AutoBase

Visual C# Developer Guide

[Ver. 2014.10.30]

Intelligent Touch Panel Computer

autobaseTouch

목 차

1 장 ATS-M1000DA 시리즈 준비사항 5

1.1 설치 전 준비사항 5

1.2 연결하기 5

1.2.1 ATS-1000DA 시리즈 전원연결 5

1.2.2 PC 와 ATS-1000DA 시리즈 연결 6

1.3 모드 설정 방법 6

1.4 응용 프로그램 AutoRun 모드 7

1.4.1 AutoRun 모드 응용 프로그램 추가, 편집 8

1.4.2 AutoRun 삭제 9

1.4.3 AutoRun 내용저장 9

2 장 개발 환경 설정 10

2.1 Microsoft Visual Studio 2008 실행 10

2.2 ActiveSync 설치 18

2.2.1 Active Sync/Windows Mobile Device Center 설치 18

2.2.2 USB 케이블 연결 18

2.2.3 Windows XP ActiveSync 설치 19

2.2.4 Windows Vista/Windows7 ActiveSync(Windows Mobile Device Center) 설치 20

3 장 Visual C# Windows CE 프로젝트 작업 21

3.1 Visual C# 프로젝트 생성 과정 21

3.2 Visual Studio 화면(UI) 구성 24

3.3 Visual Studio C# 장치 프로젝트 만들기 25

4 장 ATS 시리즈 라이브러리 파일 29

4.1 Ats 라이브러리 파일 설치 29
4.2 Ats 라이브러리 파일 등록 29
4.2.1 AtsCppUD 파일 이용 방법 31
4.3 Ats 라이브러리 파일 사용 방법 32
4.4 Ats 라이브러리 파일의 클래스 구성 33
4.5 Ats 라이브러리 파일의 클래스 구성 34

업데이트된 사용 설명서는 저희 회사 홈페이지 www.autobase.biz에서 확인 하시기 바랍니다.

본 제품을 사용하기 이전에 반드시 본 사용설명서를 충분히 읽어 본 뒤 사용하시기 바랍니다.

진행상 문의사항은 오토베이스 홈페이지의 고객지원 > Q/A 혹은 자사의 기술지원팀으로 연락 주시기 바랍니다.

1장 ATS-M1000DA 시리즈 준비사항

ATS-M1000DA 시리즈는 사용자가 직접 프로그램을 개발/배포 하여 이용하실 수 있습니다.
본 사용 설명서는 Visual C#을 이용하시는 개발자에게 빠른 이해와 쉬운 개발 환경 구축을 안내하며, Visual C#에 대한 사전지식이 필요 합니다.

1.1 설치 전 준비사항

ATS-M1000DA 시리즈를 통해 제품 프로그램 개발을 하기 위해서는 먼저, ATS-M1000DA 시리즈의 실행 및 PC와의 연결이 되어야 하며, 필요한 구성품은 <그림 1-1>과 같습니다.

<그림 1-1> ATS-M1000 시리즈/전원 어댑터/USB2.0 케이블

1.2 연결하기

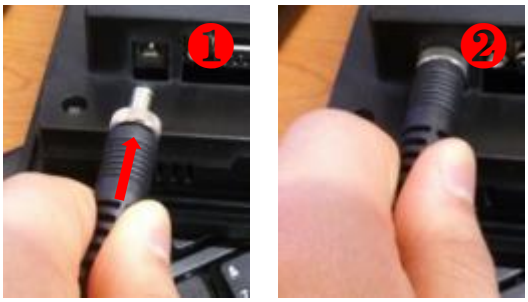
ATS-M1000DA 시리즈는 DC 어댑터에 12V DC 전원을 입력하거나, 3핀 전원입력 소켓에 24V 주전원을 입력하여 선택적으로 연결 하실 수 있고, 전원 연결을 완료 하신 후에는 PC와 ATS-M1000DA 시리즈를 연결 함으로써, 개발 환경 구축을 위한 준비를 마치실 수 있습니다..

[참 고]	매뉴얼은 DC 어댑터로 전원 연결을 한 것으로 작성 되어져 있습니다.

1.2.1 ATS-1000DA 시리즈 전원연결

<그림 1-2>와 같은 DC 어댑터 전원 입력 소켓에, <그림 1-3>처럼 어댑터를 연결 시켜주십시오.

<그림 1-2> DC 어댑터용 전원입력 소켓의 모습



<그림 1-3> **DC 어댑터 전원 연결 모습**

1.2.2 PC 와 ATS-1000DA 시리즈 연결

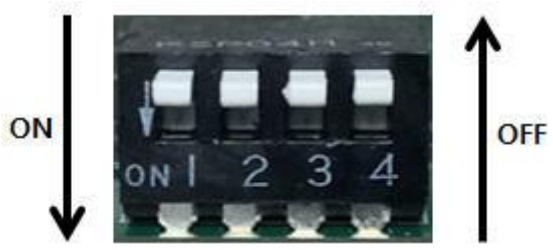
전원 입력이 완료 되신 후 ATS-1000DA 시리즈에 <그림 1-4>와 같이 USB 케이블을 연결함으로써, 개발하신 프로그램이 배포될 수 있습니다.

[참 고]	USB 케이블 설치 후 관련 프로그램 셋팅에 관련된 사항은 매뉴얼 2.2장을 참조해주시기 바랍니다.

<그림 1-4> USB 케이블과 ATS-1000DA 시리즈 연결 모습

1.3 모드 설정 방법

ATS-M1000DA 시리즈는 Mode 스위치의 설정에 따라 부팅방법, Debug, Auto Run 등의 여러 동작 모드를 지원하고 있습니다. <그림 1-5>는 Mode 스위치의 ON/OFF 방향에 대한 설명이고, <표 1-1>은 Mode 스위치의 상태에 따른 내용입니다.



<그림 1-5> Mode 스위치의 ON/OFF 방향

스위치 번호	ON 시 설정내용	비고
1	Nand Memory Boot Mode	일반적인 경우 항상 Off로 설정해야 함, On 시 부팅 안됨
2	USB Device Boot Mode	사용안함, 항상 Off 로 설정, On 시 부팅 안됨
3	COM3를 Debug 포트로 사용	일반적인 경우 항상 Off 로 설정, On으로 설정 시 COM3 사용불가
4	Auto Run 설정의 사용	Auto Run 사용 시 ON으로 설정 Auto Run = 부팅 시 실행할 프로그램을 등록하여 실행하는 유틸리티

<표 1-1> Mode 스위치의 상태에 따른 내용

[주 의]	Mode 스위치의 1, 2번을 On 하면 윈도우 부팅이 되지 않습니다. 또한 4번 스위치를 On하고 'Explorer.exe'를 Auto Run에 넣지 않으면 윈도우 기본 화면이 보이지 않습니다.

1.4 응용 프로그램 AutoRun 모드

AutoRun 모드는 ATS-M1000DA 시리즈 부팅 시 실행되는 모드를 의미합니다. 해당 설정은 ATS-M1000DA시리즈에서 지원하는 UtilMain 프로그램을 이용하여 설정 가능합니다. AutoRun은 Mode 스위치의 4번이 On 상태로 설정하고 윈도우 부팅 시 실행되는 프로그램을 말합니다. AutoRun 은 <표 1-2>와 같은 파일이름, 작성형식, 실행 조건으로 되어 있습니다.

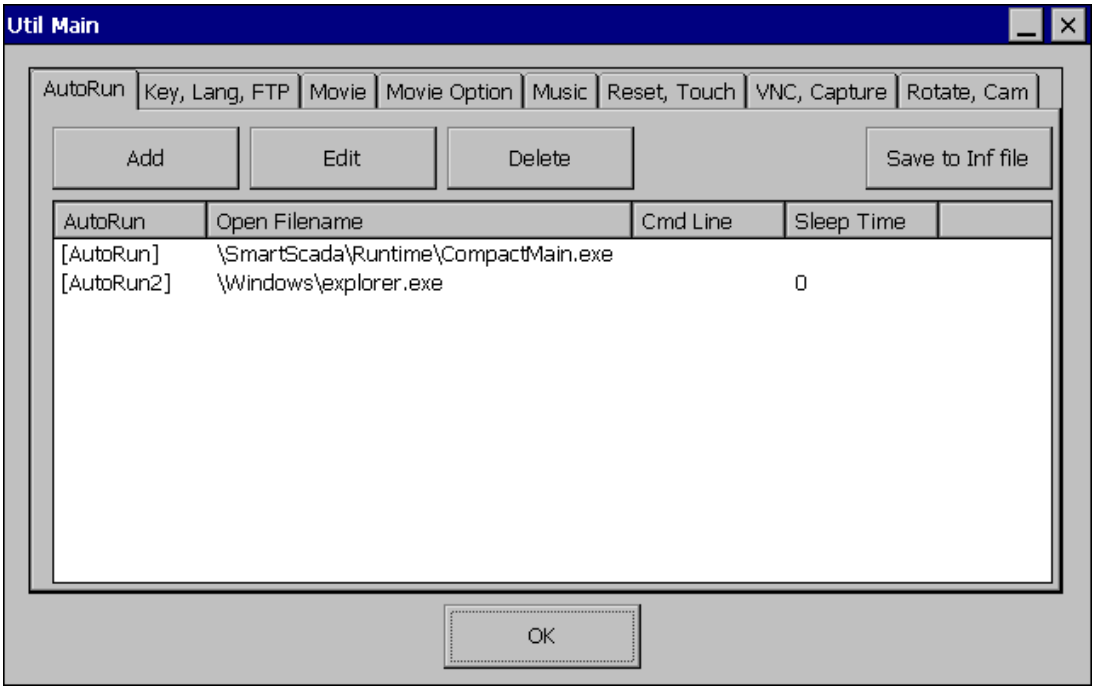
구분	내용
파일이름	₩AutoRun.inf
작성형식	Text 형식으로 작성
실행 조건	Mode 스위치의 4번이 On 일 때

<표 1-2> AutoRun 의 파일이름, 작성형식 및 조건에 대한 내용

[참 고]	AutoRun 은 Text 에디터로도 실행할 파일을 간단한 형식으로 입력할 수도 있지만 <그림 1-6>와 같은 UtilMain 프로그램으로 설정하는 것이 좋습니다.

ATS-M1000DA 실행 시킨 후 바탕 화면의

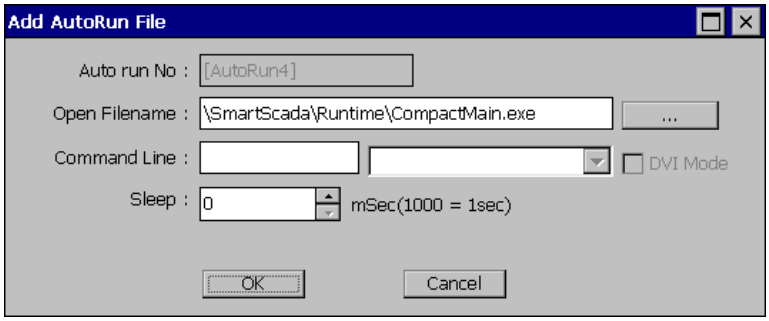
아이콘을 선택하시면 <그림 1-6>과 같은 UtilMain 프로그램의 AutoRun탭이 나타납니다.



<그림 1-6> UtilMain 프로그램 'AutoRun' 탭의 예

1.4.1 AutoRun 모드 응용 프로그램 추가, 편집

<그림 1-6>에서 Add 또는 Edit 버튼을 누르면 <그림 1-7>과 같은 추가, 편집 대화상자에서 AutoRun으로 실행할 파일을 설정할 수 있습니다.



<그림 1-7> AutoRun 추가, 편집 대화상자의 예

<그림 1-7>의 'Open Filename' 에디터에 실행할 파일의 폴더와 이름을 입력하거나 ... 아이콘을 선택하여 나타나는 파일선택 대화상자에서 실행할 파일을 선택할 수 있습니다. Command Line 영역은 'HdmiOut.exe' 과 같은 프로그램처럼 실행파일 뒤에 명령인자가 필요한 경우 설정하고 Sleep는 부팅 후 실행파일이 실행될 때 기다리는 시간을 입력합니다. <표 1-3>는 AutoRun 추가/편집 대화상자의 설정인자 및 내용입니다.

[주 의]	여러 개의 AutoRun 파일을 등록하고 앞쪽 실행파일에 Sleep 이 설정되어 있는 경우 뒤쪽의 실행파일도 앞쪽의 Sleep 만큼 기다린 후 실행되므로 주의를 요합니다.

설정인자	내용	비고
Open Filename	실행파일 폴더 및 이름	
Command Line	실행파일의 명령인자	HdmiOut.exe 등에 사용
Sleep	실행 대기시간	CeVncServer.exe 는 Sleep 등은 필요함

<표 1-3> AutoRun 추가/편집 대화상자의 설정인자 및 내용

<표 1-4>는 AutoRun 설정 시 주의해야 할 실행파일과 내용입니다.

실행파일 이름	내용
Explorer.exe	₩Windows₩Explorer.exe 를 AutoRun에 입력하지 않으면 윈도우 기본 화면이 생성되지 않습니다. 이 때는 실행파일을 종료하면 윈도우를 사용할 수 없습니다. 사용자 실행 프로그램에서 '₩Windows₩Explorer.exe'를 실행하여 윈도우 기본화면을 사용할 수 있습니다.
HdmiOut.exe	₩Tools₩HdmiOut.exe 실행 시 설정 해상도와 DVI 모드여부는 실행파일 뒤의 명령인자로 설정함.
CeVncServer.exe	Ethernet 주소 유형이 DHCP일 때 ₩Tools₩CeVncServer.exe 는 20000 mSec 정도의 Sleep 가 필요함

<표 1-4> AutoRun 설정 시 주의해야 할 실행파일과 내용

1.4.2 AutoRun 삭제

<그림 1-6>의 AutoRun이 설정된 항목을 선택 후,

버튼으로 등록된 AutoRun 파일을 삭제할 수 있습니다.

1.4.3 AutoRun 내용저장

<그림 1-6>의

버튼을 눌러서 설정한 AutoRun 파일을 레지스트리에 저장할 수 있습니다. 수정한 AutoRun 내용저장은 UtilMain 프로그램 종료를 위한

버튼을 눌러도 저장됩니다.

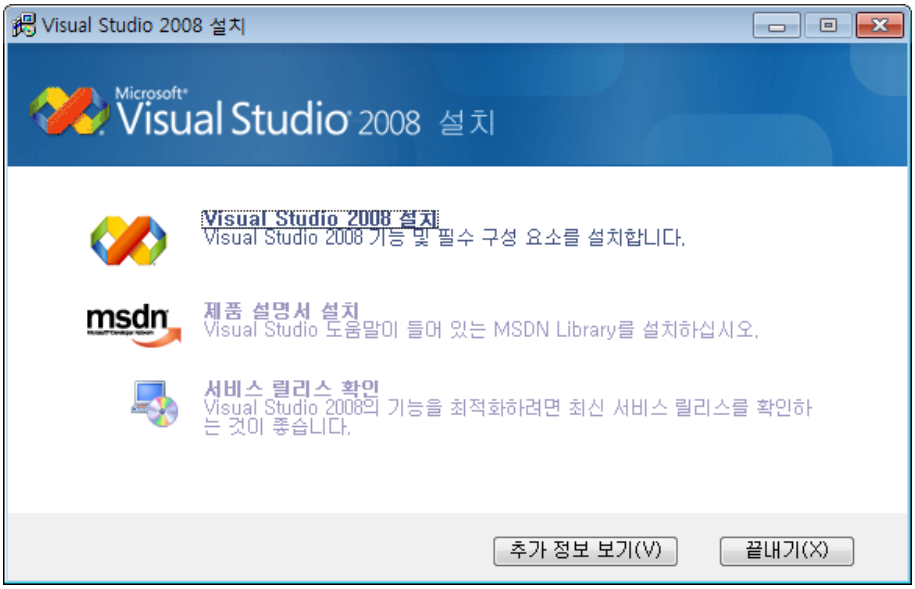
2장 개발 환경 설정

개발 환경 설정을 위해서는 Microsoft 사의 Visual Studio의 설치가 필수적입니다.
해당 프로그램의 제품 군은 MS Visual Studio Standard 2008, MS Visual Studio Professional 2008, MS Visual Studio Team Edition 2008 등으로 나뉘며, Standard 버전의 경우에는 장치용 응용 프로그램 개발을 할 수 없으므로, Professional 혹은 Team Edition 제품을 사용 하시기 바랍니다.

