

EISC–Studio3 Manual

Extendable Instruction Set Computer

Korean version 1.0

2008.xx.xx

Advanced Digital Chips Inc.

EISC-Studio3 Manual

©Advanced Digital Chips Inc.

All right reserved.

No part of this document may be reproduced in any form without written permission from Advanced Digital Chips Inc.

Advanced Digital Chips Inc. reserves the right to change in its products or product specification to improve function or design at any time, without notice.

Office

8th Floor, KookMin 1 Bldg.,
1009-5, Daechi-Dong, Gangnam-Gu, Seoul, 135-280, Korea.

Tel: +82-2-2107-5800

Fax: +82-2-571-4890

URL: <http://www.adc.co.kr>

목차

1. Installation.....	10
1.1 About the EISC-Studio3	10
1.2 About the Handbook	10
1.3 Printed Version	11
2. Installation.....	12
2.1 How to obtain EISC-Studio3	12
2.2 Requirement	12
2.3 Starting EISC-Studio3	오류! 책갈피가 정의되어 있지 않습니다.
3. Programs	13
3.1 프로젝트에 포함 될 기본적인 파일	13
3.2 Make and Makefile	13
3.3 elf 실행 파일	13
3.4 컴파일러 설치 확인	13
4. Overview.....	15
4.1 Project Explorer Window	15
4.2 Tags View	16
4.3 Text Editor Window	16
4.4 Output View Window	17
4.5 Menu Bar	18
4.5.1 File	18
4.5.2 Edit	19
4.5.3 View	19
4.5.4 Project.....	20
4.5.5 Build.....	20
4.5.6 Tools.....	20
4.5.7 Windows	20
4.6 Tool Bar	20
4.6.1 Standard Toolbar	21
4.6.2 Editor Toolbar	21
4.6.3 Build Toolbar	22
4.6.4 Full screen Toolbar	22
4.6.5 View Toolbar	22
4.6.6 Debug Toolbar	22

4.7	Status Bar.....	23
5.	Working with the Editor.....	24
5.1	Line Cut	24
5.2	Line Delete	24
5.3	Goto Line	24
5.4	Goto Matching Brace (Ctrl+]).....	25
5.5	Column Editor	25
5.6	Find.....	25
5.7	Find Next/Previous	26
5.8	Find in Files.....	26
5.9	Replace.....	28
5.10	Duplicate Current Line.....	28
5.11	Complete Word	29
5.12	Bookmark.....	29
5.13	Indentation.....	30
5.14	Move Up/Down Current Line.....	30
5.15	Change Case	30
5.16	Comment	31
5.17	View Menu	31
5.18	End of Line.....	35
6.	Project.....	36
6.1	New Project	36
6.2	Open Project & Close Project.....	37
6.3	Adding and Removing Files.....	38
6.4	Properties	40
7.	Build	44
8.	Debug.....	45
8.1	Target of Debugger.....	45
8.2	Start Debugger	45
8.3	Debug Options	45
8.4	Debugging interface.....	46
9.	Tools	48
9.1	Macro.....	48
9.2	Source Code Formatter.....	48
9.3	CSCOPE.....	49

9.4	Options	50
9.5	EISC Studio Command Prompt	53
9.6	External Tools	53
10.	Window	55
10.1	New Window	55
10.2	Cascade/Tile	55
11.	Help	56
11.1	About ES3...	56
11.2	Update Check	56
12.	그 외의 기능들	58
12.1	Project Explorer Window/Output View Window 위치 지정	58
12.2	New Horizontal Tab Group/New Vertical Tab Group	58
12.3	Move to Next Tab Group	59
12.4	활성화된 Text Editor Window로 이동 (Ctrl+Tap, Ctrl+F6)	60
12.5	각 Active Tool Window으로의 이동 (Alt+F7, Alt+F6)	60
13.	Example Program	62
13.1	Create New Project	62
13.2	Completed Project	63
13.3	Build Project	63
13.4	Debugging	64
14.	Help System	67

그림 목차

[그림 4-1] Project Tree	15
[그림 4-2] Tags View Window.....	16
[그림 4-3] Text View Window	16
[그림 4-4] Output View	17
[그림 4-5] Find(1) View	17
[그림 4-6] CSCOPE Result View	18
[그림 4-7] Tool Bar.....	20
[그림 4-8] Customize	21
[그림 4-9] Standard Toolbar.....	21
[그림 4-10] Editor Toolbar	21
[그림 4-11] Build Toolbar.....	22
[그림 4-12] Full screen Toolbar.....	22
[그림 4-13] View Toolbar	22
[그림 4-14] Debug Toolbar.....	22
[그림 5-1] Goto Line	24
[그림 5-2] Goto Matching Brace	25
[그림 5-3] Column Editor.....	25
[그림 5-4] Find	26
[그림 5-5] Find in Files	27
[그림 5-6] Find(1) View	28
[그림 5-7] Replace Text	28
[그림 5-8] Duplicate Current Line	29
[그림 5-9] Complete Word	29
[그림 5-10] Bookmark	30
[그림 5-11] Block Comment.....	31
[그림 5-12] Stream Comment.....	31

[그림 5-13] Show White Space and Tab	32
[그림 5-14] Show End of Line	32
[그림 5-15] Show Indent Guide	32
[그림 5-16] Fold All	33
[그림 5-17] Collapse the current level	34
[그림 5-18] Uncollapse the current level	34
[그림 5-19] Hide Lines.....	35
[그림 6-1] Project Type	36
[그림 6-2] General EISC Project.....	37
[그림 6-3] Startup Page	38
[그림 6-4] Add Exist File	39
[그림 6-5] Add New File	39
[그림 6-6] Project Properties (General Properties).....	40
[그림 6-7] Project Properties (Files)	41
[그림 6-8] Project Properties (Directory).....	42
[그림 6-9] Project Properties (Build Options).....	43
[그림 8-1] Debugger Options	45
[그림 8-2] Debugging 환경 (Source insight)	46
[그림 8-3] Debugging 환경 (Command Window)	47
[그림 9-1] Macro	48
[그림 9-2] CSCOPE Result View (1)	49
[그림 9-3] CSCOPE Result View (2)	49
[그림 9-4] CSCOPE Result View (3)	50
[그림 9-5] CSCOPE Result View (4)	50
[그림 9-6] Option (Editor Font & Color).....	51
[그림 9-7] Option (Editor Global Option)	51
[그림 9-8] Option (Editor Directories)	52
[그림 9-10] External Tools	53

[그림 9-11] Argument	54
[그림 10-1] New Window	55
[그림 11-1] About EISC Studio 3	56
[그림 11-2] Auto Updater.....	56
[그림 11-3] Auto Updater.....	57
[그림 11-4] Auto Updater.....	57
[그림 12-1] Project Explorer Window 위치 지정.....	58
[그림 12-2] New Horizontal Tab Group/New Vertical Tab Group	59
[그림 12-3] Move to Next Tab Group	59
[그림 12-4] 활성화된 Text Editor Window로 이동 (Ctrl+Tap)	60
[그림 12-5] 각 Active Tool Window으로의 이동 (Alt+F7).....	61
[그림 13-1] Create New Project.....	62
[그림 13-2] General EISC Project.....	62
[그림 13-3] Add New File/Add Exist Files	63
[그림 13-4] Build-ae32000_hello	63
[그림 13-5] Debugging 환경 (Source Insight).....	64
[그림 13-6] Debugging Start.....	65
[그림 13-7] Debugging.....	65

1. Installation

1.1 About the EISC-Studio3

EISC-Studio3는 Windows 환경에서 EISC CPU를 사용하는 개발자들을 위한 통합 개발 환경이다. EISC-Studio3는 EISC CPU의 특징을 최대한 살릴 수 있는 기능들로 구성되어 있다.

Feature

- Integrated Development Environment (IDE)
 - Works in Windows 2000/XP/VISTA
 - Friendly User Interface like Microsoft Visual-Studio
 - Source Edit/Compile
 - Automatic Windows docking
 - Project file Tree view
 - Class / Function / Global variable Tree view
 - Auto Generate Makefile
 - Graphical Symbol Browser
 - Support All EISC compilers

- Convenient Embedded Editor
 - Syntax highlighting
 - Column Editor
 - Code Completion
 - Smart Auto-indent
 - Find and replace facilities
 - Bookmark
 - Move Up/Down Current Line
 - Support for collapsible nodes (outlining)
 - Supports various fonts and styles

1.2 About the Handbook

이 사용자 매뉴얼은 EISC-Studio3를 사용하는 사용자들에게 제공된다. 정확한 프로그래밍 정보를 원한다면 EISC Software Developer Guide를 참조하기 바란다.

이 문서는 다음과 같이 구성되어 있다.

-
- EISC-Studio3 설치 방법
 - Programs 프로그램을 하기 위해 알아 두어야 할 사항들
 - Overview EISC-Studio3의 전체적인 구성과 기능
 - Working with the Editor EISC-Studio3의 editor 기능과 사용
 - 프로젝트 생성
 - 프로그램 빌드 하기
 - Sample Program

1.3 Printed Version

이 문서는 EISC-Studio3와 함께 설치 된다.

2. Installation

2.1 How to obtain EISC-Studio3

EISC-Studio3는 홈페이지(<http://www.adc.co.kr>)에서 다운로드 받거나 ADChips에서 제공되는 CD로 얻을 수 있다.

2.2 Requirement

지원 O/S: Windos200, Windows XP, Windows Vista,

3. Programs

성공적으로 EISC-Studio3를 설치하고 시작했다면, 프로그램 작성시에 필요한 파일들과 프로젝트 빌드 후 생성되는 파일들에 대해 알아본다.

3.1 프로젝트에 포함 될 기본적인 파일

- crt0.S: Stack
 - pointer register 초기화
 - 시스템 레지스터 설정(option)
 - Data 영역 Copy (필요시 : 저장영역이 ReadOnly 영역일 경우)
 - 초기화 되지 않은 전역변수 초기화
- Linker Script: 메모리 영역 설정, Stack 설정
 - Core-name.x (se3208.x, ae32000.x ...) 형태로 제공되므로, 이 파일을 기본으로 필요한 기능을 추가 또는 수정하여 사용한다. (확장자는 상관없음)

3.2 Make and Makefile

프로젝트를 생성하게 되면 "Makefile.mk"의 파일명으로 Makefile이 생성된다. EISC-Studio3는 이 Makefile을 이용하여 프로젝트를 컴파일하거나 빌드한다. Makefile에 대한 간단한 설명은 EISC Software Developer Guide를 참조하기 바란다.

3.3 elf 실행 파일

EISC Compiler 로 프로젝트를 빌드하면 최종결과 파일은 ELF (Executable and Linkable Format) 형식이다. ELF 파일을 생성 한 이후 Binary 파일을 추출 하여 사용 할 수 있다. ELF 의 구조는 여기서 다루지 않는다.

3.4 컴파일러 설치 확인

- ① EISC-Studio3에서 우선으로 참조되는 컴파일러
 - EISC-Studio3에서 컴파일을 할 경우 우선 컴파일러 설치가 잘 되었는지 확인한다. 컴파일러 설치 경로는 EISC-Studio3를 설치한 경로 안에 있다. 예를 들어 "C:\Program Files\ADChips"에 설치 했을 경우 그 안에 "EISC Studio 3\local" ("C:\Program Files\ADChips\EISC Studio 3\local")에 컴파일러가

설치되어있다. EISC-Studio3에서는 이 경로에 있는 컴파일러를 우선으로 참조하여 실행한다. 우선순위는 도구>option 에서 변경 할 수 있다.

② DOS 또는 Cygwin shell 상에서 참조되는 컴파일러

- System path에 잡혀있는 컴파일러를 우선으로 참조하여 실행되기 때문에 Cygwin의 설치가 c:\cygwin 일 경우 "c:\cygwin\usr\local\bin" (EISC software Tool chain 의 설치 디렉터리)과 "c:\cygwin\bin" 두 경로를 추가 한다.
- System path 수정방법
MS windows 2000 및 NT 계열 의 경우 "제어판->시스템->고급->환경 변수 -> path"에서 수정 할 수 있다. Path를 추가 시켰다면 다음과 같이 확인한다.

- 1) Dos 프롬프트 또는 Cygwin창을 하나 띄운다.
- 2) ae32000-elf-gcc를 실행시켜본다.
- 3) "No input files" 라는 메시지가 뜬다면 컴파일러가 정상적으로 동작 하는 것이다.
- 4) 만약 ae32000-elf-gcc가 실행 되지 않는다면 ae32000-elf-gcc를 설치한 폴더를 살펴본다. Default로 설치 되는 경로는 "c:\cygwin\usr\local\bin" 이다.
이곳에 ae32000-elf-gcc가 존재하지 않는다면 컴파일러 설치를 다시 한다. 만약 설치 경로에 ae32000-elf-gcc가 존재하는데 실행되지 않는다면, System path 가 정확히 설정되어 있는지 확인한다.

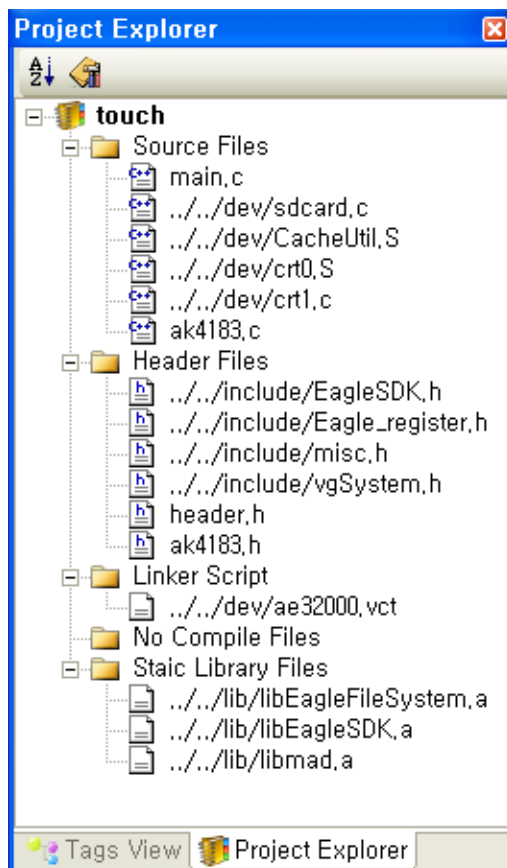
4. Overview

EISC-Studio3는 사용의 편리함을 위해 User Interface를 다음과 같이 구분한다.

- Project Explorer Window, Tags View Window, Text Editor Window, Output window, Menu Bar, Tool Bar, Status Bar.

이 장에서는 각 window 기능과 특징들에 대해 알아 본다.

4.1 Project Explorer Window




Project에 포함되는 파일들을 Source Files, Header Files, Linker Script File, No Compile Files, Static Library Files로 분류해 각각의 파일을 보여준다.


각각의 파일 분류명 위에 커서를 두고 마우스 오른쪽 버튼을 누르면 "Add Exist Files"와 "Add New File" 메뉴를 사용 할 수 있다.

"Add Exist Files" 메뉴는 바로 "열기" dialog를 열어 해당 파일들을 추가 할 수 있고 "Add New File" 메뉴는 새로운 파일을 만들어 원하는 디렉터리에 저장하면 자동으로 분류되어 프로젝트에 바로 추가된다.

파일을 빠르게 두 번 클릭하면 Text Editor Window에 파일이 열린다.

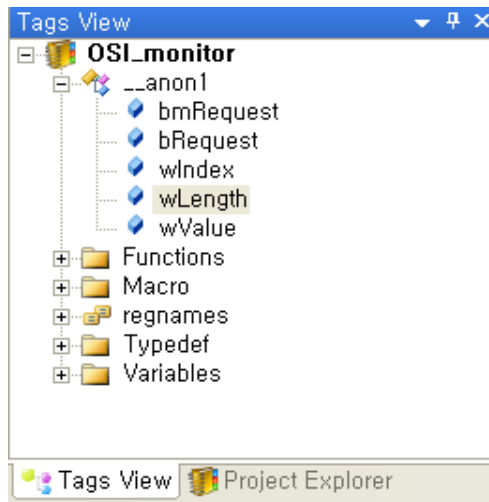
[그림 4-1] Project Tree

 : 각각의 Files안에 있는 파일들을 알파벳 순으로 정렬하여 나타냄.

