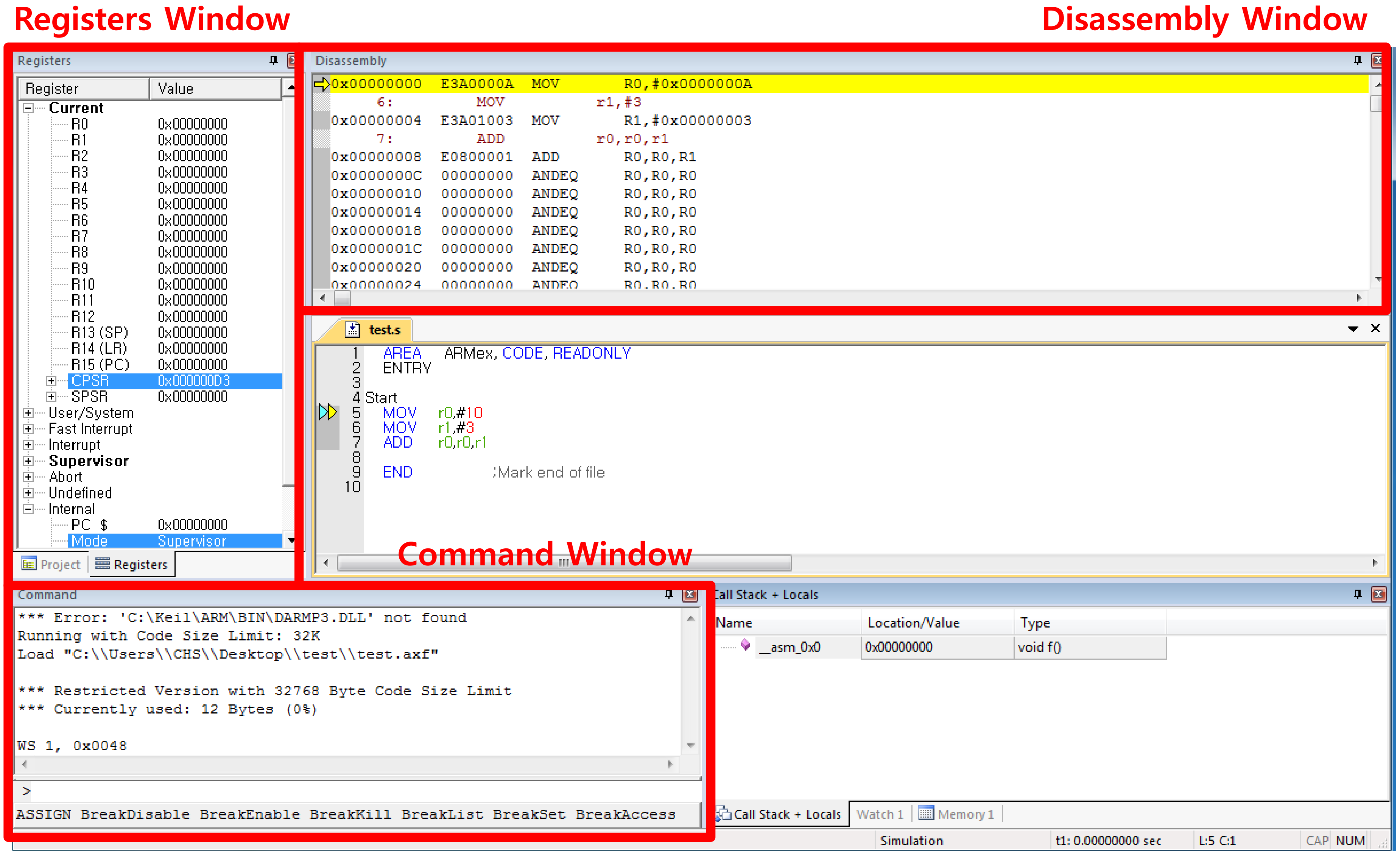
- Check the ‘Build Output’



Build Output

```
Build target 'Target 1'
assembling test2.s...
linking...
Program Size: Code=12 RO-data=0 RW-data=0 ZI-data=0
".\test.axf" - 0 Errors, 0 Warning(s).
```

Debug (1/6)



Register	Value
Current	
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)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000000
Mode	Supervisor

```
0x00000000  E3A0000A  MOV      R0,#0x0000000A
     6:          MOV          r1,#3
0x00000004  E3A01003  MOV      R1,#0x00000003
     7:          ADD          r0,r0,r1
0x00000008  E0800001  ADD      R0,R0,R1
0x0000000C  00000000  ANDEQ    R0,R0,R0
0x00000010  00000000  ANDEQ    R0,R0,R0
0x00000014  00000000  ANDEQ    R0,R0,R0
0x00000018  00000000  ANDEQ    R0,R0,R0
0x0000001C  00000000  ANDEQ    R0,R0,R0
0x00000020  00000000  ANDEQ    R0,R0,R0
0x00000024  00000000  ANDEQ    R0,R0,R0
```

```
	AREA	ARMex, CODE, READONLY
	ENTRY

Start
	MOV	r0,#10
	MOV	r1,#3
	ADD	r0,r0,r1

	END		;Mark end of file
```

