

AIC Studio

이 사용 설명서와 AIC Studio 에 관한 모든 저작권은 ㈜오토베이스에 있습니다.

© Autobase, Inc. All rights reserved.

<주의>

본 설명서의 내용은 사전 통보 없이 변경될 수 있습니다.

2018 년 3 월 2 일 개정

차 례

1. 소개	6
1.1 실행 방법	6
1.1.1 시작 메뉴에서 실행.....	6
1.1.2 탐색기에서 실행	6
1.2 화면 구성	7
2. 기능별 도움말	8
2.1 새파일	8
2.2 편집하기	10
2.3 컴파일	22
2.4 통신 설정	23
2.5 제어기로 다운로드	26
2.6 제어기 설정	27
2.7 값보기(디버깅)	41
3. 메뉴별 도움말	43
3.1 파일 메뉴	43
3.1.1 새파일 메뉴.....	43
3.1.2 열기 메뉴.....	43
3.1.3 저장 메뉴.....	45
3.1.4 다른이름으로 저장 메뉴.....	45
3.1.5 닫기 메뉴.....	45
3.1.6 모두 닫기 메뉴.....	45
3.1.6 종료 메뉴.....	45
3.2 편집 메뉴	46

3.2.1 실행 취소 메뉴.....	46
3.2.2 다시 실행 메뉴.....	46
3.2.3 복사 메뉴.....	46
3.2.4 잘라내기 메뉴.....	46
3.2.5 붙여넣기 메뉴.....	46
3.2.6 지우기 메뉴.....	46
3.2.7 글로벌 변수 설정 메뉴.....	47
3.2.8 로컬 변수 설정 메뉴.....	54
3.2.9 M 영역 설정 메뉴.....	58
3.3 보기 메뉴	62
3.3.1 프로젝트 메뉴.....	62
3.3.2 제어기 정보 메뉴.....	67
3.3.3 모두보기 메뉴.....	68
3.3.3 모두숨기기 메뉴.....	68
3.4 도구 메뉴	69
3.4.1 컴파일 메뉴.....	69
3.5 온라인 메뉴	69
3.5.1 통신 설정 메뉴.....	69
3.5.2 통신 디버깅 메뉴.....	69
3.5.3 제어기로 다운로드 메뉴.....	73
3.5.4 제어기에서 업로드 메뉴.....	73
3.5.5 CPU 제어 메뉴.....	74
3.5.6 메모리 메뉴.....	75
3.5.7 콘솔 메뉴.....	79
3.5.8 제어기 설정 메뉴.....	82
3.5.9 값보기 메뉴.....	82
3.6 도움말 메뉴	83
3.6.1 프로그램 정보 메뉴.....	83
3.6.2 도움말 메뉴.....	83

부록 1 : FUNCTION 설명	84
4.1 수치연산	84
4.2 비교	87
4.3 비트연산	89
4.4 비트이동	91
4.5 선택	92
4.7 각도 변환	93
4.8 데이터 교환	94
4.9 데이터 처리	95
4.10 배열연산	98
4.11 카운터	101
4.12 타이머	103
4.13 감시/제어	105
부록 2 : 메모리 주소 입력법	108
부록 3 : MODBUS 통신을 위한 메모리 맵	110
부록 4 : 사용자 평선 작성법	112
부록 5 : C 언어 사용법	126

AIC Studio

1. 소개

AIC Studio 는 Autobase 에서 제작한 Controller 의 설정 및 로직 제작을 위해 사용하는 프로그램입니다.

