

Pocket PC의 Windows CE용 프로그램을 개발하기 위한 Microsoft SDK 사용법

김대희

1. Window CE를 탑재한 다양한 이동 단말기의 형태

■ WinCE PDA

정해진 용도에 맞추어 사용되도록 기본 프로그램만 탑재되어 있으며, 특별한 서비스에 맞는 프로그램이 탑재되도록 고안되어진 것으로 마이크로소프트가 시장에서 주도적으로 관심을 가지고 있는 디바이스가 아니다. 사이버뱅크등에서 PDA에 Phone기능을 부가한 것은 이러한 특별한 목적으로 전화기능을 부가한 것이다. 따라서 운영체제 수준에서 전화와 PDA가 결합된 것이 아니고 PDA가 Phone기능을 수용하는 중간 단계 디바이스로 곧 사라질 것으로 전망된다.

■ Pocket PC

마이크로소프트가 Palm-size PC와 경쟁적으로 사용하기 위해 출시된 것으로 WinCE 방식의 PDA보다 응용프로그램이 많고, 2003년 6월 현재 아시아 시장에서 Palm-Size PC를 제치고 시장 점유율을 1위를 차지했다. 많은 응용프로그램 특히 게임 프로그램들은 GAPI(Game API)를 이용하여 개발되고 있다. Pocket PC에서 동영상 플레이어를 개발한다고 하면 GAPI의 사용은 필수적이다. GAPI는 그래픽뿐만 아니라 키보드까지 제어한다.

■ Pocket PC Phone Edition

현재 한국이 GSM 방식을 채용하고 있지 않아 사용되지 않지만 조만간 한국의 CDMA방식을 지원하는 Phone 기능이 운영체제 차원에서 지원되어 PDA와 이동전화가 합쳐지는 서비스 창출이 가능할 것으로 기대된다.

■ Smart Phone

Pocket PC Phone Edition이 사이즈가 크기 때문에 이것을 축소화 시켜 이동전화 사이즈로 만든 것이다. 삼성이 조만간 MS와 계약을 맺고 출시할 것으로 예정되어 있다.

2. Embedded Visual Tool의 다양한 버전

■ Embedded Visual Tools 3.0

- i. Window CE 3.0을 위한 개발툴이다.
- ii. PC에서 Emulator를 관리하기가 용이하다.

■ Embedded Visual Tools 3.0 - 2002

- i. Microsoft Windows Pocket PC 2002를 위한 개발툴로 3.0과 호환된다.
- ii. PC에서 Emulator를 관리하기가 불편하다.

■ Embedded Visual Tools 4.x

- i. Window CE.Net를 위한 개발툴이다.
- Pocket PC 2003 SDK
 - i. Visual Studio .NET 2003과 통합이 진행중이다.
 - ii. Windows CE .NET 2003은 Windows Mobile로 이름이 출시와 동시에 바뀌었다. Microsoft의 .NET 전략의 마지막 축인 Mobile Device에서 컴퓨팅 영역을 다루며, 일반 PC의 .NET Framework과 비교하여 .NET Compact Framework를 갖는다.

3. 개발환경

- PDA용 프로그램 개발은 그림 1과 같은 환경 아래에서 개발된다. 그림에서 보듯이 별도의 PDA가 없을 경우에는 SDK에 같이 패키지로 따라오는 Emulator를 사용하여 개발할 수 있다. 실제로 PDA에 연결하여 테스트할 때는 USB 포트로 PC와 PDA를 연결하고 ActiveSync라는 프로그램을 통하여 PPC와 PC 사이에 데이터 및 프로그램을 복사하거나 이동시킬 수 있다. 본 문서에는 eMbedded Visual C/C++ 3.0(Embedded Visual Tools 3.0)을 사용하여 프로그램을 개발하고, ActiveSync 3.6(Microsoft의 홈페이지에서 얻을 수 있음)을 이용하여 PC와 PDA를 연결하였다. PDA는 기본적으로 COMPAQ의 iPAQ시리즈를 사용하고 PPC의 운영체제는 Window CE 3.0이다.