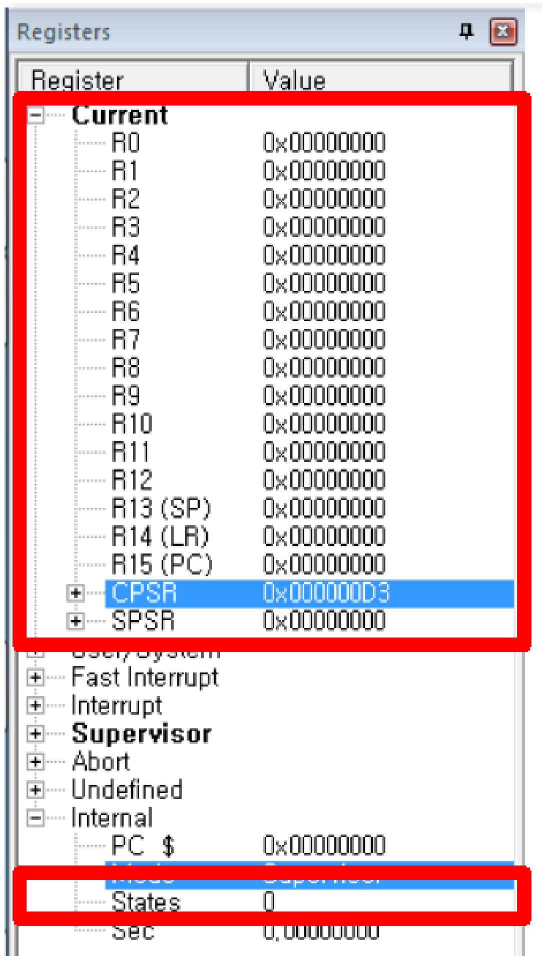
```
*** Error: 'C:\Keil\ARM\BIN\DARMP3.DLL' not found
Running with Code Size Limit: 32K
Load "C:\\Users\\CHS\\Desktop\\test\\test.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 12 Bytes (0%)

WS 1, 0x0048
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Name	Location/Value	Type
__asm_0x0	0x00000000	void f()

Simulation | t1: 0.00000000 sec | L:5 C:1 | CAP NUM

Debug (2/6)

- Registers Window
  - To see register's values
  - Execution time or cycles
    - The value of State in the Registers window

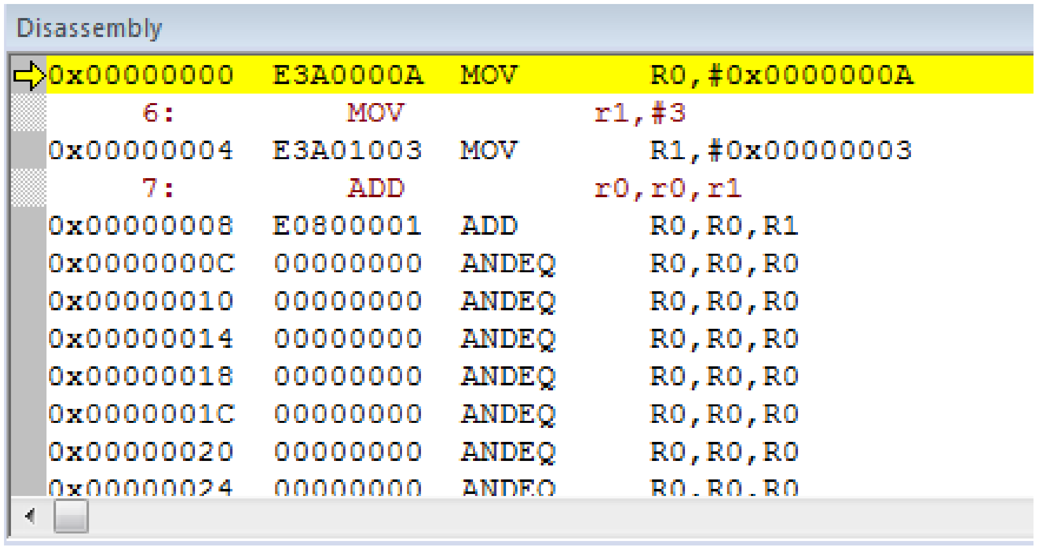


Registers

Register	Value
Current	
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)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000000
Mode	Supervisor
States	0
Sec	0.00000000

Debug (3/6)

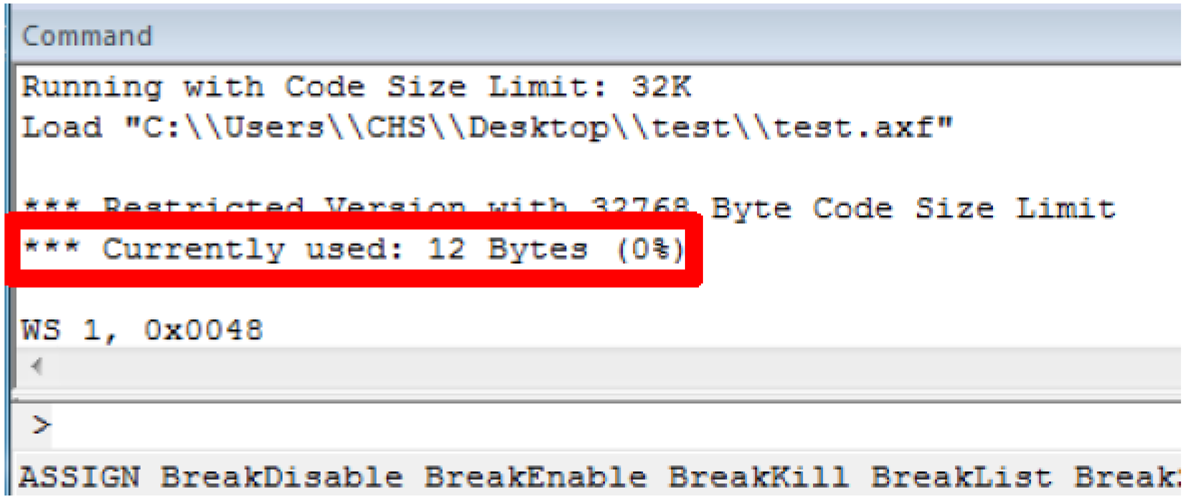
- Disassembly Window
  - Can see machine language



```
0x00000000   E3A0000A   MOV         R0,#0x0000000A
      6:          MOV          r1,#3
0x00000004   E3A01003   MOV         R1,#0x00000003
      7:          ADD          r0,r0,r1
0x00000008   E0800001   ADD         R0,R0,R1
0x0000000C   00000000   ANDEQ       R0,R0,R0
0x00000010   00000000   ANDEQ       R0,R0,R0
0x00000014   00000000   ANDEQ       R0,R0,R0
0x00000018   00000000   ANDEQ       R0,R0,R0
0x0000001C   00000000   ANDEQ       R0,R0,R0
0x00000020   00000000   ANDEQ       R0,R0,R0
0x00000024   00000000   ANDEQ       R0,R0,R0
```

Debug (4/6)

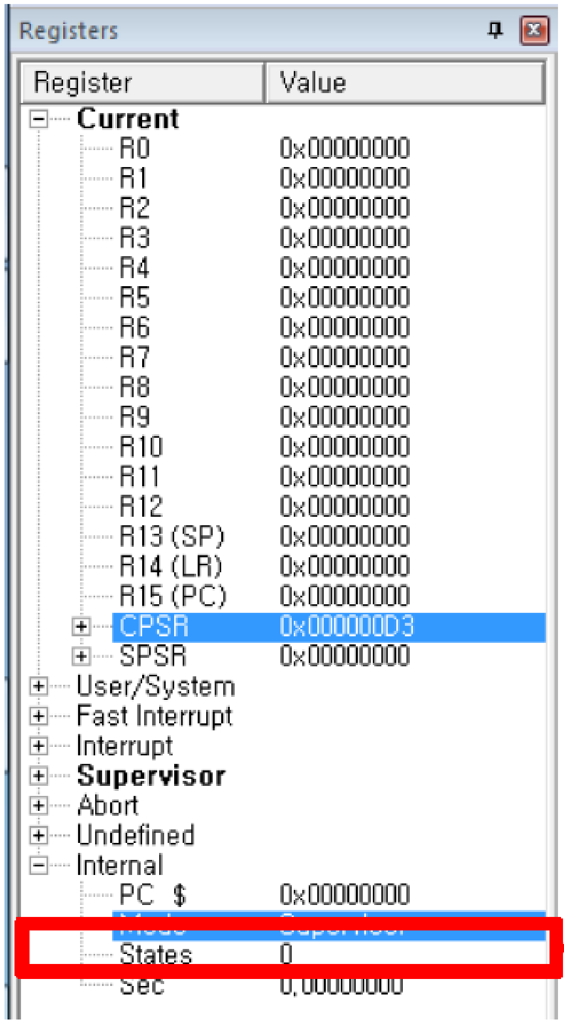
- Command Window
  - Check performance
    - Whenever you check **code size**