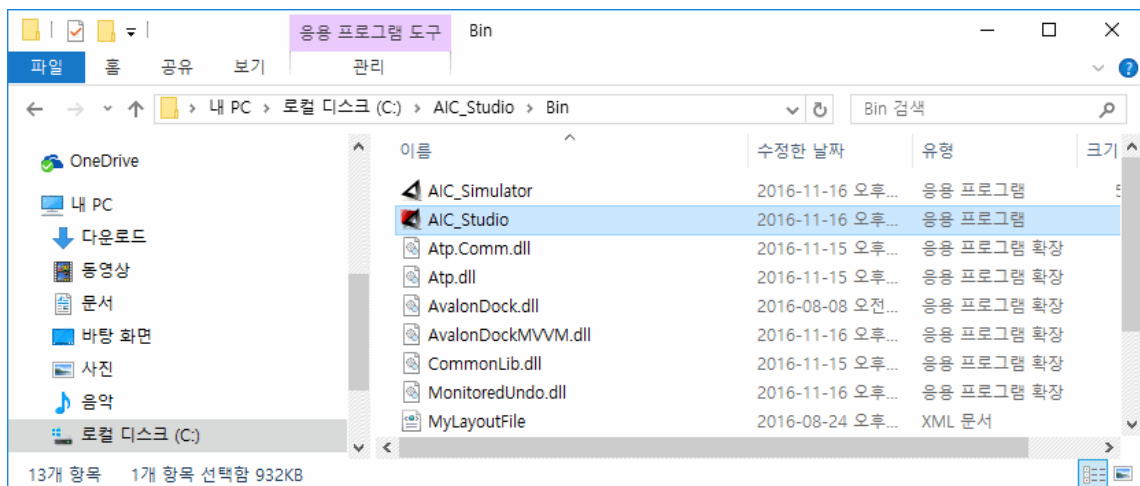
1.1 실행 방법

1.1.1 시작 메뉴에서 실행

윈도우 시작 -> 프로그램 -> AIC_Studio 의 AIC_Studio 를 선택하면 AIC Studio 가 실행됩니다

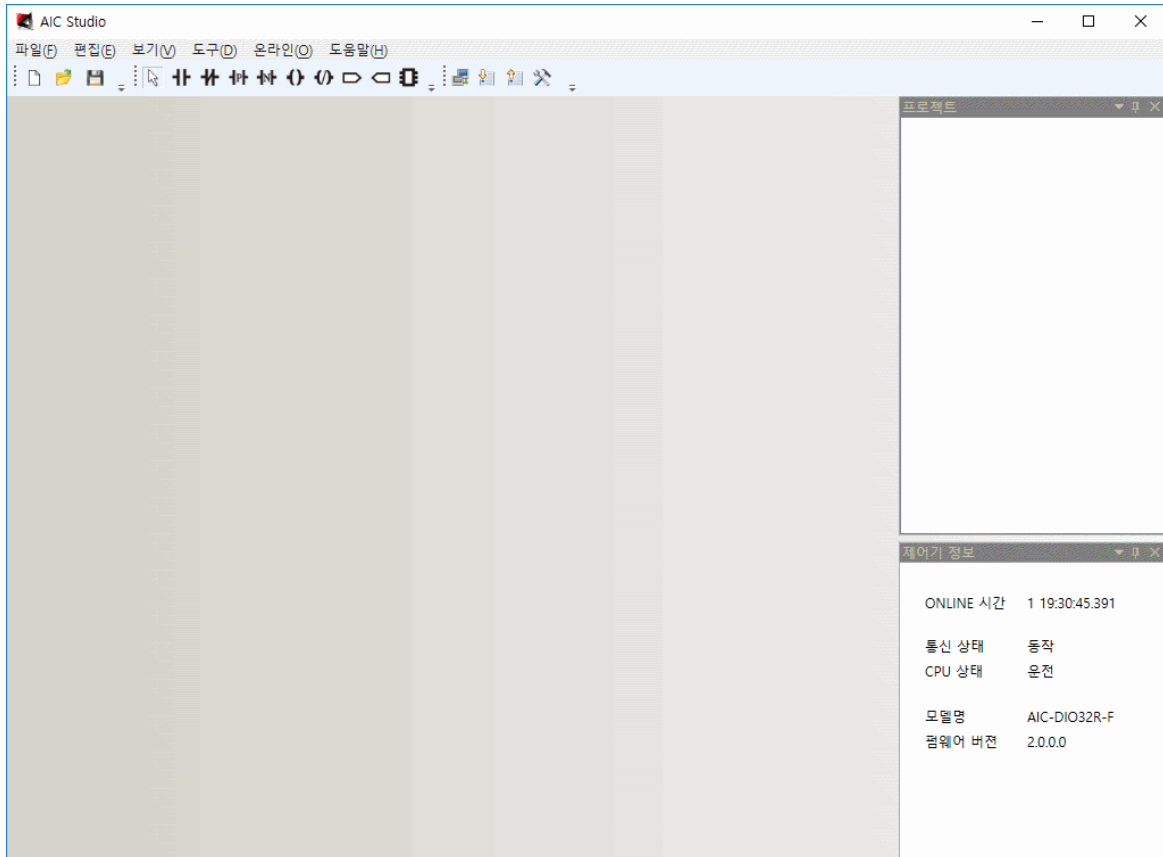
1.1.2 탐색기에서 실행

탐색기를 실행하여 C:WAIC_StudioWBin 폴더의 AIC_Studio.exe 를 실행하면 AIC Studio 가 실행됩니다.



1.2 화면 구성

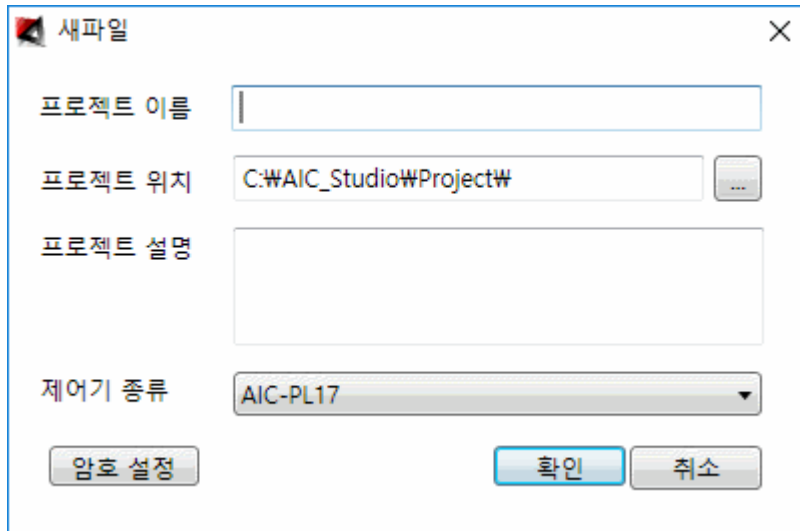
AIC Studio 를 실행하면 다음과 같이 화면에 나타납니다.



2. 기능별 도움말

2.1 새파일

파일 메뉴의 부메뉴 새파일을 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.



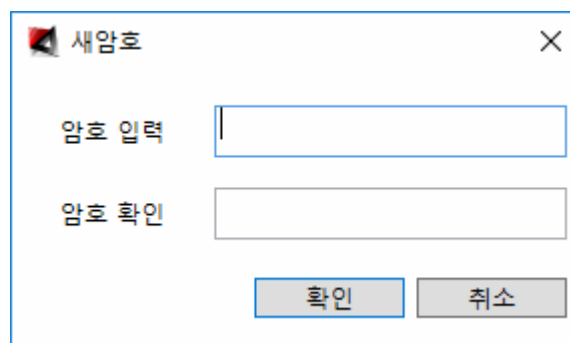
프로젝트 이름 : 프로젝트 이름을 입력합니다(프로젝트 이름은 가능하면 영문으로 입력하기 바랍니다).

프로젝트 위치 : 프로젝트 위치를 입력합니다(한번 설정하면 그 폴더를 기반으로 관리가 됩니다).

프로젝트 설명 : 프로젝트 설명을 입력합니다

제어기 종류 : 제어기 종류를 선택합니다. 제어기 종류마다 포인트 수가 다르므로 작성하고자 하는 제어기에 맞추어 설정합니다.

암호 설정 : 암호 설정을 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.



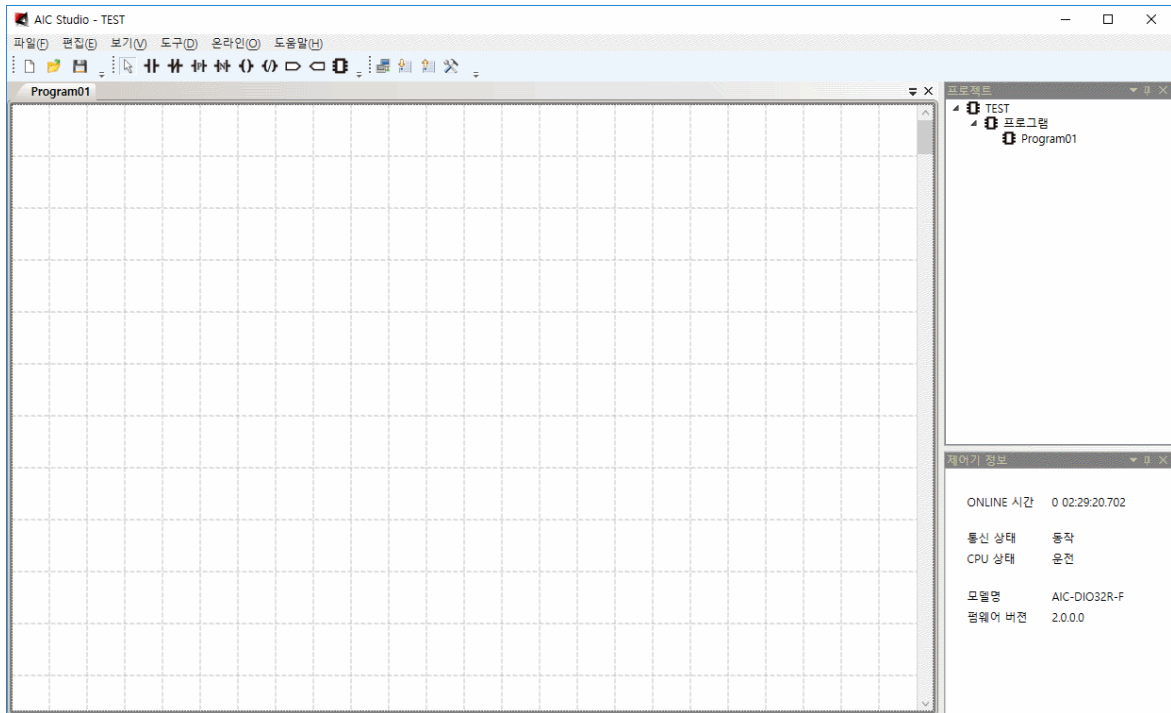
암호 입력 : 암호를 입력합니다.