 : "Project Properties" dialog가 나타난다. 이 dialog는 프로젝트에 적용할 옵션들을 설정, 변경할 수 있게 한다. 이에 대한 자세한 설명은 chapter 6장을 참조 하길 바란다.

4.2 Tags View

Project에서 활성화된 파일들을 구조체, Function, Typedef, Variables 등 각 분류에 맞게 나열해 보여준다.



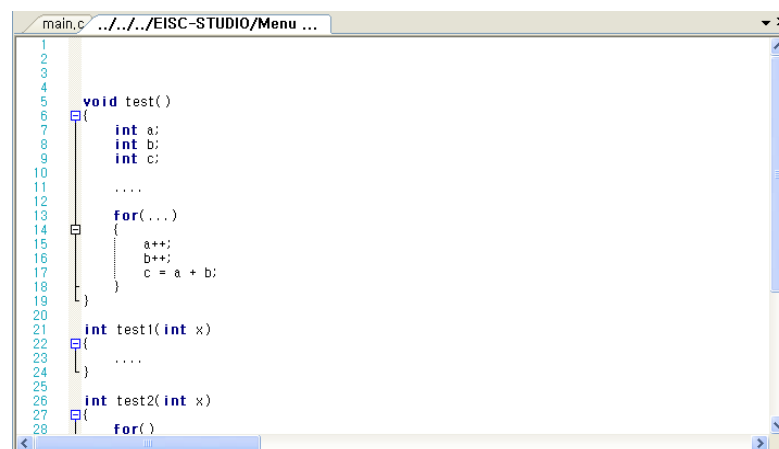
이동하고 싶은 함수나 변수 등을 두 번 클릭하면 Text Editor Window에 파일이 활성화 되면서 해당 라인으로 이동하게 된다.

Tags View Window는 파일을 저장 할 때마다 Refresh된다.

[그림 4-2] Tags View Window

4.3 Text Editor Window

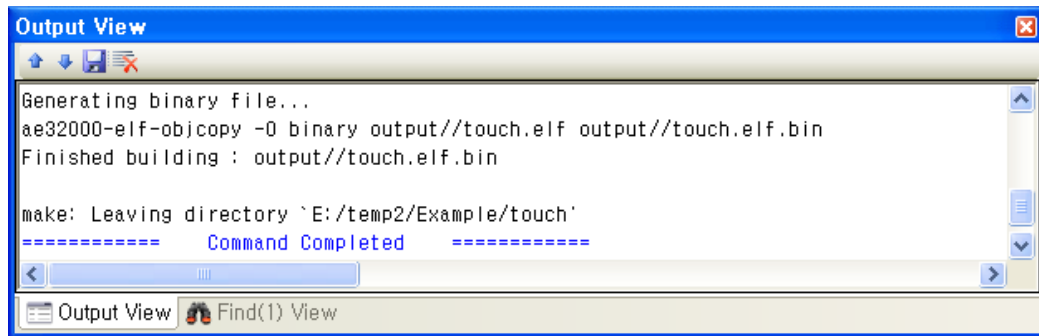
Text View Window는 사용자가 원하는 만큼 파일을 열어 코드를 쓰고 수정할 수 있는 window이다. 예약어, 주석, 문자열등 파일을 이루는 코드들을 분류해 color와 font를 달리해서 보여주므로 editing을 손쉽게 할 수 있다. Text View Window에서 사용할 수 있는 자세한 기능들은 Chapter 5. Working with the Editor에서 다룬다.



[그림 4-3] Text View Window

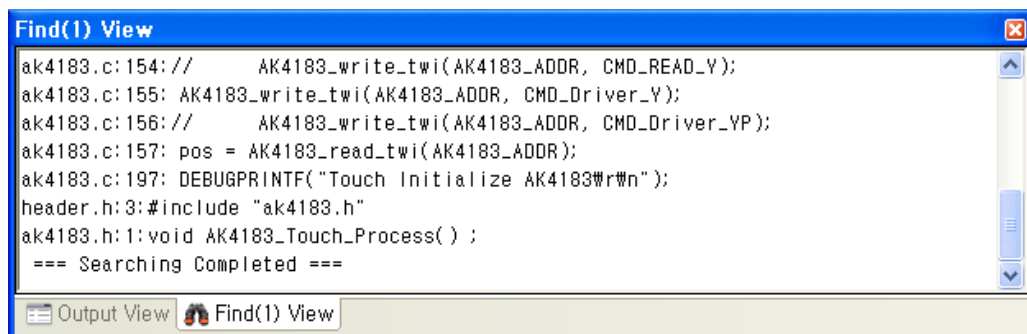
4.4 Output View Window

Output View Window는 Output View와 Find View, CSCOPE Result View로 구성된다.



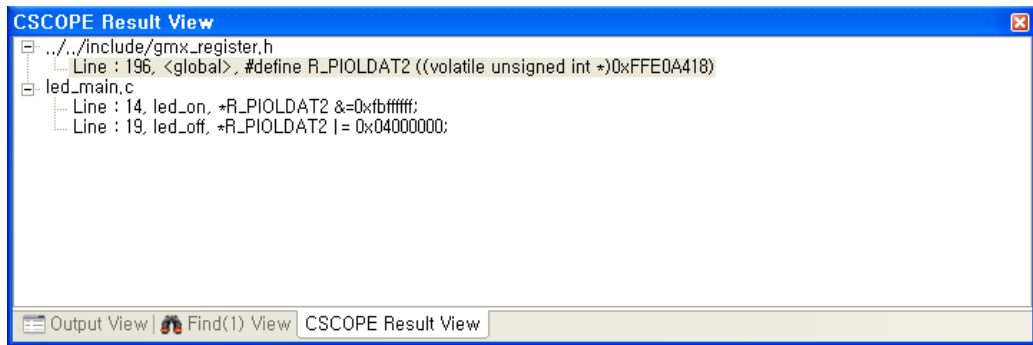
[그림 4-4] Output View

- Output View: 컴파일 과정에서 수행되는 명령들, 에러 메시지와 에러 코드의 위치를 보여준다. 에러 메시지를 클릭하게 되면 커서가 에러 메시지에 해당하는 코드로 이동된다.
 - : (Goto Previous Error Line), (Goto Next Error Line) 단축키:F4
 - : Save All Message as file (해당 project 폴더 안에 build_log.txt 파일이 생성됨)
 - : Clear All Message



[그림 4-5] Find(1) View

- Find View: Text Search 기능 – Find, Replace, Find in Files를 통해 검색한 결과를 보여준다. 검색된 결과를 더블 클릭하면 커서가 해당코드로 이동한다.



[그림 4-6] CSCOPE Result View


- CSCOPE Result View: 실행된 project 폴더 안에서만 찾고 찾은 정보는 그림 4-6과 같이 보여준다. 자세한 내용은 8장을 참조하기 바란다.

4.5 Menu Bar

Menu Bar 중 File과 View에 대한 내용을 설명하고 각 메뉴에 대하여 자세히 설명된 chapter의 내용을 간단히 소개한다.


4.5.1 File

새로운 파일, 프로젝트, 열기, 저장, 프린트 하는 기능


- New
 - Files (Ctrl+N, )

선택한 파일을 연다.
 - Close


Text Editor window에 활성화 되어 있는 파일을 닫는다.
 - Close All

Text Editor window에 열려 있는 모든 파일을 닫는다.
 - Save (Ctrl+S, )

Text Editor window에 활성화 되어 있는 파일을 저장한다.
 - Save As

Text Editor window에 활성화 되어 있는 파일을 다른 이름으로 저장한다.
 - Save All, 

Text Editor window에 열려 있는 모든 파일을 저장한다.



- Open Project (Ctrl+Shift+O)
선택한 프로젝트를 연다.
- Save Project
현재 활성화 되어있는 프로젝트를 저장한다.
- Close Project
현재 활성화 되어있는 프로젝트를 닫는다.
- Print (Ctrl+P, )
Text Editor window에 활성화 되어 있는 파일을 프린트한다. Header와 footer에 파일이름, 날짜와 시간, 페이지, 에디칩스 홈페이지 주소가 출력된다.
- Page Setup
인쇄될 페이지의 용지, 방향, 여백 등을 조절한다.
- Recent Files
최근 열었던 파일 목록을 리스트로 보여주고 선택하면 Text Editor window에 연다.
- Exit
EISC-Studio3를 종료한다.

4.5.2 Edit

Text Editor Window에서 코드를 editing할 때 사용하는 메뉴이다. 잘라내기, 복사, 붙이기, do, undo 등의 일반적인 기능들과 Match Brace, Complete word, Block comment/uncomment, stream comment, increase/decrease indentation, Line number 와 Zoom 등 많은 기능을 포함한다. 그리고 기존에 있던 Search 기능을 포함했다.

4.5.3 View

EISC-Studio3를 구성하고 있는 window를 활성화 시키고, 전체적인 화면 테마를 결정 및 Toolbar를 설정하는 기능을 포함한다.

- Navigate Backward (Ctrl+Left Arrow, ) : 선택한 함수나 변수의 원형으로 가서 다시 선택한 함수나 변수가 있는 곳으로 돌아갈 수 있게 한다.
- Navigate Forward (Ctrl+Right Arrow, ) : 다시 위에서 선택한 함수나 변수의 원형으로 갈 수 있다.
- Start Page: Recent Project를 보여주는 창이 뜬다. Create New Project와 Open Other Project를 실행할 수 있다.

- cscope: CSCOPE Result view 창으로 간다.

4.5.4 Project

프로젝트에 새 파일 및 기존 파일을 추가 하기, 현재 활성화 되어 있는 프로젝트의 속성을 설정하는 기능을 포함한다.

4.5.5 Build

프로젝트에서 실행파일, binary 파일, disassemble 파일 등을 생성, Makefile등 프로젝트에 관련된 파일을 다시 생성하고 object 파일과 실행파일 등을 삭제하는 기능을 포함한다. 또한 Full Build Message기능을 추가하여 컴파일 옵션 보는 것을 선택하여 Output View창에 나타낼 수 있다.

4.5.6 Tools

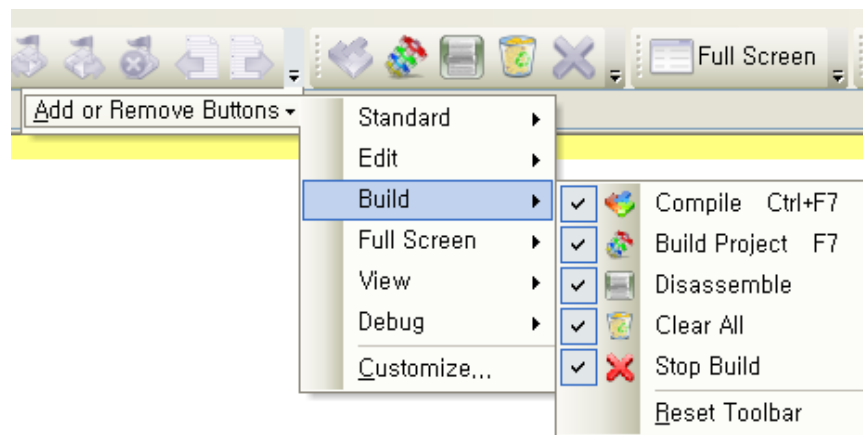
Macro 기능 및 소스 코드의 형식을 설정, 에디터와 관련된 폰트와 컬러 등을 설정하는 기능을 포함한다. 그리고 위에서 설명한 cscope기능도 포함되어 있다.

4.5.7 Windows

Text Editor window에서 열리는 파일들의 display 형태를 변경한다.

4.6 Tool Bar

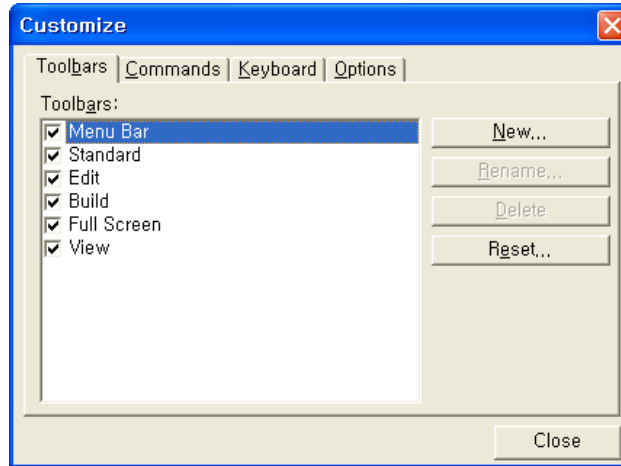
각 Toolbar에 아이콘을 추가/삭제 할 수 있다.



[그림 4-7] Tool Bar

각 Toolbar 오른쪽을 누르면 [그림 4-7]과 같이 서브 메뉴로 각 Toolbar에 포함시킬

기능들의 아이콘을 선택해 추가/삭제 할 수 있다. 또한 Menubar의 View->Toolbars를 누르면 위의 Standard, Edit등의 메뉴에 포함된 기능들의 아이콘 전부를 추가/삭제 할 수 있다.



[그림 4-8] Customize

- Customize
 - Toolbars tab: 각각의 Toolbar display 설정
 - Commands tab: Toolbar에 아이콘 추가/삭제 설정
 - Keyboard tab: 각각의 기능들에 대한 단축키 설정
 - Options tab: 아이콘의 크기 및 Toolbar Tip에 대한 설정

4.6.1 Standard Toolbar

파일 생성, 열기, 저장, 프린트를 할 수 있다.



[그림 4-9] Standard Toolbar

4.6.2 Editor Toolbar

Editor에 관련된 기능들을 Toolbar의 아이콘으로 이용할 수 있다.



[그림 4-10] Editor Toolbar

4.6.3 Build Toolbar

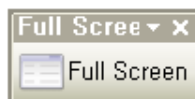
프로젝트 또는 파일 컴파일에 관련된 기능들을 Toolbar의 아이콘으로 이용할 수 있다.



[그림 4-11] Build Toolbar

4.6.4 Full screen Toolbar

Text Editor window를 전체화면으로 볼 수 있는 기능을 Toolbar의 아이콘으로 이용할 수 있다.



[그림 4-12] Full screen Toolbar

4.6.5 View Toolbar

EISC-Studio3를 구성하는 각 window를 활성화 시키는 기능을 Toolbar의 아이콘으로 이용할 수 있다.



[그림 4-13] View Toolbar

4.6.6 Debug Toolbar

Debugger start 기능을 Toolbar의 아이콘으로 이용할 수 있다.



[그림 4-14] Debug Toolbar

4.7 Status Bar

현재 선택한 메뉴의 기능, line number 그리고 column number 등의 정보를 status bar에 보여준다.

5. Working with the Editor

EISC-Studio3는 다음과 같은 editor 기능을 가지고 있다.

- Line control
- Syntax highlighting
- Column Editor
- Find and replace facilities
- Code Completion
- Smart Auto-indent
- Bookmark
- Support for collapsible nodes (outlining)

5.1 Line Cut

잘라낸 라인의 data가 메모리 버퍼에 저장됨으로 다시 붙이면 (paste, Ctrl+v) 그 라인의 내용이 복구된다.

- 실행: Edit -> Line Cut, Ctrl+L

5.2 Line Delete

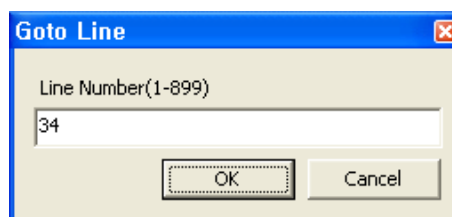
잘라낸 라인의 data가 삭제된다. Paste를 해도 잘라낸 라인이 복구되지 않는다.

- 실행: Edit -> Line Delete, Ctrl+Shift+L

5.3 Goto Line

Text Editor window에 활성화 되어 있는 파일에서 원하는 Line으로 이동한다. [그림 5-1]과 같은 dialog가 실행되면, 원하는 Line수를 입력하고 "OK" 버튼을 누르면 해당 Line으로 이동한다. 활성화된 파일의 Total Line은 dialog의 괄호 안에 표시된다.