```
Command
Running with Code Size Limit: 32K
Load "C:\\Users\\CHS\\Desktop\\test\\test.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 12 Bytes (0%)

WS 1, 0x0048
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList Break
```

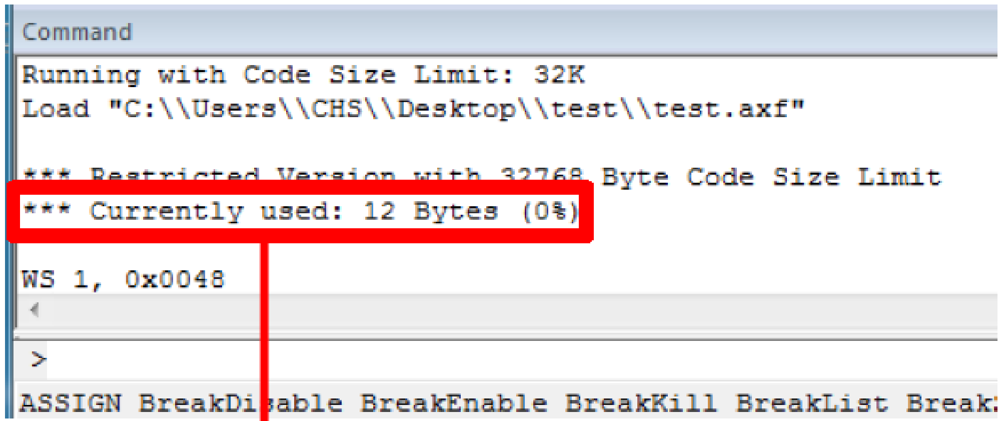
Debug (5/6)

- Check performance



Registers

Register	Value
Current	
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)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x00000000
Mode	Supervisor
States	0
Sec	0.00000000



Command

```
Running with Code Size Limit: 32K
Load "C:\\Users\\CHS\\Desktop\\test\\test.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 12 Bytes (0%)

WS 1, 0x0048

>
ASSIGN BreakDisable BreakEnable BreakKill BreakList Break
```

Code size

State

Debug (6/6)

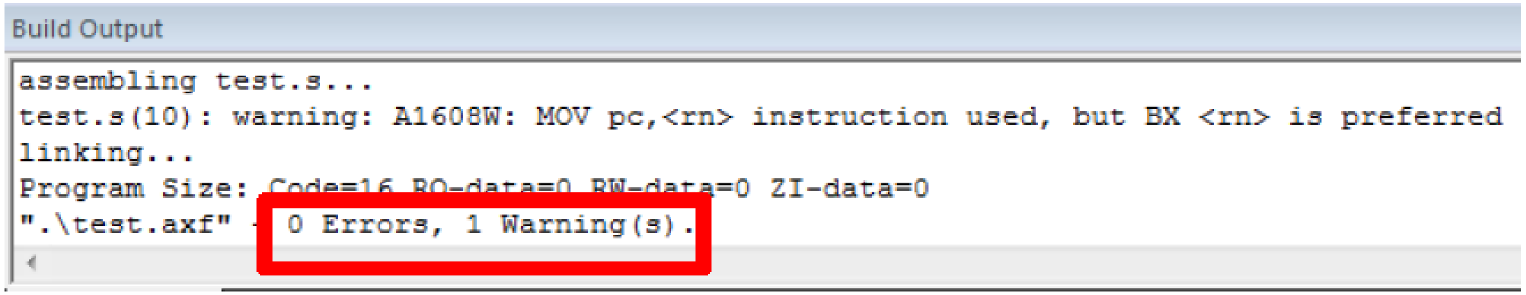
- Shortcut key
  - Start debug mode: Ctrl+F5
  - Break point: F9
  - Check line: F10
  - End debug mode: Ctrl+F5

Example (1/5)

- Code

```
    AREA    ARMex, CODE, READONLY
    ENTRY

Start
    MOV     r0,#10   ;store integer 10 to register 0
    MOV     r1,#3    ;store integer 1   to register 1
    ADD     r0,r0,r1 ;add register0's value and register1's value and store result to register 0

    MOV     pc,lr    ;go to first instruction
    END              ;Mark end of file
```

- Build

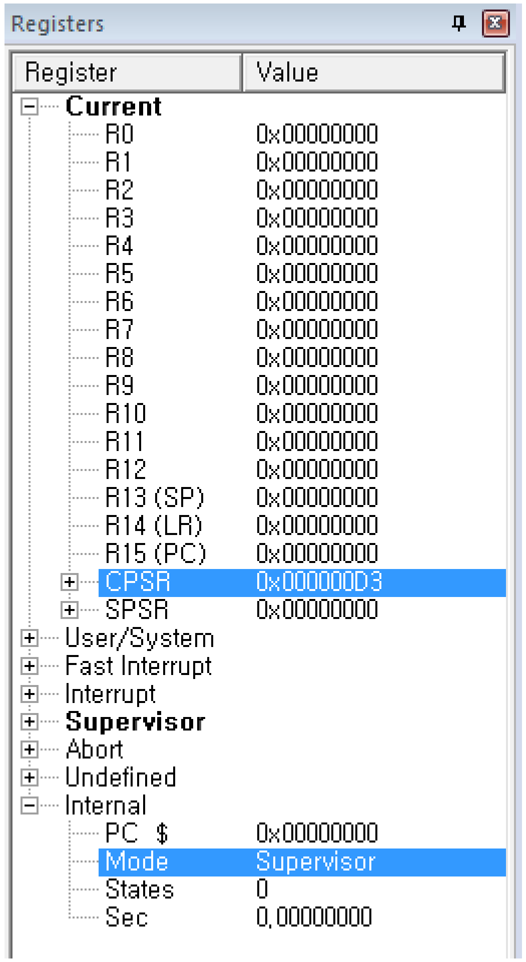


Build Output