암호 확인 : 암호 입력에 입력한 암호와 같은 암호를 입력합니다.

확인 : 확인을 선택하면 암호가 입력되며 이후 해당파일을 열기 위해서는 암호를 알아야만 열 수 있으므로 잘 기억하기 바랍니다(암호를 입력하지 않으면 아무나 열어 편집할 수 있게 됩니다).

변수 자동 추가 : 초기 변수를 자동으로 추가해 줄것인지를 선택합니다.

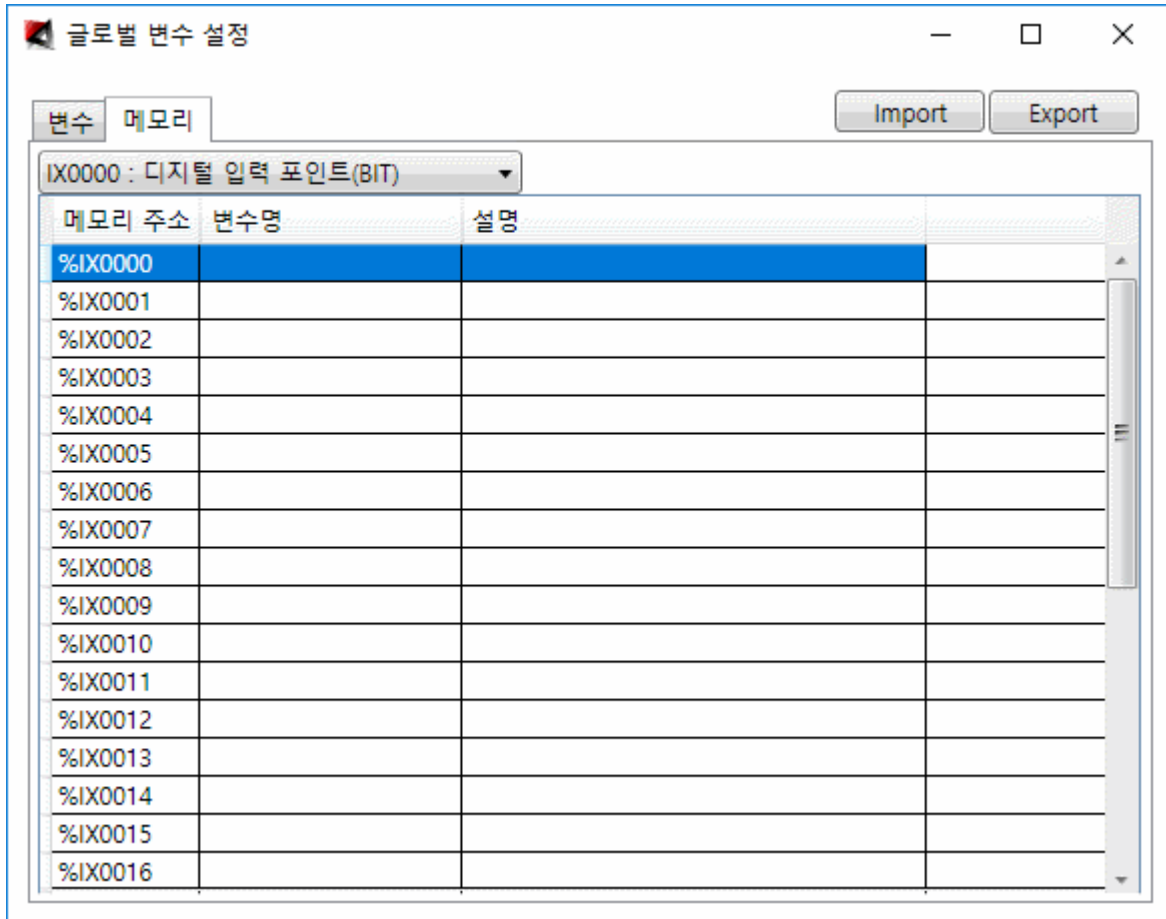
확인 : 확인을 선택하면 다음과 같이 새파일이 화면에 나타납니다.