- 실행: Edit -> Goto Line, Ctrl+G

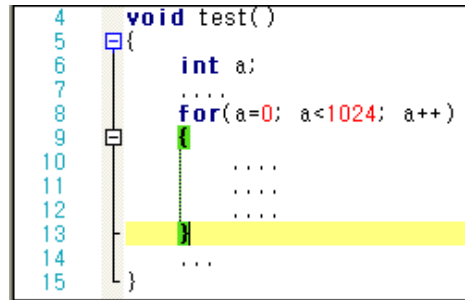


[그림 5-1] Goto Line

5.4 Goto Matching Brace (Ctrl+J)

'(', ')', '[', ']', '{', '}' 여기에 해당하는 brace 들에 커서를 가져가면 [그림 5-2] 같이 짝이 맞는 brace 와 함께 highlighting 된다. brace 한쪽에 커서를 두고 Goto Matching Brace 기능을 이용하면 짝이 되는 brace로 이동할 수 있다.

- 실행: Edit -> Goto Matching Brace, Ctrl+J

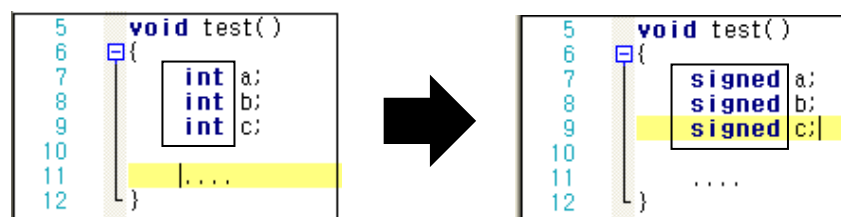


[그림 5-2] Goto Matching Brace

5.5 Column Editor

소스 코드의 특정 부분에 사용자가 원하는 단일 코드를 여러 행을 삽입하거나 수정하고자 할 때 코드를 행 단위로 바꾸어 주는 기능이다. Column 단위 선택을 위해선 Alt 키와 마우스 왼쪽을 동시에 누른 상태에서 사용자가 원하는 부분을 드래그 하거나 Shift + Alt 를 누른 채 방향키를 이용해 원하는 부분을 선택하면 된다.

- 실행: Edit -> Column Editor, Alt+C



[그림 5-3] Column Editor

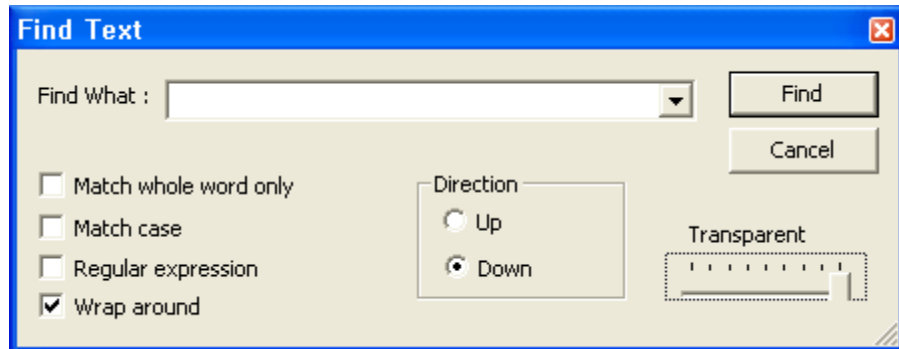
5.6 Find

선택된 파일 내에서만 Find 기능이 적용된다.

- ① 메뉴의 Edit → Find, "Ctrl + F" 또는  을 선택한다.
- ② [그림 5-4]와 같은 "Find Text" dialog의 "Find what" 부분에 찾고자 하는 패턴을

넣는다.

- ③ “Find” 버튼을 누르면 커서가 있던 부분부터 패턴을 찾아가 반전 시킨다.



[그림 5-4] Find

※ **Transparent** 기능은 **Dialog Box** 뒤쪽의 글씨를 보여주기 위해 추가된 기능이다.

- Match whole word only
 - “Find What”에 넣은 Text 패턴과 완전히 일치하는 것을 찾는다.
- Match case
 - “Find What”에 넣은 Text 패턴을 대/소문자 구분하여 찾는다.
- Regular Expression
 - “Find What”에 정규 표현 식을 사용할 수 있다.
- Wrap around
 - Text 패턴을 순환해서 찾는다.
- Direction
 - Text 패턴을 찾는 방향을 설정한다

5.7 Find Next/Previous

“Find Text” dialog 를 종료한 후 이전에 찾은 패턴 사이를 이동할 때 사용하는 기능이다.

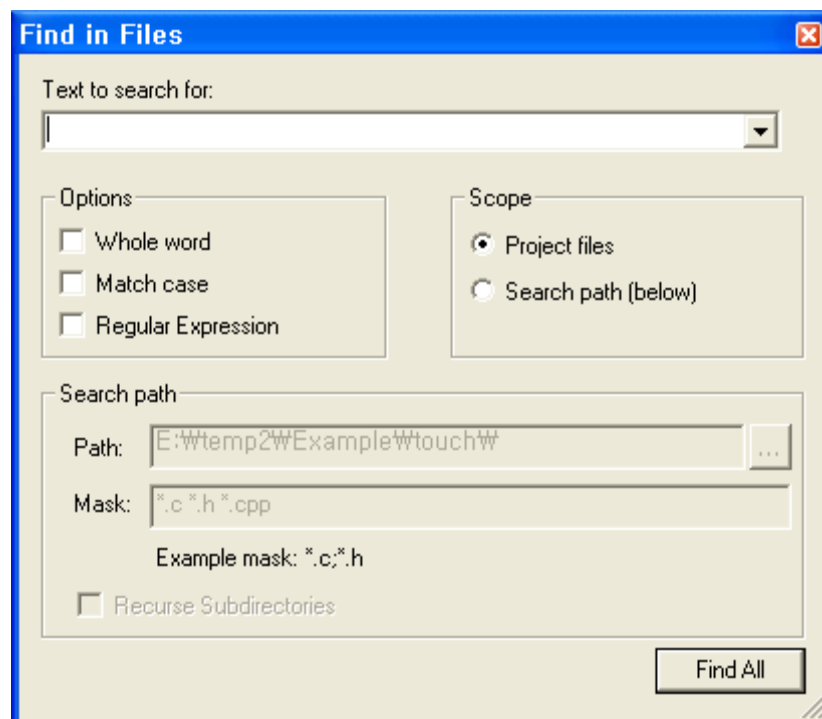
- 실행: Edit -> Find Next, F3
- 실행: Edit -> Find Previous, Shift+F3

5.8 Find in Files

선택한 Project 나 사용자가 지정하는 디렉터리에서 동일한 패턴을 찾기 위한 기능이다.

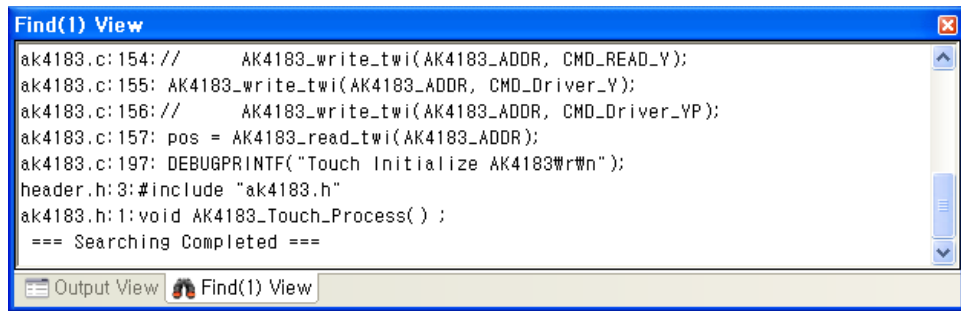
- ① 메뉴의 Edit ➔ Find in Files 또는 “Ctrl+Shift+F” 를 실행한다.

- ② [그림 5-5]같은 dialog의 "Text to search for" 부분에 찾고자 하는 text 패턴을 넣고 Scope를 선택한다.
- A. Project files: 현재 활성화 되어있는 Project 내에서 검색을 진행한다.
 - B. Search path(below): 사용자가 지정하는 디렉터리에서 검색을 진행한다.
 - Path: 사용자 지정 경로 설정
 - Mask: 사용자 지정 파일 type 설정
- ③ "Find All" 버튼을 누르면 [그림 5-6]과 같이 output window → Find View 을 통해 검색한 결과를 볼 수 있다.
- ④ output window에 출력된 라인을 더블 클릭하면 해당 파일을 Text Editor window에 활성화 시키고 패턴이 있는 라인에 커서를 위치시킨다.



[그림 5-5] Find in Files

- Options
 - Whole word: Text 패턴과 완전히 일치하는 것을 찾는다.
 - Match case: Text 패턴을 대/소문자 구분하여 찾는다.
 - Regular Expression: 정규 표현 식을 사용할 수 있다.
- Scope
 - Project files: 현재 열린 Project 내의 모든 파일을 검색을 한다.
 - Search path(below): 사용자가 원하는 디렉터리 아래의 모든 파일을 검색한다.

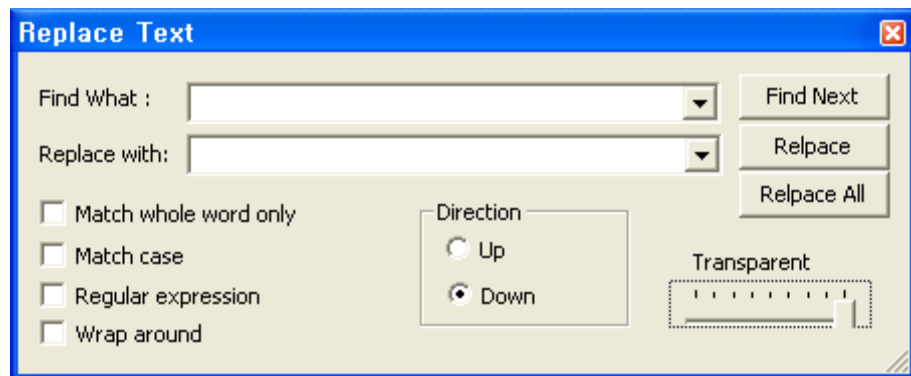


[그림 5-6] Find(1) View

5.9 Replace

선택된 파일 내에서만 Replace 기능이 적용된다.

- ① 메뉴의 Edit ➔ Replace 또는 "Ctrl+H"를 실행한다.
- ② [그림 5-7]과 같은 "Replace Text" dialog의 "Find what" box에 찾고자 하는 text 패턴을 넣고, "Replace with" box에 변경 시킬 패턴을 넣어준다.
- ③ "Find Next" 버튼을 눌러서 찾고자 하는 패턴이 찾아지면, "Replace" 버튼을 눌러 변경 시킬 패턴으로 바꾼다. 이러한 과정을 자동으로 해주는 것이 "Replace All" 기능이다.



[그림 5-7] Replace Text

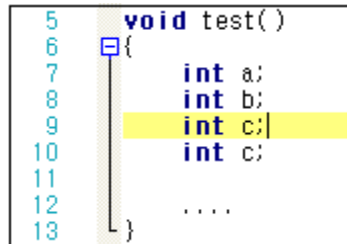
※ Transparent 기능은 Dialog Box 뒤쪽의 글씨를 보여주기 위해 추가된 기능이다.

Replace 의 각 옵션들은 Find 의 옵션과 동일하다.

5.10 Duplicate Current Line

사용자가 원하는 소스 코드 라인을 바로 복사하는 기능이다.

- 실행: Edit -> Duplicate Current Line, Ctrl+D

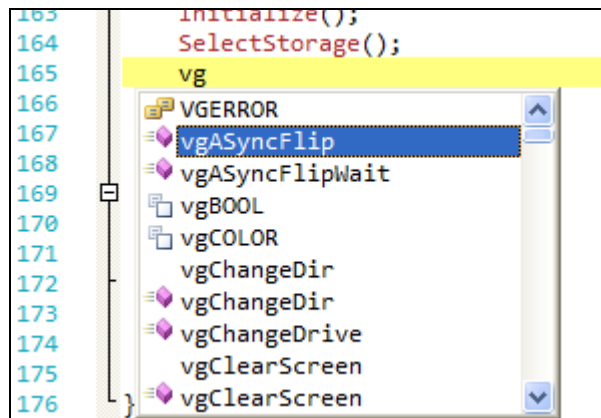


[그림 5-8] Duplicate Current Line

5.11 Complete Word

현재 입력중인 단어를 완성시켜 주는 기능이다. [그림 5-9]와 같이 단어의 첫 몇 자를 입력하면 complete word 기능은 자동적으로 해당 단어를 찾아 표시해준다. 해당 단어가 여러 개라면 리스트가 뜨고 하나라면 곧바로 단어가 완성된다. 현재 작업중인 파일에서만 적용된다.

- 실행: Edit -> Complete Word, Ctrl+Space






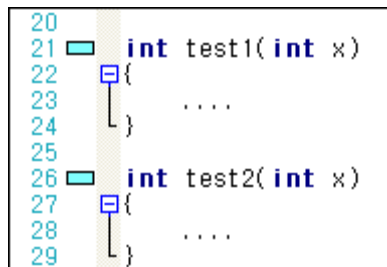
[그림 5-9] Complete Word

5.12 Bookmark

- Toggle Bookmark, (Ctrl+F2, )



원하는 소스 라인에 Bookmark를 [그림 5-10]과 같이 생성하거나 이미 생성된 곳이라면 삭제한다.

- Next Bookmark, (F2, )
 설정된 bookmark 들 중 다음 bookmark로 이동한다.
- Previous Bookmark, (Shift+F2, )
 설정된 bookmark 들 중 이전 bookmark로 이동한다.
- Clear All Bookmark, ()
 Text Editor window에 활성화 되어 있는 파일에 생성된 Bookmark를 모두 삭제한다.



[그림 5-10] Bookmark

5.13 Indentation

- Increase Indentation ()
 선택한 코드에 들여쓰기를 늘리는 기능이다.
- Decrease Indentation ()
 선택한 코드에 들여쓰기를 줄이는 기능이다.

5.14 Move Up/Down Current Line


사용자가 선택한 소스 코드 Line 의 위치를 위/아래로 변경한다.

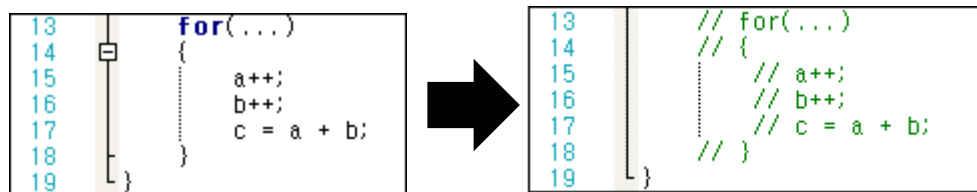
- 실행: Edit -> Move up Current Line, Ctrl+Shift+Up Arrow
- 실행: Edit -> Move down Current Line, Ctrl+Shift+Down Arrow

5.15 Change Case

- Upper Case
 사용자가 선택한 영역의 영문을 모두 대문자로 변경한다.
- Lower Case
 사용자가 선택한 영역의 영문을 모두 소문자로 변경한다.

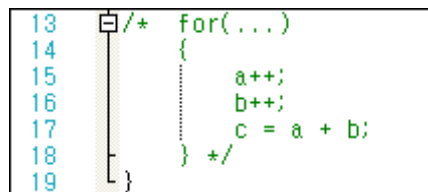
5.16 Comment

- Block Comment/Uncomment ()
 - Block Comment
주석 처리 할 코드를 block 선택한 후 이 기능을 사용하면 "//" 를 이용해 [그림 5-11]과 같이 주석처리 한다.
 - Block Uncomment
"Block Comment" 를 적용해 주석 처리한 코드를 block 선택한 후 이 기능을 사용하면 주석처리를 해제한다.



[그림 5-11] Block Comment

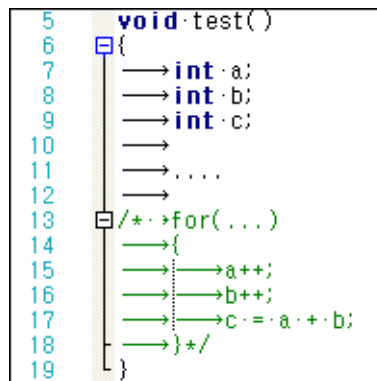
- Stream Comment
코드를 block 설정한 후 이 기능을 사용하면 "/*..... */" 를 이용해 [그림 5-12]와 같이 주석 처리 한다.