[참 고]	매뉴얼은 Professional 버전을 기준으로 작성되어져 있습니다.

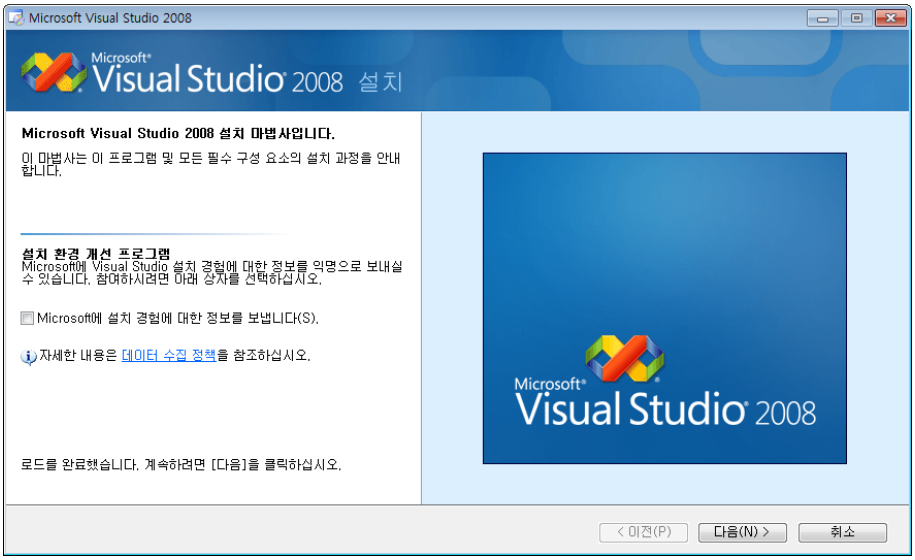
2.1 Microsoft Visual Studio 2008 실행

설치는 Visual Studio 2008 Professional 설치, 서비스 팩 1 설치, Visual Studio 2008 update for Windows Embedded Compact 7 설치, ATS-M1000 SDK의 설치(C++사용 시에 해당) 등, 총 4단계로 진행 됩니다.
첫 번째 단계, Visual Studio 2008 Professional 설치를 우선적으로 해줍니다. 설치를 실행 하면 아래 <그림 2-1>과 같은 화면이 나타납니다.

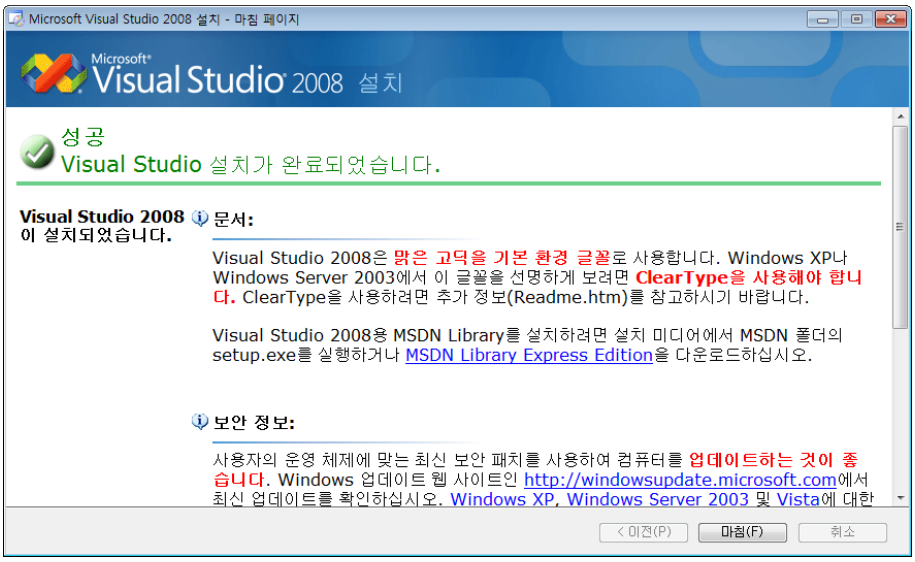


<그림 2-1> Visual Studio 2008 설치 시작 화면

<그림 2-1>에서 Visual Studio 2008 설치 항목을 선택하면, <그림 2-2>와 같은 화면이 나타납니다.
'다음' 버튼을 누를 때마다 나타나는 항목들을 설정해 주시면서 설치를 완료해 주시면 <그림 2-3>과 같은 설치 완료 화면이 나타납니다.



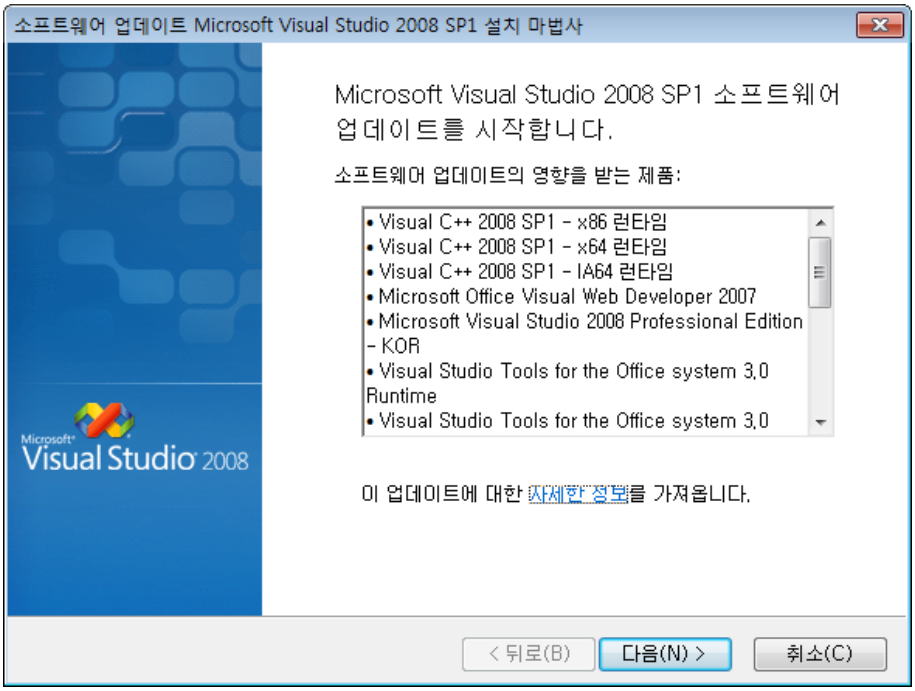
<그림 2-2> Visual Studio 2008 설치 진행 화면



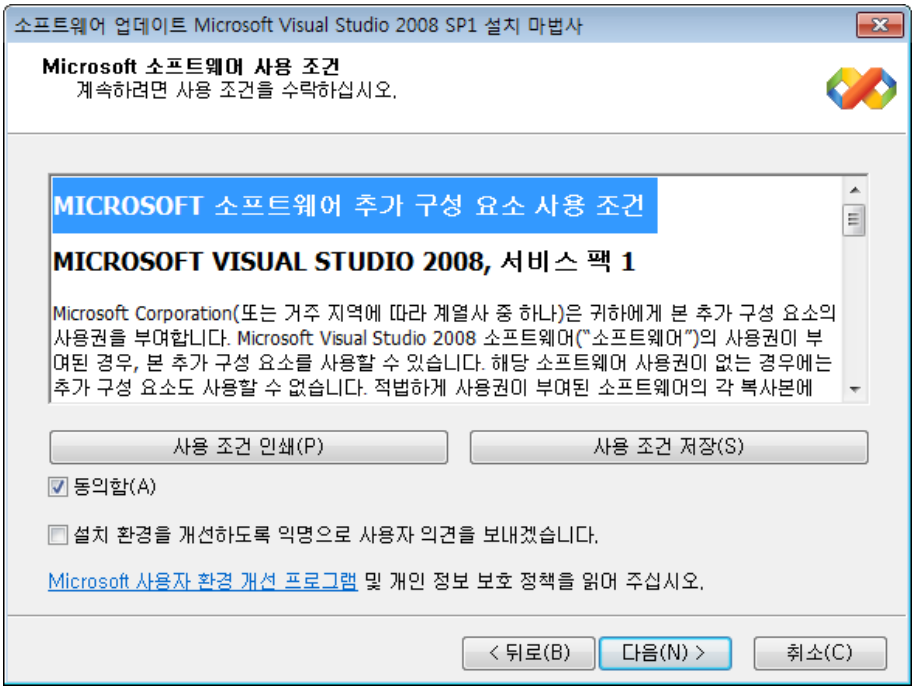
<그림 2-3> Visual Studio 2008 설치 완료 화면

두 번째 단계, Visual Studio SP1(서비스 팩 1)을 설치해 주시면 되겠습니다. 해당 프로그램 설치 역시, 첫 번째 단계와 마찬가지로 '다음' 버튼을 눌러 진행을 시켜 주시면 되겠습니다.

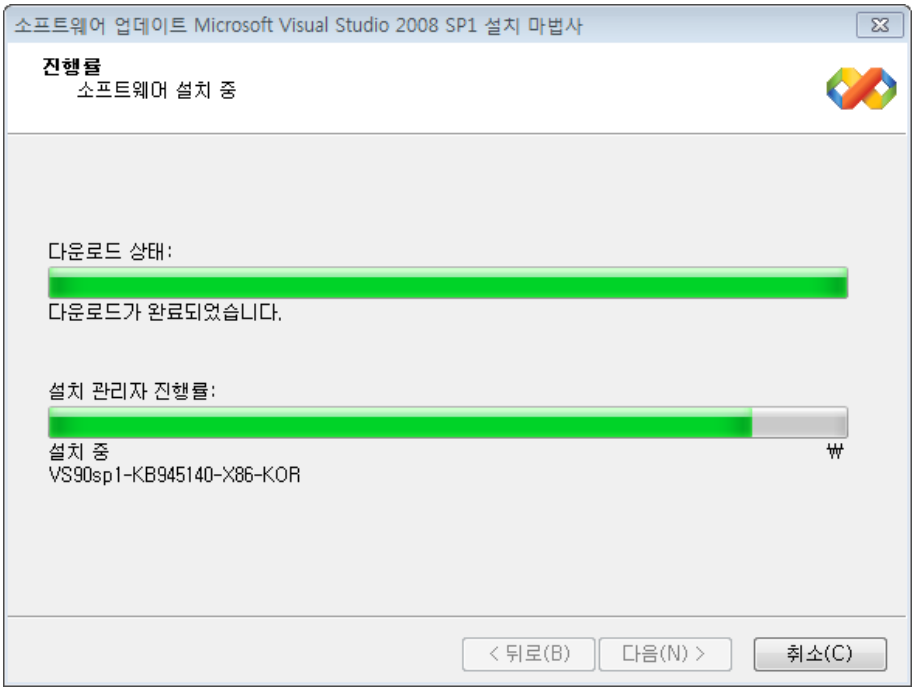
<그림 2-4>~<그림 2-6>은 Visual Studio SP1(서비스 팩 1) 설치 진행부터 완료까지의 화면입니다.



<그림 2-4> Visual Studio 2008 SP1(서비스 팩 1) 설치 시작 화면

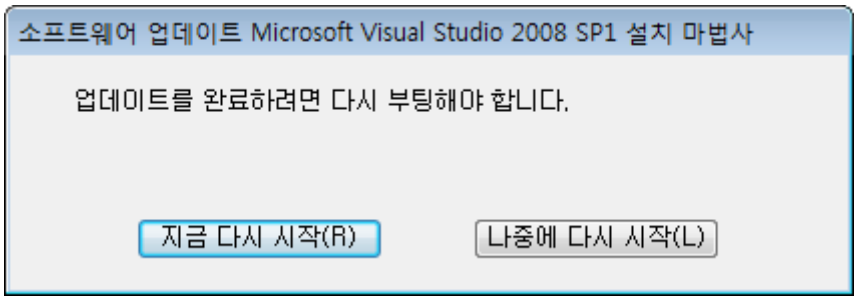


<그림 2-5> Visual Studio 2008 SP1(서비스 팩 1) 설치 진행 화면



<그림 2-6> Visual Studio 2008 SP1(서비스 팩 1) 설치 진행 화면

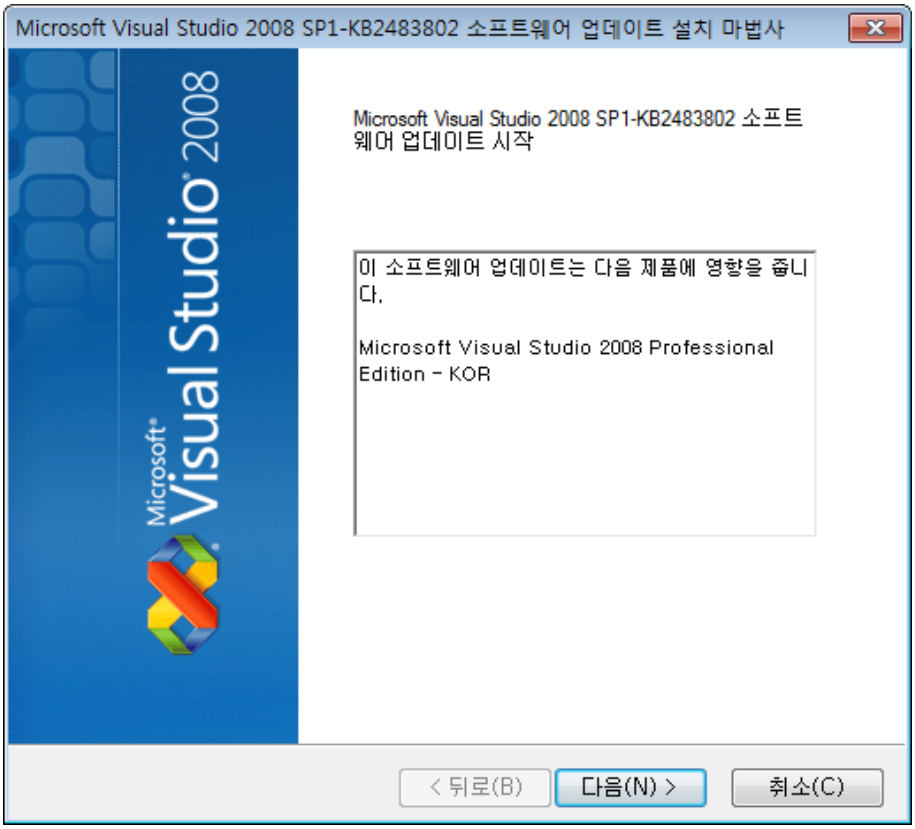
<그림 2-6>과 같은 설치 진행을 완료 하면, <그림 2-7>과 같은 설치 마법사 화면이 나타나게 되고, PC를 재 부팅 시켜주시면 됩니다.



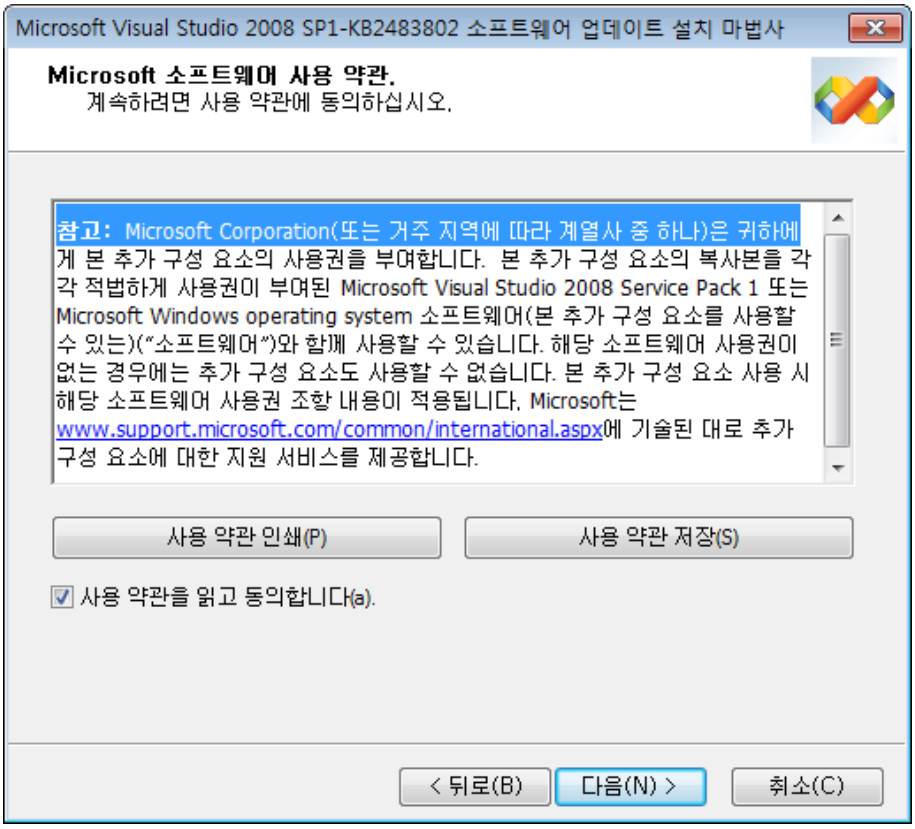
<그림 2-7> Visual Studio 2008 SP1 설치 마법사 화면

세 번째 단계, Visual Studio 2008 update for Windows Embedded Compact 7 설치를 진행 시켜주시면 됩니다. 해당 프로그램 설치 역시, 첫 번째, 두 번째 단계와 마찬가지로 '다음' 버튼을 눌러 진행을 시켜 주시면 되겠습니다.