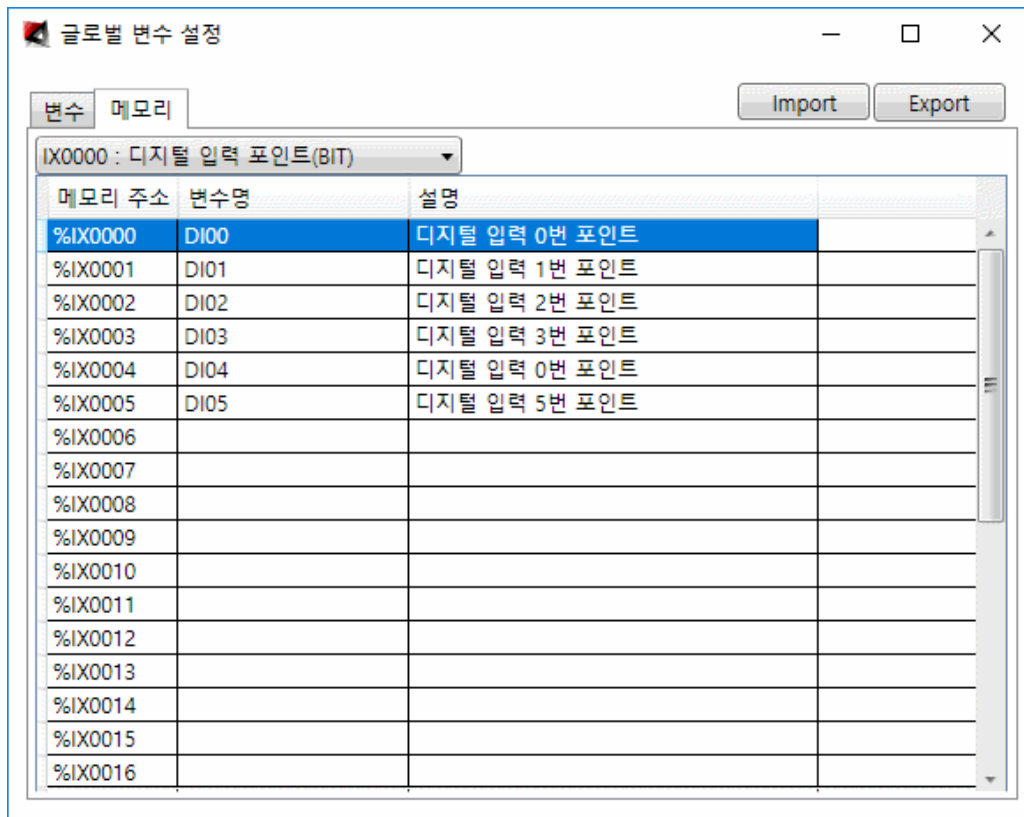
2.2 편집하기

편집하기 위해서는 먼저 변수를 등록해야 합니다(새파일을 만들때 변수 자동 추가를 체크 하면 자동으로 추가가 되어 변수명을 수정하여 사용하면 되나 여기서는 정확한 사용법을 설명하기 위해 변수 자동 추가를 체크하지 않는 경우를 가정하여 설명합니다).

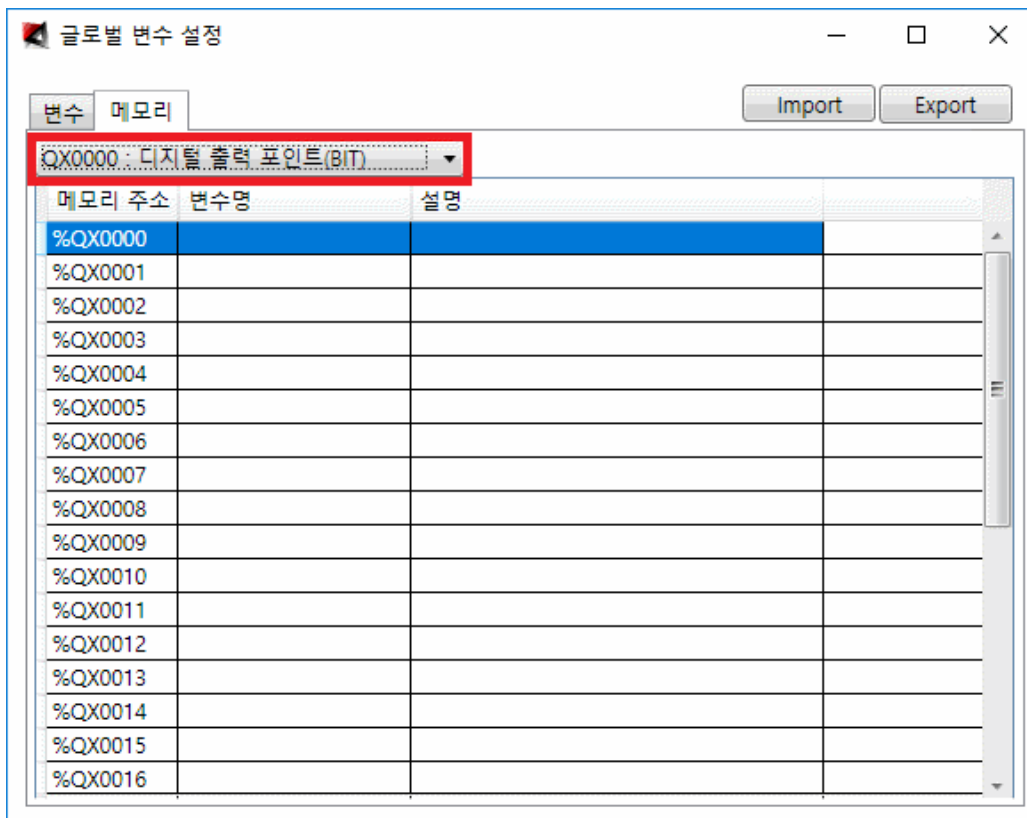
편집메뉴의 부메뉴 글로벌 변수 설정을 선택하여 메모리 탭을 누르면 다음과 같이 화면에 나타납니다.



다음 화면과 같이 디지털 입력 포인트 DI00~DI05 를 입력합니다.



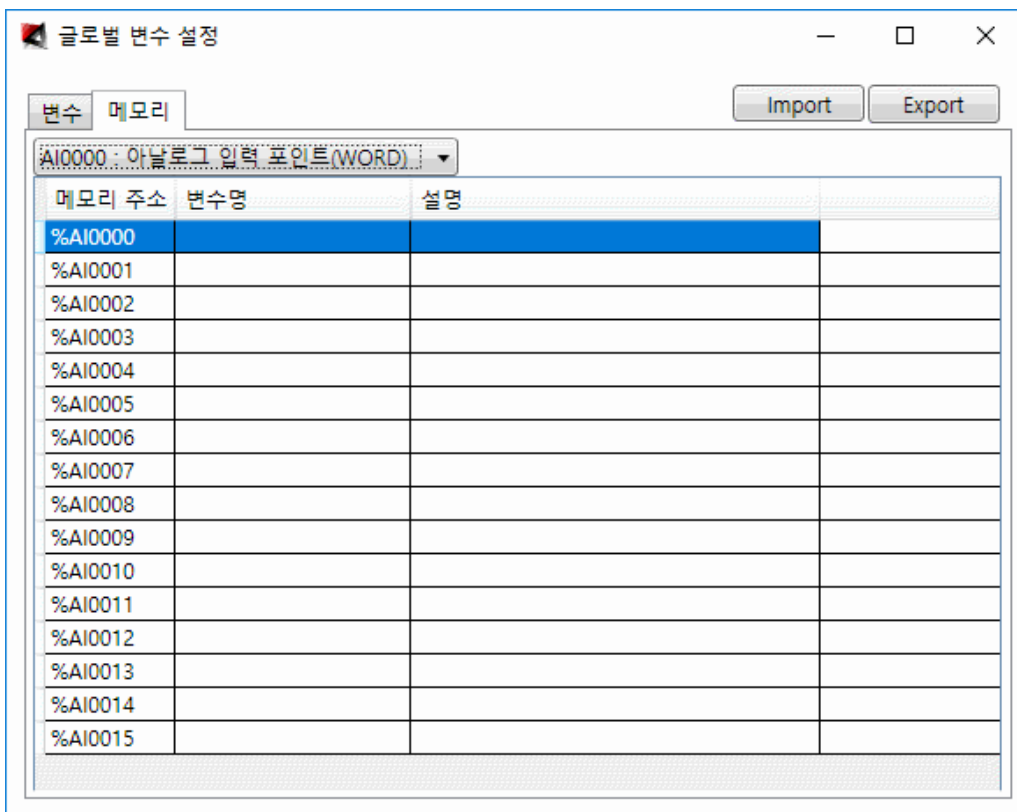
메모리 종류를 디지털 출력 포인트(BIT)로 선택하면 다음과 같이 화면이 바뀝니다.



