

제6장 개발 장비의 사용법 (MPLAB-IDE 기준)

PC의 바탕화면에 있는 MPLAB IDE 아이콘을 클릭한다. 그러면 다음 그림과 같은 화면이 나타난다.

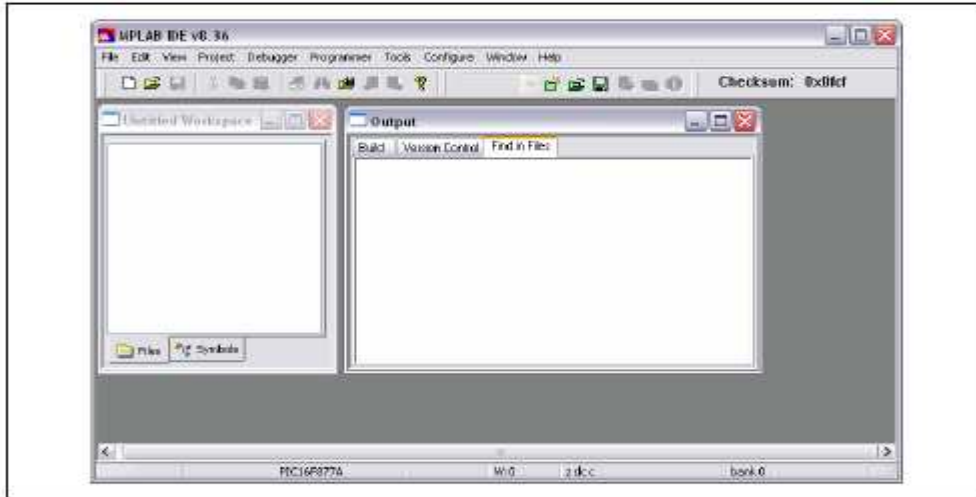


그림 6-1. 개발장비 초기 화면도

이 화면에서 다음 순서에 따라서 개발장비를 사용하며, 초기의 설정은 일반적으로 동일 프로젝트 안에서는 한번만 해주면 된다.

- Setting Up The Environment
- Creating the Application Code
- Running the Project Wizard
- Viewing the Project
- Creating a Hex File
- Viewing Debug Options
- Setting Up the Demo Board
- Loading Program Code For Debugging
- Running Debug Code
- Debugging Code Using Breakpoints
- Debugging Code Using A RunTime Watch
- Debugging Code Using Native Trace

- Programming the Application

1. Selecting the Device

MPLAB REAL ICE는 microchip사의 범용 개발장비이므로 먼저 개발 하고자 하는 마이컴의 종류를 결정해 주어야 한다.

1. Select *Configure>Select Device*.

2. In the Device Selection dialog, choose *"PIC16F876"* from the Device list box. The light icon next to *"MPLAB REAL ICE"* in the *"Microchip Tool Programmer/Debugger Tool Support"* sections should be green.

3. Click **OK**.

2. Selecting the Emulator as a Debugger

다음으로는 프로그램을 개발하는 디버깅 장비로 사용할 것인가, 아니면 프로그램을 마이컴에 기억시키는 programmer 로 사용할 것인지를 결정해야 한다. 초기에는 디버거로 사용하도록 설정하자.

1. Select MPLAB REAL ICE in-circuit emulator as a debugger, select *Debugger>Select*

Tool>REAL ICE. Then:

그러면 혹시 개발장비에 사용되는 프로그램이 변경되었거나, 또는 update 되었으며 자동으로 최종 프로그램을 down-load 하게 된다. 보통의 경우 이 상황은 디바이스가 변경되지 않으며, 초기 한번으로 끝난다.

3. RUNNING THE PROJECT WIZARD

그 다음으로 작업할 프로젝트를 새롭게 만들고 작업 환경을 설정해야 한다. 프로젝트란 한가지 일을 여러개의 단위 프로그램으로 나누어 작업할 때 사용하는 방식으로 하나의 작업에 하나의 프로그램만 존재하는 것이 아니고 여러개의 프로그램이 존재하며, 다시말하면 C로 작성된 프로그램도 있을 수 있으며, 어셈블리로 작성한 프로그램도 있을 수 있다. 따라서 이러한 프로그램들이 사용할 컴파일러가 존재하는 위치를 결정해 주어야 한다.