```
assembling test.s...
test.s(10): warning: A1608W: MOV pc,<rn> instruction used, but BX <rn> is preferred
linking...
Program Size: Code=16 RO-data=0 RW-data=0 ZI-data=0
".\test.axf" - 0 Errors, 1 Warning(s).
```

Example (2/5)

- Start debug

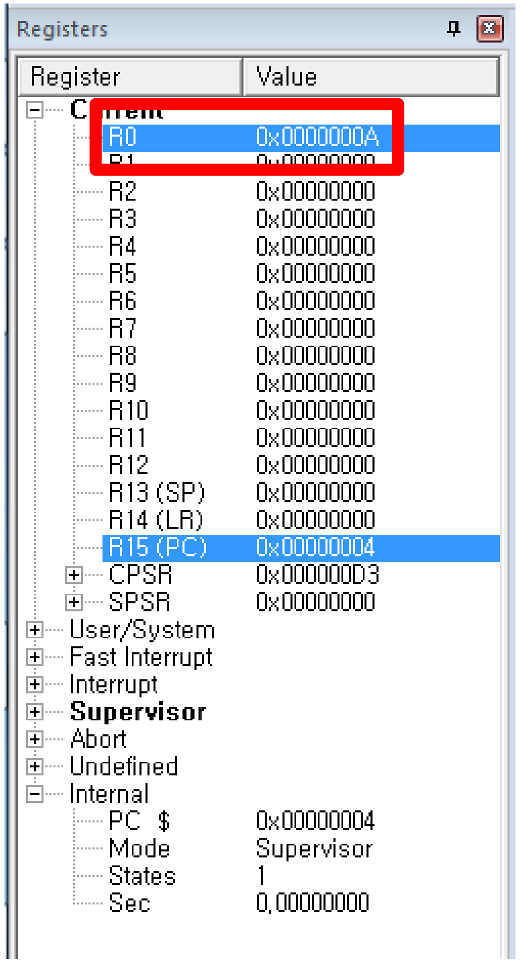


```
Registers
Register          Value
Current
    R0            0x00000000
    R1            0x00000000
    R2            0x00000000
    R3            0x00000000
    R4            0x00000000
    R5            0x00000000
    R6            0x00000000
    R7            0x00000000
    R8            0x00000000
    R9            0x00000000
    R10           0x00000000
    R11           0x00000000
    R12           0x00000000
    R13 (SP)      0x00000000
    R14 (LR)      0x00000000
    R15 (PC)      0x00000000
    CPSR          0x000000D3
    SPSR          0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
    PC  $         0x00000000
    Mode          Supervisor
    States        0
    Sec           0,00000000
```

```
    AREA    ARMex, CODE, READONLY
    ENTRY

Start
    MOV     r0,#10  ;store integer 10 to register 0
    MOV     r1,#3   ;store integer 1   to register 1
    ADD     r0,r0,r1 ;add register0's value and register1's value and store result to register 0

    MOV     pc,lr   ;go to first instruction
    END             ;Mark end of file
```

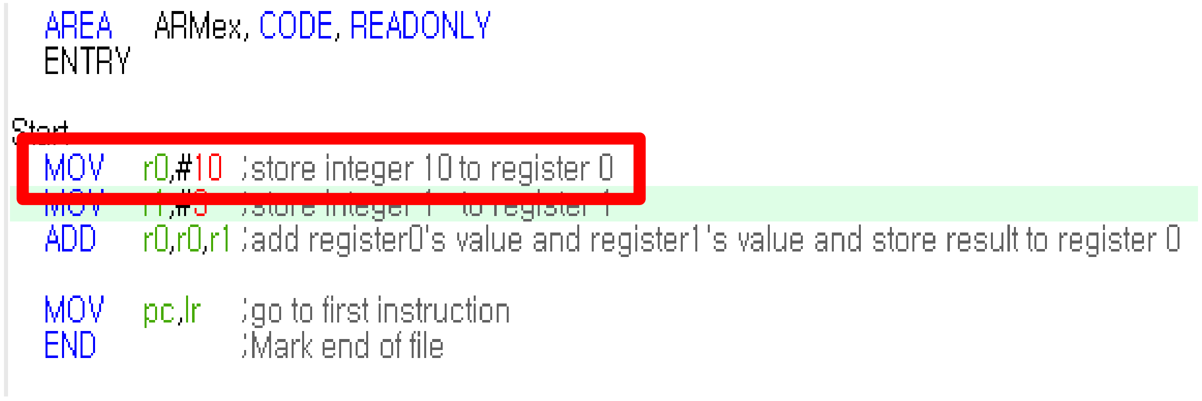
Example (3/5)

- R0=10



```
Registers
Register        Value
⊟ Current
    R0          0x0000000A
    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)    0x00000000
    R14 (LR)    0x00000000
    R15 (PC)    0x00000004
  ⊞ CPSR        0x000000D3
  ⊞ SPSR        0x00000000
⊞ User/System
⊞ Fast Interrupt
⊞ Interrupt
⊞ Supervisor
⊞ Abort
⊞ Undefined
⊟ Internal
    PC  $       0x00000004
    Mode        Supervisor
    States      1
    Sec         0.00000000
```

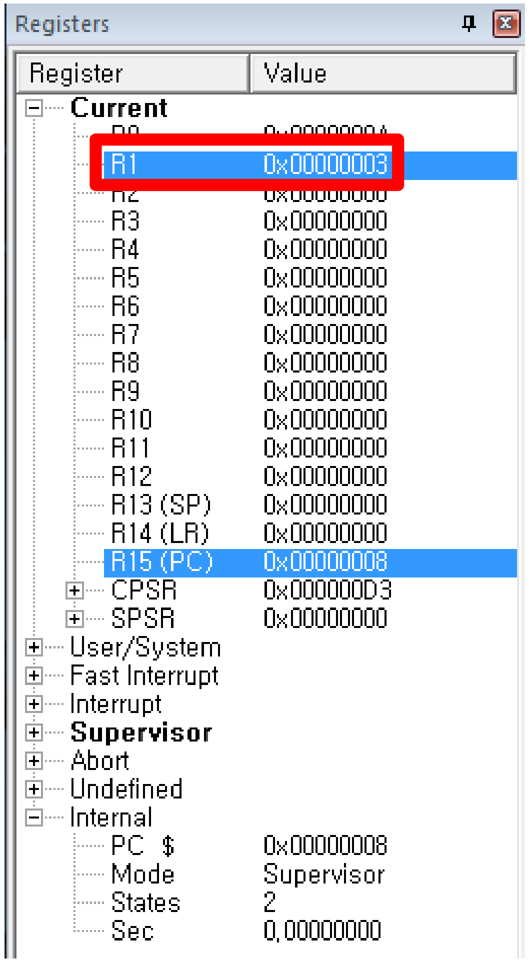
R0=10(0xA)