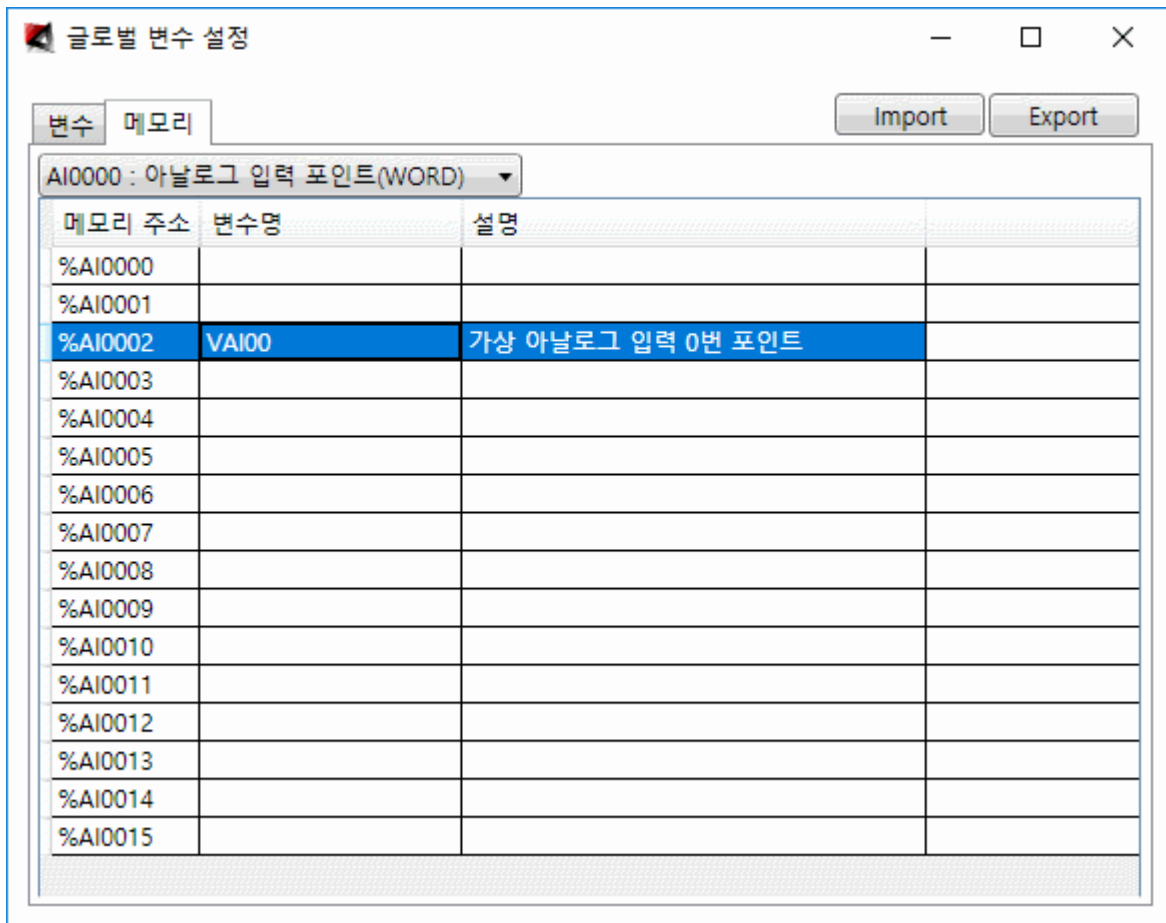
다음 화면과 같이 디지털 출력 포인트 DO00~DO03 을 입력합니다.



메모리 종류를 아날로그 입력 포인트(WORD)로 선택하면 다음과 같이 화면이 바뀝니다.

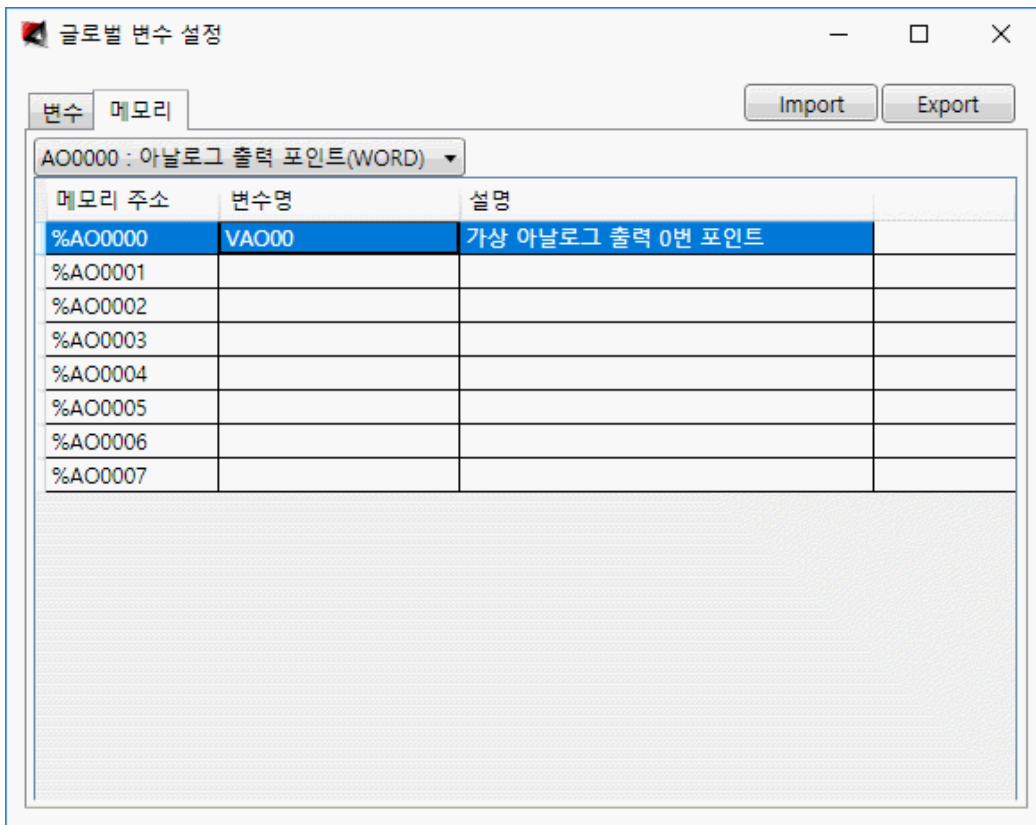


다음 화면과 같이 아날로그 입력 포인트 VAI00 을 입력합니다.