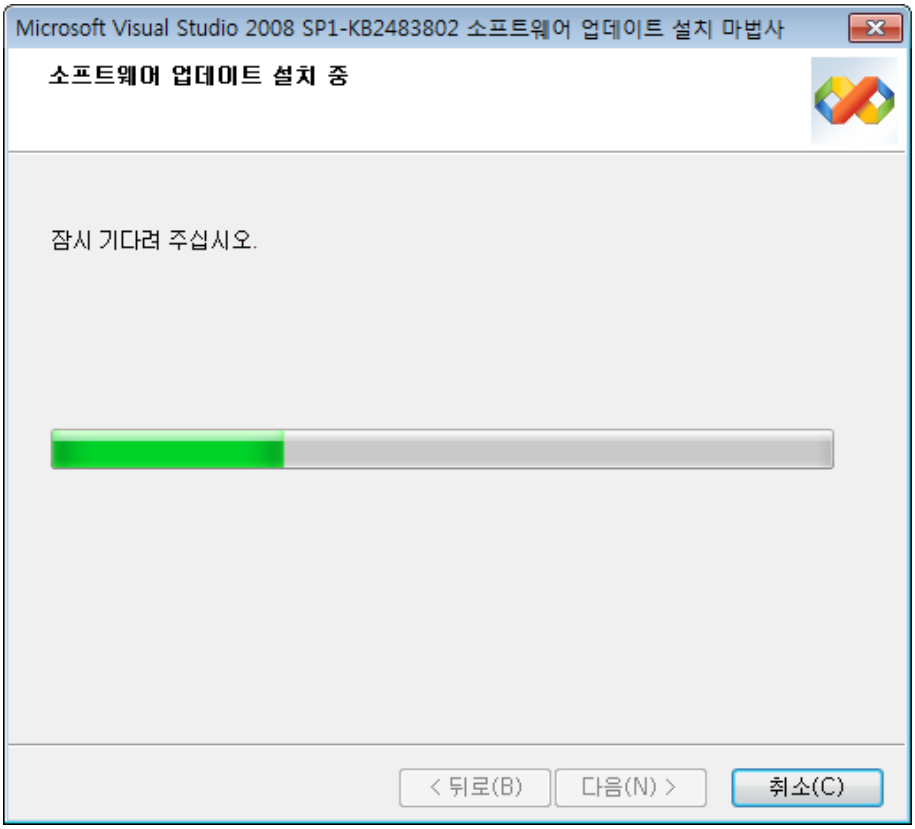
<그림 2-8>~<그림 2-11>은Visual Studio 2008 update for Windows Embedded Compact 7의 설치 진행부터 완료까지의 화면입니다.



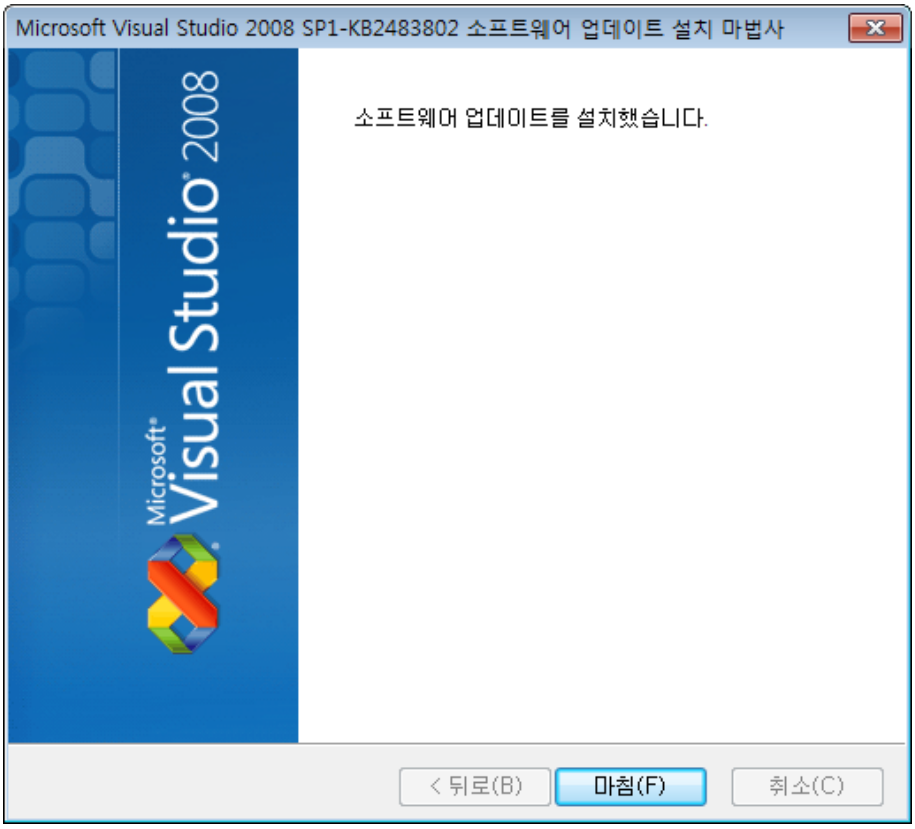
<그림 2-8> Visual Studio 2008 update for Windows Embedded Compact 7 설치 시작 화면



<그림 2-9> Visual Studio 2008 update for Windows Embedded Compact 7 설치 진행 화면



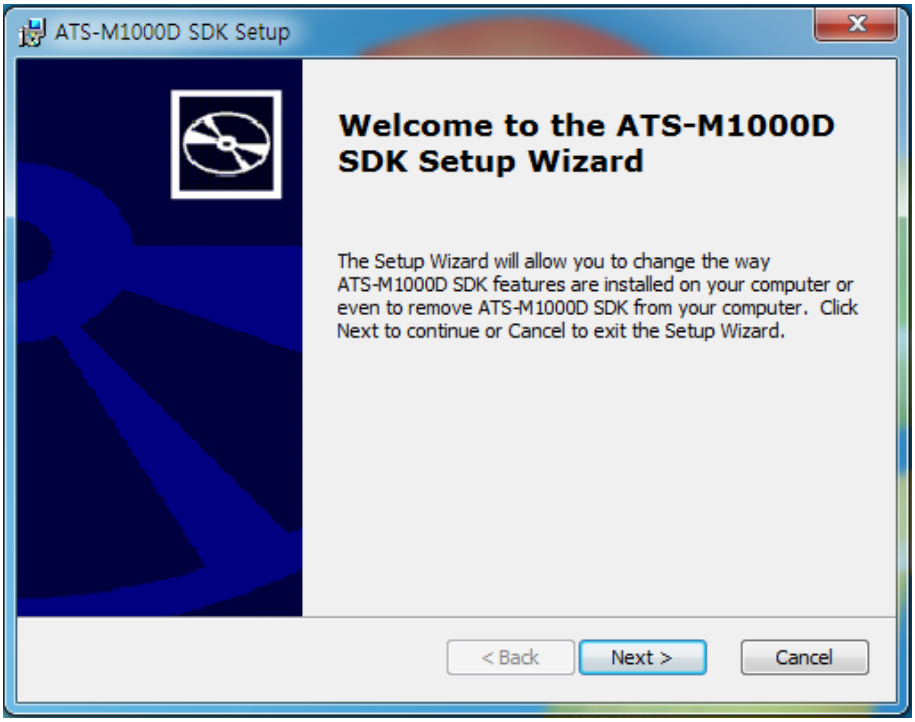
<그림 2-10> Visual Studio 2008 update for Windows Embedded Compact 7 설치 진행 화면



<그림 2-11> Visual Studio 2008 update for Windows Embedded Compact 7 설치 완료 화면

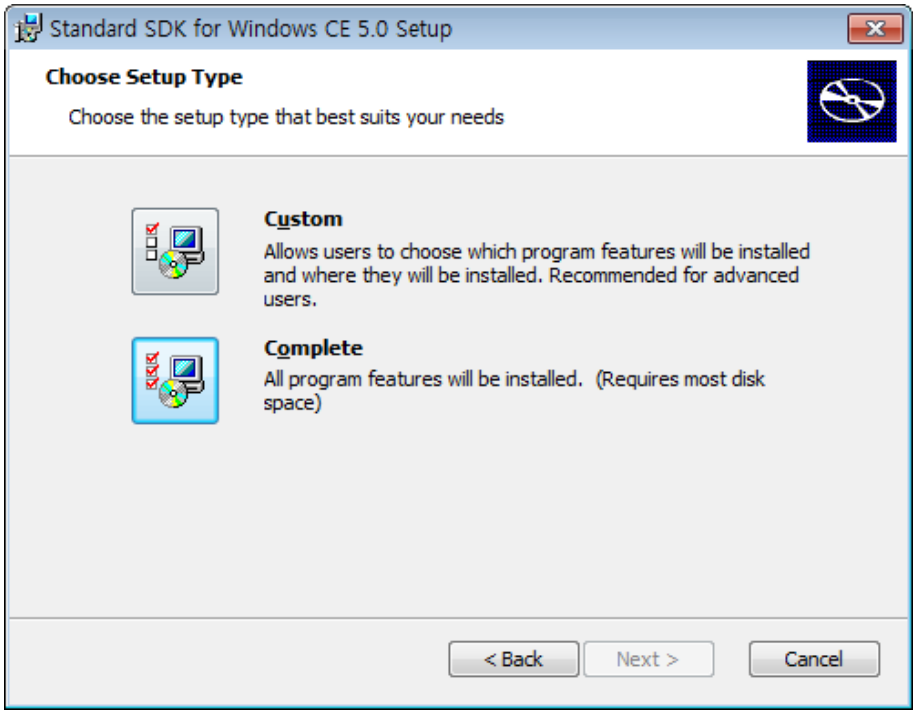
네 번째 단계는 ATS-M1000 SDK 설치로써, 이는 C++을 이용하여 장치 응용 프로그램 개발 시에 설치를 하시면 되며, C#과 Basic을 이용하여 프로그램 개발을 하실 경우에는 설치를 하지 않으셔도 무방합니다.

<그림 2-12>~<그림 2-14>는 ATS-M1000D SDK 설치 과정에 대한 화면입니다.



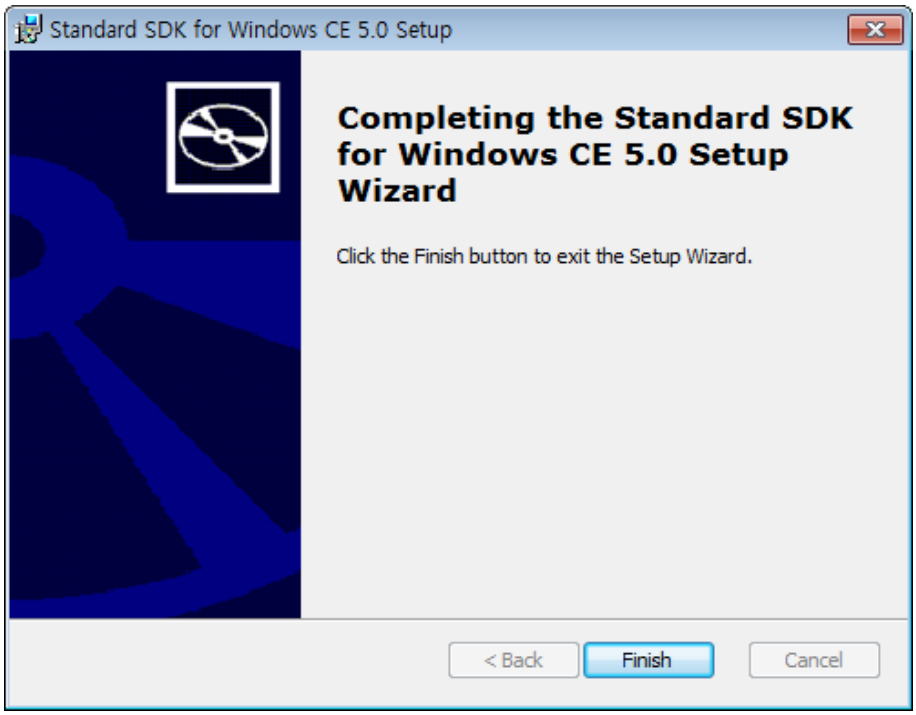
<그림 2-12> ATS-M1000D SDK 설치 시작 화면

<그림 2-13>은 설치 종류 선택 대화 상자 입니다. Custom을 선택할 경우 설치 항목을 사용자가 선택적으로 설치가 가능하며, Complete를 선택할 경우 프로그램 전체를 설치하게 됩니다. (본 매뉴얼에서는 Complete설치 선택)



<그림 2-13> ATS-M1000D SDK 설치 종류 선택 대화 상자

<그림 2-13>에서 설치 종류를 선택 하신 후 'Next'버튼을 눌러, 설치를 완료 시키시면 됩니다. <그림 2-14>는 설치 완료 대화 상자입니다.



<그림 2-14> ATS-M1000D SDK 설치 완료 화면

2.2 ActiveSync 설치

Smart 제품과 사용자 PC와의 호환을 위해선 동기화 프로그램의 설치가 반드시 이루어져야 합니다. 일반적으로 ATS-M1000DA 시리즈와 사용자 컴퓨터를 연결 하면 ActiveSync 프로그램이 자동으로 실행됩니다.

2.2.1 Active Sync/Windows Mobile Device Center 설치

동기화 프로그램은 OS 마다 조금씩 명칭이 다르며, 그 내용은 아래 <표 2-1>과 같습니다.

	Windows XP	Windows Vista / Windows 7, 8 이상
명칭	ActiveSync	Windows Mobile Device Center
다운 경로	마이크로소프트 홈페이지에서 다운 가능 (http://www.microsoft.com/en-us/search/DownloadResults.aspx?q=activesync) 혹은 Autobase SCADA OEM DVD에 포함된 프로그램으로 설치 가능	OS에 탑재되어 있음, 설치되어 있지 않은 경우 마이크로 소프트 홈페이지에서 다운 가능
주 의 : ActiveSync/Windows Mobile Device Center는 Window Media Player가 설치 되어 있어야 합니다.		

<표 2-1> 사용자 컴퓨터의 OS 별 연력 프로그램 명칭 및 경로

2.2.2 USB 케이블 연결

ATS-M1000DA 시리즈와 사용자 컴퓨터와의 연결을 위해서는 <그림 2-15>와 같은 USB Device 용 케이블을 각각 연결하여야 합니다.

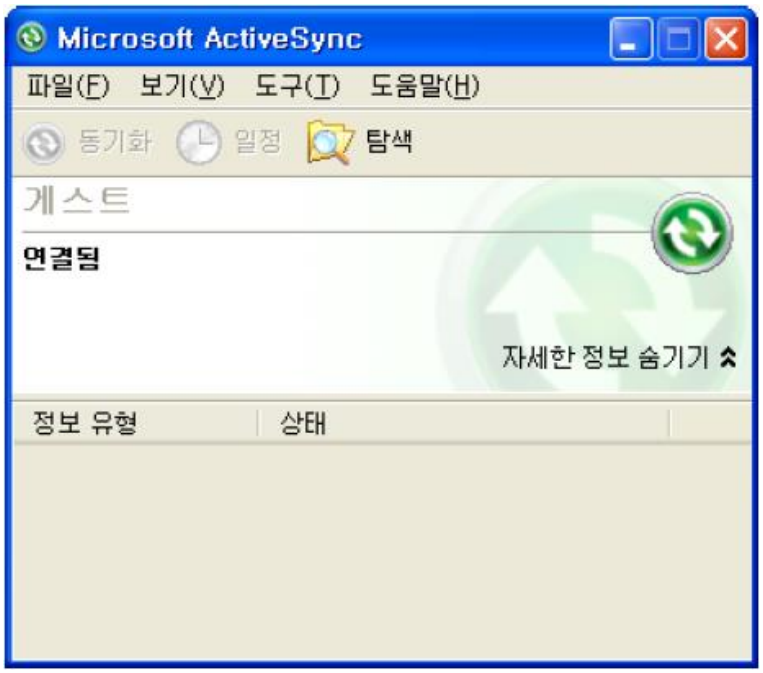
<그림 2-16>과 같은 ATS-M1000DA 시리즈 USB Device 포트에 <그림 2-15>의 B Type 컨넥터 ( )를 연결하고 사용자 컴퓨터에는 A Type 컨넥터( ) 를 연결합니다.