```
    AREA    ARMex, CODE, READONLY
    ENTRY

Start
    MOV     r0,#10  ;store integer 10 to register 0
    MOV     r1,#5   ;store integer 5 to register 1
    ADD     r0,r0,r1 ;add register0's value and register1's value and store result to register 0

    MOV     pc,lr   ;go to first instruction
    END             ;Mark end of file
```

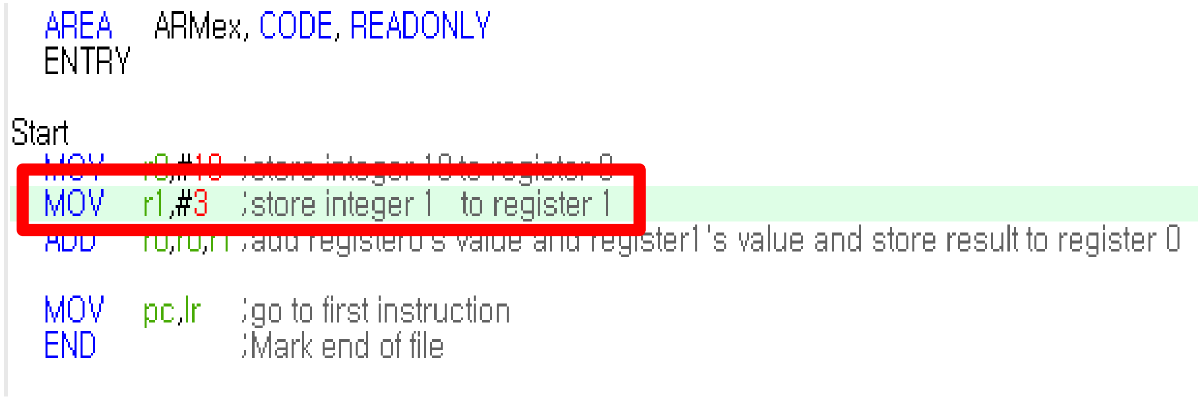
Example (4/5)

- R1=3



```
Registers
Register          Value
Current
  R0              0x0000000A
  R1              0x00000003
  R2              0x00000000
  R3              0x00000000
  R4              0x00000000
  R5              0x00000000
  R6              0x00000000
  R7              0x00000000
  R8              0x00000000
  R9              0x00000000
  R10             0x00000000
  R11             0x00000000
  R12             0x00000000
  R13 (SP)        0x00000000
  R14 (LR)        0x00000000
  R15 (PC)        0x00000008
  CPSR            0x000000D3
  SPSR            0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
  PC  $           0x00000008
  Mode            Supervisor
  States          2
  Sec             0.00000000
```

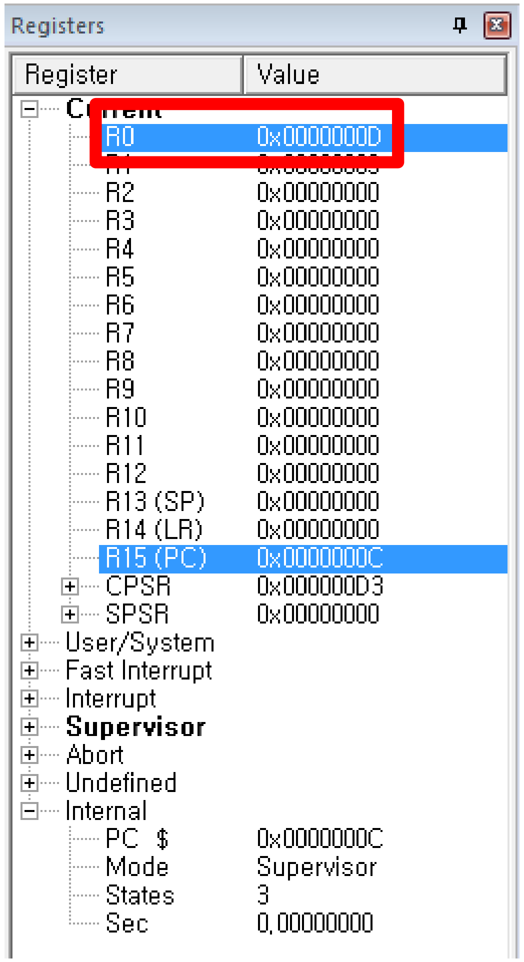
R1=3(0x3)



```
    AREA    ARMex, CODE, READONLY
    ENTRY

Start
    MOV     r0,#10   ;store integer 10 to register 0
    MOV     r1,#3    ;store integer 1  to register 1
    ADD     r0,r0,r1 ;add register0's value and register1's value and store result to register 0