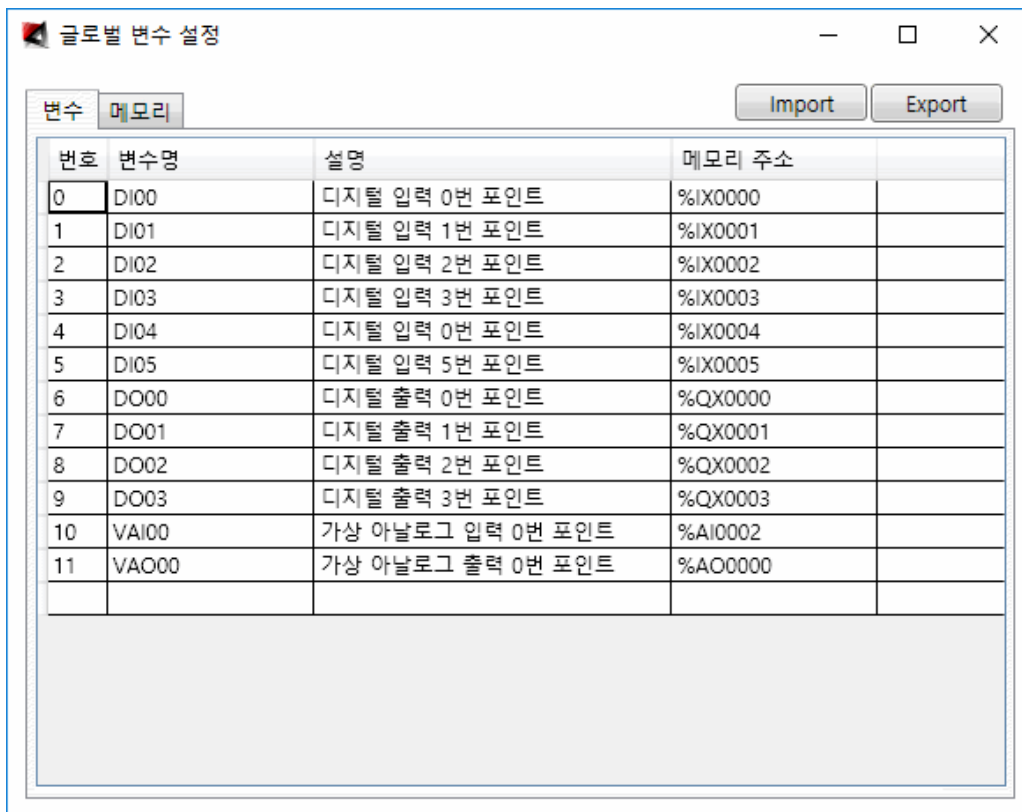


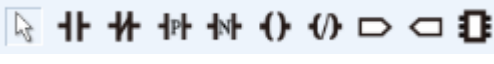
VAI00 은 아날로그 입력 포인트의 세 번째 위치에 입력했는데 본 VAI00 을 카운터로 사용하기 위하여 가상포인트로 입력한 것이며, Sample 프로젝트로 작성한 장비 AIC-PL17-F 모델은 실 아날로그 입력 2 포인트를 지원하므로 첫 번째 두 번째 포인트가 실포인트로 매핑이 됩니다.


메모리 종류를 아날로그 출력 포인트(WORD)로 선택하여 다음과 같이 VAO00 을 입력합니다.




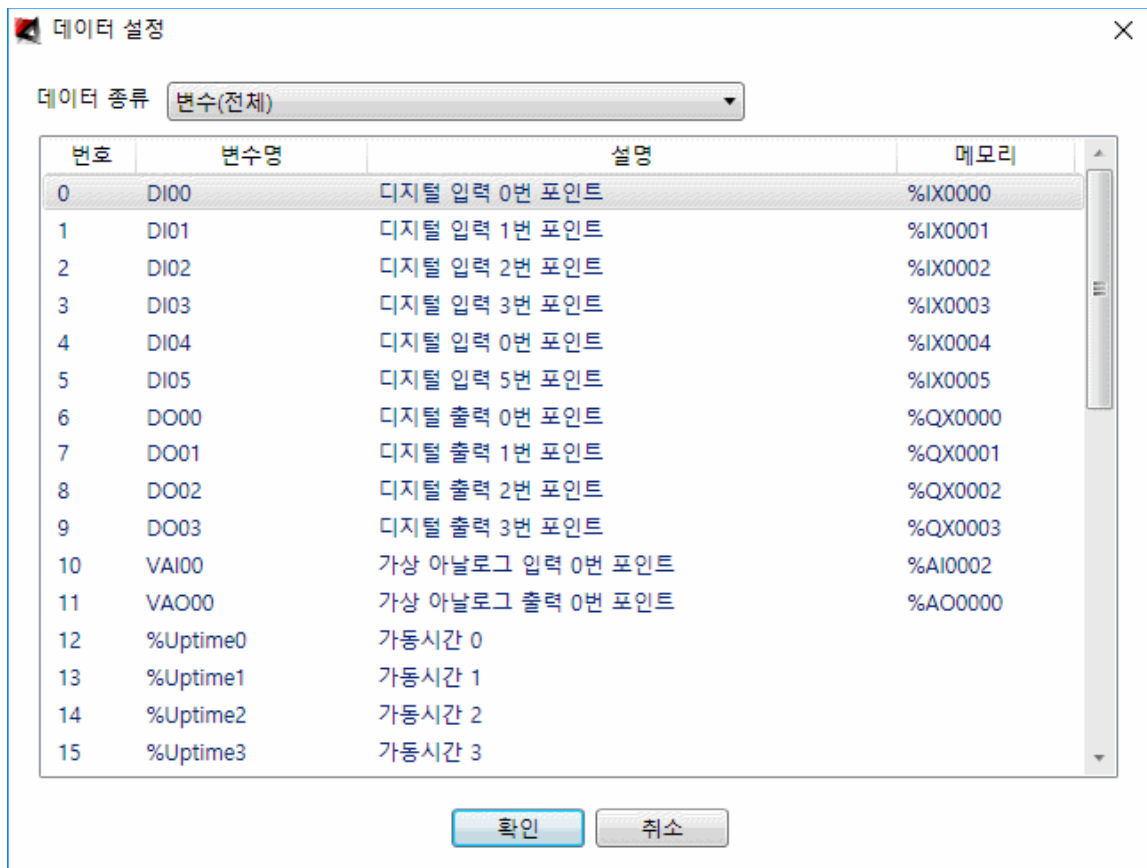
변수탭을 선택하면 다음과 같이 입력한 데이터 전체를 보여주며 여기에서도 편집이 가능합니다.