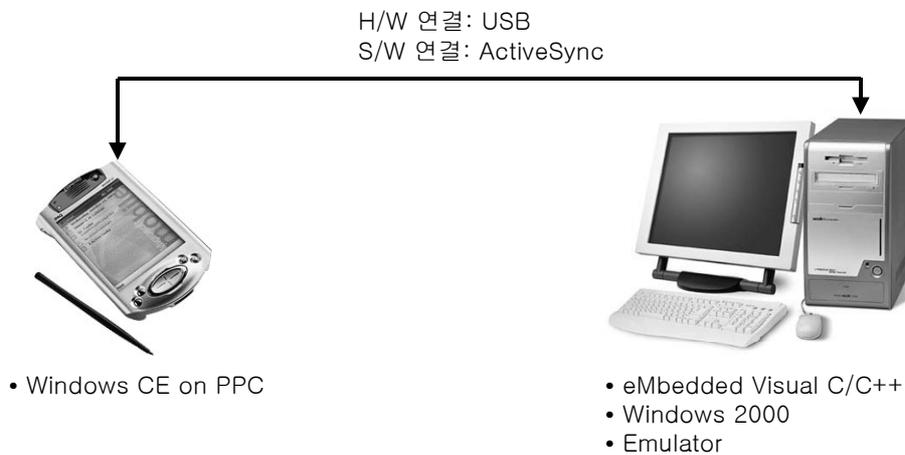


그림 1 개발환경

4. eMbedded Visual C/C++ 에서 Project 생성, 컴파일 및 실행

- Embedded Visual Tools 3.0은 Microsoft 홈페이지에서 다운 받을 수 있으며, 기본적인 eMbedded Visual C/C++ 3.0의 사용법은 Microsoft Visual C/C++와 같다. 프로젝트를 만들기 위해서 File->New를 선택하면 그림 2와 같은 화면이 나온다. 여기서 PPC용 응용프로그램을 만들기 위해선 WCE Pocket PC

Application을 선택한다. 그리고 그림의 오른쪽 하단에 사용하고 싶은 CPU를 선택하도록 되어 있는데 가능한 많은 CPU를 포함하여 나중에 쉽게 다른 CPU에서 돌아가는 프로그램을 만들기 용이하게 하는 것이 좋다. Cassio 계열의 PPC는 MPPS를 선택해야 하고, Compaq의 PPC는 ARM을 선택해야 한다. 그리고 그림에서 보듯이 x86em은 Emulator를 위한 CPU이다. 일반적인 PC가 x86계열이므로 그 위에서 구동 가능한 Emulator용 프로그램을 만들기 위해서 필요하다. 그리고 나머지 프로젝트를 생성하기 위한 과정은 Visual C/C++와 유사하다.