    MOV     pc,lr    ;go to first instruction
    END              ;Mark end of file
```

Example (5/5)

- R0=R0+R1



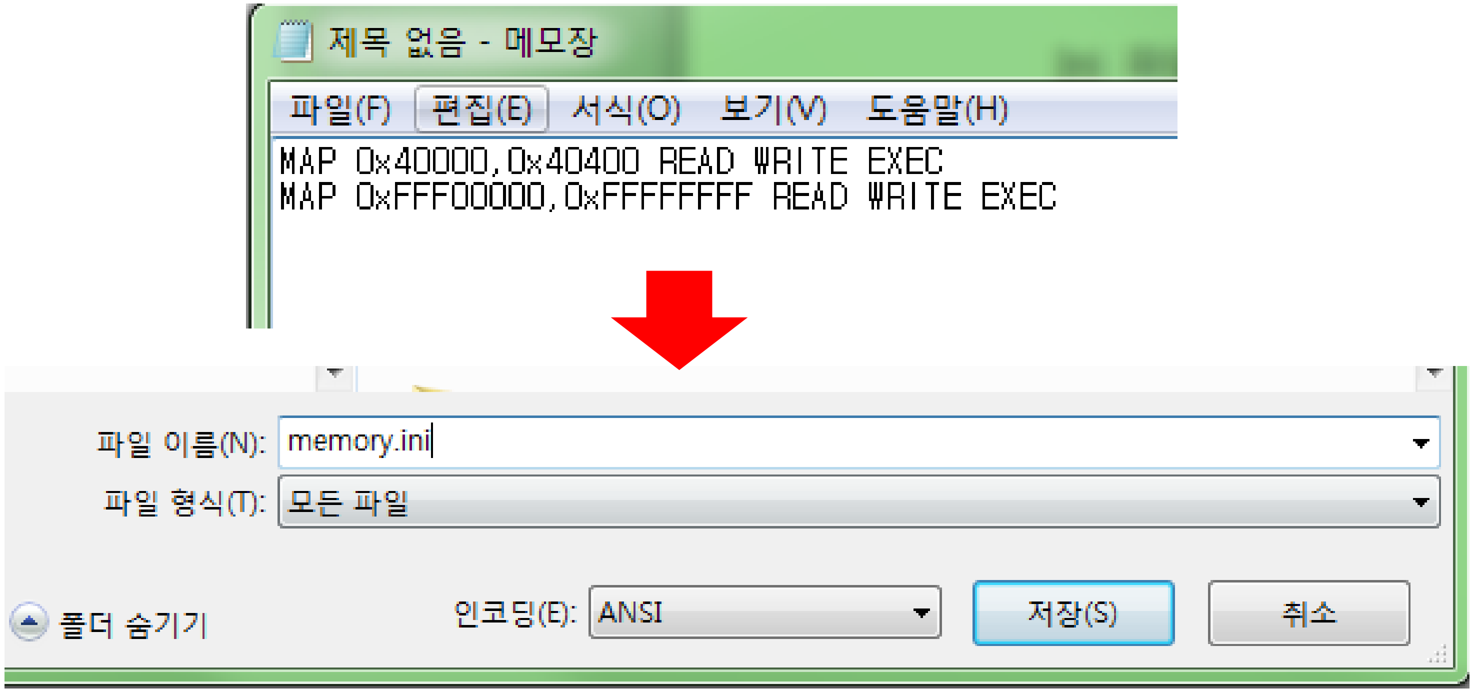
```
Registers
Register          Value
Current
  R0              0x0000000D
  R1              0x0000000D
  R2              0x00000000
  R3              0x00000000
  R4              0x00000000
  R5              0x00000000
  R6              0x00000000
  R7              0x00000000
  R8              0x00000000
  R9              0x00000000
  R10             0x00000000
  R11             0x00000000
  R12             0x00000000
  R13 (SP)        0x00000000
  R14 (LR)        0x00000000
  R15 (PC)        0x0000000C
  CPSR            0x000000D3
  SPSR            0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
  PC  $           0x0000000C
  Mode            Supervisor
  States          3
  Sec             0.00000000
```

R1=13(0xD)

```
    AREA    ARMex, CODE, READONLY
    ENTRY

Start
    MOV     r0,#10   ;store integer 10 to register 0
    MOV     r1,#13   ;store integer 13 to register 1
    ADD     r0,r0,r1 ;add register0's value and register1's value and store result to register 0

    MOV     pc,lr    ;go to first instruction
    END              ;Mark end of file
```

Register INI File (1/6)

- Make ini file
  - Debug모드 진입할 때 읽게 되는 파일
  - 임의의 메모리 영역에 대해 read, write 또는 exec권한 부여
  - 접근하는 영역에 read 권한이 없을 경우 메모리로부터 load불가, write 권한이 없을 경우 메모리로 store 불가

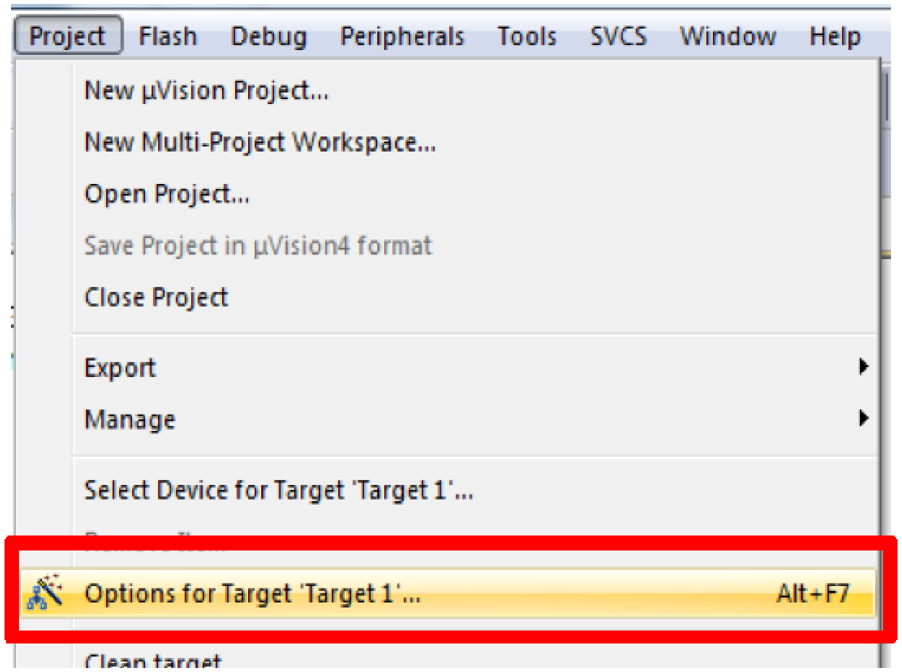