[그림 5-12] Stream Comment

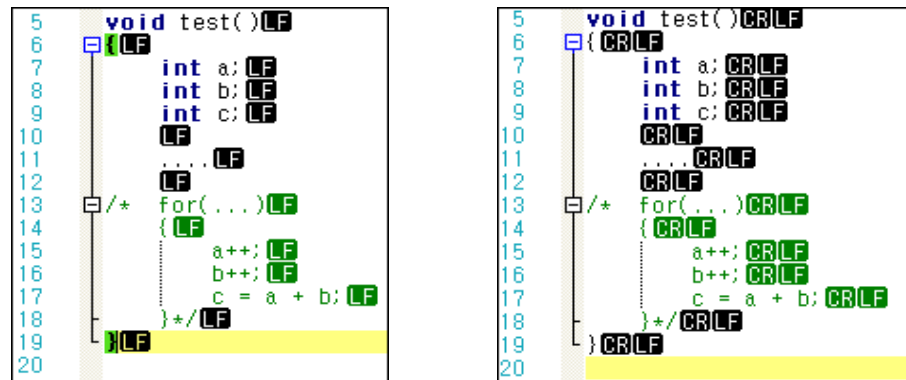
5.17 View Menu

- Show White Space and Tab
Text Editor window에 활성화 되어 있는 파일에 [그림 5-13]과 같이 Tab이 사용된 부분은 "→" 로, Space 가 사용된 부분은 "." 로 나타낸다.



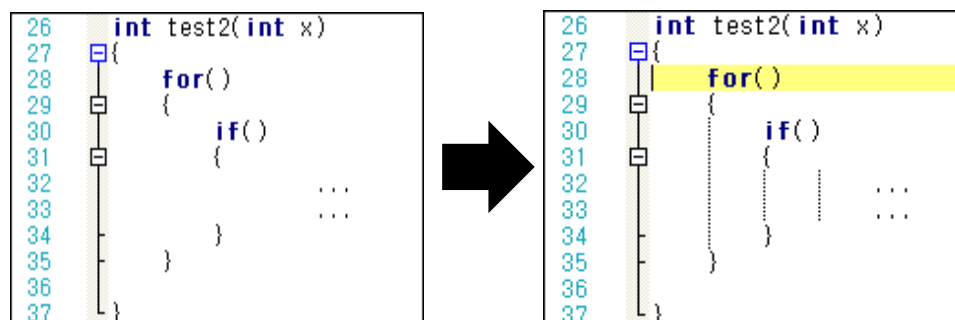
[그림 5-13] Show White Space and Tab

- Show End of Line
End Line의 Character를 나타낸다.



[그림 5-14] Show End of Line

- Show Indent Guide
Matching Brace 간의 연결 및 Indentation을 나타내는 선을 보여준다.



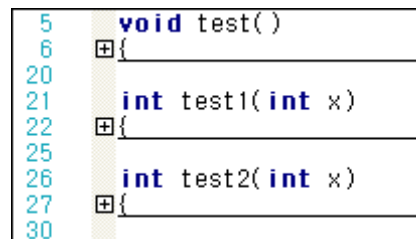
[그림 5-15] Show Indent Guide

- Zooming
Text Editor window에 활성화 되어 있는 파일에만 적용되는 기능이다. 폰트의 특성에 따라 확대, 축소 적용이 한 단계 식 적용되지 않고 여러 번 실행해야 하는 경우도 있다.
- Zoom In
화면을 확대한다. (ctrl + 마우스 scroll up)
- Zoom Out
화면을 축소한다. (ctrl + 마우스 scroll down)
- Folding
Code folding 기능은 용량이 큰 소스 파일에서 작업을 할 때 특정 영역을 숨기거나 현재 작업중인 부분만을 보여주는 기능이다. 이 기능은 사용자에게 복잡함을 덜어주고 많은 양의 코드를 효율적으로 관리하게 해준다.

Folding 된 코드를 복사, 잘라내기를 한 후 붙여넣기 작업을 수행할 경우 코드는 Unfolding 된 상태가 된다.

- Fold All

현재 작업중인 소스 코드의 모든 Folding Points 의 코드를 Folding 한다. Folding Point 내의 하위 Folding Point 도 같이 Folding 된다.

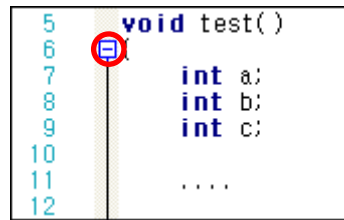


[그림 5-16] Fold All

- Collapse the current level

현재 작업중인 소스 코드의 단락에 속해있는 Folding Point 의 코드만을 Folding 한다.

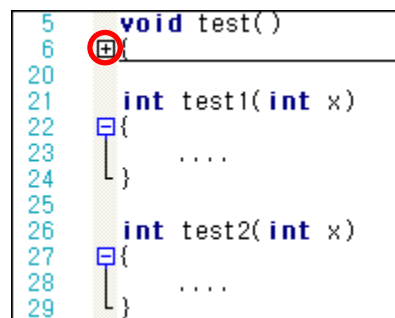
또한 아래 그림에서 빨간색으로 표시된 Folding Point를 Click 해 현재 단락을 Folding 할 수 있다. 상위 Folding Point를 Folding 할 때 내부의 하위 Folding Point 도 같이 Folding 할 경우 Ctrl 을 누른 채 마우스로 Click 하면 하위에서도 Folding 이 적용된다.



[그림 5-17] Collapse the current level

- Uncollapse the current level

현재 작업중인 소스 코드의 단락에 속해있는 Folding Point 의 코드만을 Unfolding 한다. 또한 아래 그림에서 빨간색으로 표시된 Folding Point를 Click 해 현재 단락을 Unfolding할 수 있다. 상위 Folding Point에서 Unfolding할 때 내부의 하위 Folding Point도 같이 Unfolding 할 경우 Ctrl을 누른 채 마우스로 Click 하면 하위에서도 Unfolding 이 적용된다.



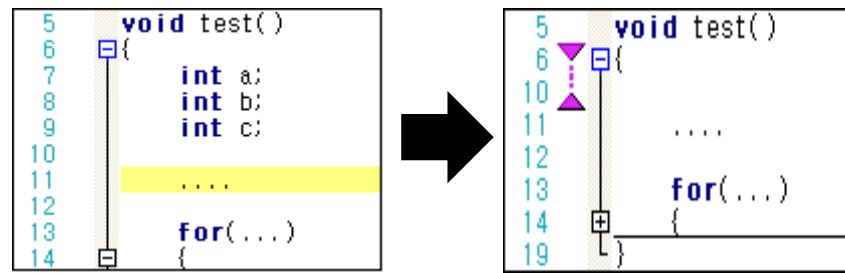
[그림 5-18] Uncollapse the current level

- Unfold All

현재 작업중인 소스 코드의 모든 Folding Points 의 코드를 Unfolding한다.

- Hide Lines

Folding Point 내의 코드가 많을 때 사용자가 원하는 Line 을 선택해 Folding 한다. Hide Line 시에 오른쪽 그림같이 감춰진 구간에는 표시가 나타난다. Hide Line 을 해제할 때는 표시된 부분을 마우스로 Click 하면 된다.



[그림 5-19] Hide Lines

5.18 End of Line

파일을 DOS 모드로 지정하면 라인 끝에 CR(Carriage Return)과 LF(Line Feed) 가 붙게 되고, UNIX 모드로 지정하면 LF만 붙게 된다.

6. Project

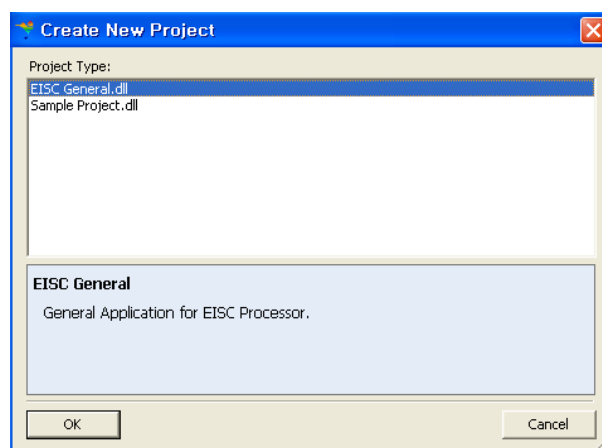
EISC-Studio3를 이용해 프로젝트 생성, 관리 시 필요한 기본적인 사항들은 다음과 같다.

- 구성 파일들
 - *.epx: 프로젝트 파일, 삭제하지 않도록 주의 한다.
 - Makefile.mk: EISC-Studio3에서 사용하는 Makefile
- 이전 버전과의 호환

이전 버전의 EISC-Studio2로 생성한 프로젝트도 EISC-Studio3에서 그대로 사용할 수 있다. 이전 버전의 프로젝트를 열 경우 "File → Open → Project" 를 통해서 열어야 한다. 프로젝트를 선택하면 이전 버전에서 생성된 프로젝트를 EISC-Studio3에서 사용할 수 있도록 새로운 포맷으로 자동 변경된다. 이전의 프로젝트 파일은 그대로 남게 되고 EISC-Studio3에서 사용되는 *.epx 파일이 생성된다.

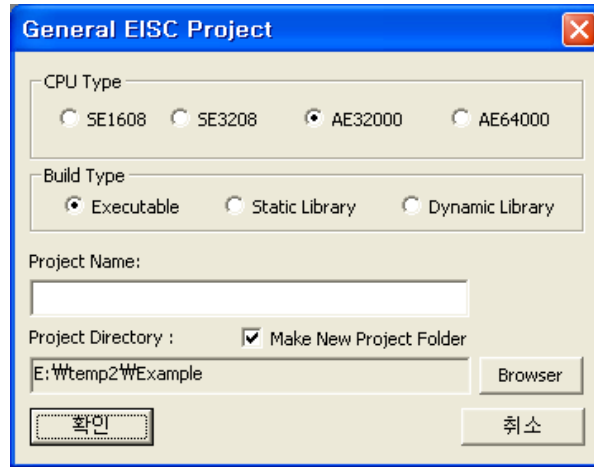
6.1 New Project

- ① 메뉴의 File → New → Project 또는 Start Page의 "Create New Project" 버튼을 선택한다. [그림 6-1]과 같이 "Create New Project" Dialog가 실행되면 Project Type 을 선택한다.
 - EISC General.dll: EISC Processor에 대한 일반적인 어플리케이션을 만들 때 선택함.
 - Sample Project.dll: EISC-Studio3에서 제공하는 Sample 어플리케이션인 "Hello World"실행 시 선택함.



[그림 6-1] Project Type

- ② [그림 6-2]와 같이 "General EISC Project" Dialog에 다음 사항들을 설정한다.



[그림 6-2] General EISC Project

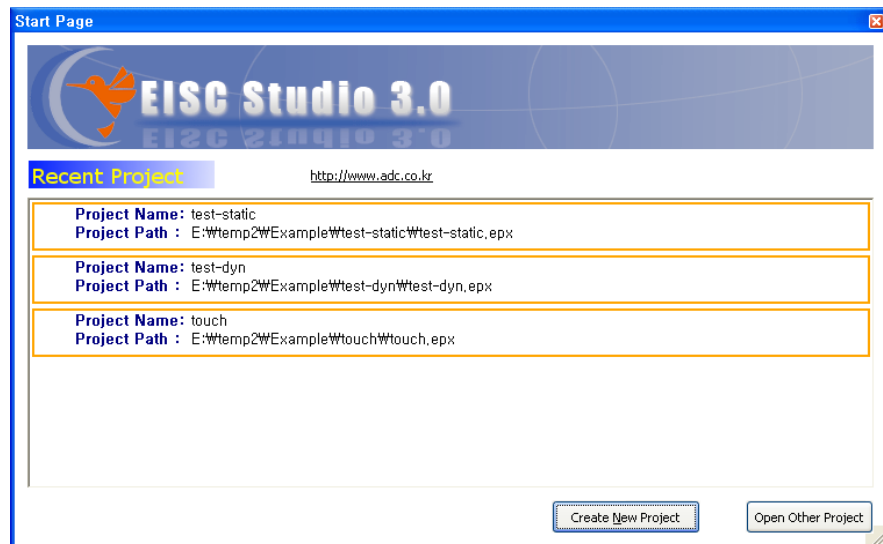
- CPU Type
원하는 Target type을 설정한다.
- Build Type
 - Executable: 프로젝트를 Build 하면 "*.elf" 형식의 실행파일을 생성한다.
 - Static Library: 프로젝트를 Build 하면 "*.a" 형식의 정적 library를 생성한다.
 - Dynamic Library: 프로젝트를 Build 하면 "*.so" 형식의 동적 library를 생성한다.
- Project Name
생성될 Project 이름을 입력한다.
- Make New Project Folder
"Make New Project Folder" 메뉴를 선택하면 "Project Name"과 동일한 Folder를 "Project Directory" 하위에 생성한다.
- Project Directory
Project가 생성될 디렉토리를 결정한다.

6.2 Open Project & Close Project

현재 Project가 열려있는 상태라면 사용중인 Project를 닫아야만 새로운 Project를 생성하거나, 다른 Project를 열 수 있다.

- Open Project & Close Project

- ① 메뉴의 File ➔ Close Project을 선택한다.
 - ② 편집이나 생성 후 저장하지 않은 프로젝트 파일들이 있으면 저장할 것인지를 묻고 프로젝트를 닫는다.
 - ③ 메뉴의 File ➔ Open Project를 선택한다.
 - ④ "열기" dialog가 뜨게 되면 원하는 " *.epx " Project 파일을 선택한다.
- Startup Page
 - ① EISC-Stuio3를 실행 시킨다.
 - ② [그림 6-3] 같이 생성된 Startup Page의 "Resent Project" 목록에서 원하는 Project를 선택한다.
 - ③ "Resent Project" 목록에서 원하는 Project 가 없다면, "Open Other Project"를 선택한다.
 - ④ "열기" dialog가 뜨게 되면 원하는 " *.epx " Project 파일을 선택한다.



[그림 6-3] Startup Page

6.3 Adding and Removing Files

새로운 프로젝트를 생성하거나 기존의 프로젝트를 활성화 한 후, 프로젝트에 파일을 추가/삭제 하는 방법을 설명한다.

프로젝트에 속하는 파일들은 Source Files, Header Files, Linker Script File, No Compile Files, Static Library Files로 분류된다.

파일은 두 가지 방법으로 프로젝트에 추가/삭제 할 수 있다.

- Adding Files

위에서 설명한 3번 항목과 동일한 과정으로 파일을 추가할 수 있다.

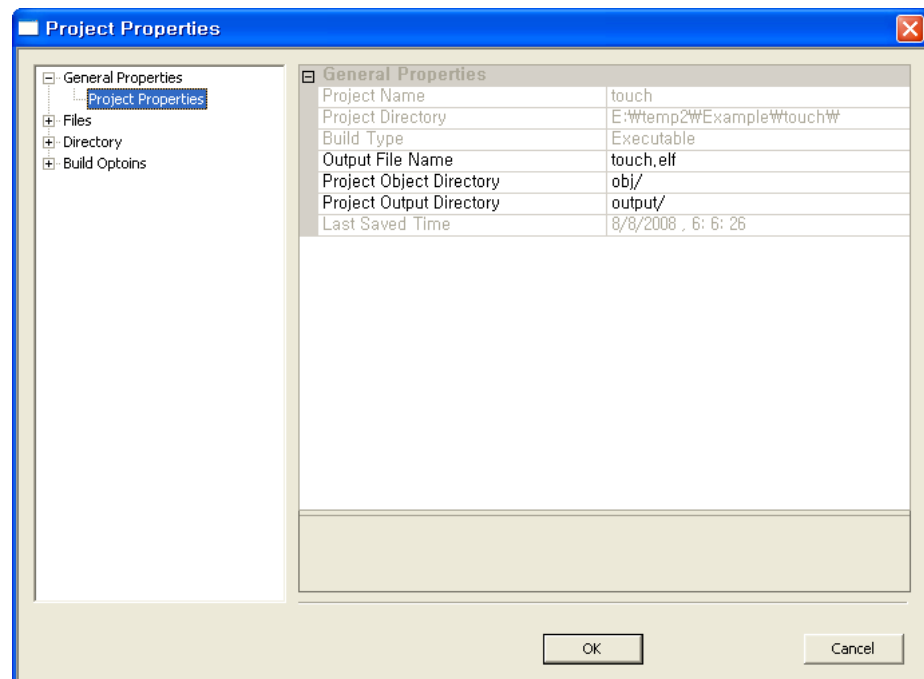
- Removing Files

프로젝트에서 더 이상 필요하지 않은 파일은 삭제 해줘야 프로젝트 관리가 편하다. Project Explorer window에서 삭제 하고 싶은 파일을 선택한 후 "Delete"키를 이용하여 삭제하면 된다. 또한 마우스를 이용해 해당 파일을 오른쪽 Click 하고 "Remove" 메뉴를 선택하면 해당 파일을 삭제할 수 있다.