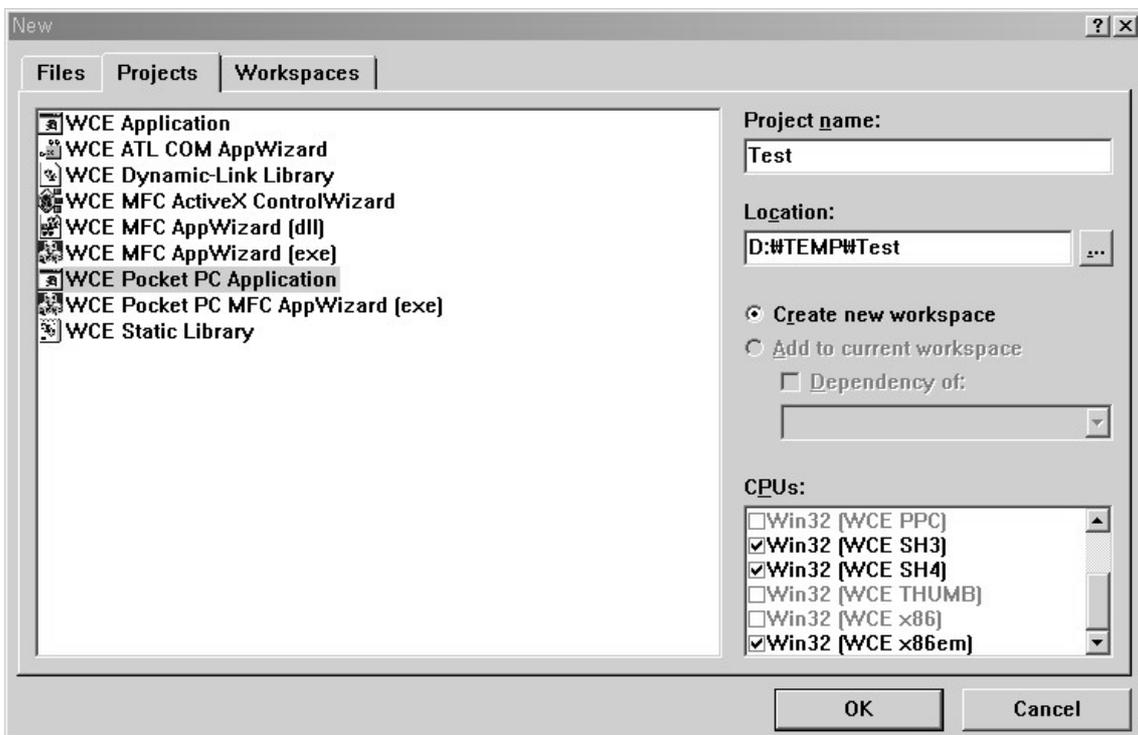


그림 2 프로젝트 생성

- 프로그램을 작성하고 컴파일할 때 사용할 PDA의 기종을 그림 3과 같이 Combo 박스에서 Palm PC인지, Handheld PC인지, Pocket PC 인지 결정한다. 그리고 그림 4에서 사용하고자 하는 PDA의 CPU 타입을 선택한다. 이때 PC에서 에뮬레이션을 원할 경우에는 Win32 (WCE x86em)을 선택한다. 에뮬레이션 모드일 경우는 Pocket PC Emulation 모드라는 표시가 다음 Combo 박스에 나타난다.
- 프로그램을 작성하고 컴파일 및 실행방법은 기존의 Visual C/C++와 동일하다. 그리고 실행 시킬 때 CPU 타입에 관련되는 오류메시지 박스가 나타날 수 있지만 무시하고 넘어가도 실행시 아무런 문제가 없다. PC에서 실행 시킬 경우에

는 실행파일을 emulator가 관리하는 디렉토리로 복사해야 한다. PPC에 연결하여 실행시킬 경우에는 ActiveSync로 실행 파일을 PPC로 복사하여 PPC에서 실행시킬 수 있고, eMbedded Visual C/C++의 실행버튼을 눌러 직접적으로 PPC에서 실행시킬 수도 있다.