<그림 2-15> ATS-M1000DA 시리즈와 연결을 위한 USB 케이블의 예

<그림 2-16> USB Device 포트의 모습

2.2.3 Windows XP ActiveSync 설치

사용자 컴퓨터의 OS 가 Windows XP 일 때 ATS-M1000DA 시리즈와 연결을 위해 사용하는 프로그램은 ActiveSync 입니다. <그림 2-15>와 같은 Device 용 USB 케이블을 ATS-M1000DA 시리즈와 사용자 컴퓨터에 각각 연결하면 <그림 2-17>과 같은 ActiveSync 프로그램이 자동으로 실행됩니다.



<그림 2-17> ActiveSync 4.5 버전의 실행 예

Windows XP 환경에서 <그림 2-17>와 같은 ActiveSync 자동으로 실행되지 않을 경우에는 <표 2-2>와 같은 사항들을 체크하여 사용합니다.

체크 사항	확인 내용
케이블 연결	USB 케이블 연결을 다시 확인
케이블 연결위치	ATS-M1000DA 시리즈에는 과 같은 B Type Connector, 사용자 컴퓨터에는 과 같은 A Type Connector 를 연결
인식불량	ATS-M1000DA 시리즈 재 시작(Power Reset) 또는 사용자 컴퓨터 재 시작
Windows Mobile Device	MS 사 홈페이지에서 다운로드하여 설치

Center 미설치	
기타 연결불량	ATS-M1000DA 시리즈 서비스센터로 연락해 주세요

<표 2-2> ATS-M1000DA 시리즈와 연결을 위한 프로그램의 연결 불량 체크사항

2.2.4 Windows Vista/Windows7 ActiveSync(Windows Mobile Device Center) 설치

사용자 컴퓨터의 OS 가 Windows Vista, Windows 7, Windows 8 등일 때 ATS-M1000DA 시리즈와 연결을 위해 사용하는 프로그램은 Windows Mobile Device Center 입니다. <그림 2-15>와 같은 Device 용 USB 케이블을 ATS-M1000DA 시리즈와 사용자 컴퓨터에 각각 연결하면 <그림 2-18>과 같은 Windows Mobile Device Center 프로그램이 자동으로 실행됩니다.

<그림 2-18> Windows Mobile Device Center 프로그램의 예

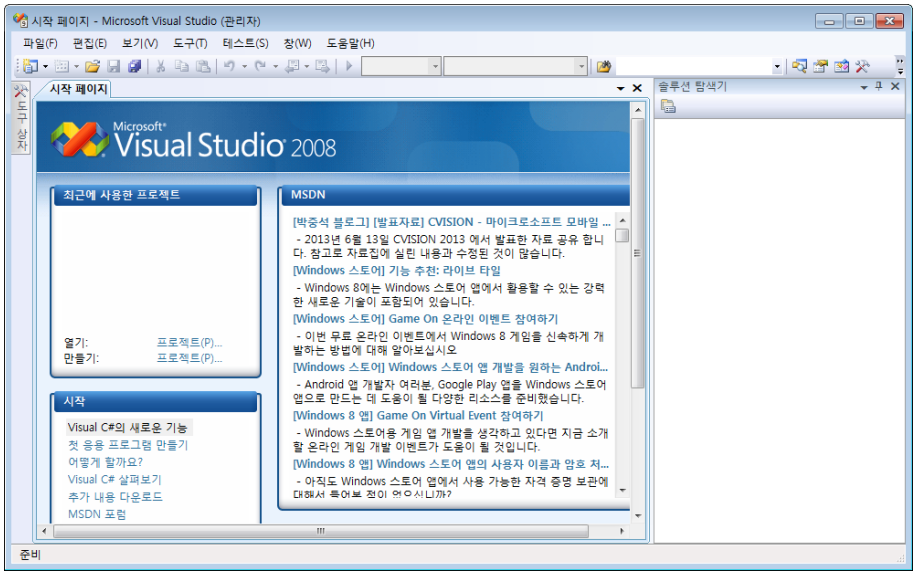
만약 <그림 2-18>과 같은 Windows Mobile Device Center 가 실행되지 않을 경우에는 <표 2-2>와 같은 사항들을 체크하여 사용합니다.

3장 Visual C# Windows CE 프로젝트 작업

ATS-M1000DA 시리즈에 사용할 장치 응용 프로그램을 개발할 경우 Visual C# 프로젝트를 Windows용이 아닌 스마트 장치 프로젝트로 생성 시켜 주어야 하며, 개발 시 반드시 PC와 ATS-M1000DA 시리즈가 ActiveSync로 연결된 상태여야 합니다.

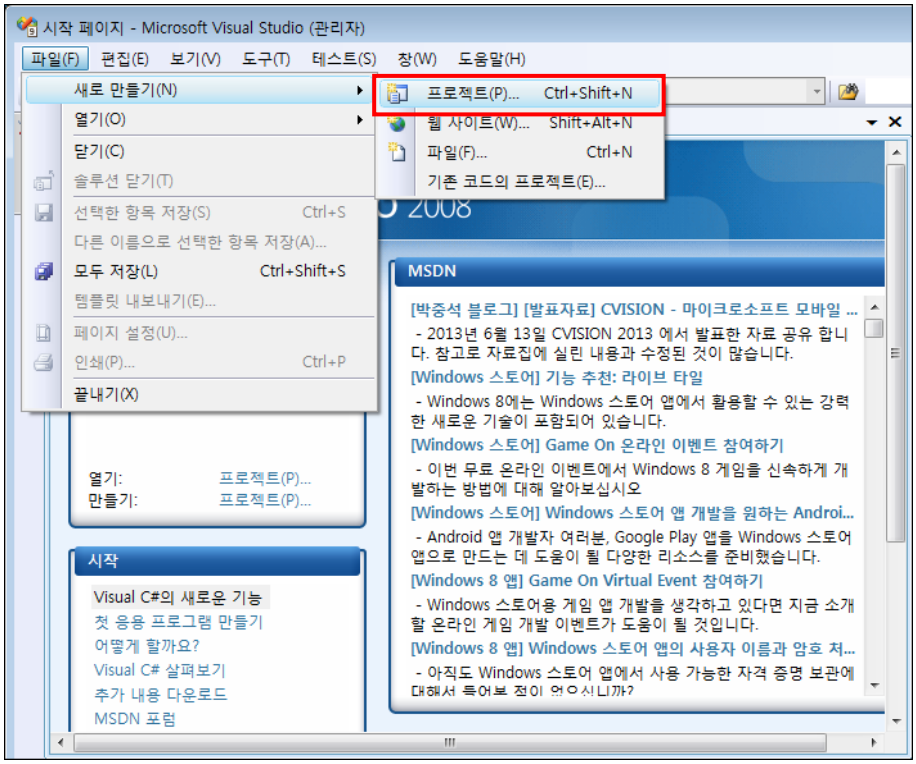
3.1 Visual C# 프로젝트 생성 과정

CE 프로젝트 생성을 위해 아이콘을 눌러, Visual Studio 2008을 실행 시킵니다. <그림 3-1>는 Visual Studio 2008 실행 시 나타나는 첫 화면입니다.



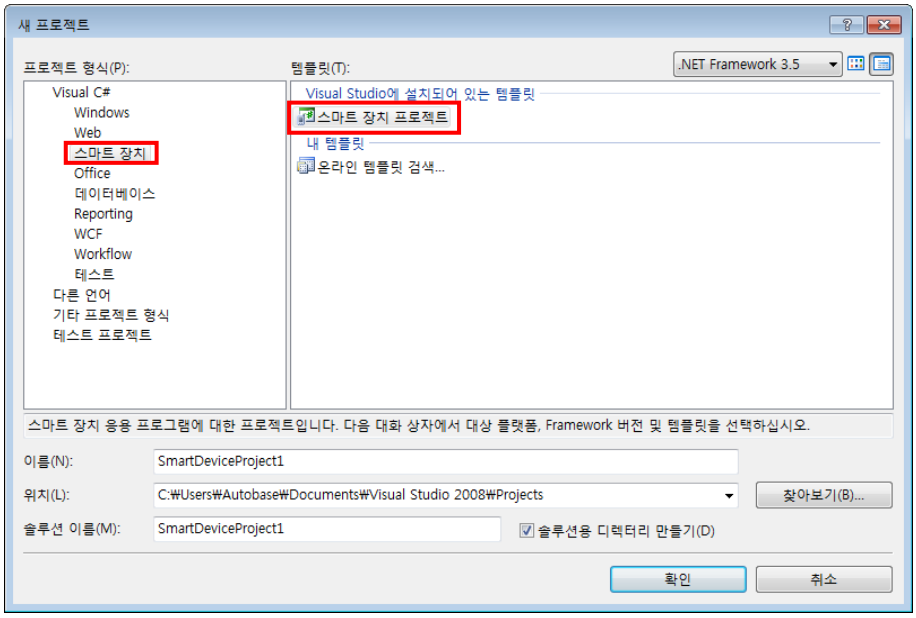
<그림 3-1> Visual Studio 2008 실행 시 첫 화면

<그림 3-2>과 같이 <그림 3-1>항목에서 '파일₩새로 만들기₩프로젝트' 선택하고, <그림 3-3>~<그림 3-5>까지의 새 프로젝트를 생성 과정을 따라 주시면 됩니다.



<그림 3-2> Visual Studio 2008 새 프로젝트 생성 메뉴

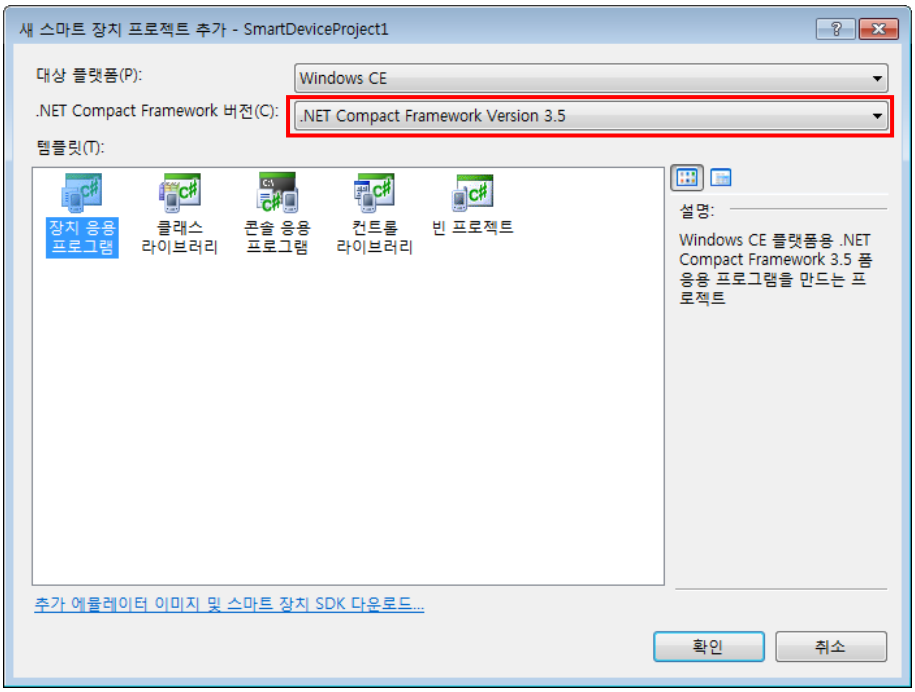
<그림 3-2>과 같이 선택을 하면, <그림 3-3>과 같은 새 프로젝트 대화상자가 나타나며, 해당 대화상자의 프로젝트 형식 항목에서 Visual C# - 스마트 장치, 템플릿 항목에서 스마트 장치 프로젝트를 선택하여 주고 '확인'버튼을 눌러 주시면 됩니다.



<그림 3-3> Visual C# 새 프로젝트 대화상자

<그림 3-3>에서 설정을 마친 후 '확인' 버튼을 누르게 되면, <그림 3-4>과 같은 새 스마트 장치

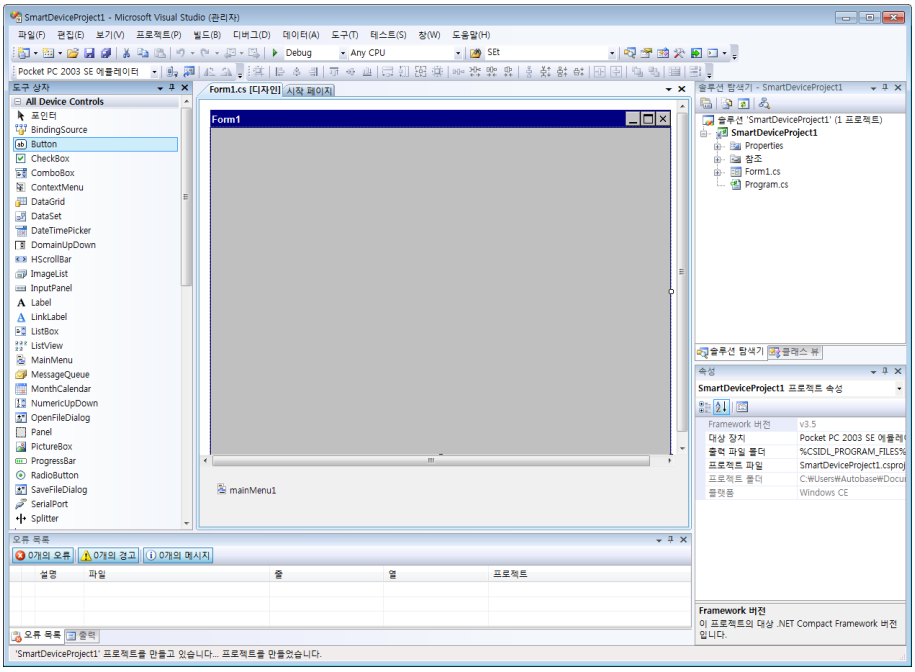
프로젝트 추가 대화 상자가 열리게 됩니다.



<그림 3-4> Visual C# 새 스마트 장치 프로젝트 추가 대화 상자

대상 플랫폼은 <그림 3-4>과 같이 Windows CE로 선택하여 주며, 템플릿은 응용 프로그램을 만들고자 하실 경우 '장치 응용 프로그램' 항목을, dll 형식의 파일을 만들고자 하실 경우엔 '클래스 라이브러리' 항목을 선택하여 프로젝트를 생성 시켜주시면 됩니다.

<그림 3-4>의 과정을 마치신 후 '확인' 버튼을 누르실 경우, <그림 3-5>와 같이 새 프로젝트 화면이 나타나게 됩니다.