```
MAP 0x40000,0x40400 READ WRITE EXEC
MAP 0xFFF00000,0xFFFFFFFF READ WRITE EXEC
```

파일 이름(N): memory.ini
파일 형식(T): 모든 파일
인코딩(E): ANSI

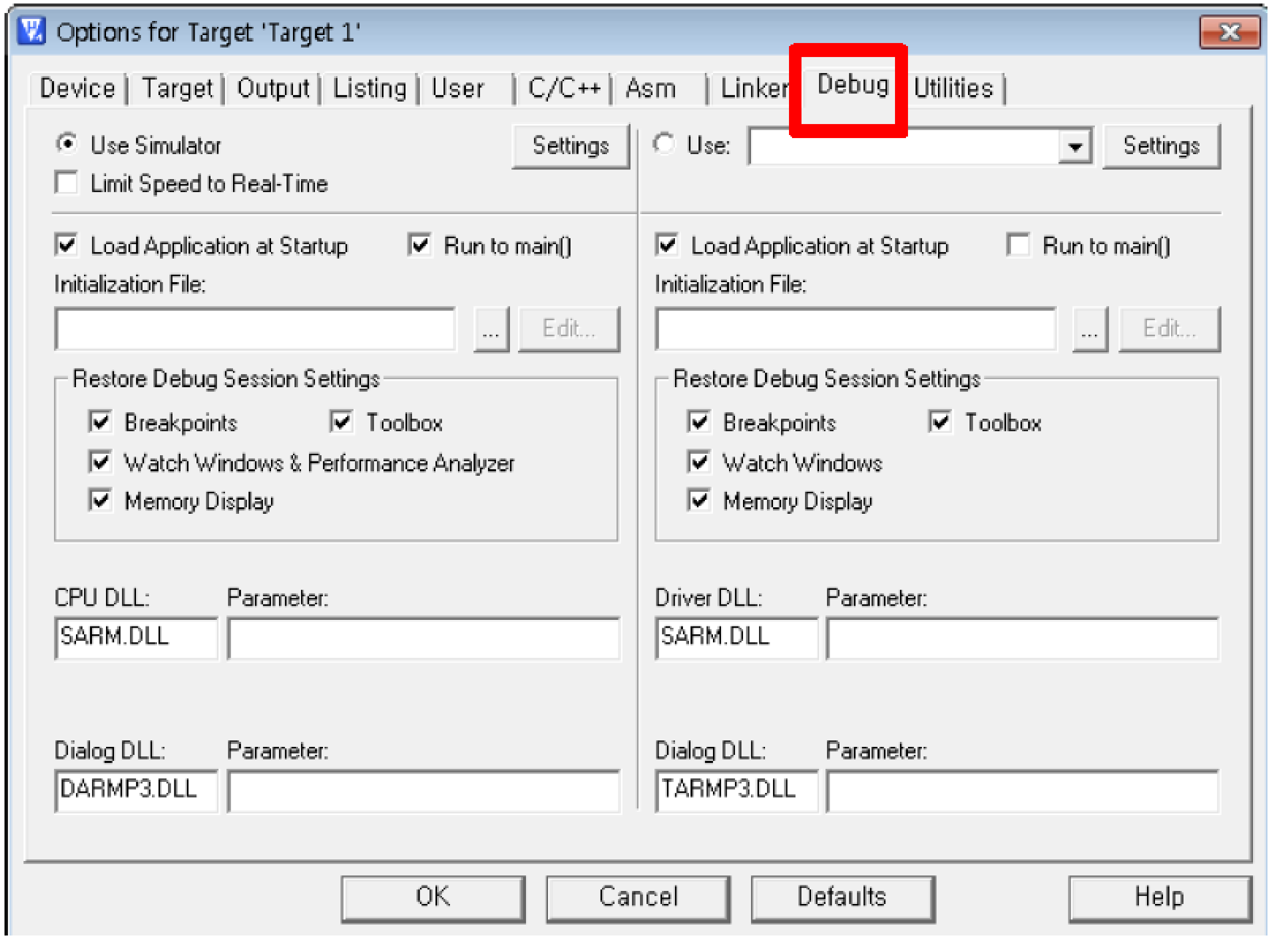
Register INI File (2/6)

- Activate Project tab → Click Options for Target


Project
Flash
Debug
Peripherals
Tools
SVCS
Window
Help
New µVision Project...
New Multi-Project Workspace...
Open Project...
Save Project in µVision4 format
Close Project
Export
Manage
Select Device for Target 'Target 1'...
Options for Target 'Target 1'...
Alt+F7
Clean target

Register INI File (3/6)

- Debug tab Click



Options for Target 'Target 1'

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

(•) Use Simulator — 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

CPU DLL: SARM.DLL Parameter:
Dialog DLL: DARMP3.DLL Parameter:

( ) Use: — Settings

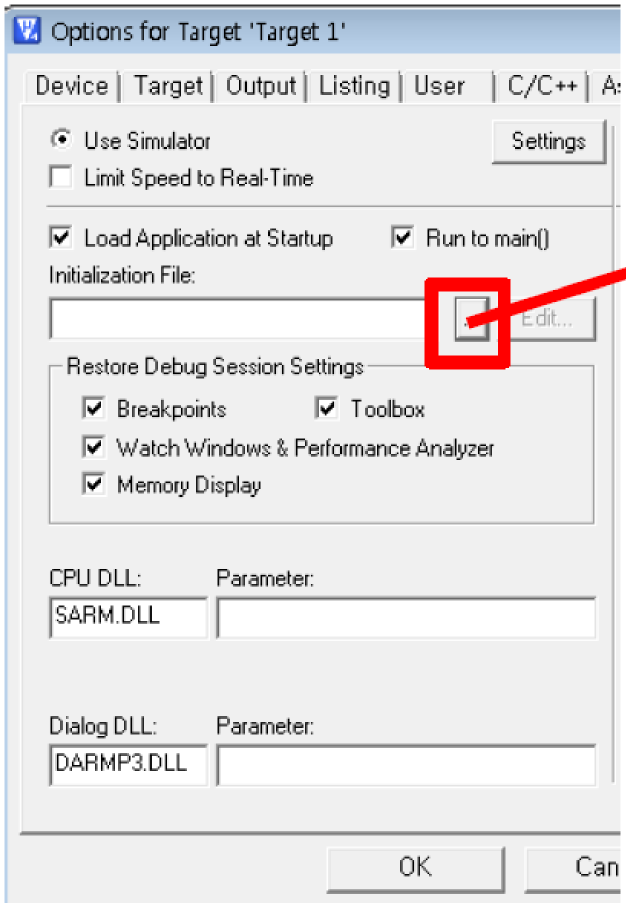
☑ Load Application at Startup ☐ Run to main()
Initialization File:

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

Driver DLL: SARM.DLL Parameter:
Dialog DLL: TARMP3.DLL Parameter:

Register INI File (4/6)

- Click "..." Button and register ini file


Options for Target 'Target 1'
Device | Target | Output | Listing | User | C/C++ | A
Settings