일반적으로 표준으로 제공되는 컴파일러가 존재하는 위치는 아래와 같으며, 따라서 이 위치로 설정을 해야만 정상적으로 동작이 된다.

C:\Program Files\Microchip\MPLAB C30\bin. MPLAB C30 should be

pointing to pic30-gcc.exe and MPLAB LINK30 should be pointing to pic30-ld.exe.



그림 6-2. 프로젝트 마법사 화면도

다음으로는 프로젝트를 관리할 위치를 결정한다.

이제 이 프로젝트에 사용할 프로그램들을 프로젝트 파일 아래로 모으는 작업을 한다. 화면에서 **Project>Project Wizard**.를 선택한다. 그러면 아래와 같은 그림이 나타난다.

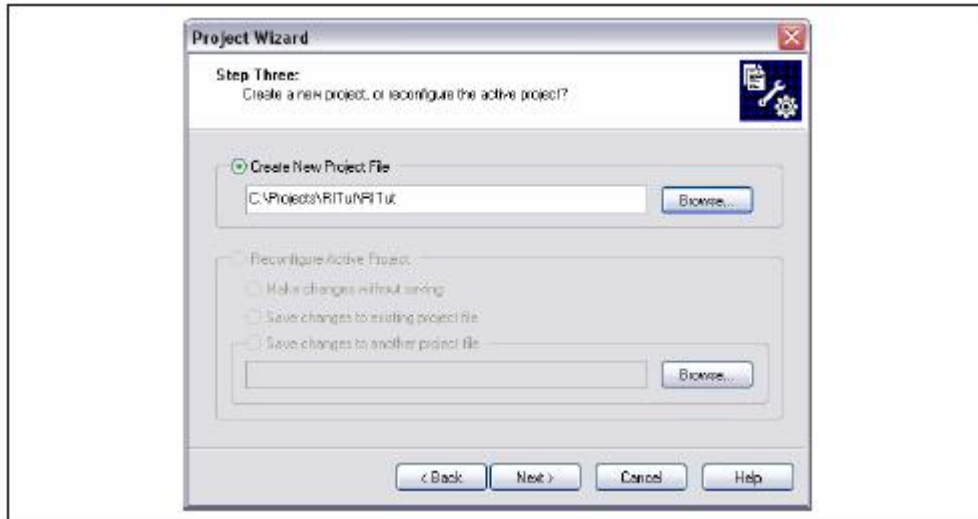


그림 6-3. 프로젝트 관리장소 결정 화면

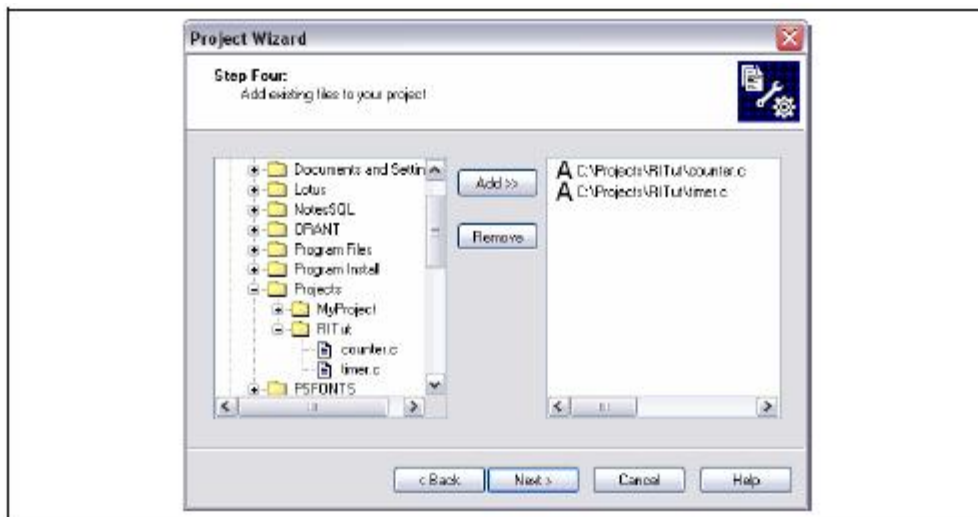


그림 6-4. 프로젝트 소스파일 불러오기 화면

4. 기타 필요한 환경 설정하기

위의 순서에 의한 작업 파일 만들기가 끝나면, ???.mcw 라는 파일이 생성된다. 이 파일은 프로젝트 파일로 안에는 아래 그림과 같은 Source files, Header files, Object files, Library files, linker script 그리고 Other files 로 구

성된다.