<그림 3-5> Visual C# 새 프로젝트 화면

3.2 Visual Studio 화면(UI) 구성

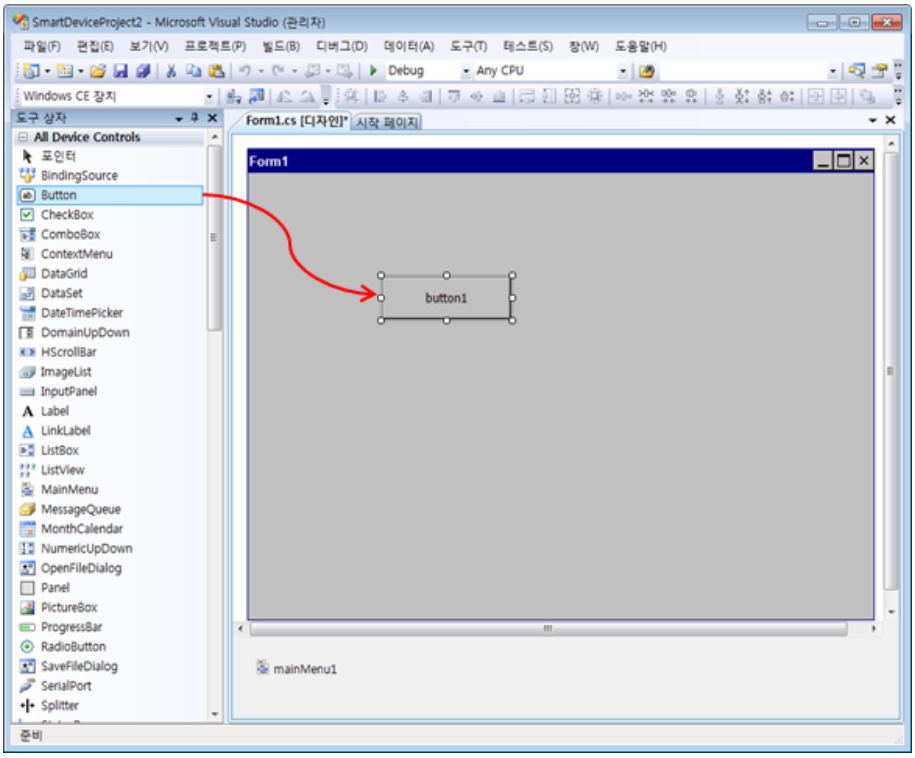
ATS-M1000DA 시리즈의 응용 프로그램 개발 시 Visual Studio의 개발 환경 화면 구성 및 내용은 <표 3-1>과 같습니다.

화면 구성	내용
디자인 화면	도구상자에서 디자인 화면으로 활동을 끌어 워크플로를 만드는데 사용 하는 화면.
도구 상자	그래픽을 디자인 화면으로 드래그 앤 드롭하여 디자인으로 표현.
속성	디자이너 화면에서 작업 중인 컴포넌트의 속성을 설정 시 사용.
솔루션 탐색기	워크플로 프로젝트를 계층적으로 볼 수 있도록 트리 형식으로 표시하며 프로젝트 관련 기능에 액세스 시 사용.
디버깅 창	스레드, 호출, 스택, 중단점, 로컬 도구 및 조사식 도구 창과 같은 표준 Visual Studio 디버깅 창.

<표 3-1> Visual Studio 의 개발 환경 화면 구성 및 내용

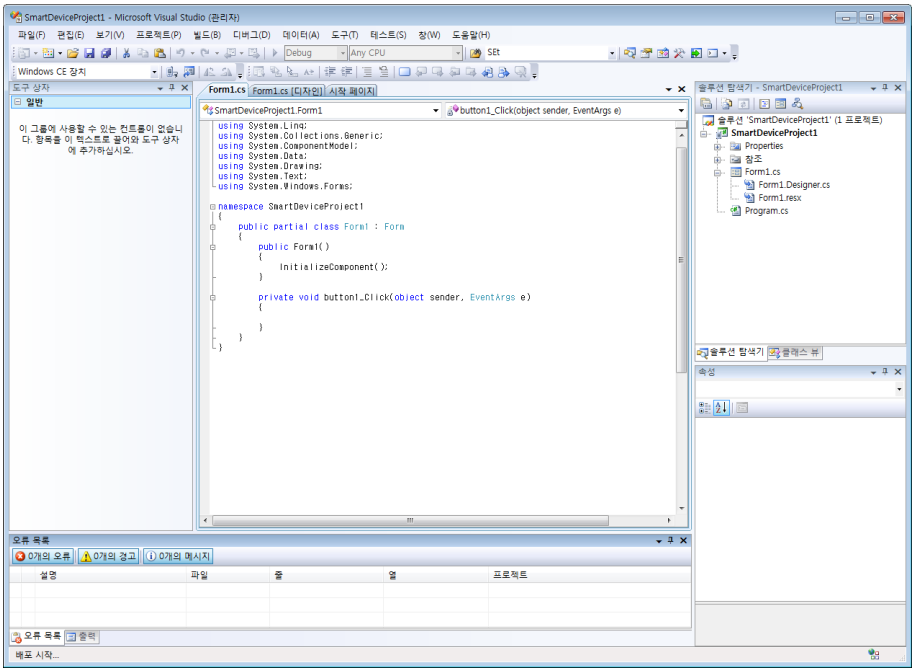
3.3 Visual Studio C# 장치 프로젝트 만들기

<그림 3-5>와 같이 새 프로젝트가 생성 되면, <그림 3-6>처럼, 도구 상자의 Button을 선택하여 Form1에 드래그 앤 드롭하여 줍니다.

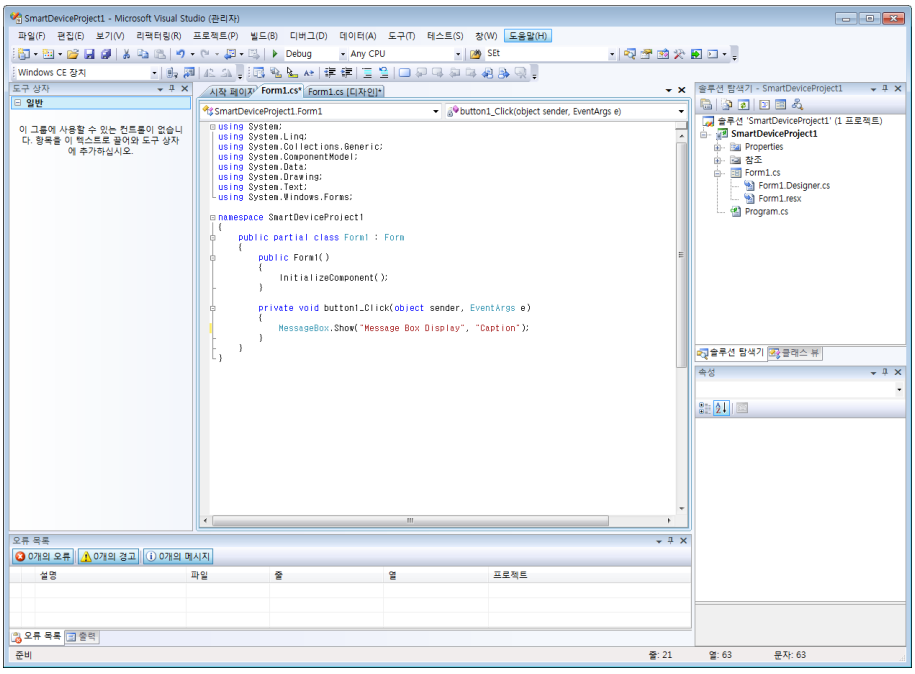


<그림 3-6> Visual C# 도구 상자의 Button 을 Form1 에 삽입한 화면

삽입된 Button을 마우스로 더블클릭 하여 주면, <그림 3-7>과 같은 코드 작성 화면이 나타나게 되며, <그림 3-8>과 같이 메시지 박스 표시 코드를 작성해주시면 됩니다.

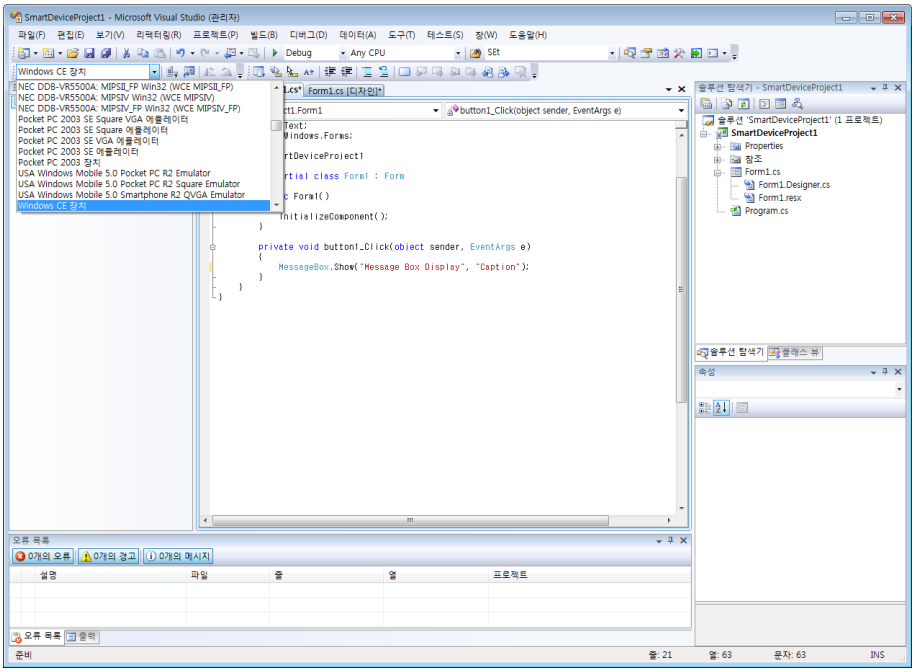


<그림 3-7> Visual C# 코드 작성 화면의 예

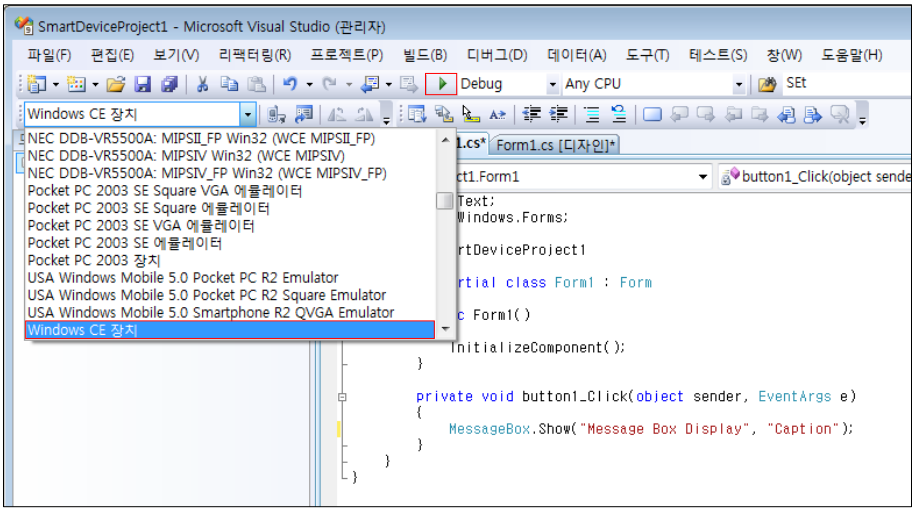


<그림 3-8> Visual C# 메시지 박스 표시 코드 작성 예

코드 작성이 완료 되면, <그림 3-9>과 같이 콤보 박스에서 배포할 장치 종류를 선택 하여 주신 후, <그림 3-10>와 같이 버튼(F5)을 눌러, Debug를 실행 시켜주시면 됩니다.

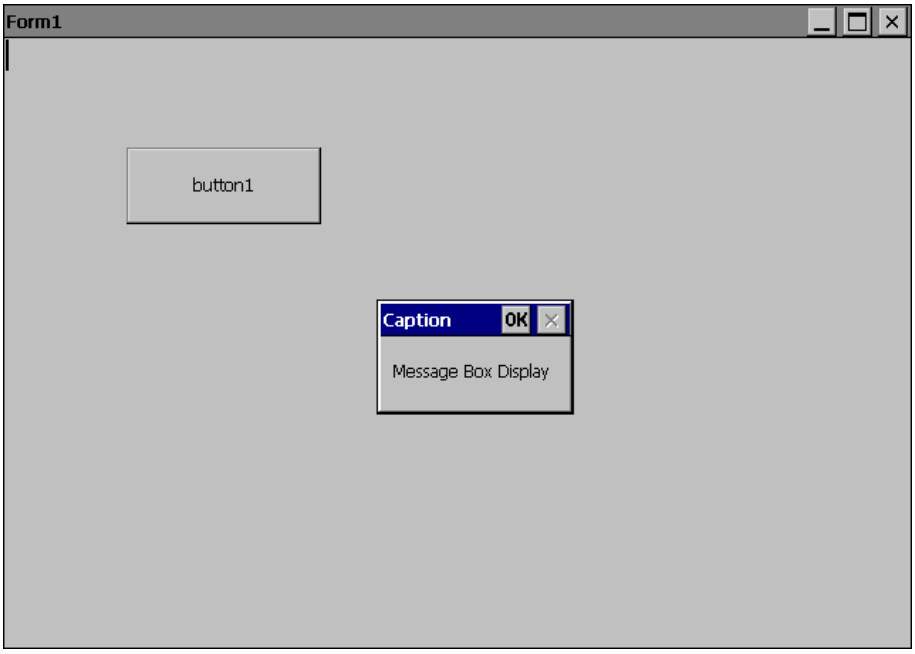


<그림 3-9> Visual C# 장치 종류 선택 콤보 박스의 예



<그림 3-10> Visual C# Debug 실행 버튼 화면

디버깅이 누르면, ATS-M1000DA 시리즈로 배포가 진행되고, 개발하신 프로그램이 화면에 실행되어 나타나게 됩니다. <그림 3-11>은 위에서 작성하신 프로그램의 버튼을 누른 예입니다.



<그림 3-11> 배포한 프로그램의 버튼을 클릭한 예

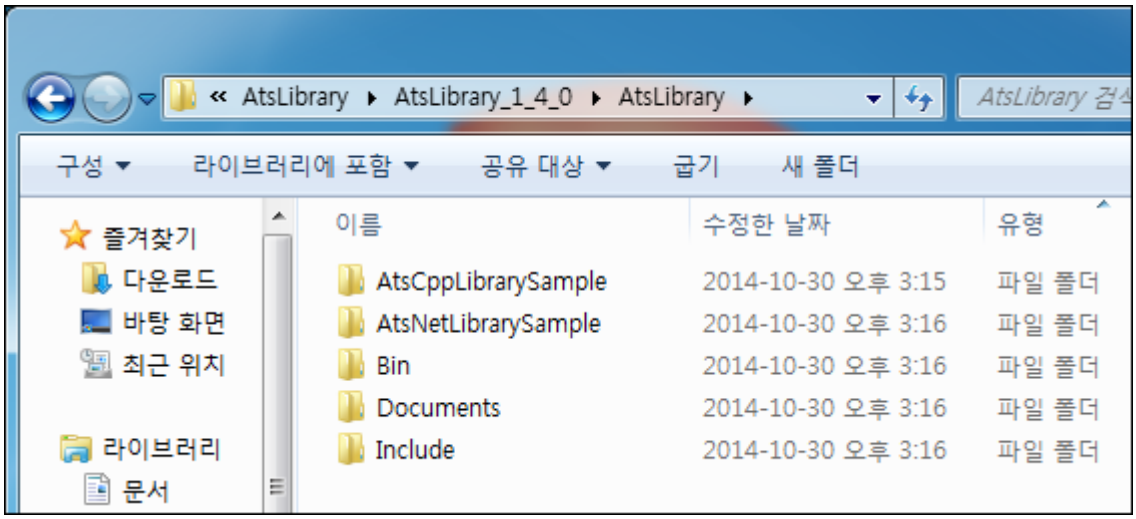
<그림 3-11>과 같이 테스트가 끝난 실행 파일은 Compact 장치에 복사해서 사용하시면 됩니다.

4장 ATS 시리즈 라이브러리 파일

Ats 시리즈의 라이브러리 l파일은 ATS 시리즈의 하드웨어 제어, Autobase SCADA CE 데이터 공유 및 기타 .NET용 일반 라이브러리 함수들로 구성되어 있습니다.

4.1 Ats 라이브러리 파일 설치

Ats 라이브러리 파일은 오토베이스 홈페이지의 다운로드₩Touch Smart 자료 항목에서 다운 받으실 수 있습니다. 다운로드 하신 파일을 실행 시키시면 <그림 4-1>과 같이 C:₩AutoBase₩AtsLibrary₩(버전정보) 폴더 하위에 AtsLibrarySample, Bin, Documents 세 개의 폴더가 생성되어 있는 것을 확인하실 수 있습니다.