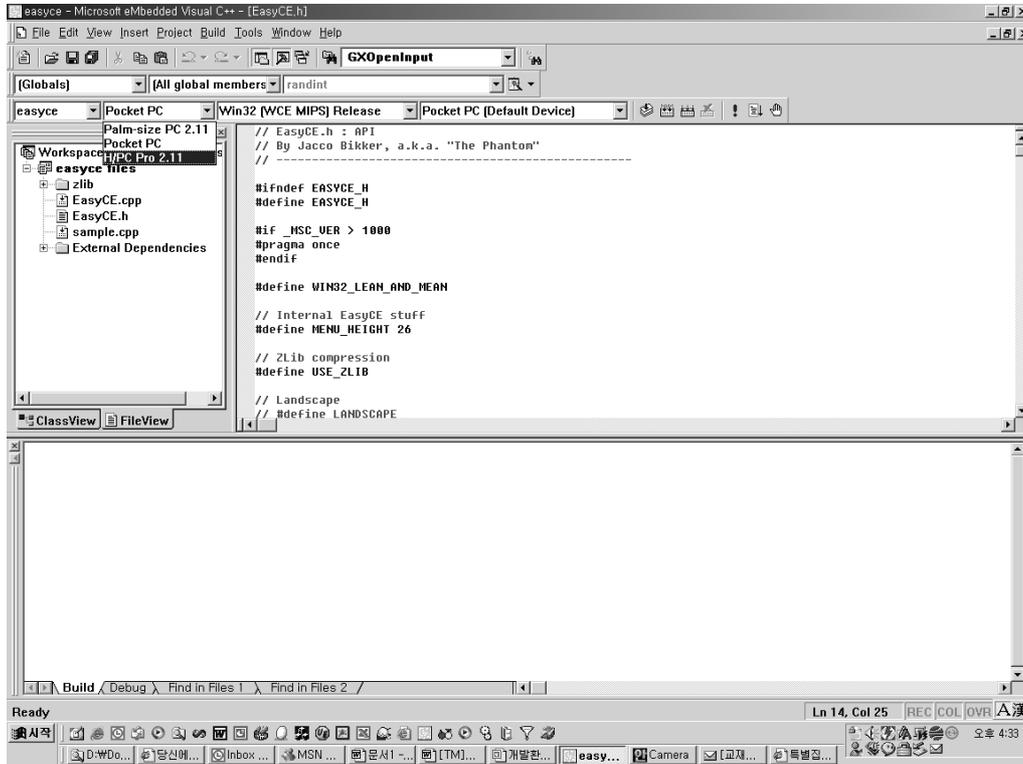


그림 3 PDA 종류 선택

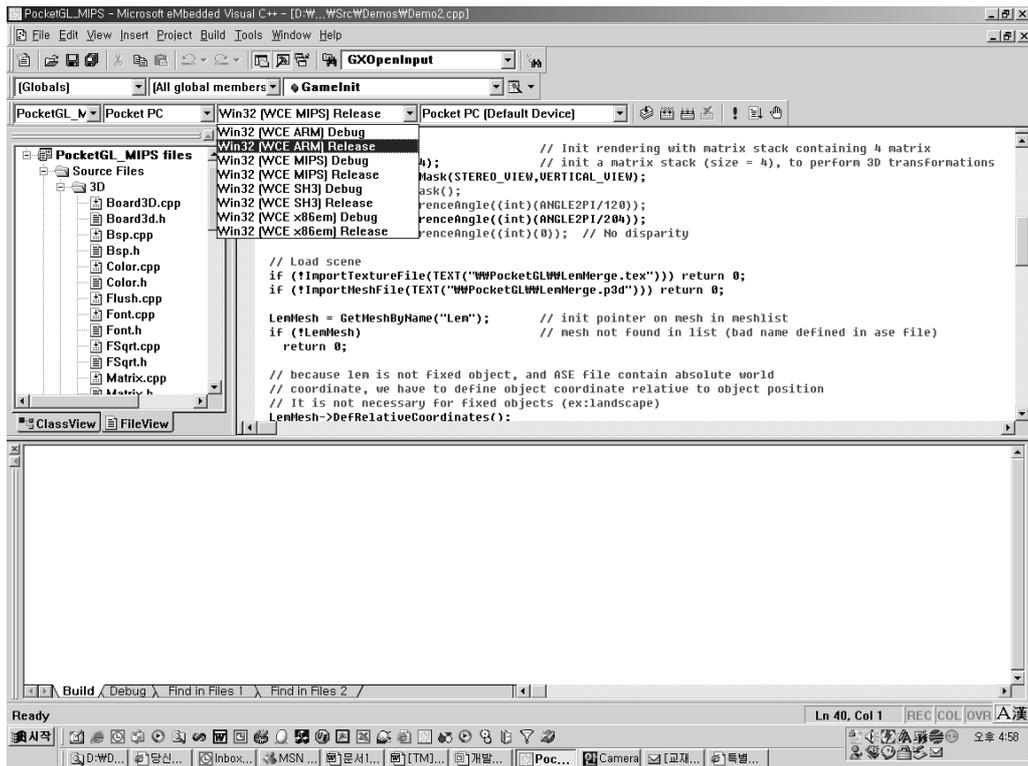


그림 4 CPU와 컴파일 모드 선택

5. Emulator의 사용

- PC에서 PDA 장비 없이 Emulator를 사용할 수 있고, 그림 5는 PPC Emulator를 구동한 화면이다. 그림에서 보듯이 실제 PPC와 유사한 화면을 제공하고 있다.