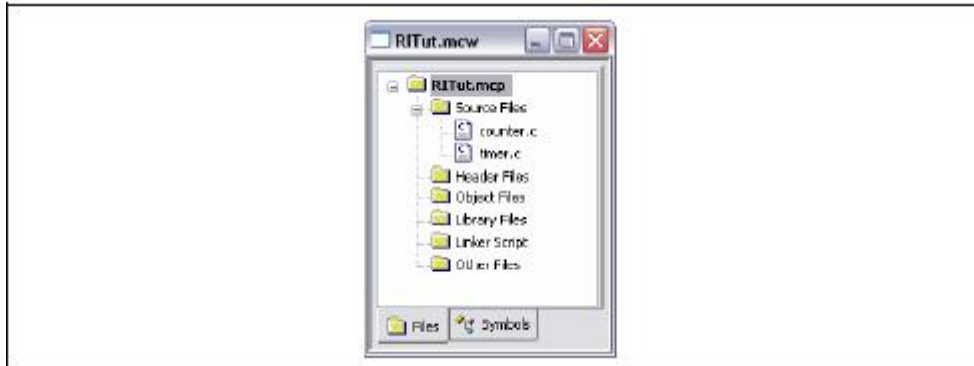


그림 6-5. 프로젝트 파일의 내부 구조도

위 파일 그룹에 개별 파일이 들어있으며 별도로 표시되나 내용이 없으면 표시되는 것이 없다. 그러면 개별적인 파일의 용도와 의미를 이해해보자

Source file: 여러분이 작성하는 프로그램이다.

Header files: 프로그램으로 공통으로 사용되는 선언문 등을 넣어두는 파일이다.

Object files: 기계어로 번역되어 있는 프로그램으로 작성한 **source** 프로그램과 결합하여 사용할 프로그램이다. 이 프로그램은 다른 곳에서 다른 작업자가 만들어 준 것을 그대로 사용하는 경우에 사용된다.

Library files: 일반적으로 고급 언어로 프로그램을 할 경우는 복잡한 기능은 기호로 작성을 한다. 그러나 프로그램이 수행될 경우는 기호가 아닌 기호에 해당하는 기능을 가지는 프로그램 집합체로 대치가 되며, 이를 위하여 만들어 둔 복잡한 개별 프로그램의 집합체가 들어 있는 곳이다.

linker script: 여러 종류의 프로그램이 하나로 뭉쳐서 한 프로그램이 되려고 하면, 프로그램이 들어가는 기억장치의 위치나 데이터가 위치할 주소 등이 결정되어야 하며, 이를 지정해 주는 것이 **linker**이다.

◆ 주의: 마이컴의 하드웨어를 이해하려면 이 프로그램을 이해해야 하고, 그 내용은 마이컴 마다 다르고, 동일 마이컴에서도 개발환경에 따라서 변경된다. 즉 개발장비를 **programmer**로 사용할 때와 **debugger**로 사용할 때 이 파일이 다르게 적용되며, **Release** 와 **Debug**를 선택할 때도 변경된다.

그리고 앞에 보이는 프로젝트 파일에서 개별 파일의 내용을 삭제/추가 하

려면 그 항목을 선택하고 마우스 오른쪽 키를 누르면 된다. 예로 **source file**에 새로운 파일을 추가하려면, **source file**를 선택한 후 오른쪽 마우스를 누르면 메뉴가 나온다. 그리고 기존에 들어있는 **source files**를 제거하려면, 제거 하려는 **file**를 선택하고 **delete key**를 누르면 된다. 또 그 프로그램을 화면에 올릴려고 하면 더블 클릭하면 된다.

5. 프로그램을 수행시키기 위한 준비 단계

앞의 단계에 따라서 작성한 프로그램을 실제로 수행시키기 위해서는 컴파일하고 링크하여 선택된 마이컴의 메모리 구조에 맞게 배치해야 한다. 그러려면 구체적인 마이컴의 이름을 결정해 주어야 한다. 개발장비 기본화면에서 **Configure** 항목을 누르면 **Devices select**, **configuration bit**, **settings** 등의 항목이 나온다. 여기서 **Devices select**를 누르면 **Device**를 선택할 수가 있다. 이 작업은 한번만 하면 된다.