6.4 Properties

- General Properties (🔧)

Project properties에는 [그림 6-6] 같이 프로젝트의 일반적인 정보를 보여주고 수정할 수 있다. 비 활성화 되어 있는 필드들은 수정이 불가능한 정보들이다. 각각의 필드를 선택하면 그에 대한 설명이 Dialog 하위에 보여진다.

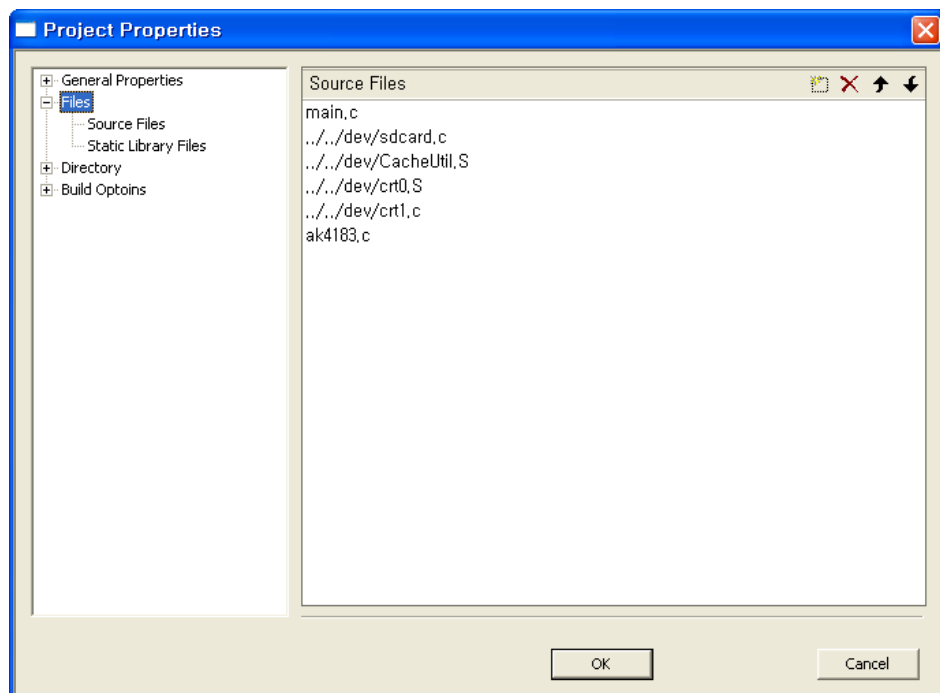


[그림 6-6] Project Properties (General Properties)

- Project Name
Active 된 Project Name 이다.
- Project Directory
Project가 생성된 전체 경로이다.

- Build Type
Project의 Build Type 이다.
 - Output File Name
Build Project를 실행하면 생성되는 실행 파일 Name이다.
 - Project Object Directory
Build 하면 생성되는 Object 파일들이 위치하는 Project Directory의 상대 경로이다.
 - Project Output Directory
Build 하면 생성되는 파일들(*.elf, *.bin)이 위치하는 Project Directory의 상대 경로이다.
 - Last Saved Time
프로젝트가 마지막으로 저장된 시간이다.
- Files



[그림 6-7]의 Dialog에서는 프로젝트에 포함된 파일 및 Static Library 파일을 나타낸다. Dialog의 우측에 나타난 프로젝트 파일 목록들은 Project Directory의 상대 경로로 나타난다.



[그림 6-7] Project Properties (Files)

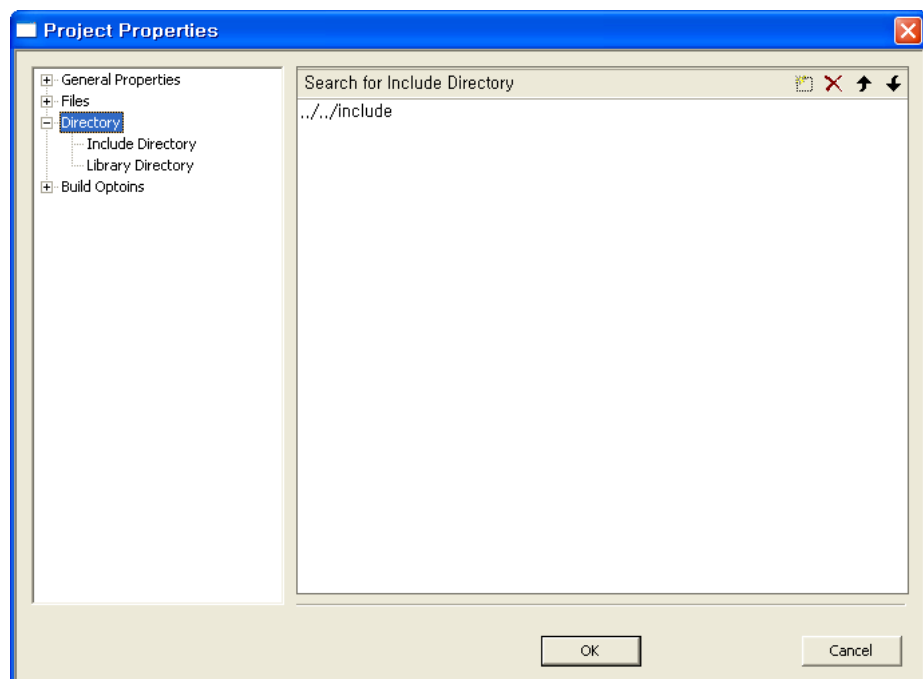
파일이 추가 된 순서에 따라 컴파일 되는 순서가 결정된다. 간혹 crt0.S 와 같은

파일은 컴파일이 먼저 되지 않아 에러가 발생할 수 있다. 그러므로 파일 추가 시 crt0.S 파일은 가장 상위에 두는 것이 바람직하다. 이때 추가된 파일들의 순서를 바꿀 수 있는 기능이 있다

- ① 메뉴의 Project ➔ Properties를 선택한다.
- ② 파일 추가 과정에서 열린 "Project Properties" dialog 가 열린다. Files ➔ Source Files를 선택한다.
- ③ 특정 파일을 선택하고  또는  을 이용하여 파일의 위치를 바꿔 준다.
- ④ 확인 버튼을 누르면 Project Explorer window에서도 파일의 위치가 바뀐 것을 확인 할 수 있다.

- Directory

[그림 6-8] dialog에서는 프로젝트에 추가 할 Include, Library 디렉터리를 지정할 수 있다. 디렉터리 추가/삭제 에는 다음 아이콘의 기능들을 사용한다.



[그림 6-8] Project Properties (Directory)

- Include Directory
gcc 컴파일 시 "-I" 옵션에 해당하는 기능을 한다. Include 경로를 추가 할 때 사용한다.
- Library Directory

gcc 컴파일 시 "-L" 옵션에 해당하는 기능을 한다. 사용자 라이브러리 경로를 설정할 때 사용한다

 : Directory 추가

 : Directory 삭제

  : 추가된 Directory 의 위치 변경

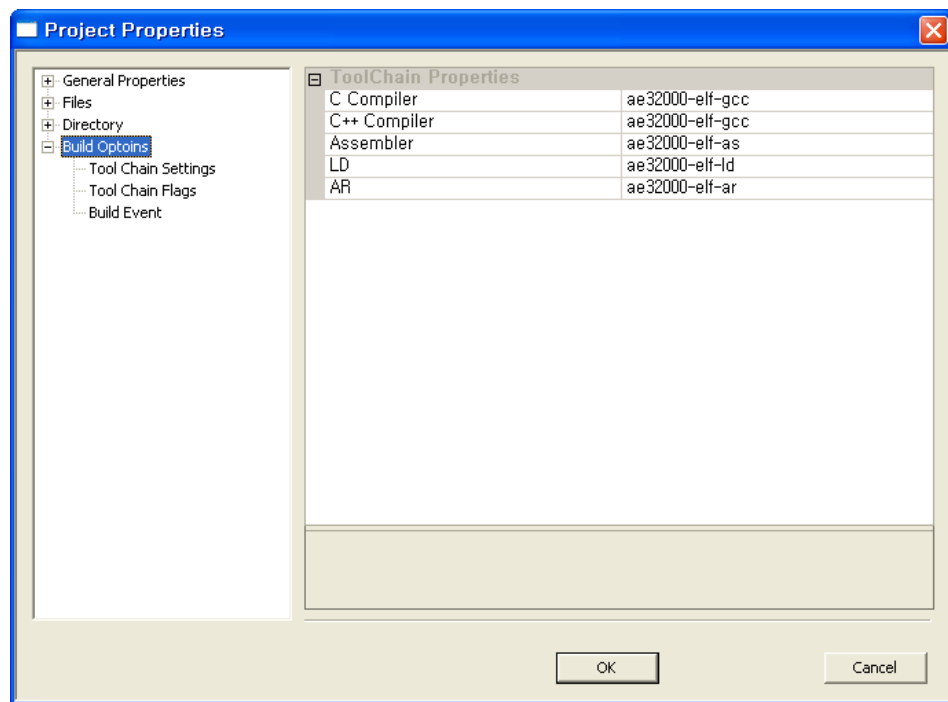
- Build Options

[그림 6-9] dialog에서는 프로젝트를 Build 할 때 사용되는 여러 옵션들이 각 과정에 따라 분류되어 있다. 자주 사용되는 옵션들은 True/False 중 선택한다.

사용자가 추가 하고 싶은 옵션들은 "Others" Edit Box에 추가한다.

dialog 하단에 각 옵션에 대해 간단히 설명 되어 있다.

컴파일 옵션에 관한 자세한 정보는 GCC manual 을 참조하라



[그림 6-9] Project Properties (Build Options)

7. Build

Build에 해당되는 기능들은 대부분이 Build Toolbar에 아이콘으로 되어있다. 이 매뉴얼에 서는 각 메뉴가 가지는 기능들만 간단히 소개한다. 컴파일 되는 좀더 자세한 과정과 옵션 들의 기능을 원한다면, EISC Software Developer Guide를 참조하여라.

- Build ➔ Build Project, F7, 
Makefile을 실행하여 *.elf 파일을 생성한다.
- Build ➔ Compile File, Ctrl+F7, 
현재 Text Editor Window에 활성화 되어있는 파일만을 컴파일 한다.
- Build ➔ Rebuild Project, Ctrl+F8
Makefile.mk 파일을 재 생성하고 기존의 목적 파일을 삭제하고 다시 Build한다.
- Build ➔ Clean All, 
생성된 목적 파일, 실행 파일을 모두 삭제 한다.
- Build ➔ Stop the Build, 
현재 컴파일 되고 있는 과정을 중단한다.
- Build ➔ Disassemble file, 
Makefile을 실행하여 *.elf.dis 파일을 생성한다.
- Build ➔ Full Build Message
이 메뉴를 선택하면 현재 컴파일이 되고 있는 모든 컴파일 옵션들을 "Output View" 창에 보여준다.


8. Debug

Debugging을 하기 위해서는 컴파일 과정에서 "-g" 옵션을 반드시 넣어줘야 실행 파일 (*.elf)에 debugging정보가 들어간다. Debugging 정보는 실행 파일 크기에만 영향을 주고 실제 target board에서 수행되는 binary 파일 크기에는 영향을 주지 않으므로 디버깅을 위해 추가 하도록 한다.

8.1 Target of Debugger

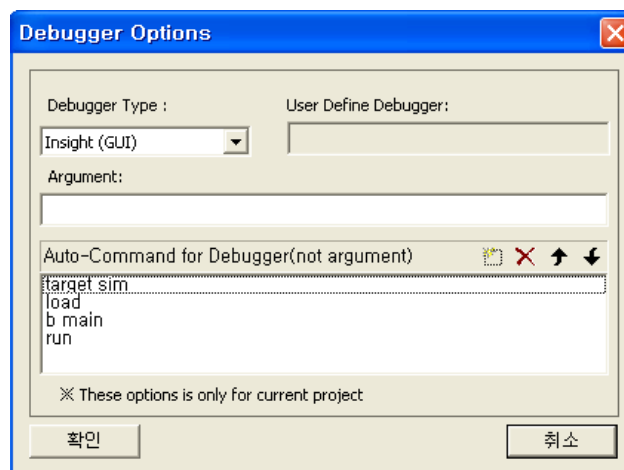
- Simulation
PC에서 target의 환경에서 실행 하듯 simulation하며 debugging하는 방법이다.

8.2 Start Debugger

- ① File→Open Project 메뉴를 통해 프로젝트를 연다.
- ② Debug→Start Debugger 메뉴, "F5" 또는 를 통하여 debugger를 실행한다.

8.3 Debug Options





- Debug를 실행하기 전에 [그림 8-1]과 같이 debugger에 필요한 option들을 setting 해준다.



[그림 8-1] Debugger Options


Debugger Type은 3가지(Insight(GUI), GDB(CLI), User Define)가 있는데 이중 하나를

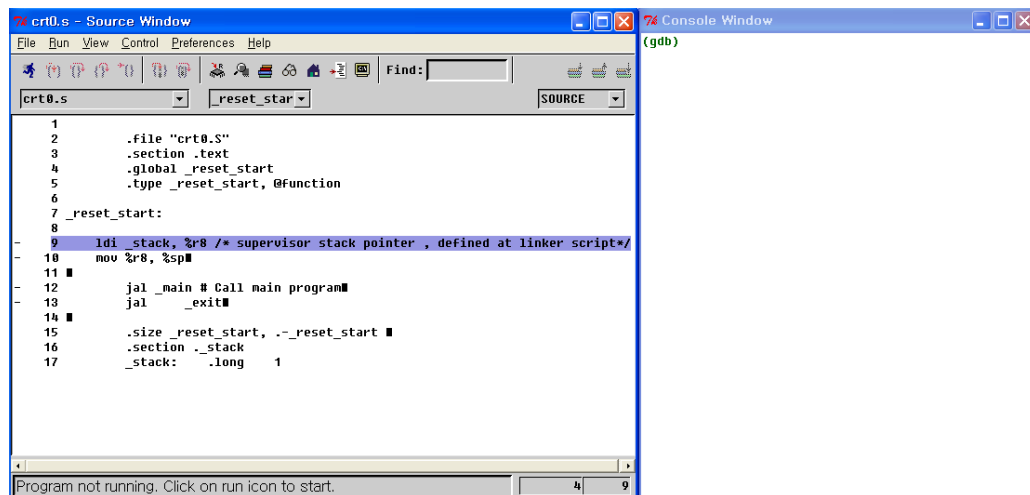
선택한다. Argument는 필요하면 setting 해준다.

Auto-Command for Debugger 라인은 아래 그림과 같이 debugging을 하기 전에 계속 사용되는 명령어들을  을 눌러 직접 입력하여 자동으로 실행되게 한다. 입력된 명령어는  버튼으로 삭제 가능하며   버튼으로 위아래 방향을 움직일 수 있다.