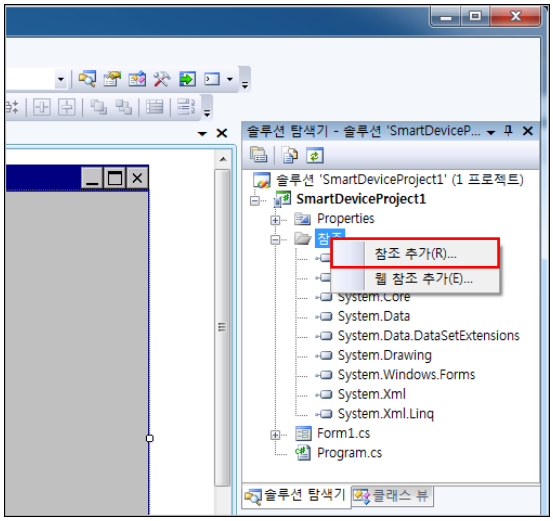


<그림 4-1> AtsLibrary 파일이 설치 시 경로 및 생성 폴더

AtsLibrarySample 폴더 내부에는 Ats 라이브러리를 이용하여 만든 예제 파일이 존재 하고, Bin 폴더에는 실제 라이브러리 내의 함수에 대한 정의가 저장되어 있는 dll 파일들이 존재합니다. Documents 파일은 라이브러리에서 지원하는 함수에 대한 설명 및 사용 예제 파일 포함되어 있습니다.

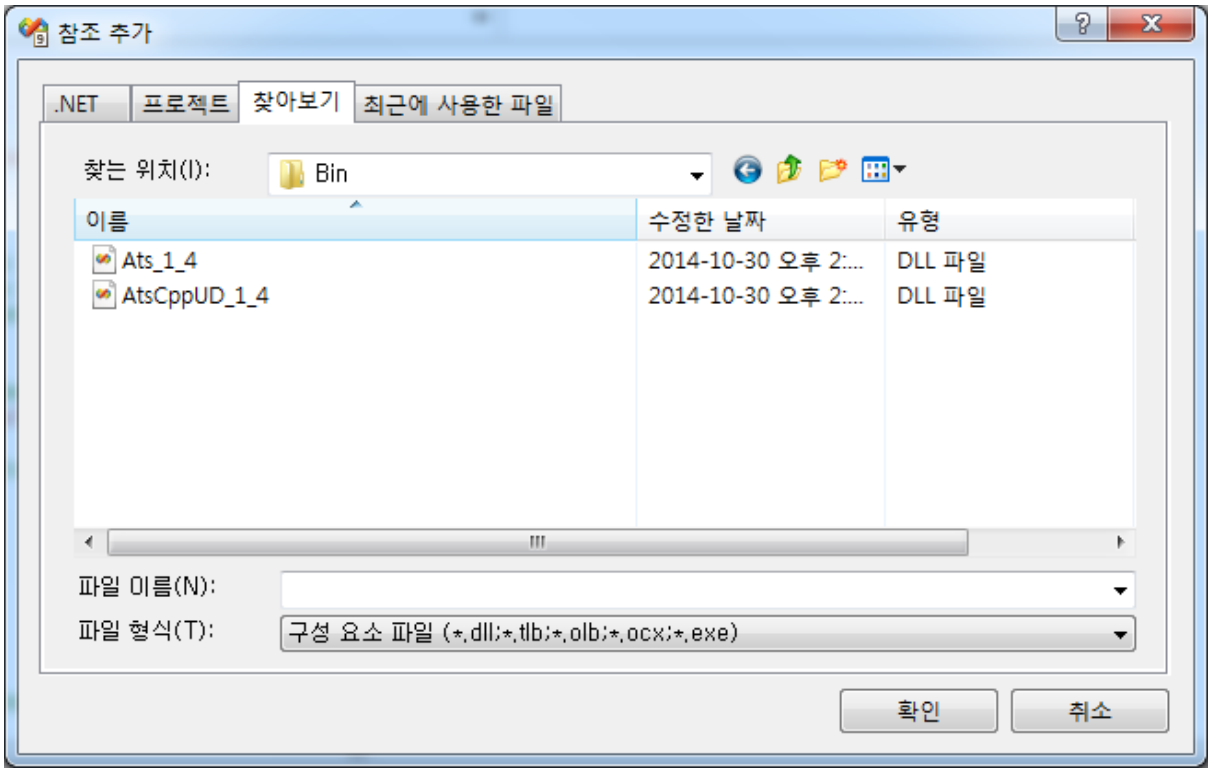
4.2 Ats 라이브러리 파일 등록

제공되는 dll 파일을 참조하기 위해선 해당 파일을 프로젝트에 추가하는 과정이 필요 합니다. <그림 4-2>는 새로 만든 프로젝트의 솔루션 탐색기 영역에서 Ats 라이브러리 파일을 참조 추가하는 과정을 나타내고 있습니다.



<그림 4-2> 솔루션 탐색기에서 Ats 라이브러리 파일 참조 추가

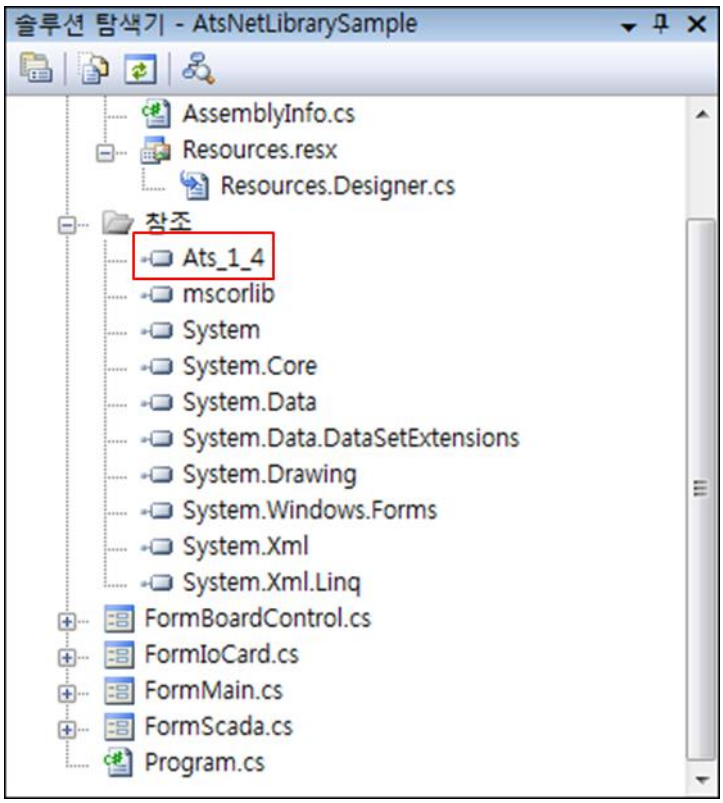
<그림 4-2>와 같이 솔루션 탐색기 트리에서 마우스 오른쪽 버튼을 누르면 참조할 dll 파일을 추가할 수 있는 메뉴가 나타나며, 해당 메뉴에서 참조 추가 메뉴를 눌러주시면 <그림 4-3>과 같은 참조 추가 대화상자가 나타나게 됩니다.



<그림 4-3> 참조 추가 대화 상자의 예

해당 대화상자의 찾아 보기 탭에서 <그림 4-1>과 같이 AtsLibrary 파일이 설치된 경로로 찾아가 Ats_x_x 파일을 선택하신 후 확인 버튼을 누르면, Ats 라이브러리 함수를 포함하고 있는 dll 파일 등록이 완료 됩니다.

Ats_x_x.dll 파일을 참조 추가를 완료하면, <그림 4-4>와 같이 dll 파일이 솔루션 탐색기에 추가 되는 것을 확인할 수 있으며, Ats_x_x.dll 파일에 내장된 함수들을 이용하실 수 있습니다.

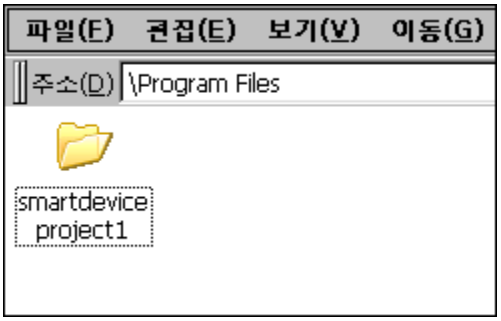


<그림 4-4> Ats_x_x.dll파일의 등록 완료 모습

4.2.1 AtsCppUD 파일 이용 방법

AtsCppUD 파일은 Extension I/O 보드를 제어하기 위한 기능들이 모여 있습니다.

AtsCppUD 파일은 참조 추가하여 사용하는 것이 아니라, Ats 라이브러리 파일이 포함되어 있는 프로젝트 폴더로 복사/붙여 넣기를 해주셔야 합니다. 우선적으로, 작성한 프로젝트가 Touch Smart 장비 내부에 배포가 되어 있어야 합니다. <그림 4-5>은 Touch Smart 제품 내로 작성한 프로젝트 폴더가 배포된 예 입니다.



<그림 4-5> Touch Smart 제품 내로 작성한 프로젝트 폴더가 배포된 예

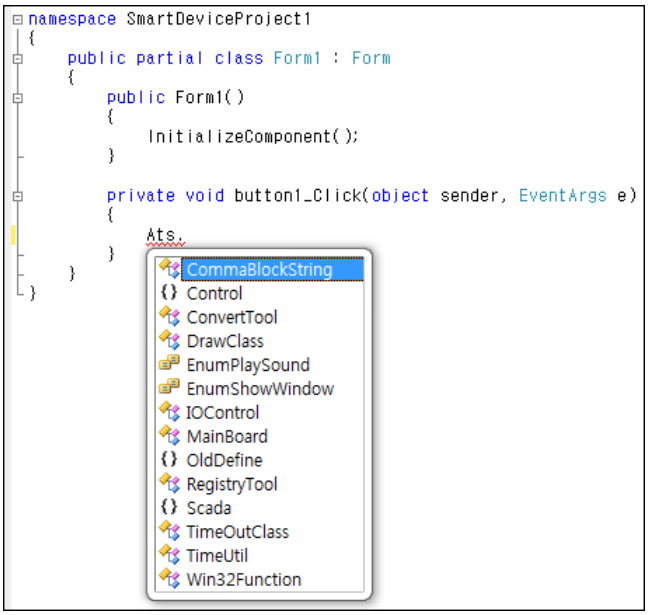
<그림 4-5>과 같이 AtsCppUD 파일을 복사/붙여 넣기를 해주기 전에, 우선적으로 작성한 프로젝트를 디버깅 하여 Touch Smart 제품으로 프로젝트 배포를 해주어야 합니다. 프로젝트 배포가 완료되면, Touch Smart 장비 내부의 'Program Files₩프로젝트 명'의 경로로 프로젝트가 생성 됩니다. 이렇게 배포된 프로젝트 폴더 내부는 <그림 4-6>과 같이 구성되어 있으며, 해당 프로젝트 내부로 AtsCppUD 파일을 복사/붙여 넣기를 해주시면 Extension I/O 보드를 제어할 함수들을 이용하실 수 있습니다. <그림 4-7>은 AtsCppUD 파일을 붙여 넣기를 한 예 입니다.

<그림 4-6> 프로젝트 폴더 내부 파일 구성 예

<그림 4-7> ATS_IOCONT.dll 파일을 붙여 넣기 한 예

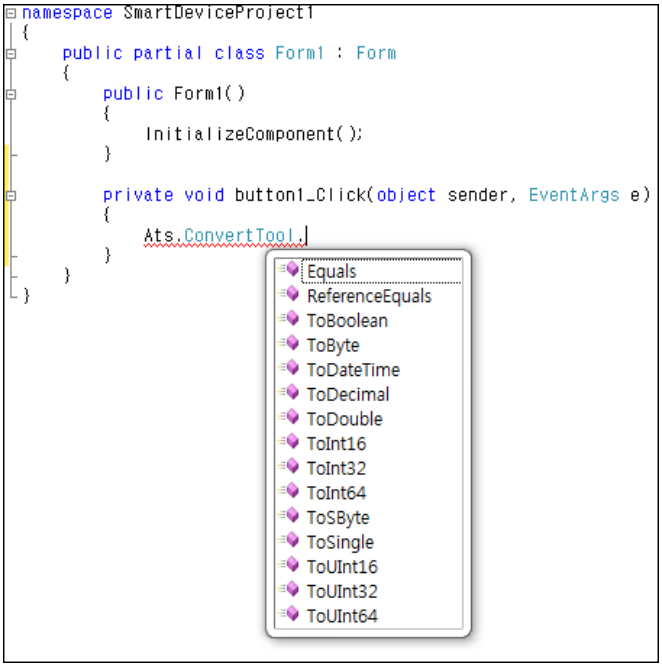
4.3 Ats 라이브러리 파일 사용 방법

Ats 라이브러리 파일을 Visual Studio의 참조 트리에 등록을 마치시면, <그림 4-8>와 같이 Ats를 기입 시, 관련된 클래스, 메서드, 열거형, 함수 명 등을 선택할 수 있도록, 나타나게 됩니다.
<그림 4-8>는 Ats 라이브러리 파일에 포함된 함수들의 예입니다.



<그림 4-8> Ats 기입 시 나타나는 클래스, 메서드, 열거형, 함수 명들의 예

각각 클래스들은 고유의 다양한 함수들로 구성되어 있으며, 관련 함수들은 '.' 문자를 기입하여 확인, 사용 가능합니다. <그림 4-9>는 ConvertTool 클래스의 함수 종류가 표시되는 예입니다.



<그림 4-9> ConvertTool 함수 종류 표시의 예

4.4 Ats 라이브러리 파일의 클래스 구성

각각 클래스들은 다른 기능을 실행하는 함수와, 공통적으로 포함되어있는 메서드들로 구성되어 있습니다. <표 4-1>은 Ats 라이브러리 파일에 구성되어있는 클래스들의 종류와 내용입니다.

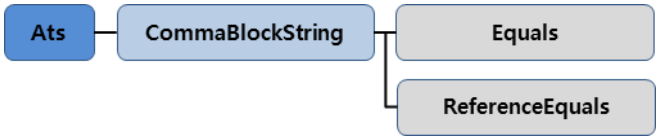
CommaBlockString	콤마로 구분된 문자열을 해석하기 위한 클래스들의 네임스페이스.
Control	윈도우 컨트롤 설정 관련 클래스.
ConvertTool	선택한 객체를 지정한 자료형으로 변환 시키는 클래스.
DrawClass	지정한 요소를 그리는 클래스.
EnumPlaySound	지정한 음성파일을 재생/정지 등에 관련된 클래스.
EnumShowWindow	윈도우 창의 크기, 위치, 표시, 활성화 등을 제어/확인하는 클래스.
IOControl	Extension I/O 모듈의 상태를 제어/확인하는 클래스.
MainBoard	메인보드에 설치된 이미지 버전의 정보 및 전원 리셋에 관련된 클래스.
OldDefine	Visual Studio에서 미리 정의되어있는 클래스들의 네임스페이스.
RegistryTool	레지스트리에 등록된 특정한 데이터를 읽고 쓰기 위한 클래스.
Scada	Autobase CE의 태그 값 및 자료 값에 관련된 클래스들의 네임스페이스.
TimeOutClass	통신 시간 초과에 관련된 클래스.
TimeUtil	날짜/시간을 가감하거나 설정할 수 있는 클래스.
Win32Function	윈도우 창 제어에 관련된 클래스.