수행시키는 방법은 개발 화면에서 **Project** 항목을 선택하고 **Build All** 를 수행시키면 된다. 그러면 아래의 화면이 나타날 것이다. 단 아래 화면은 **debugger** 상태로 설정한 경우이다.

만약 컴파일 과정에 에러가 발생되면, 에러 메시지를 띄우며, 에러가 발생한 프로그램의 줄 번호와 에러 내용을 알려주므로 이를 보고 맞도록 수정을 해야한다. 정보가 발생되는 경우는 컴파일을 했는데, 프로그램에 작성된 정의가 불충분하여 자기가 임의로 했다는 것이므로 표시하는 내용에 따라서 문제가 없는 경우도 있고, 매우 심각한 문제가 일어날 수도 있다.

이러한 정보의 예는 정수변수에 소수점이 있는 값을 넣어라고 했던가, 또는 사용할 수 있는 숫자 보다 큰 값을 넣어라고 한 경우이다. 전자의 경우는 심각할 수도 않을 수도 있으나, 후자의 경우는 범위를 넘는 값은 버리고 나머지만 넣게 되므로 매우 심각한 문제가 발생한다.

이렇게 정상적으로 프로그램이 작성되고 컴파일되고 링크되면 수행파일이 만들어진다. 수행파일의 이름은 **????.hex** 이며, 이 내용이 마이컴의 메모리에 기억되어지는 내용이다. 즉 기계어로 만들어진 프로그램이라는 것이다. 그러니 우리가 봐야 별 도움이 안되는 코드만으로 만들어진 것이다. 따라서 개발자가 디버깅 하면서 프로그램의 수행 과정을 보려고 하면 별도의 프로그램이 필요하며, 이것이 **????.cof** 파일이다.

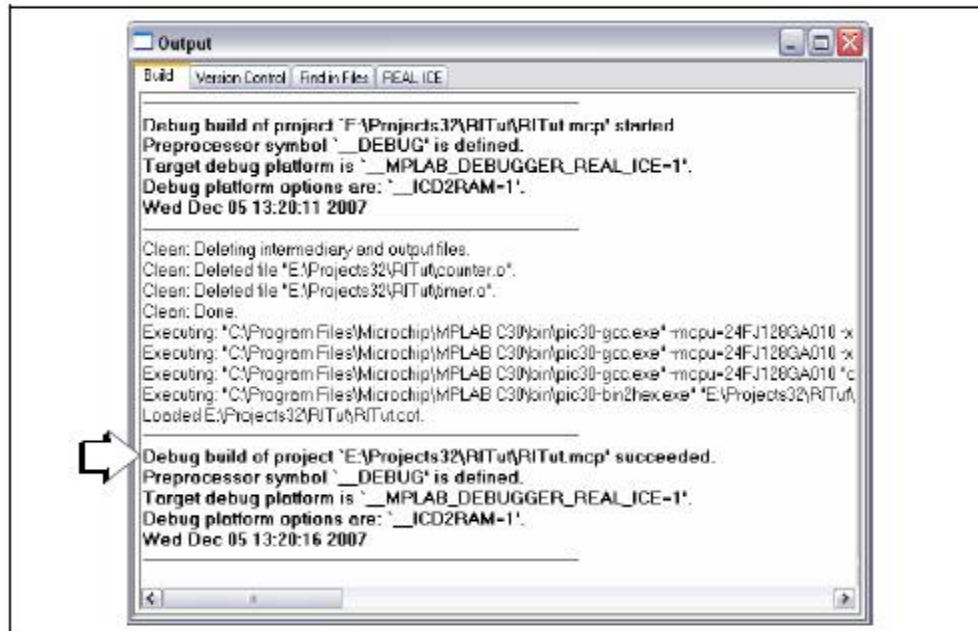


그림 6-6. 작성된 프로그램을 컴파일 한 경우의 화면

또 작성된 프로그램을 링커가 메모리에 어떻게 배치했는가를 알려주는 파일이 **???map** 이다. 따라서 이 map 파일을 관심있게 보면, 컴파일러의 의미와 프로그램이 수행되는 원리라든가 하는 것을 알 수가 있으며, 특히 고급언어로 작성한 프로그램이 정상적으로 수행되지 않을 경우의 해결책을 찾을 수 있는 좋은 실마리를 알려주는 보물창고이다. 여러분도 관심있게 보고 익숙해지기를 바란다.