그림 5. PPC Emulator

- 앞에서 Embedded Visual Tools 3.0이 Embedded Visual Tools 3.0 - 2002 보다 Emulator를 관리하기가 용이하다고 하였다. 그 이유는 Embedded Visual Tools 3.0 - 2002은 실제 구동환경과 유사하게 만드는 것에 중점을 두어 Emulator의 모든 프로그램과 시스템 소프트웨어를 바이너리 이미지로 관리한다. 따라서 사용자가 Emulator의 시스템 환경을 변경하려면 그것을 매번 저장하고 종료를 하거나, 아니면 매번 Emulator를 구동 시킬 때 마다 설정을 다시 해야 하는 번거로움이 있다. 그리고 무엇보다도 Emulator가 바이너리 이미지 형태로 관리 되기 때문에 Emulator상에서 디렉터리나 파일 관리를 eMbedded Visual C/C++의 통합환경의 메뉴 Tools->Remote Process Viewer를 통해서 관리해야 한다.
- Embedded Visual Tools 3.0은 Emulator의 파일 및 폴더 관리를 윈도우 탐색기를 통해서 손쉽게 할 수 있다. 그림 6은 Emulator를 위한 폴더를 보여주고 있다 그림에서 Windows와 Program Files 폴더는 PPC에 기본적으로 설치되어 있는 것이고, 나머지 폴더는 사용자가 만든 것이다. Embedded Visual Tools 3.0에서 PPC를 위한 Emulator의 폴더 위치는 "WINDOOWS_CE_TOOLS\wce300\WMS Pocket PC\Wemulation\Wpalm300"이다. 여기서 WINDOOWS_CE_TOOLS는 Embedded Visual Tools가 설치된

루트 폴더이다.