8.4 Debugging interface

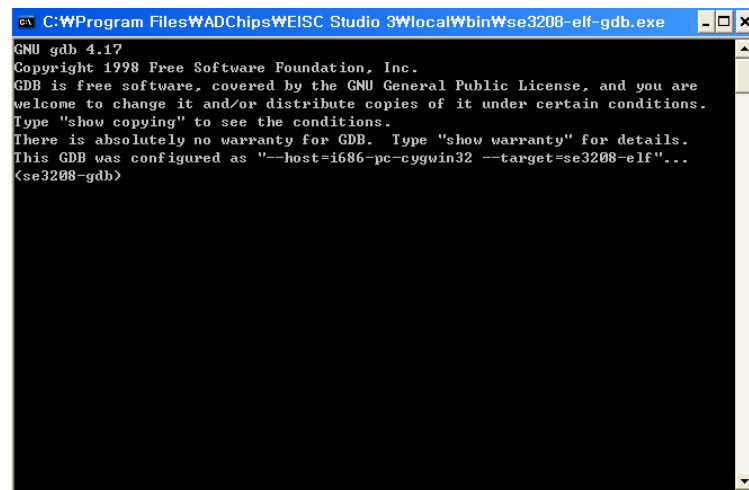
Debugging을 시작하면,

[그림 8-2]과 같이 AE32000 core일 경우, Source insight 환경이 실행된다. Console 창이 실행되지 않았을 경우 Source 창에서  을 누르면 된다.



[그림 8-2] Debugging 환경 (Source insight)

SE3208 core와 SE1608 core일 경우에는 [그림 8-3]와 같이 Command 창이 뜬다.

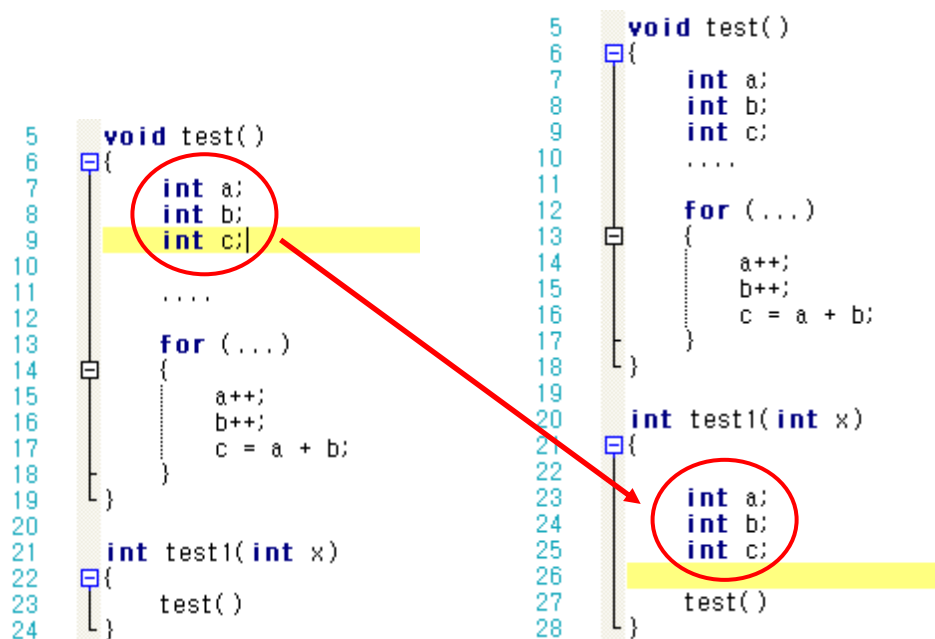


[그림 8-3] Debugging 환경 (Command Window)

9. Tools

9.1 Macro

자주 사용하는 여러 개의 명령을 묶어서 하나의 키 입력 동작으로 만들어 사용한다. 여러 번 해야 하는 일을 간단하게 수행하기 위하여 사용하기도 하지만, 문서 안의 같은 문자열을 한꺼번에 변경 할 때도 사용된다.



[그림 9-1] Macro

- Start Recording: Macro를 실행해 작성되는 소스 코드를 계속 기록하게 된다.
- Stop Recording: Macro를 멈춘다.
- Playback: 기록된 Macro를 원하는 곳에 넣는다

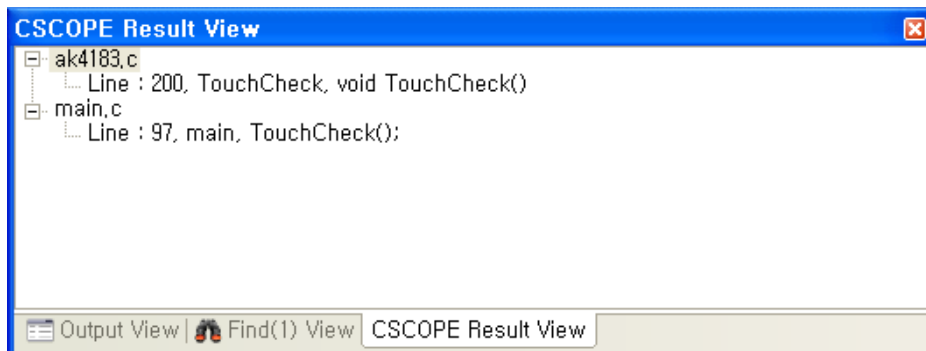
9.2 Source Code Formatter

- Format Current File
AStyle Options에서 설정한 Format을 소스 코드에 적용시킨다.
- AStyle Options
소스 코드의 Format을 설정한다.
각 설정 메뉴의 사용법은 [Http://astyle.sourceforge.net/](http://astyle.sourceforge.net/) 을 참고하면 된다.

9.3 CSCOPE

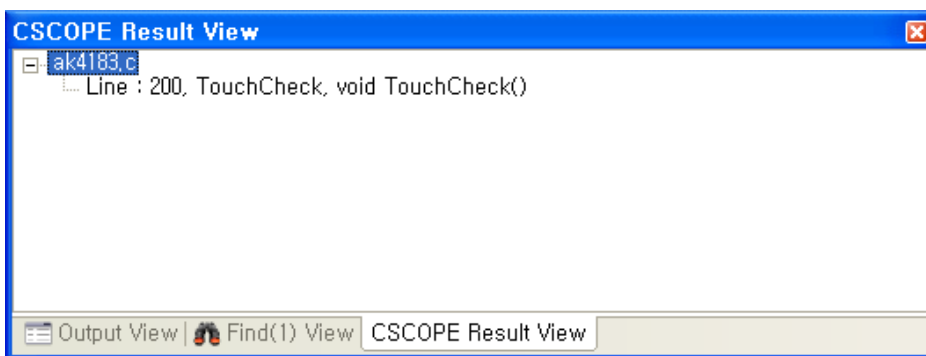
Cscope 는 소스 코드를 검색하기 위한 개발자 툴이다.

- Find this C symbol
찾고자 하는 C symbol을 전부 찾아서 보여준다.



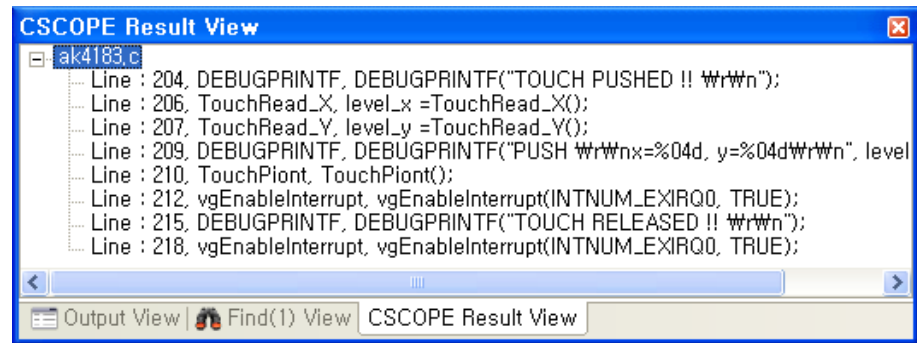
[그림 9-2] CSCOPE Result View (1)

- Find this function definition
찾고자 하는 함수의 원형이 있는 .C파일의 위치를 찾아서 보여준다.



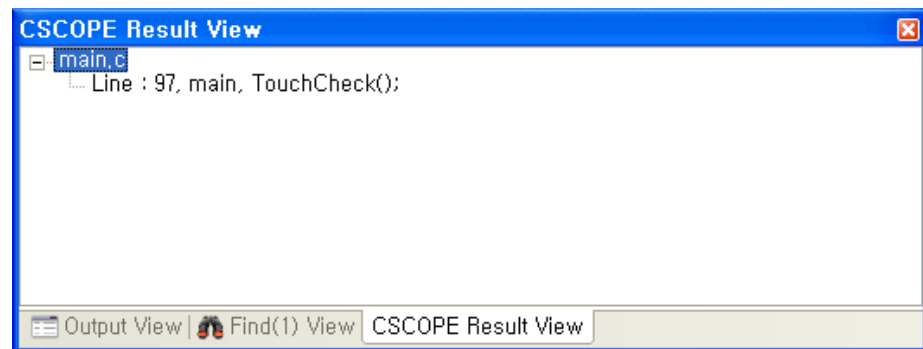
[그림 9-3] CSCOPE Result View (2)

- Find functions called by this function
선택한 함수 내에 포함된 함수들을 찾아 보여준다.



[그림 9-4] CSCOPE Result View (3)

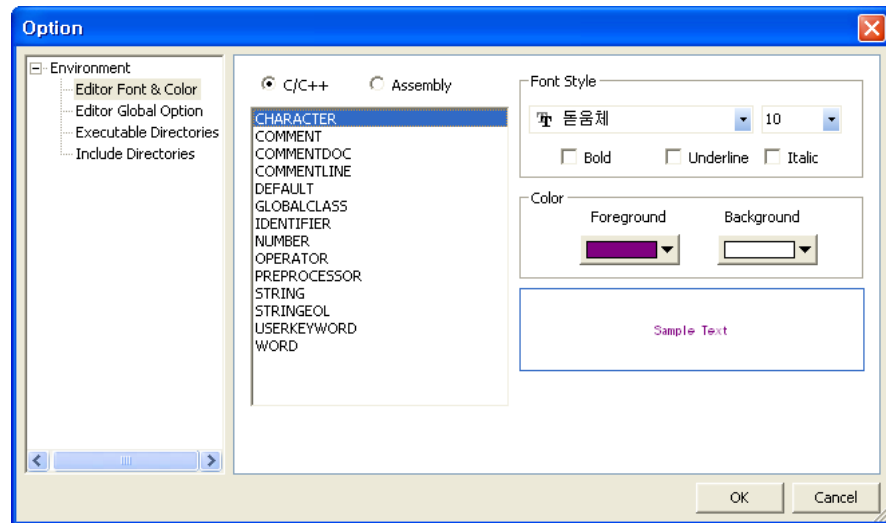
- Find functions calling by this function
선택한 함수를 호출하는 곳을 모두 찾아 보여준다.



[그림 9-5] CSCOPE Result View (4)

9.4 Options

- Editor Font & Color
Keywords 나 주석과 같이 Text의 여러 구성 요소를 다른 색으로 나타낸다.
 - ① 메뉴의 Tools → Preferences를 선택해 [그림9-6]과 같은 Dialog를 띄운다.
 - ② "Font Style" 와 "Color"의 서브메뉴를 선택해 사용자가 원하는 Font, size Color를 설정한다.

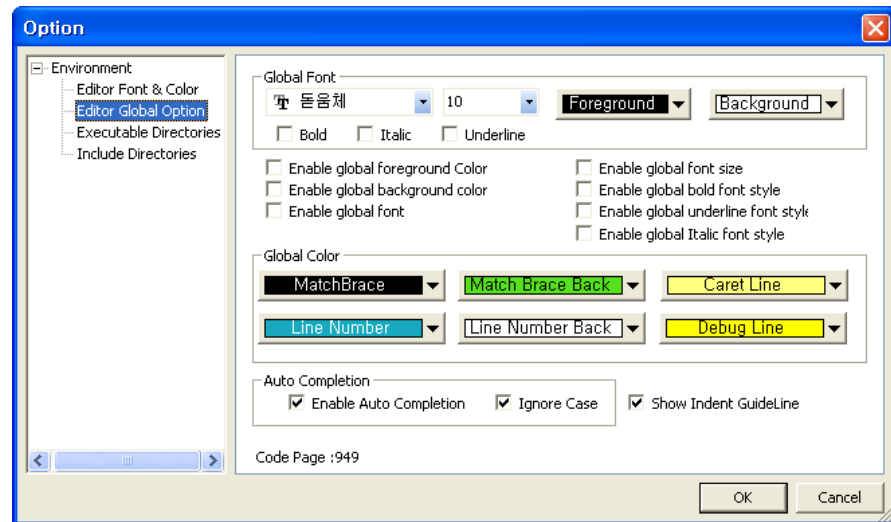


[그림 9-6] Option (Editor Font & Color)

- Editor Global Option

[그림 9-7]에서 설정한 것은 가장 우선시 되어 Text Editor Window에 적용된다.

- ① “Global Font”의 서브메뉴를 선택하여 사용자가 원하는 Font, Size, Ground color를 설정한다. 설정한 것을 적용하려면 “Enable..”란에 체크하면 된다.




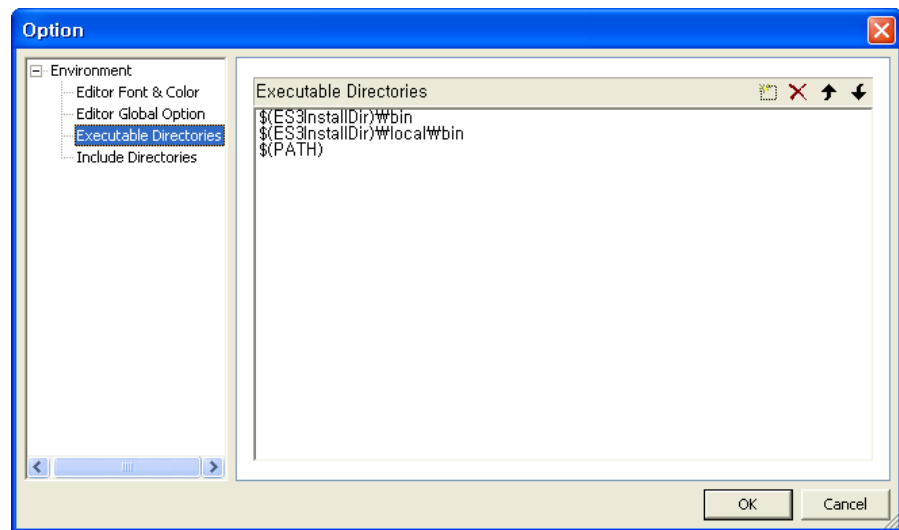
[그림 9-7] Option (Editor Global Option)

- ② “Global Color” 메뉴는 Match Brace나 Line Number등의 색을 서브메뉴에서 선택하여 바꿀 수 있다. [그림 9-7]에서 보이는 색이 default 이다.
- ③ “Auto Completion”는 5.11절에서 설명한 “Complete Word” 사용유무를 체크

하는 메뉴이다.


- Executable Directories

EISC-Studio3는 컴파일러가 내장되어있어 EISC-Studio3를 설치할 때 컴파일러가 설치된다. 이렇게 설치된 컴파일러는 [그림 9-8]과 같이 셋팅 되어있으며 맨 위에 설치된 경로에서 가장 우선적으로 처리된다. 컴파일러를 직접 설치해서 사용할 경우  을 눌러서 경로를 지정해주면 된다.



[그림 9-8] Option (Editor Directories)

- Include Directories

 을 눌러서 포함시킬 디렉터리를 지정한다.

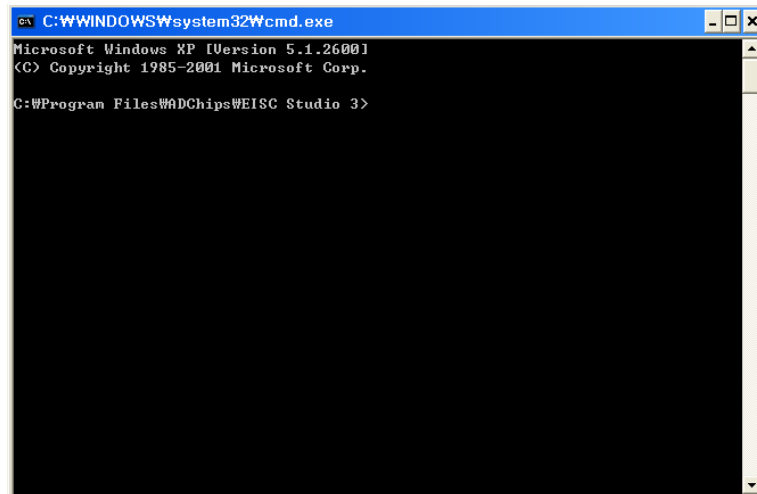
- editorconfig.cfg 파일

셋팅을 끝내고 저장하면 EISC-Studio3가 설치된 위치 (예>C:\Program Files\ADChips\EISC Studio 3\config)의 config폴더에 editorconfig.cfg이 바뀌어 있다. EISC-Studio3를 다시 시작하면 바뀐 셋팅 대로 실행된다.