작성된 프로그램을 수행하기에 앞서 마지막 단계이면서 매우 중요한 것이 아직 남아있다. 해당 마이컴에 대한 하드웨어적인 구조를 설정해 주어야 한다. 이 과정은 한번만 해 주면 이 프로젝트 안에서는 항상 같은 조건으로 동작된다. 그러면 프로그램과 별개로 왜 이러한 설정을 해 주어야 하는가? 이는 한 마이컴이라고 하여도 다양한 조건에서 동작될 수 있도록 하기 위해서이며, 대표적인 것이 동작 속도를 결정해 주는 것이다.

개발장비 기본화면에서 **Configure** 항목을 누르면 **Devices select, configuration bit, settings** 등의 항목이 나오며, 여기서 **configuration bit**를 누르

면 설정항목이 나타난다. 이 작업은 한번만 하면 된다.

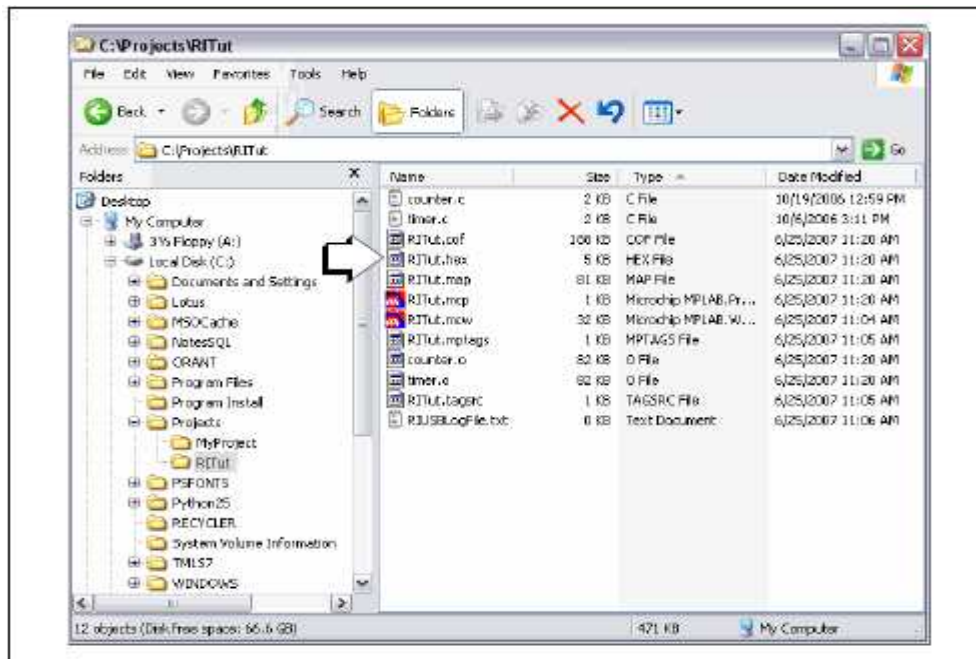


그림 6-7. 컴파일이 완료된 프로젝트 파일의 파일 구조

PIC16F876A 마이컴에서는 총 8개의 설정 항목이 있으며, 설정치를 변경하고자 하면 **setting** 항목을 클릭하면 다른 조건들이 나타난다.