<표 4-1> Ats 라이브러리 파일에 구성되어있는 클래스들의 종류와 내용

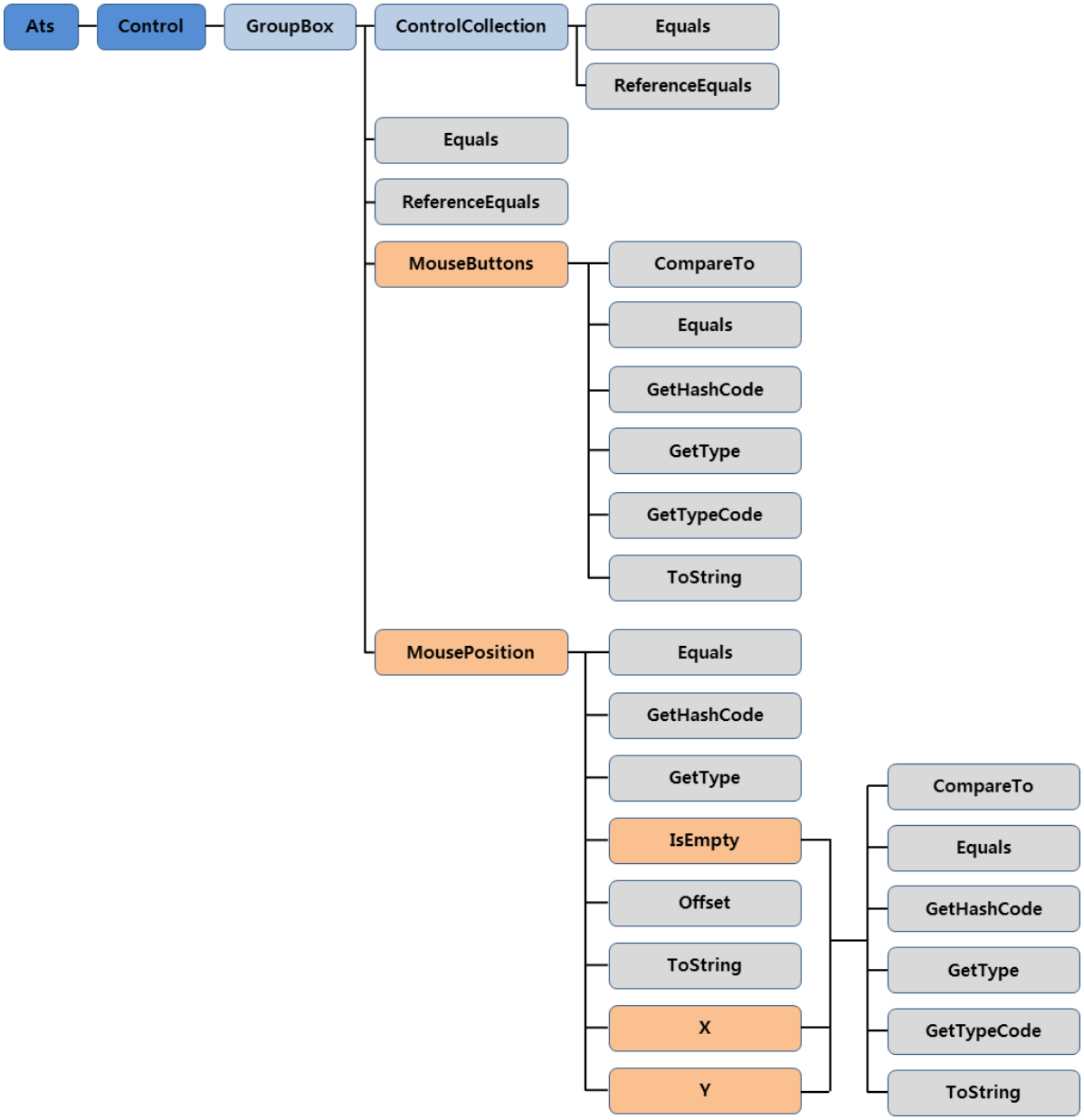
[참 고]	각 함수들의 형식, 인자, 반환 값 등에 관한 자세한 사항은 Ats Library 문서를 참조 하시기 바랍니다.

4.5 Ats 라이브러리 파일의 클래스 구성

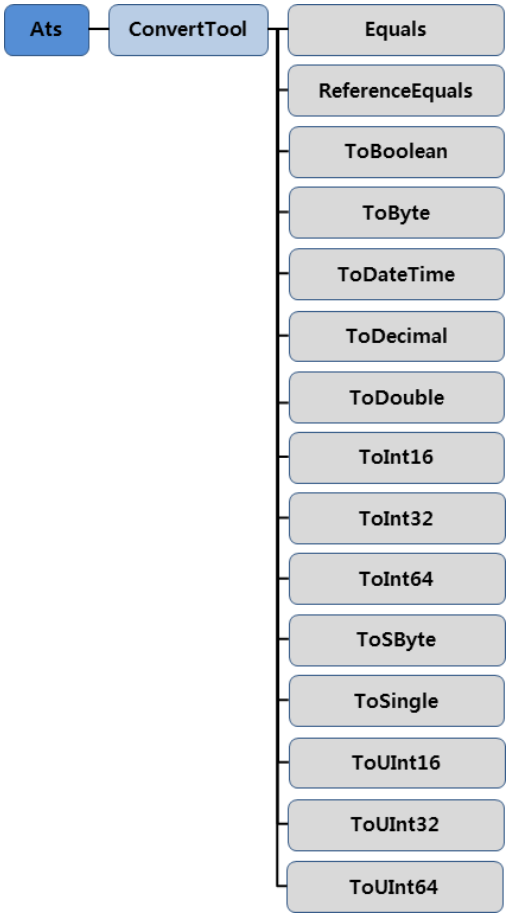
Ats 라이브러리 파일이 포함하고 있는 함수들의 구성은 <그림 4-10>~<그림 4-23>와 같습니다.