편집은 툴바의 편집 아이콘  을 이용하여 편집합니다.


-  [Select] : 객체를 선택하거나 이동 시 사용합니다. 또한 선을 추가로 그릴 때도 사용합니다.


-  [Normal Contact] : 디지털상태 값을 입력 받습니다. Contact 객체를 선택하고 추가하고자 하는 자리에 마우스로 클릭하면 다음과 같이 데이터 설정 대화 상자가 화면에 나타납니다.








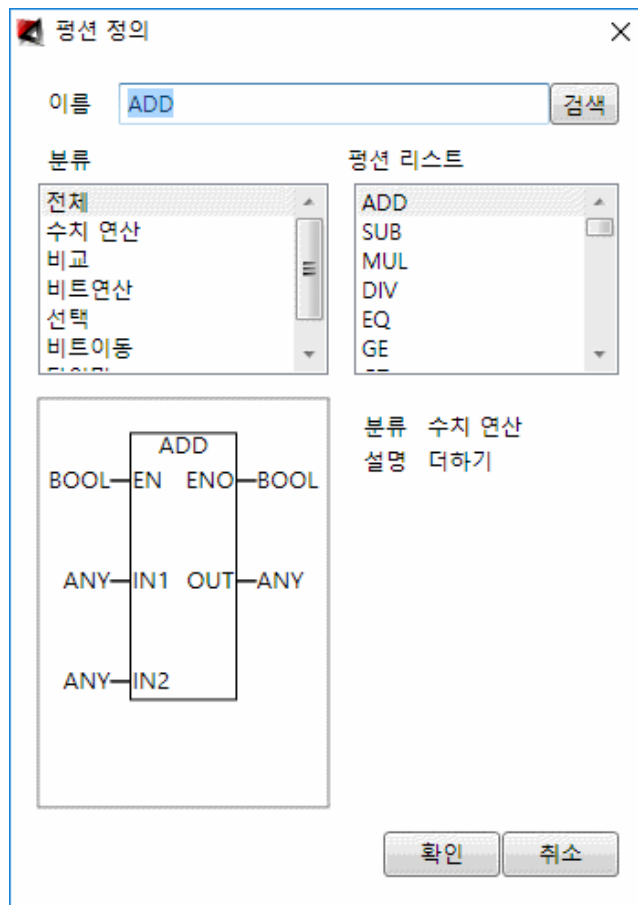
사용하고자 하는 포인트를 선택한 후 확인을 누르면 해당 자리에 Contact 가 들어갑니다

-  [Negative Contact] : 디지털 상태값의 NOT 을 하여 입력 받습니다.

-  [Positive transition-sensing Contact] : Positive transition 을 검출하는 contact 입니다. 값이 0 에서 1 로 바뀔 때 1 이 됩니다.


-  [Negative transition-sensing Contact] : negative transition 을 검출하는 contact 입니다. 값이 1 에서 0 으로 바뀔 때 1 이 됩니다.

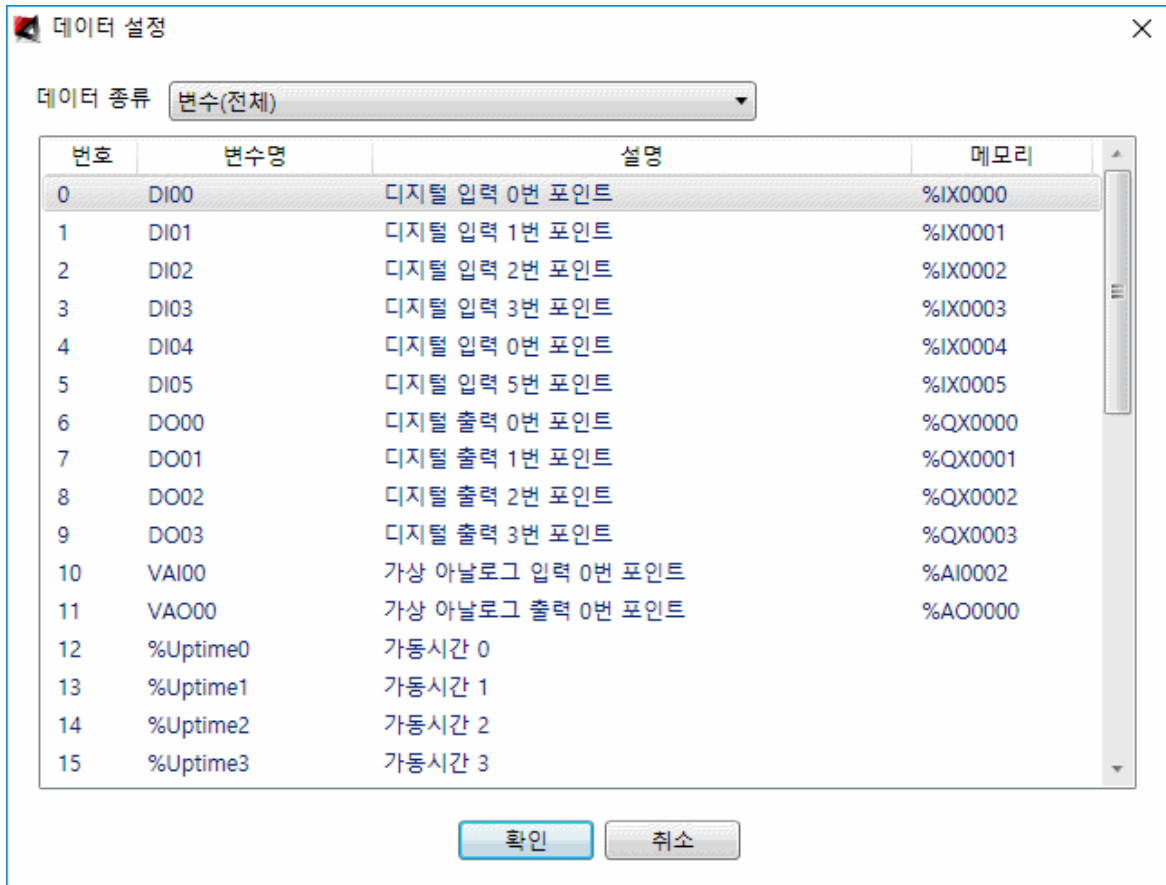
-  [Normal Coil] : 원편의 계산결과가 True 이면 1 이 됩니다.
-  [Negative Coil] : 원편의 계산결과가 False 이면 1 이 됩니다.
-  [input Variable] : 함수블록의 입력에 데이터를 공급하기 위해 사용됩니다.
-  [Output Variable] : 함수블록의 출력으로부터 데이터를 받기 위해 사용됩니다.
-  [Function] : Function 객체를 배치할 때 사용합니다. Function 객체를 선택하고 추가하고자 하는 자리에 마우스로 클릭하면 다음과 같이 평선 정의 대화 상자가 화면에 나타납니다