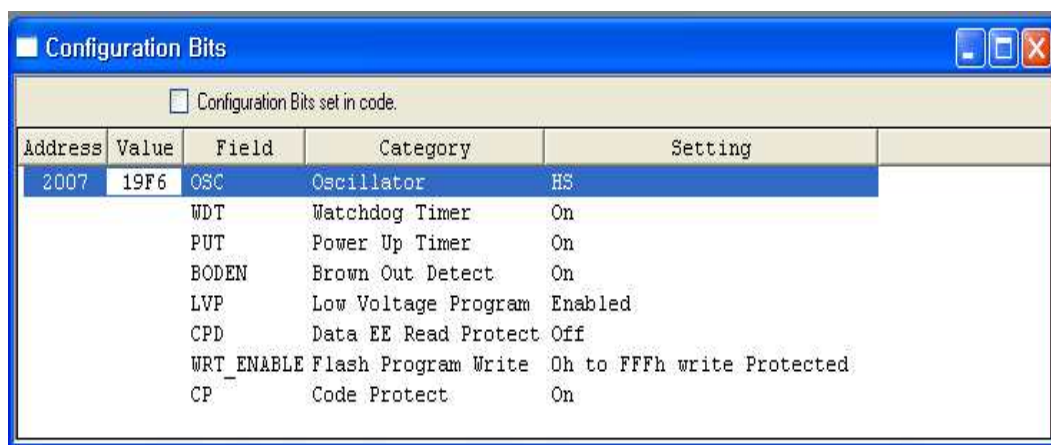


그림 6-8. 마이컴 하드웨어 설정 화면

여기서 모든 항목을 자세히 설명하기는 한계가 있으므로 기초적인 내용과 의미만 설명한다.

- **OSC 선택: RC/LP/XT/HS 4 종류가 있다.** 즉 마이컴에 사용되는 클럭의 주파수 범위에 따라서 이를 구동하는 회로가 다르기 때문에 사용하는 것이다. **RC**는 마이컴 내부에 들어있는 **R**과**C**를 이용하여 자체적으로 클럭을 만들어 사용하겠다는 의미이다. 따라서 이 경우는 외부에 오실레이터를 부착할 필요가 없다. 그러나 이 방식은 낮은 주파수 클럭과 불안정한 클럭이 만들어지므로 간이형 제어기에서 만 사용 가능하다. **LP**는 전력 소모가 문제가 되는 곳에서 전기 소모를 최소화하는 방식으로 구동되도록 만들어진 클럭이다. 따라서 리모콘 등의 특수한 응용에서만 사용된다. **XT**는 **4Mhz** 이하의 오실레이터나 크리스탈을 사용하는 경우에 사용되며, **HS**는 **4Mhz** 이상의 주파수에서 사용된다. 따라서 이것의 설정이 잘못되면 동작 자체가 안될 수가 있다. 우리는 **HS** 모드를 사용할 것이다.

BODEN(brown out detect)는 마이컴에 가해지는 전원이 특정 전압 이하가 되며, 마이컴을 리셋시키는 기능이다. 이는 낮은 전압에서 마이컴이 불안정한 동작을 하여 오동작을 하는 것을 막기 위한 것이다. 마지막 **CP(code protect)**는 힘들게 작성한 프로그램을 남이 그대로 복사하여 사용하지 못하도록 만드는 기능이다. 즉 이것을 **ON** 상태로 두면 마이컴에 기록된 프로그램을 어떠한 수단을 동원해도 읽을 수가 없다는 것이다,. 그래도 읽어보면 전부가 **00**인 코드가 올라온다.

따라서 이 부분은 자신이 생길 때까지는 절대로 **건드리지 말아야할 부분**이다.

6. 프로그램을 작성하기

에디터 상태에서 사용할 수 있는 기능키들은 다음과 같은 종류가 있습니다.

블록 설정과 관계된 기능키

SHIFT+커서키	블록을 설정한다.(PGUP등도 사용 가능)
CTRL+X	현재 선택된 영역을 클립보드로 이동
CTRL+C	현재 선택된 영역을 클립보드로 복사
CTRL+V	클립보드의 내용을 커서 위치로 복사

파일처리와 관계된 기능키

CTRL+O	다른 파일을 오픈
CTRL+S	현재 작업중인 내용을 세이브
단순 기능키	
커서키 UP	(커서를)위로 이동
커서키 DOWN	(커서를)아래로 이동
커서키 LEFT	(커서를)좌로 이동
커서키 RIGHT	(커서를)좌로 이동
HOME	(커서를)한 문장의 제일 처음으로 이동
END	한 문장의 제일 끝으로 이동
CTRL+HOME	문서의 제일 처음으로 이동
CTRL+END	문서의 제일 끝으로 이동
PGUP	한 페이지 위로 이동
PGDN	한 페이지 아래로 이동
CTRL+F	단어 찾기
F3	찾기 반복
SHIFT+F3	뒤로 찾기
CTRL+H	REPLACE
CTRL+Z	UNDO

위의 기능키들은 에디팅 작업 중에 자주 사용하게 되는 것이므로 외워두면 편리합니다. 특히 **CTRL+S키(세이브)**는 **에디팅 중에 자주 사용하는 것이 좋습니다.**

☞ 주의 : 각자의 source program은 디스켓에 각자가 보관하여 파괴로부터 보호합니다.