만약 원 상태(Default)로 바꾸고 싶다면 editorconfig.cfg을 지우면 되고, 셋팅한 상태를 다른 사용자가 사용하고자 한다면 editorconfig.cfg을 덮어쓰면 된다.

editorconfig.cfg 파일은 EISC-Studio3를 시작하면 생긴다.

9.5 EISC Studio Command Prompt

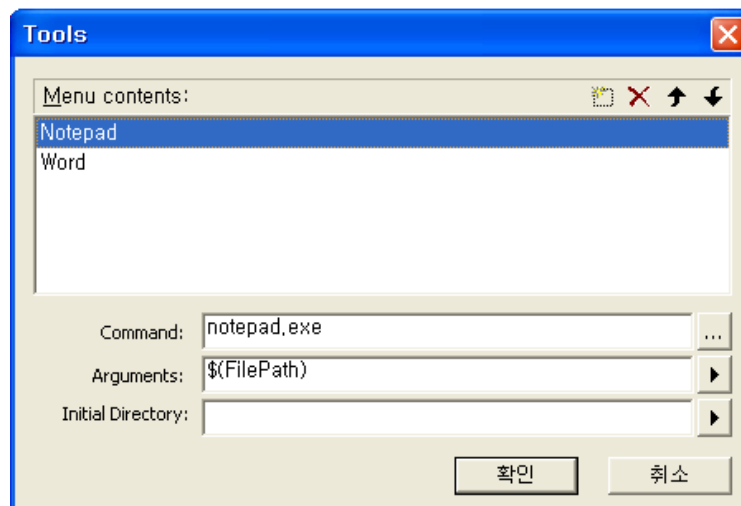


[그림 9-9] Command Prompt

- EISC Studio Command Prompt에는 EISC Studio에서 필요한 컴파일러 및 모든 라이브러리의 올바른 경로가 자동으로 설정되므로 일반적인 명령어 프롬프트 창 대신 이 창을 사용합니다.

9.6 External Tools

- 이 기능은 사용자가 원하는 외부 툴로 파일을 열 수 있다.



[그림 9-10] External Tools

- [그림 9-10]와 같이 Menu Contents란에 사용하고자 하는 외부 툴의 name value를 쓰고 Command란은 그 툴의 실행 파일의 경로를 넣을 수 있다.
Arguments란은 사용자가 [그림 9-11]과 같이 미리 외부 툴의 argument를 지정할 수 있고 Initial Directory역시 마찬가지로 지정 가능 하다.

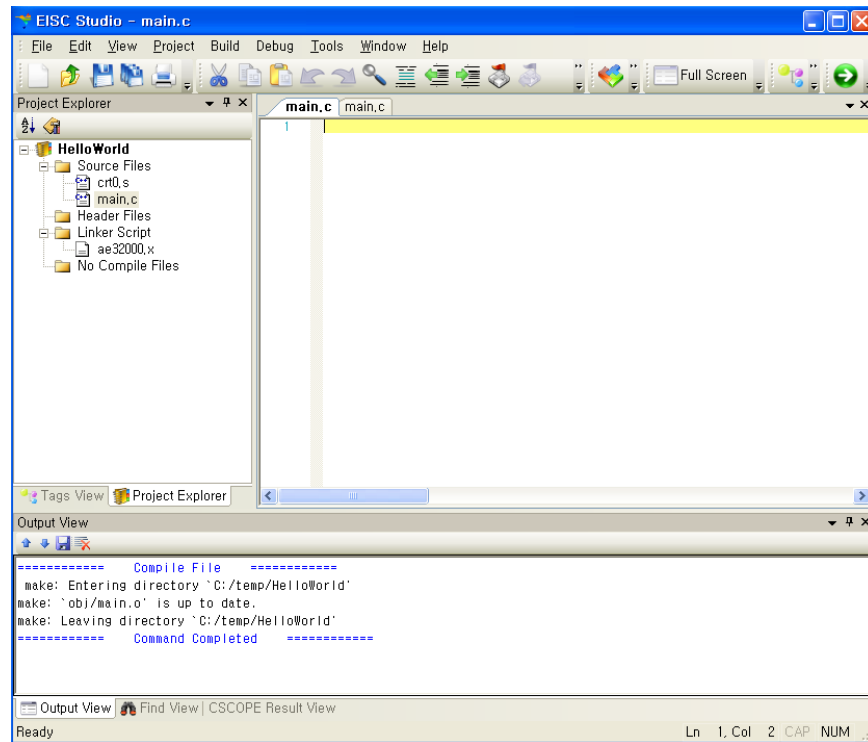


[그림 9-11] Argument

10. Window

10.1 New Window

현재 Text Editor Window에 활성화 되어있는 파일의 이름을 그대로 하여 새로운 Text Editor Window창이 활성화 된다.



[그림 10-1] New Window

10.2 Cascade/Tile

Text Editor window에서 열리는 파일들의 display 형태를 변경하는 메뉴이다.

11. Help

11.1 About ES3...



[그림 11-1] About EISC Studio 3


EISC Studio 3에 대한 버전 등 간략한 정보가 적혀있다.

11.2 Update Check

EISC Studio3는 Auto Update가 가능하다. Help→Update Check를 누르면 아래와 같은 그림이 나타난다.




[그림 11-2] Auto Updater

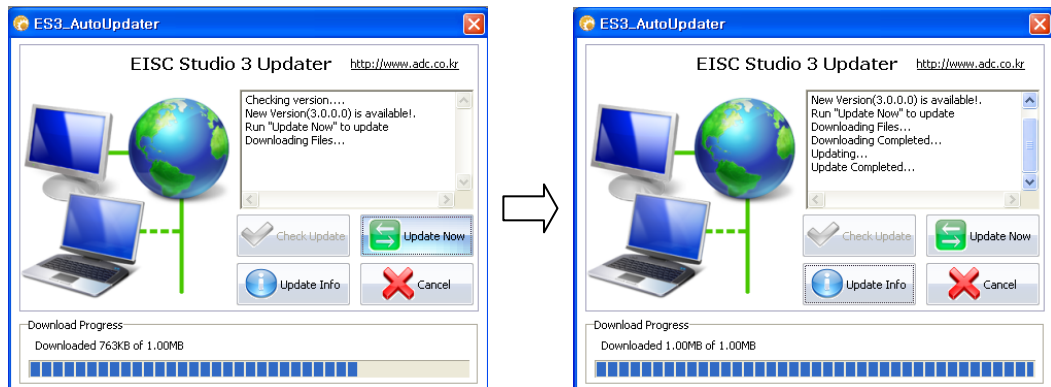
위의 그림에서  (Check Update)를 누르면 아래 그림과 같이 server에 접속된다.



[그림 11-3] Auto Updater

Server에 접속되면 버전체크를 하여 가장 최신 버전을 찾아낸다.

그리고  Update Now (Update Now)를 클릭하여 Update 한다. 이 버튼을 눌러 Update하기 전에 만약 EISC-Studio 3 또는 프로젝트가 열려있다면 꼭 닫고 Update를 실행해야 한다.



[그림 11-4] Auto Updater

[그림 11-4]는 Update 과정을 나타낸다. 올바르게 완료되었다면 "Update Completed..."

라는 메시지가 나타난다.

Update정보를 확인하려면  Update Info (Update Info) 버튼을 클릭하면 된다. 모

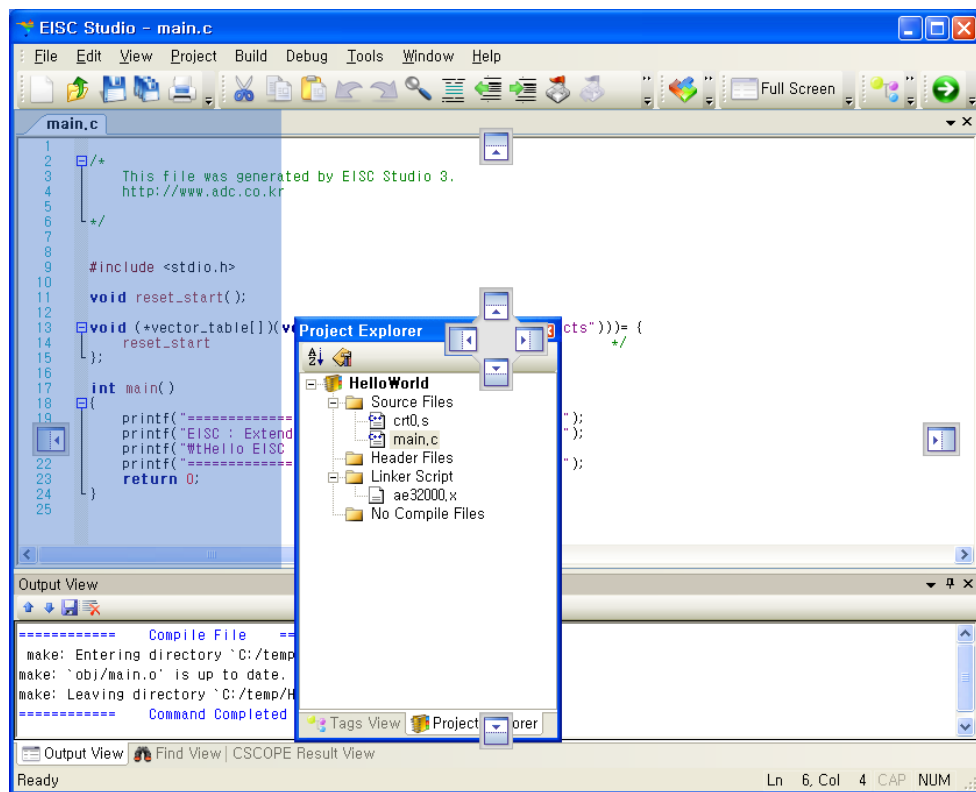
든 것이 완료되었다면  Cancel (Cancel) 버튼을 클릭하여 빠져 나온다.

12. 그 외의 기능들

이 장에서는 Menu Bar나 Tool Bar에 나와 있지는 않지만 편리한 기능들을 소개한다.

12.1 Project Explorer Window/Output View Window 위치 지정

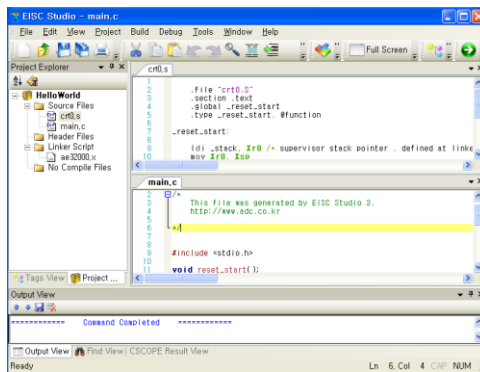
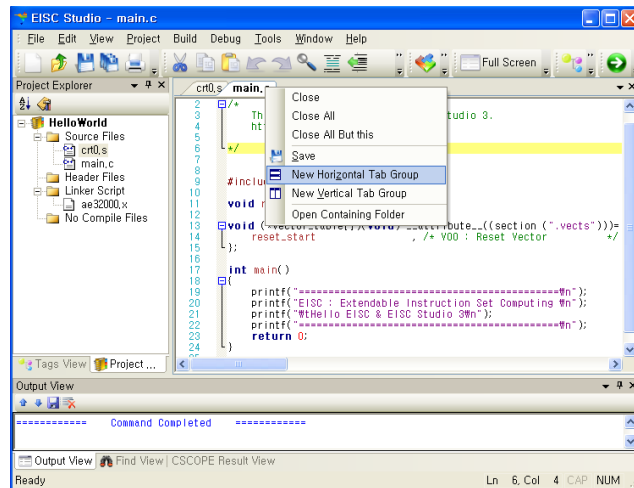
사용자는 마우스로 Project Explorer Window를 끌어서 [그림 12-1]과 같이 위치를 정할 수 있다. Output View Window의 위치를 정하는 방법도 동일하다.



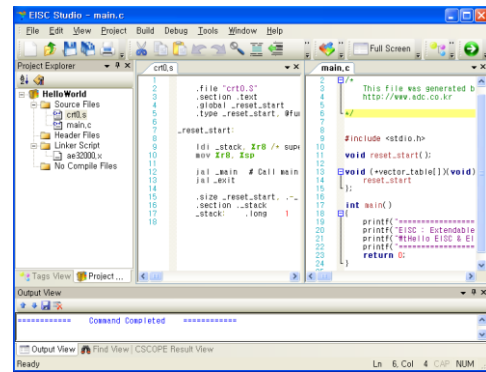
[그림 12-1] Project Explorer Window 위치 지정

12.2 New Horizontal Tab Group/New Vertical Tab Group

Text Editor Window의 display를 결정한다.



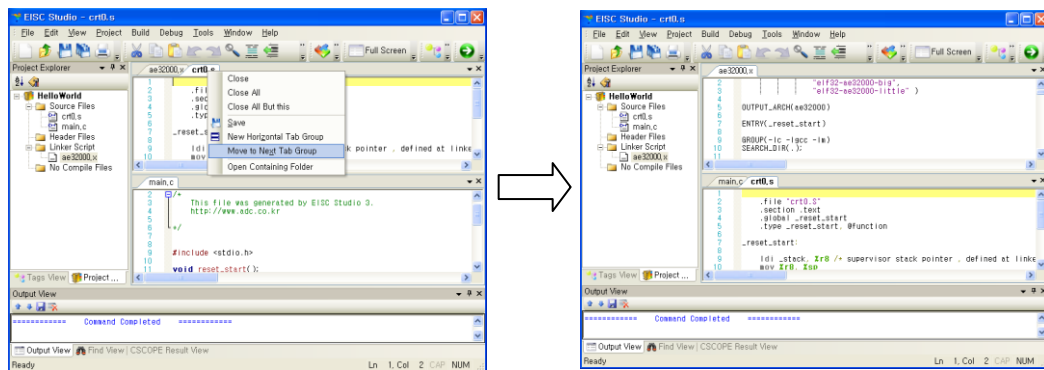
< New Horizontal Tab Group >



< New Vertical Tab Group >

[그림 12-2] New Horizontal Tab Group/New Vertical Tab Group

12.3 Move to Next Tab Group

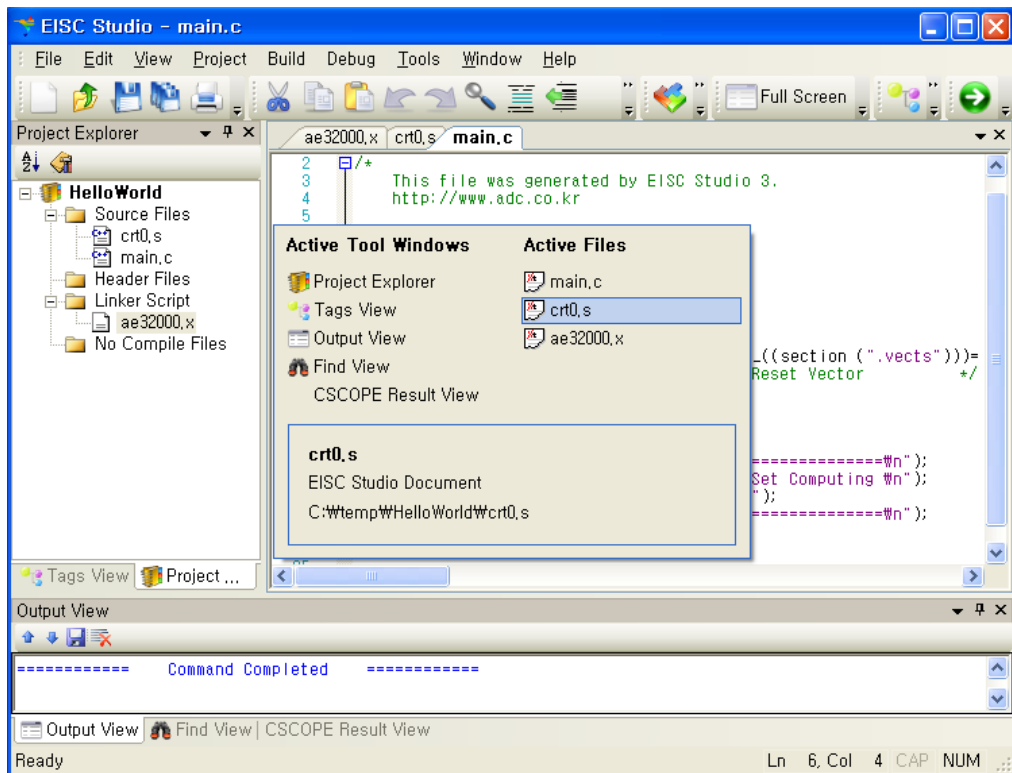


[그림 12-3] Move to Next Tab Group

12.4 활성화된 Text Editor Window로 이동 (Ctrl+Tap, Ctrl+F6)

활성화된 Texts끼리 서로 이동하기 위해서 "Ctrl+Tap"을 누르면 [그림 12-4]와 같이 나타난다.