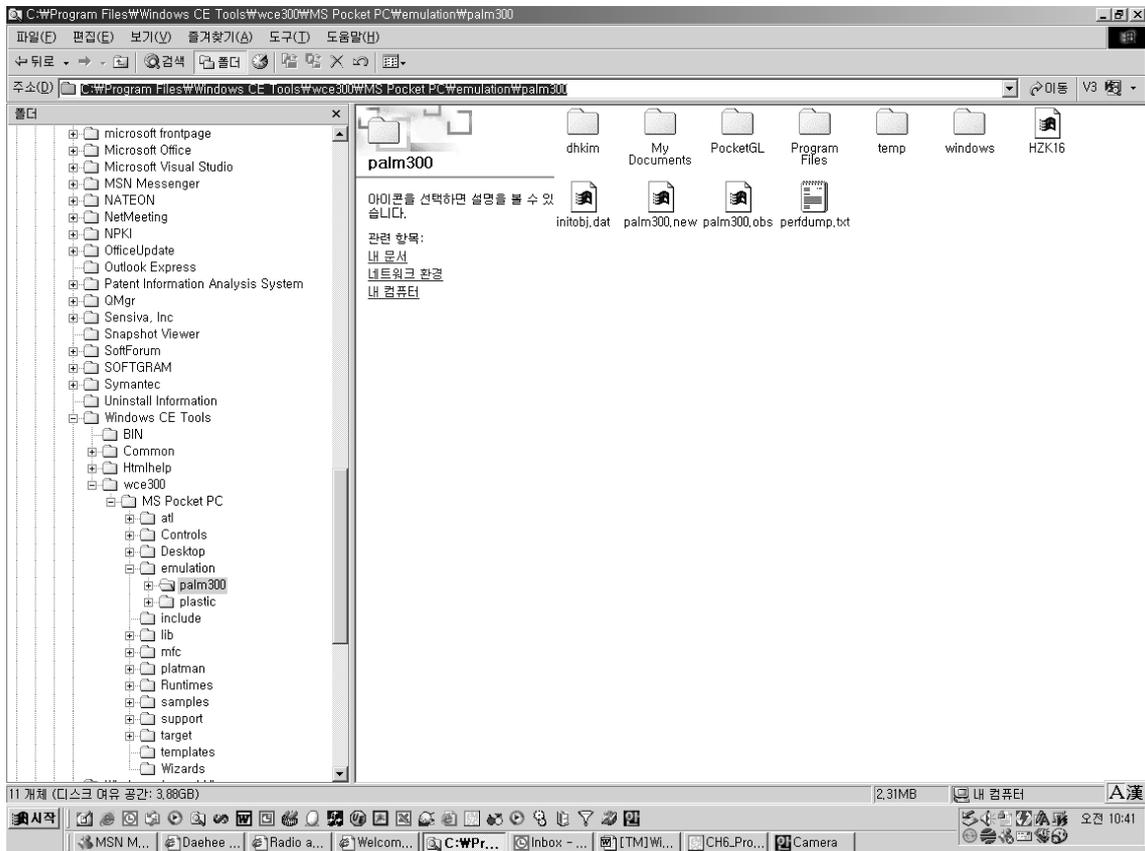


그림 6. Emulator를 위한 폴더

- GAPI와 같은 특수목적의 라이브러리를 사용하기 위해선 PPC나 Emulator의 시스템 폴더에 Dynamic Linked Library(DLL)을 복사하여야 한다. GAPI의 경우 Emulator를 위한 dll 파일이 나와 있어서 Embedded Visual Tools 3.0에서는 앞에서 말한 Emulator를 위한 폴더에서 Windows 폴더에 복사하면 되므로, Embedded Visual Tools 3.0에서는 GAPI를 이용한 프로그램을 테스트할 수 있다. 그러나 Embedded Visual Tools 3.0 - 2002에서는 바이너리 이미지로 관리되기 때문에 Emulator에서 GAPI를 이용한 프로그램을 테스트할 수 없다. 그림 7은 GAPI를 이용하여 작성된 프로그램을 PC 상의 Emulator를 이용하여 실행시킨 결과의 화면이다.

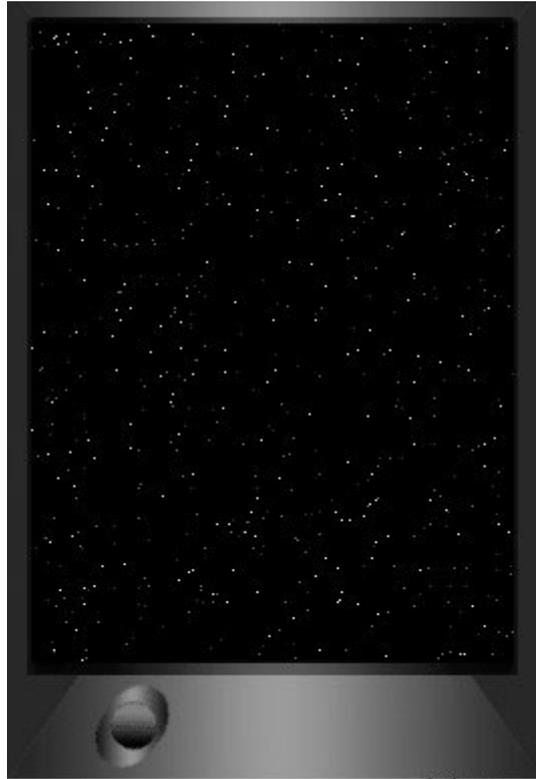


그림 7. GAPI를 이용한 프로그램을 Emulator에서 테스트한 예제

6. 특정 Library를 이용하여 컴파일하는 방법

- 본 문서에서는 GAPI를 이용하여 컴파일하는 방법을 예로 하여 특정 라이브러리를 사용하는 방법을 설명하고자 한다. 기본적인 사용법은 Visual C/C++와 사용법이 같다. 다른 점이 있다면 여러 개의 CPU를 지원하고 다양한 PDA 디바이스를 지원하기 때문에 거기에 맞는 Library와 파일을 선택하여야 한다.
- 그림 8은 특정 라이브러리를 위한 헤더 파일의 위치를 등록하기 위한 창으로 eMbedded Visual C/C++의 통합 환경에서 Tools->Options를 선택하고 Directories 탭을 선택하면 나타난다. 그림 8에서 왼쪽의 Combo 박스는 디바이스의 타입을 나타내고, 가운데 Combo 박스는 CPU의 타입을 나타내고, 마지막 Combo 박스는 현재의 창이 Header 파일을 위한 폴더를 설정하기 위한 것임을 표시한다.
- 그림 9는 Library 파일의 위치에 관한 등록을 하기위한 창으로 그림 8에서 마지막 Combo 박스에서 Library Files를 선택하면 된다. Header 파일은 CPU가 달라도 같을 수 있지만 Library 파일은 그렇지 않기 때문에 CPU에 맞는 Library인지를 확인하고, CPU 타입에 맞는 Library가 들어있는 폴더를 설정해야 한다.