☞ 주의 : File의 이름과 파일이 들어있는 디렉토리 이름(중간 path에 있는 것도 포함)의 길이가 영문자로 8자(한글 1자는 영문자 2자에 해당됨)를 초과하면 파일을 인식하지 못합니다. 이는 윈도우의 바탕화면이나 내문서 폴더에 소스파일을 저장하면 인식할 수 없음을 의미합니다. 그리고 이름에 blank, -, * 등의 특수 기호가 들어가면 안 됩니다. 따라서 source file는 실험반(분반/조) 단위로 directory를 만들어 사용하십시오.

예) 3E01 (3-E반 01조)

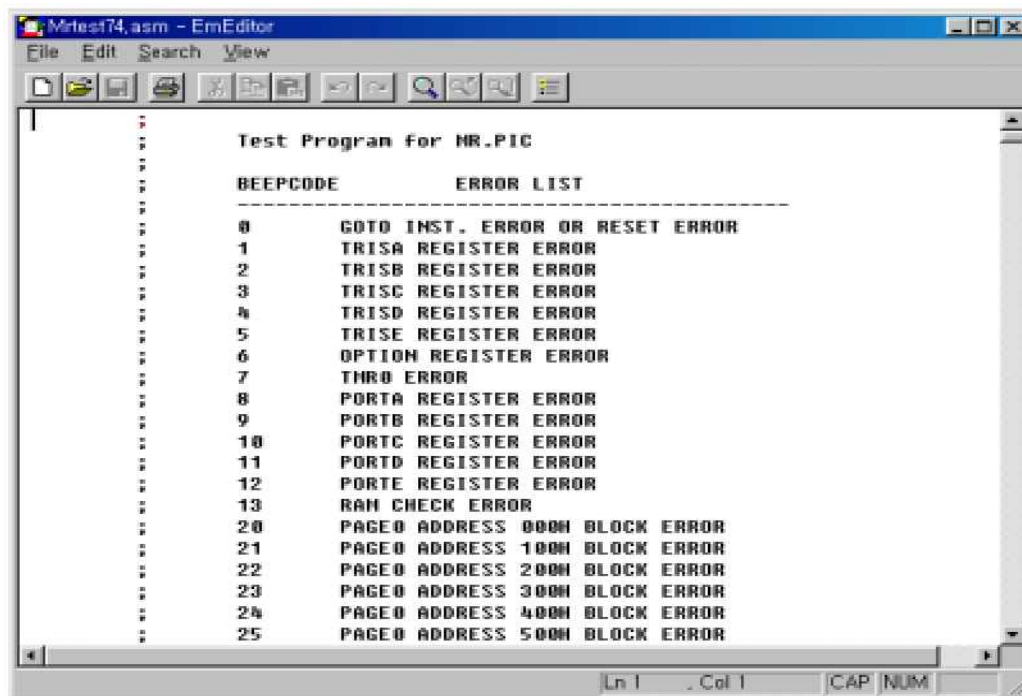


그림 6-4. 에디터 화면

7. 어셈블링하기

소스 작성 및 세이브가 끝났으면 Project 창을 펼쳐서 Build All를 선택해서 어셈블딩 해야합니다.. 어셈블 결과 생성되는 파일은 다음과 같습니다.(소스 파일이 ABC.ASM인 경우...)

파일명	설 명
ABC.COD	에뮬레이터용 코드파일
ABC.HEX	인텔 HEX 파일(MR.PIC-PGM에서 사용하는 파일입니다.)
ABC.LST	프린트할 수 있는 리스팅 파일
ABC.ERR	에러 발생시 생성되는 에러 레포트 파일

☞ 주의: 반드시 어셈블 진행 상황을 보여주었던 Output 창의 Build 윈도우가 사라진 후 다음 작업으로 진행해 주어야 합니다. 어셈블 상황 윈도우가 표시된 상태에서는 어떠한 작업도 불가능합니다. (원하는 파일들이 생성되기 전이기 때문입니다.) 어셈블 시 발생하는 WARNING이나 MESSAGES는 주로 기억소자의 페이지 전환 부분에서 주의를 환기시키는 의미에서 발생하는 것이므로 무시하셔도 좋습니다.