"Ctrl+F6"을 누르면 [그림 12-4]처럼 창은 나타나지 않지만 직접 각 활성화 되어있는 Text들로 이동할 수 있다.

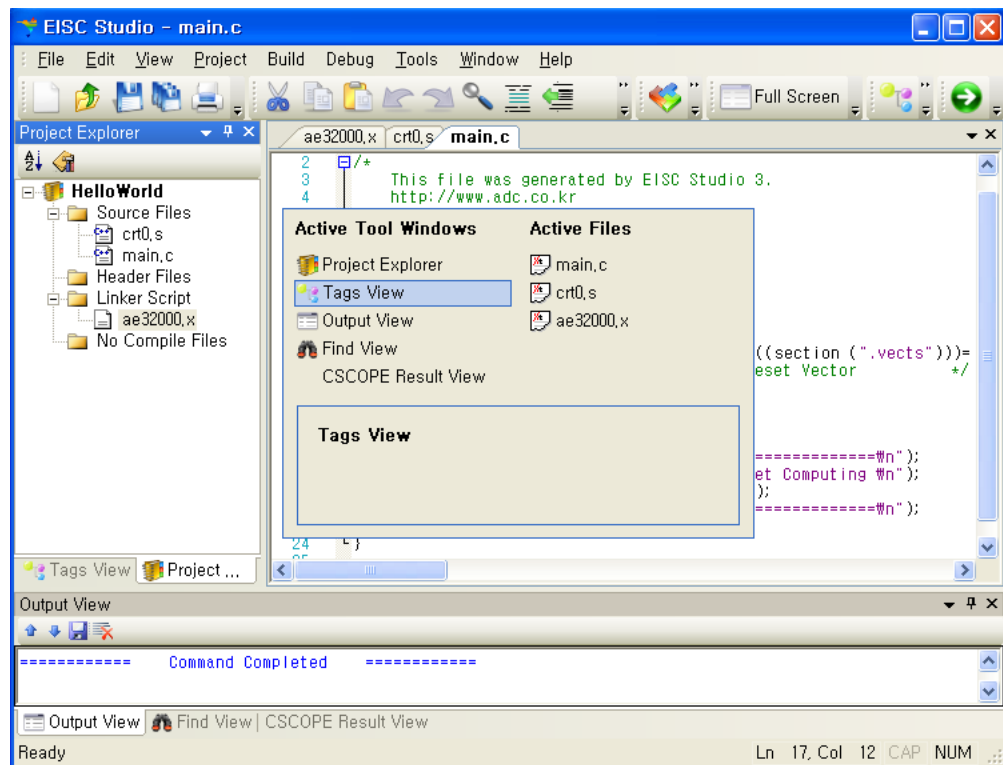


[그림 12-4] 활성화된 Text Editor Window로 이동 (Ctrl+Tap)

12.5 각 Active Tool Window으로의 이동 (Alt+F7, Alt+F6)

Active Tool Window들로 이동하기 위해서 "Alt+F7"을 누르면 [그림 12-5]와 같이 창이 나타난다.

"Alt+F6"은 [그림 12-5]와 같은 창은 나타나지 않지만 직접 원하는 Window로 이동할 수 있다.



[그림 12-5] 각 Active Tool Window으로의 이동 (Alt+F7)

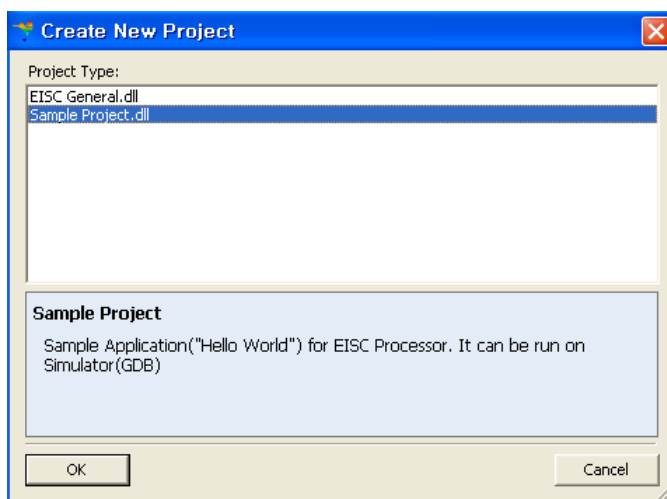
13. Example Program

EISC-Studio3의 기능을 제대로 익혔다면 Sample 프로젝트를 실행하여 보자.

Sample 프로젝트는 EISC-Studio3가 설치되면 자동으로 생성된다. 이 장에서는 프로젝트를 실행하고 디버깅하는 것까지 차례로 살펴본다.

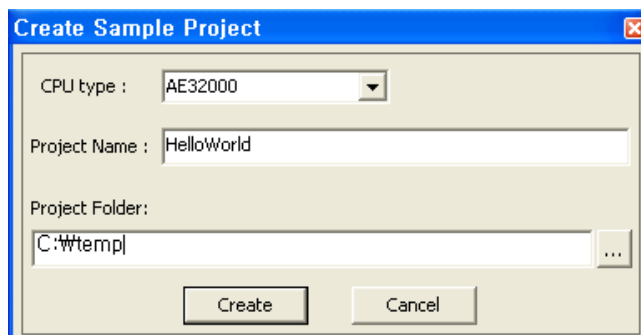
13.1 Create New Project

- ① Start Page의 "Create New Project" 또는 File → New → Project 메뉴를 선택한다.
- ② [그림 13-1]과 같이 dialog가 뜨면 Sample Project.dll을 선택하고 "OK" 버튼을 누른다.



[그림 13-1] Create New Project

- ③ 필요한 정보를 [그림 13-2]와 같이 입력하고 "Create" 버튼을 누른다.

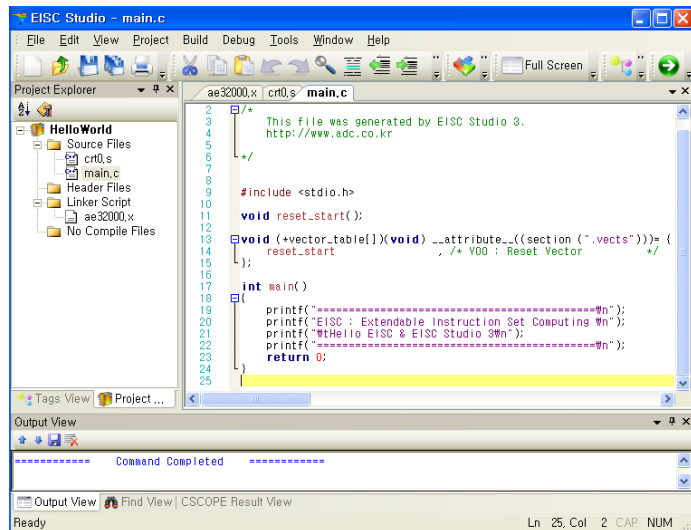


[그림 13-2] General EISC Project

13.2 Completed Project

[그림 13-3]과 같이 "HelloWorld"라는 프로젝트가 생성된다.

프로젝트의 파일은 Project Explorer window에 있는 각 파일의 분류에 맞게 이미 추가되어 있다.




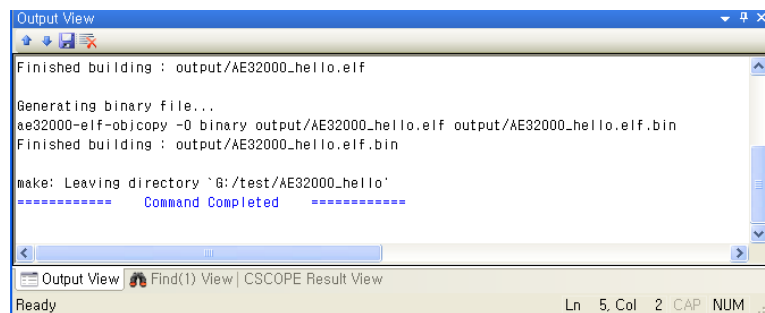
[그림 13-3] Add New File/Add Exist Files

13.3 Build Project

지금까지 제대로 실행하였다면 아래 두 방법 중 한가지로 실행 파일을 얻을 수 있다.
([그림 13-4]는 Build→Full Build Message를 체크 한 후 컴파일 한 것이다.)

실행 파일 및 binary 파일 생성: ae32000_hello.elf, ae32000_hello.elf.bin


실행: Build → Build Project, , F7

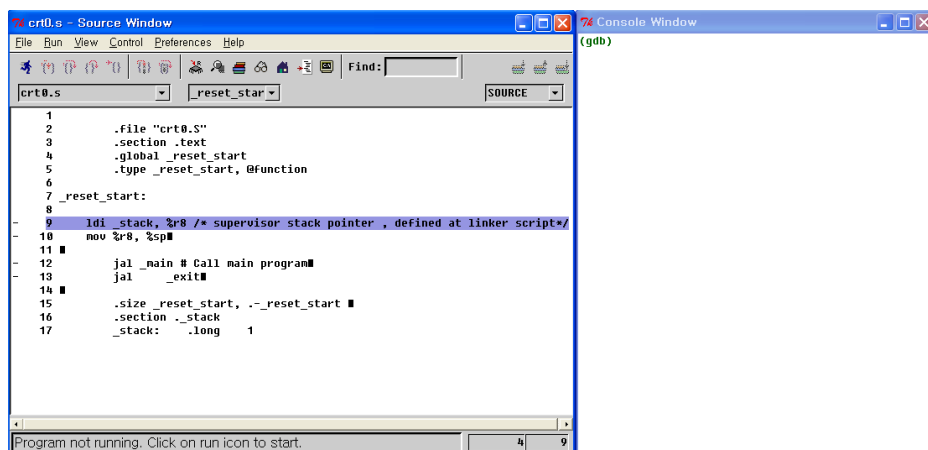


[그림 13-4] Build-ae32000_hello

13.4 Debugging

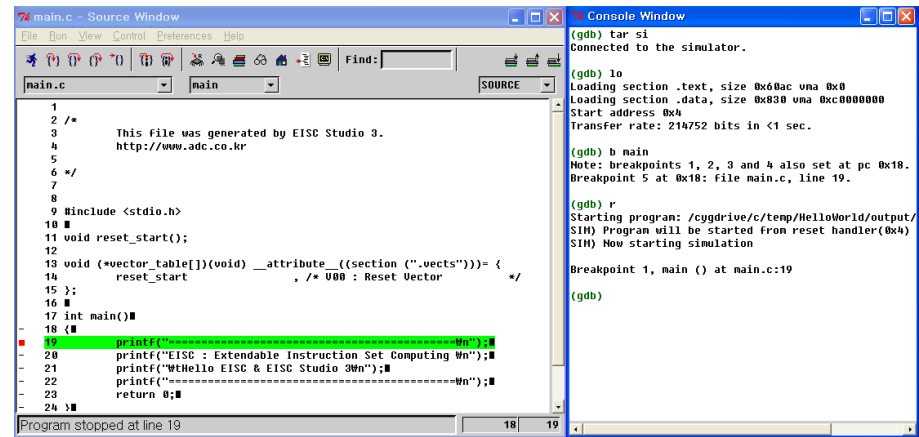
“HelloWorld” 프로젝트 컴파일이 끝났다면 Simulation을 타겟으로 Debugging을 할 수 있다.

- ① Debug→Start Debugger 또는 F5,  을 누른다.
- ② Ae32000 core를 사용하여 [그림 13-5]과 같이 Source Insight 환경이 나타난다.



[그림 13-5] Debugging 환경 (Source Insight)

- ③ Console 창에 command를 이용하여 debugging을 시작한다.
 첫째, Simulation을 타겟으로 하므로,
 (gdb) tar si => Simulation Target
 명령어를 실행한다.
 둘째, 실행파일을 로딩해야 하므로,
 (gdb) lo => 실행 파일 loading
 명령어를 실행한다.
 셋째, breakpoint를 설정하여,
 (gdb) b main => main함수에 breakpoint를 설정
 첫 breakpoint를 만나는 곳에서
 (gdb) r => run (실행)
 실행하다 멈출 수 있다.
 위의 내용은 [그림 13-6]과 같다.



[그림 13-6] Debugging Start

- ④ [그림 13-7]은 debugging을 하는 장면이다.

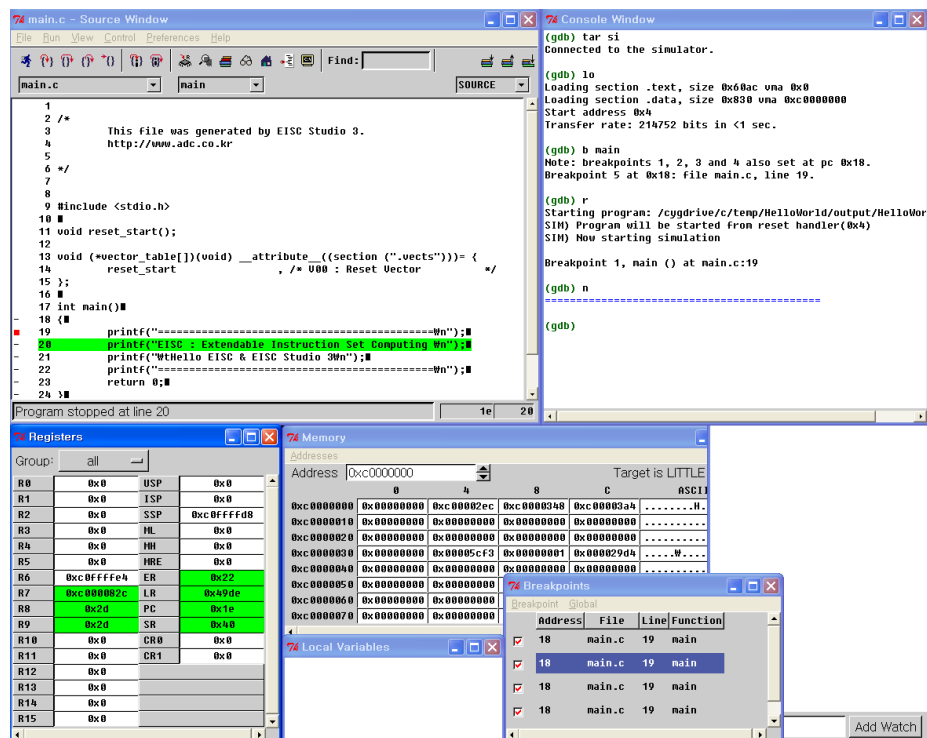
Source 창에서,



은 디버깅 Toolbar이다. 각 그림에 마우스를 가져다 대면 설명이 나온다.



은 [그림 13-7]과 같이 register, memory등을 볼 수 있게 해주는 아이콘이다. 이 그림들 또한 마우스를 가져다 대면 설명이 나온다.



[그림 13-7] Debugging

⑤ Source 설명

이 프로그램은 ae32000 core를 사용하여 만들어 졌다. 처음 프로그램이 시작 시 vector table에 의하여 crt0.s의 reset_start에서 시작한다. 이 프로그램은 Printf를 이용하여 "EISC.."에 대한 내용을 출력 하는 프로그램이다.

14. Help System

EISC-Studio3 사용 중 문제가 발생한다면 helpide@adc.co.kr 로 연락 주시기 바랍니다.