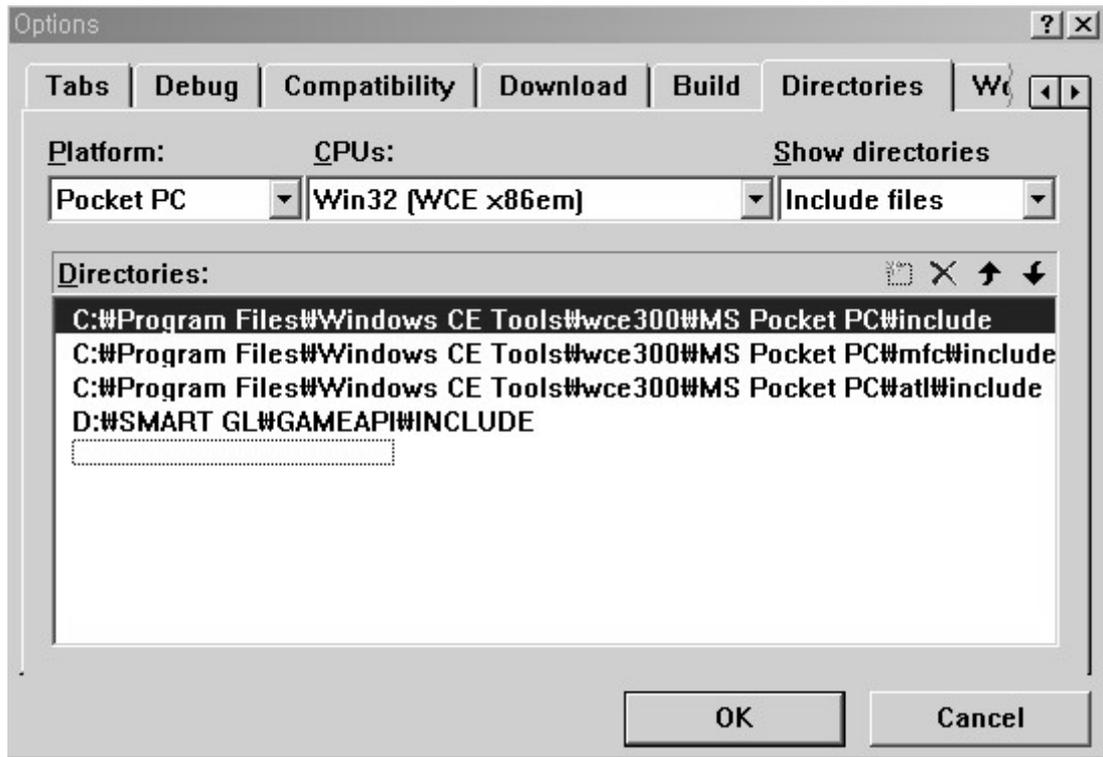


그림 8. Include 파일 위치 설정

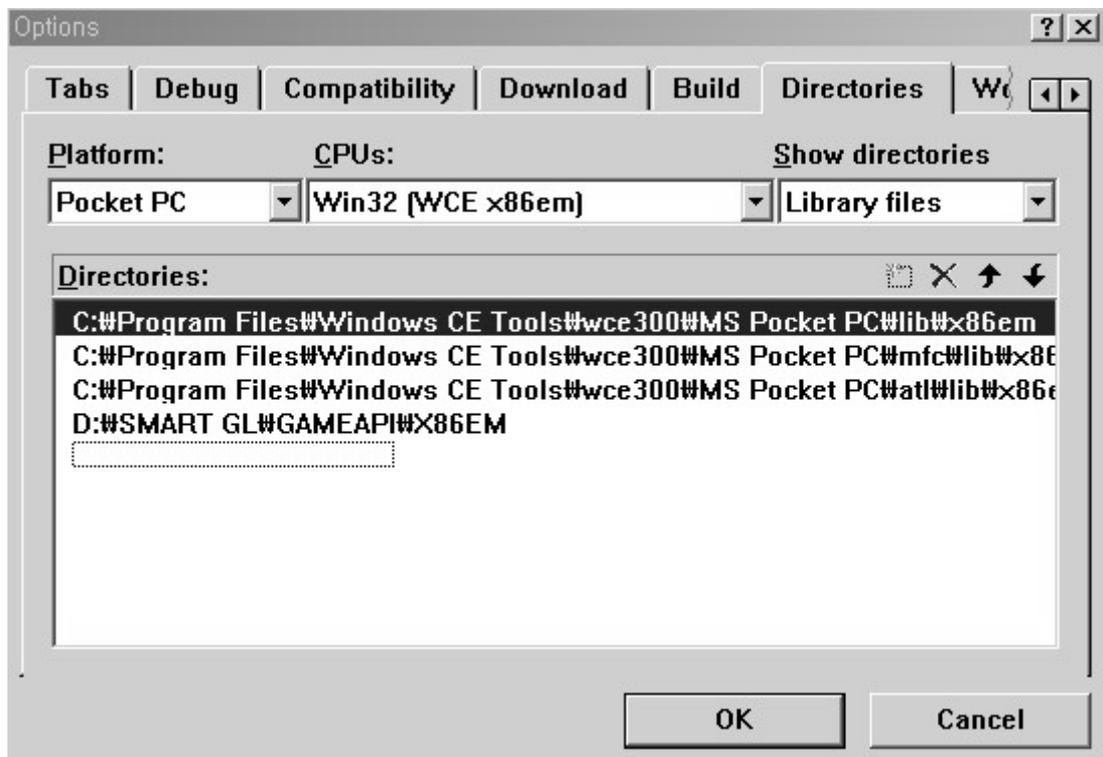


그림 9 Library 파일 위치 설정

- 컴파일러에게 해당 Library를 사용할 것을 알려주기 위해서는 그림 10과 같이 사용될 Library를 추가해야 한다. 여기서는 GAPI를 위해서 gx.lib를 추가하였다. 