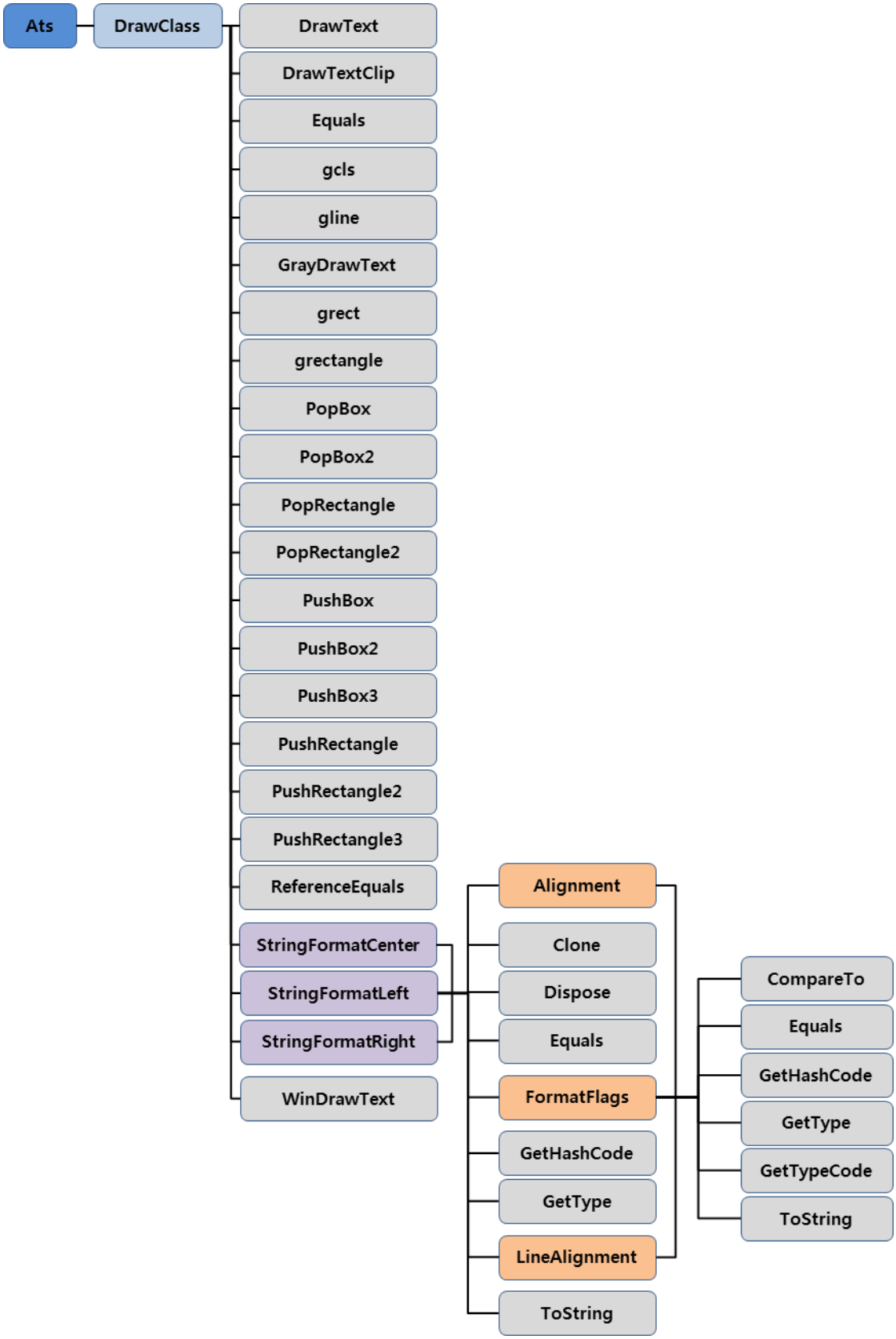
<그림 4-10> CommaBlockString 의 함수 구성



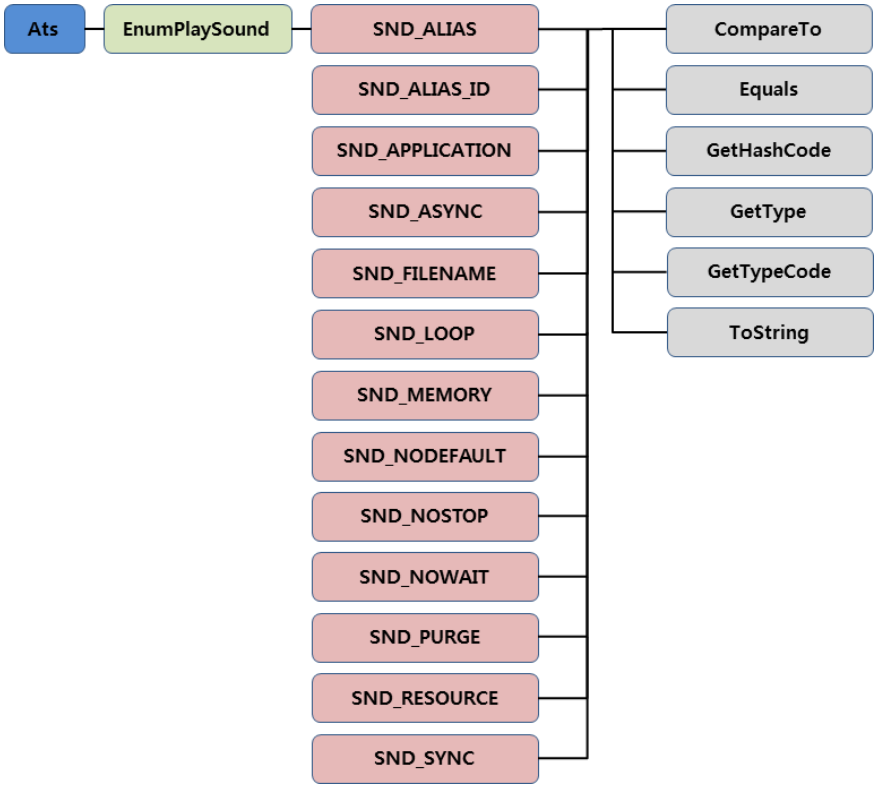
<그림 4-11> Control의 함수 구성



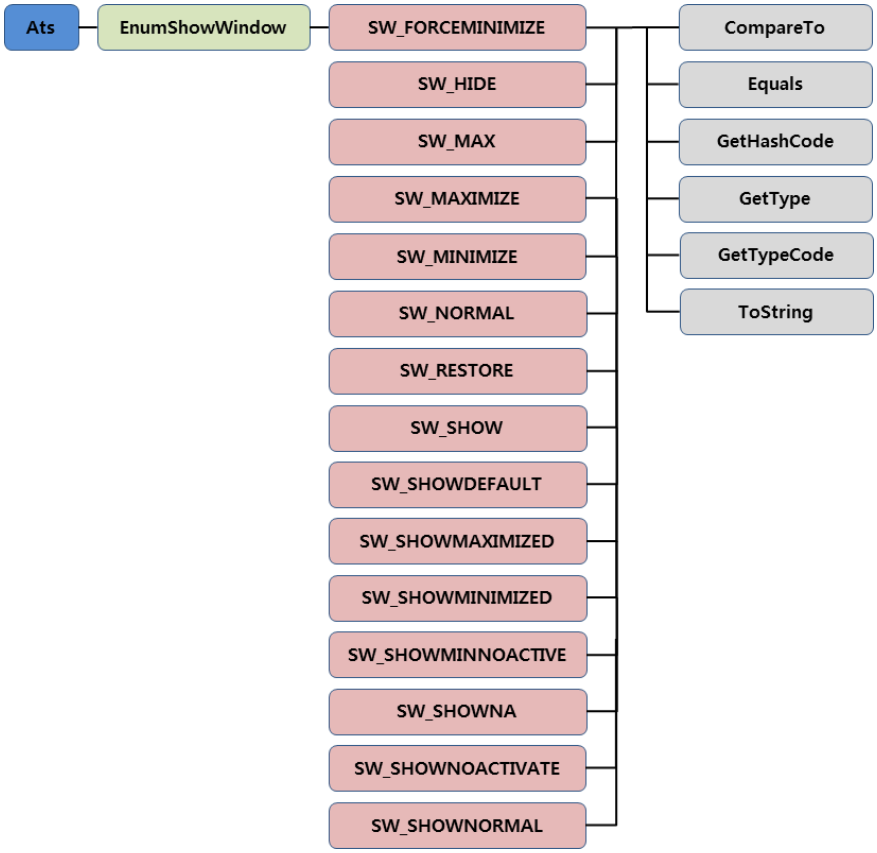
<그림 4-12> ConvertTool 의 함수 구성



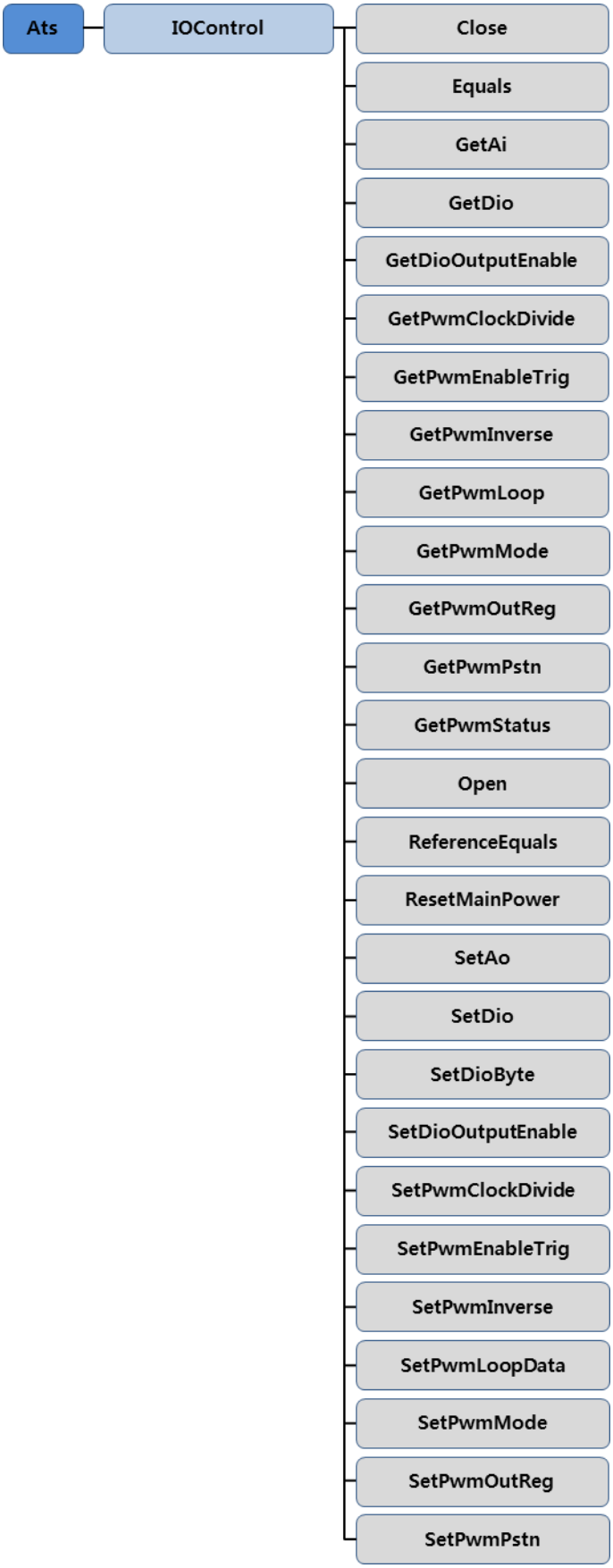
<그림 4-13> DrawClass 의 함수 구성



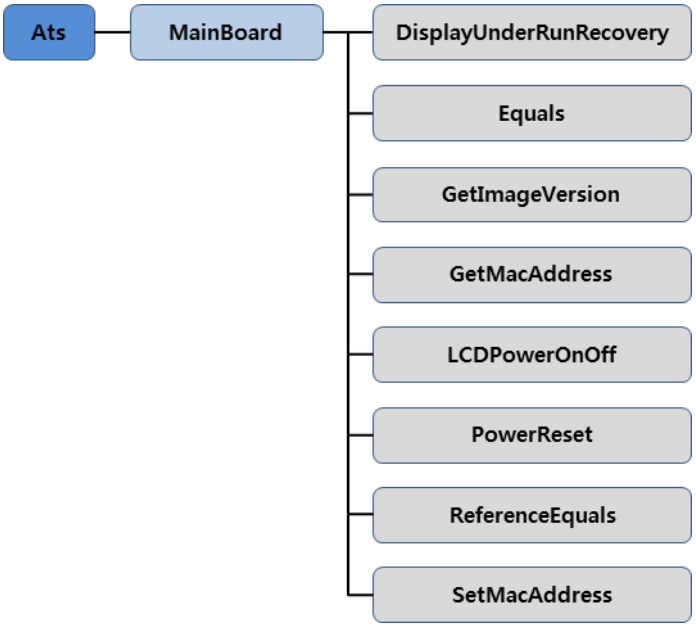
<그림 4-14> EnumPlaySound 의 함수 구성



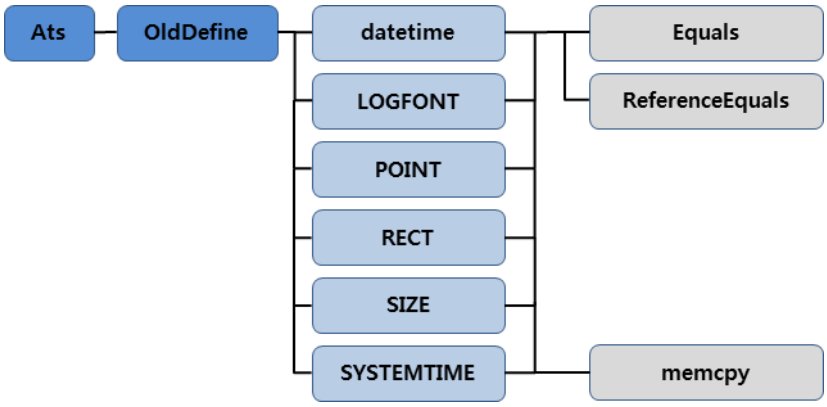
<그림 4-15> EnumShowWindow 의 함수 구성



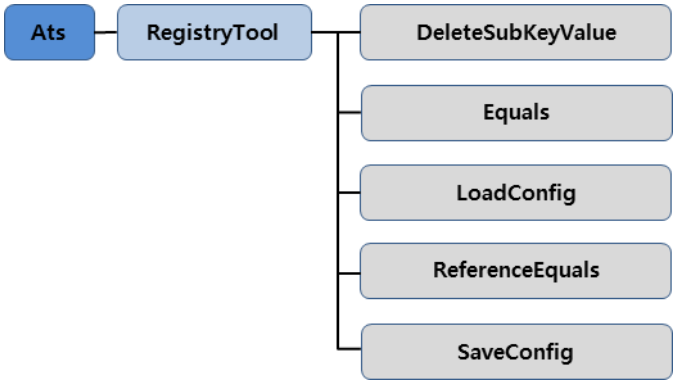
<그림 4-16> IOControl 의 함수 구성



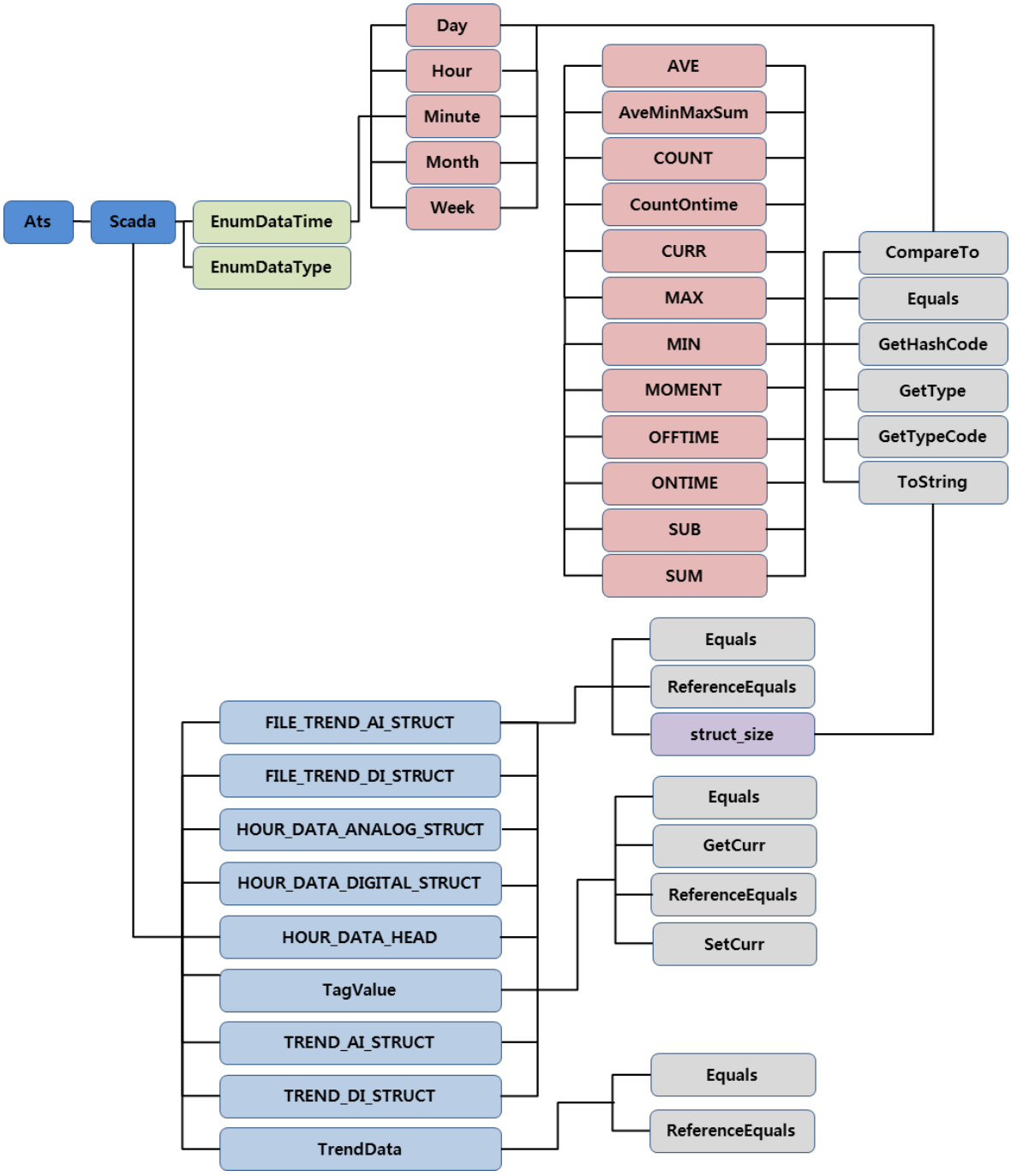
<그림 4-17> MainBoard 의 함수 구성



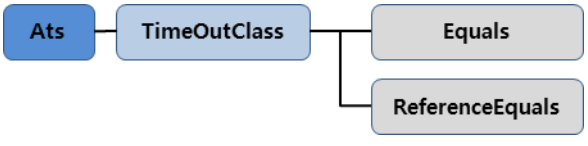
<그림 4-18> OldDefine 의 함수 구성



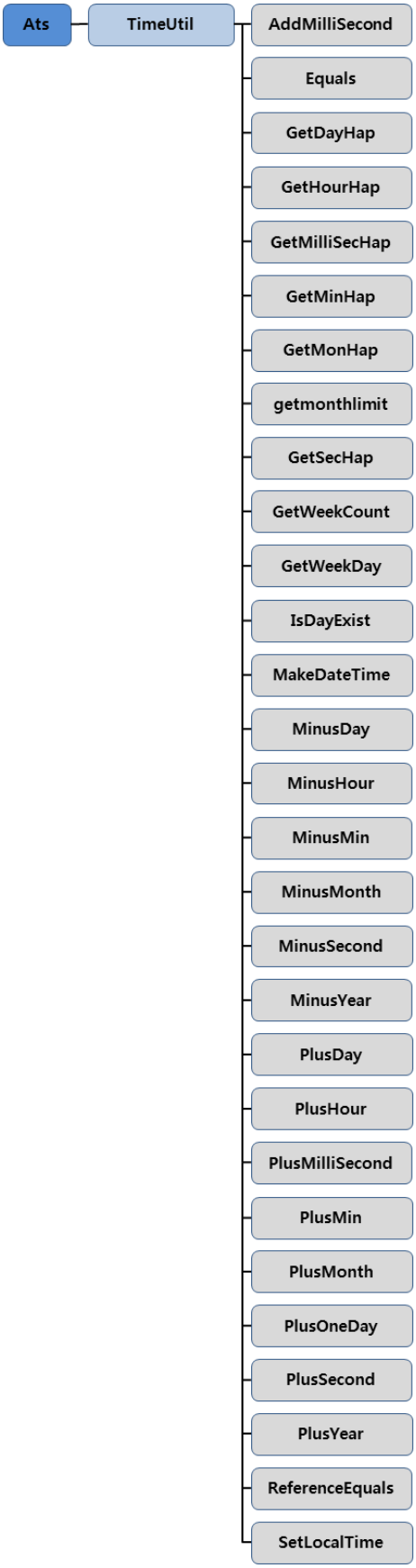
<그림 4-19> RegistryTool 의 함수 구성



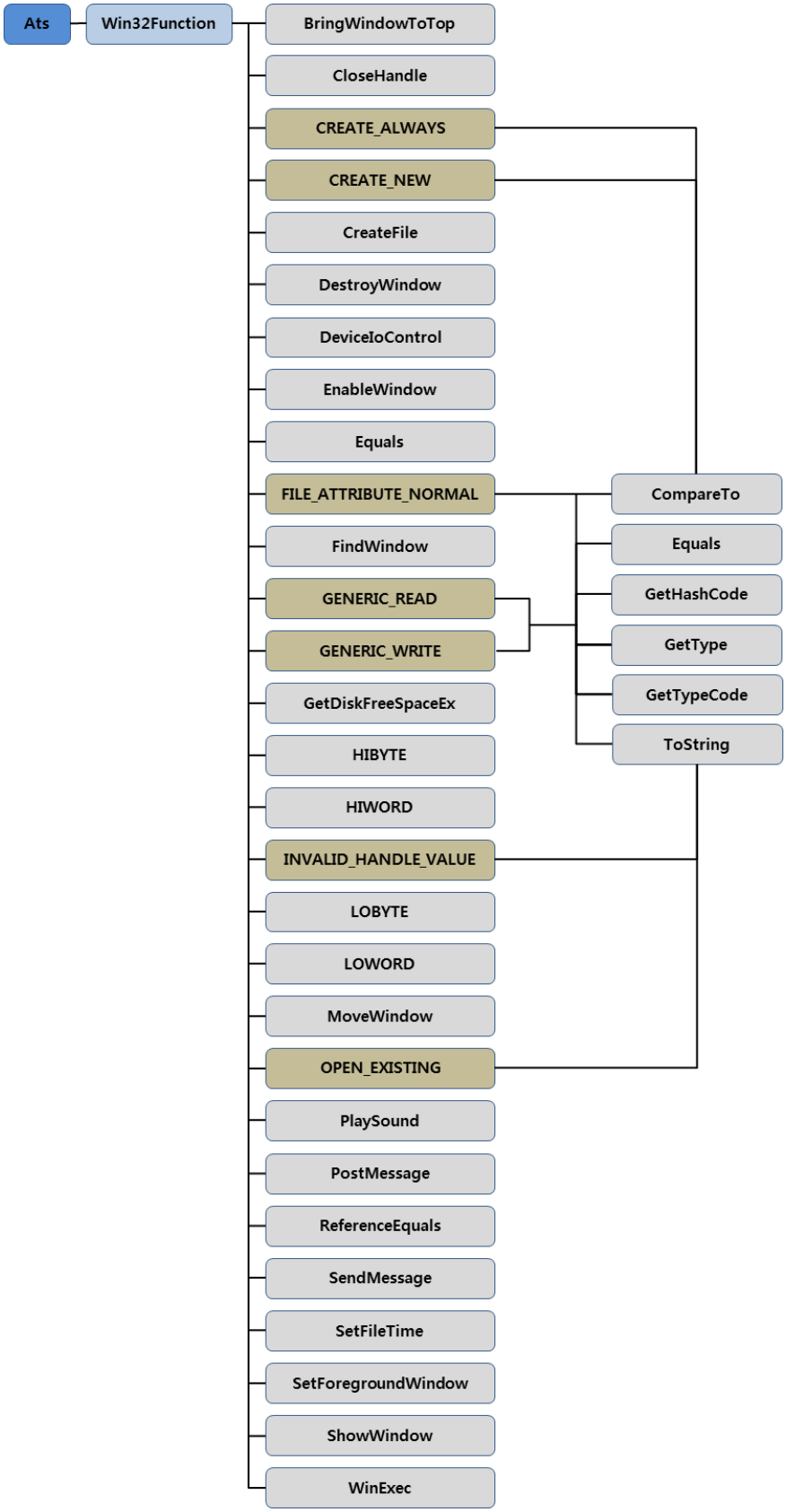
<그림 4-20> Scada 의 함수 구성



<그림 4-21> TimeOutClass 의 함수 구성

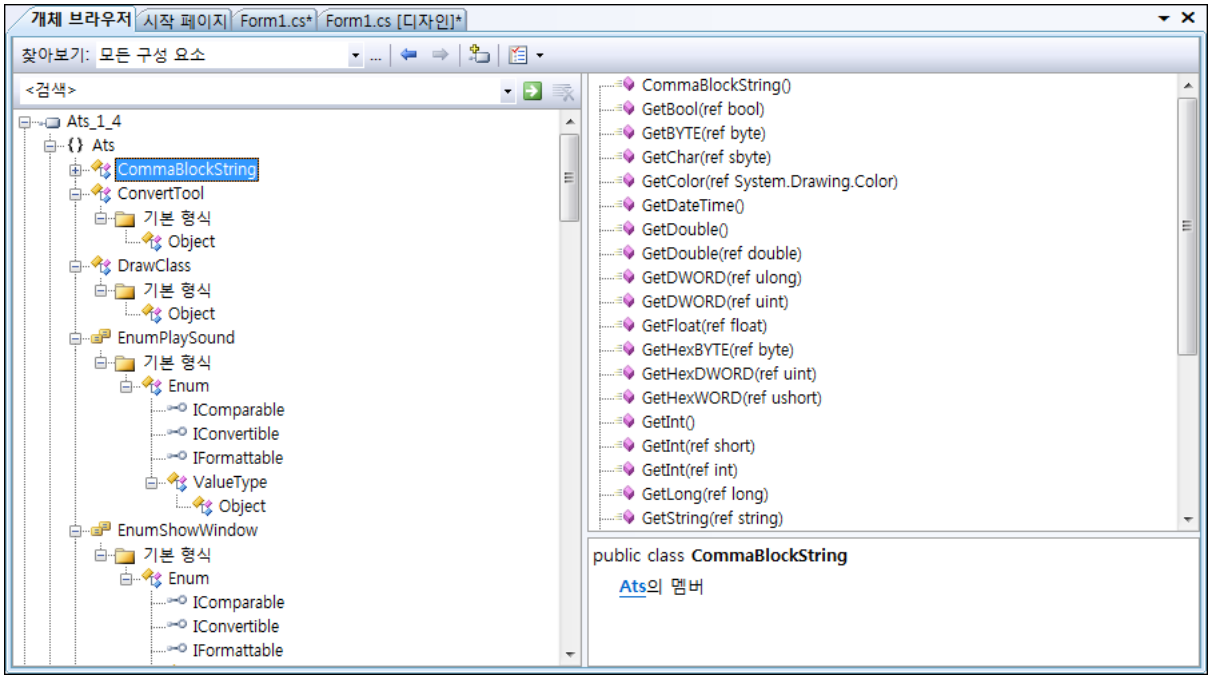


<그림 4-22> TimeUtil 의 함수 구성



<그림 4-23> Win32Function 의 함수 구성

<그림 4-10>~<그림 4-23>에 나타나있는 구성들은 Visual Studio의 보기₩개체 브라우저를 통해서도 Ats의 모든 구성 요소를 확인하실 수 있습니다. <그림 4-24>는 개체 브라우저를 통해 Ats 라이브러리 파일의 구성 요소를 확인한 예입니다.



<그림 4-24> Visual Studio 보기메뉴의 개체브라우저를 통한 Ats 라이브러리 파일의 구성요소 확인 예