Edit...
OK
Can

- [✓] Use Simulator
- [ ] Limit Speed to Real-Time
- [✓] Load Application at Startup
- [✓] Run to main()

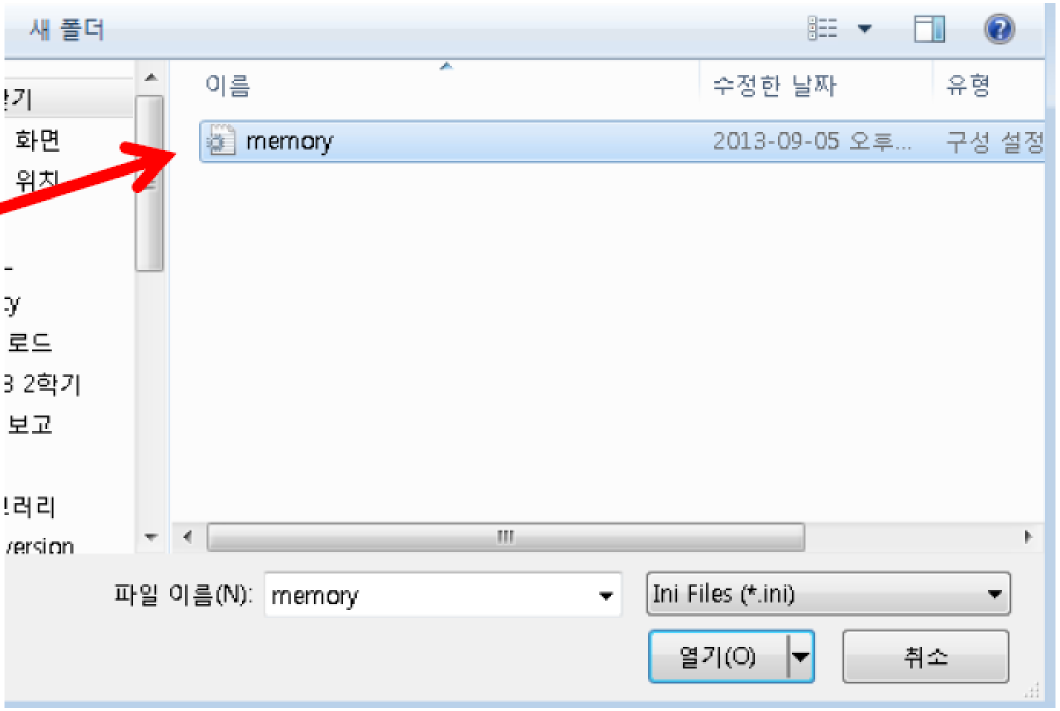
Initialization File:

Restore Debug Session Settings

- [✓] Breakpoints
- [✓] Toolbox
- [✓] Watch Windows & Performance Analyzer
- [✓] Memory Display

CPU DLL:	Parameter:
SARM.DLL	

Dialog DLL:	Parameter:
DARMP3.DLL	



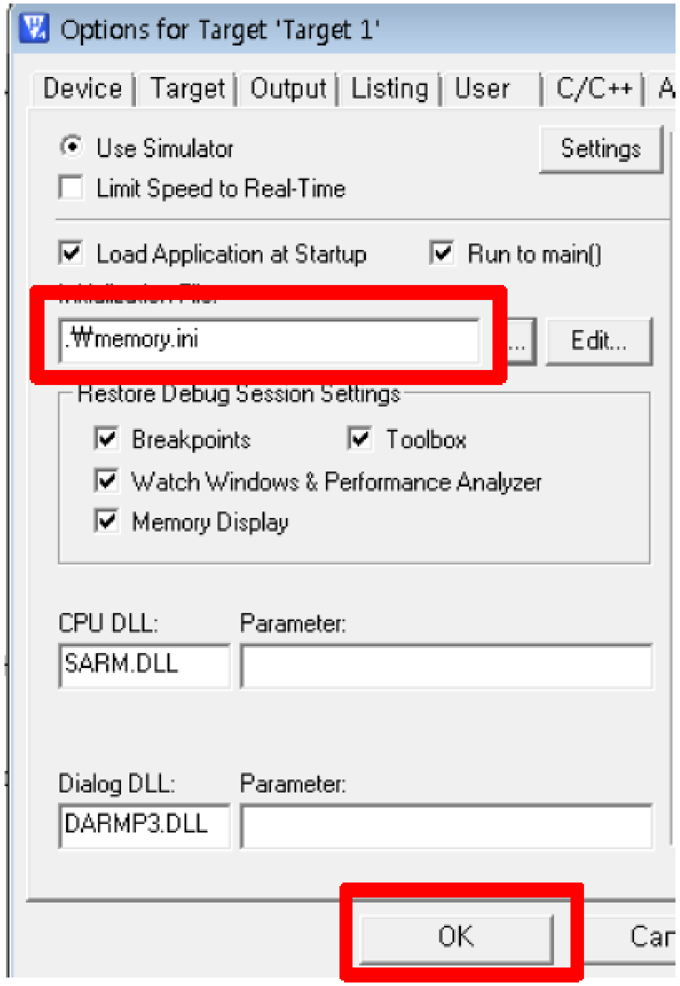
이름	수정한 날짜	유형
memory	2013-09-05 오후...	구성 설정

파일 이름(N): memory

Ini Files (*.ini)

Register INI File (5/6)

- Check and click ok



Options for Target 'Target 1'

Device | Target | Output | Listing | User | C/C++ | A

(●) Use Simulator — Settings

☐ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File: .₩memory.ini

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watch Windows & Performance Analyzer

☑ Memory Display

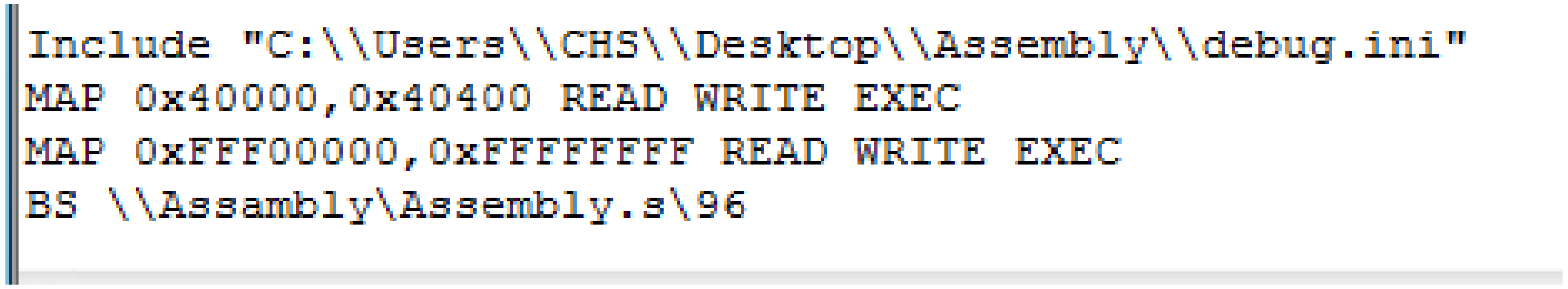
CPU DLL: SARM.DLL Parameter:

Dialog DLL: DARMP3.DLL Parameter:

OK Car

Register INI File (6/6)

- When you start debug mode, you can see command like picture



```
Include "C:\\Users\\CHS\\Desktop\\Assembly\\debug.ini"
MAP 0x40000,0x40400 READ WRITE EXEC
MAP 0xFFF00000,0xFFFFFFFF READ WRITE EXEC
BS \\Assambly\Assembly.s\96
```