컴파일이 끝나고 실행하려면 gx.dll을 PPC의 Window 폴더에 복사를 하여야 한다. 물론 Emulator로 실행할 경우에는 앞에서 언급한 Emulator를 위한 폴더에 복사를 하여야 한다. 이때 CPU의 타입에 맞는 *.dll 파일을 복사하여야 한다.

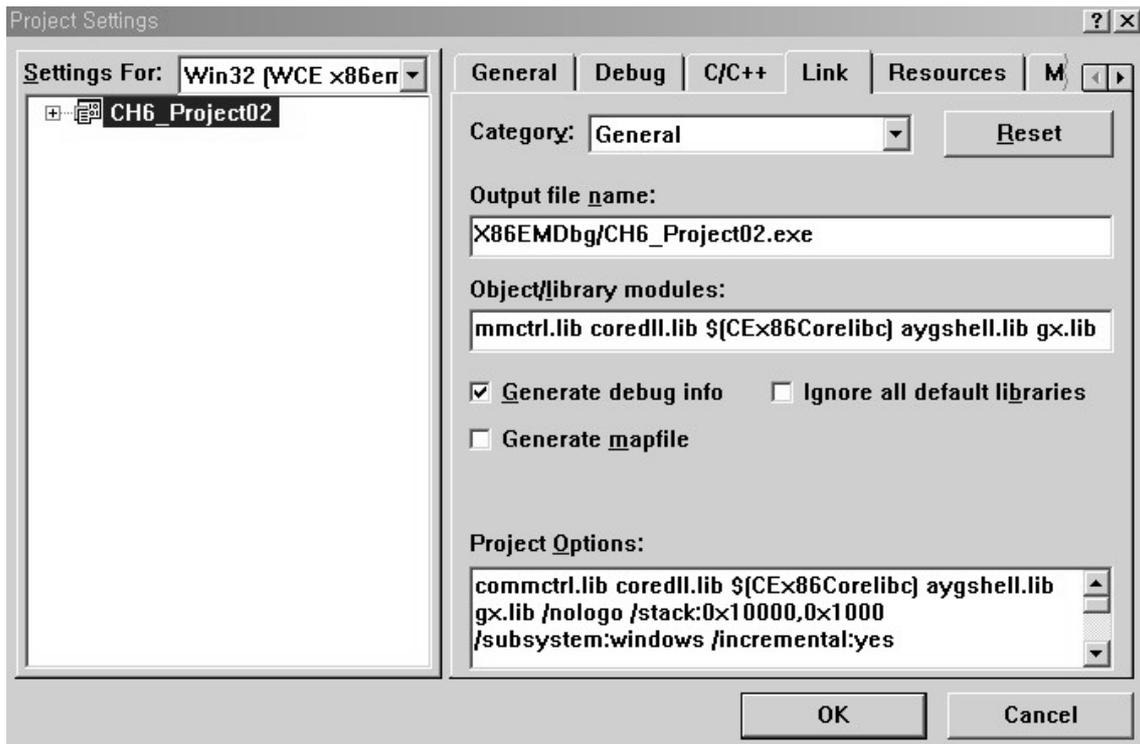


그림 10. 컴파일 옵션 선택