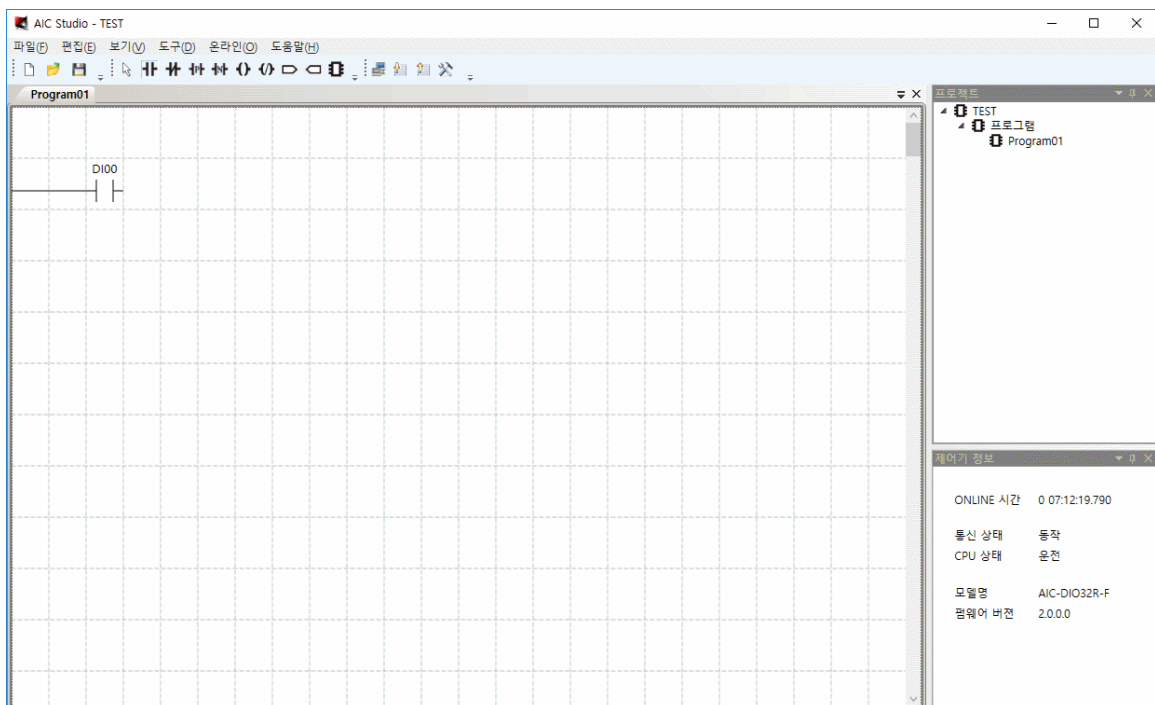
Function 에 대한 설명은 [첨부 1 Function 설명](#)을 참조하기 바랍니다.


이제 실제 편집을 해보도록 하겠습니다.

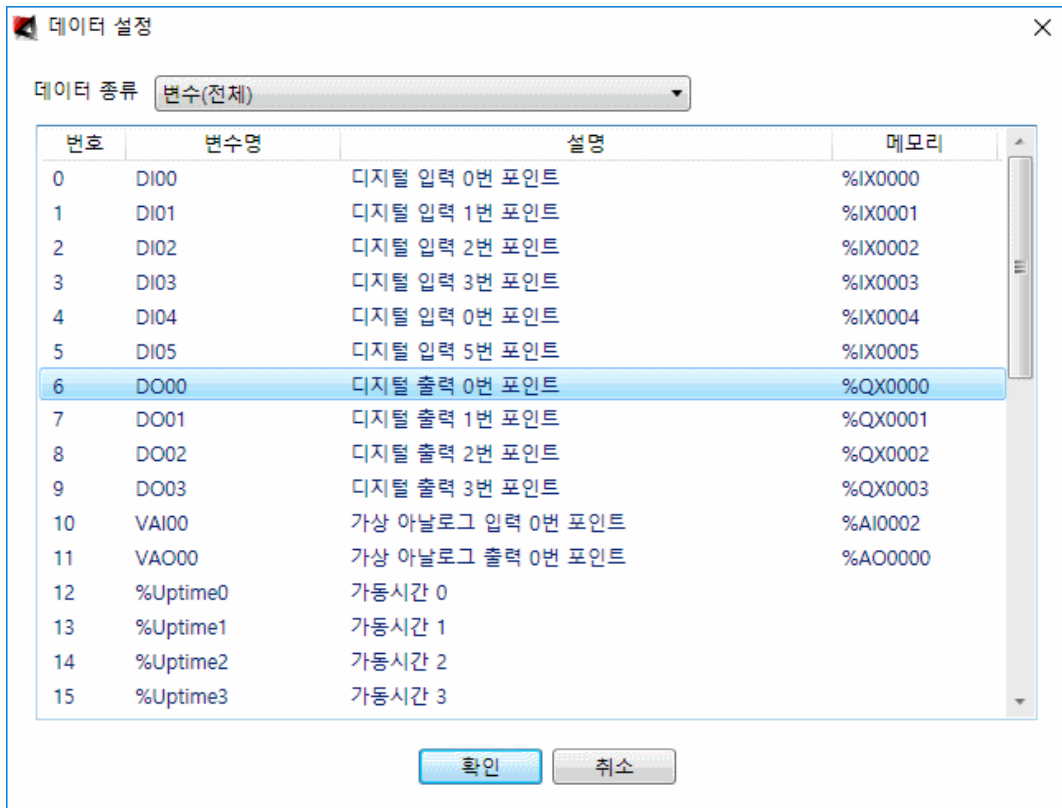
먼저  아이콘을 선택한 후 적당히 좌측 위치에 마우스를 클릭하면 다음과 같은 대화 상자가 생성됩니다.





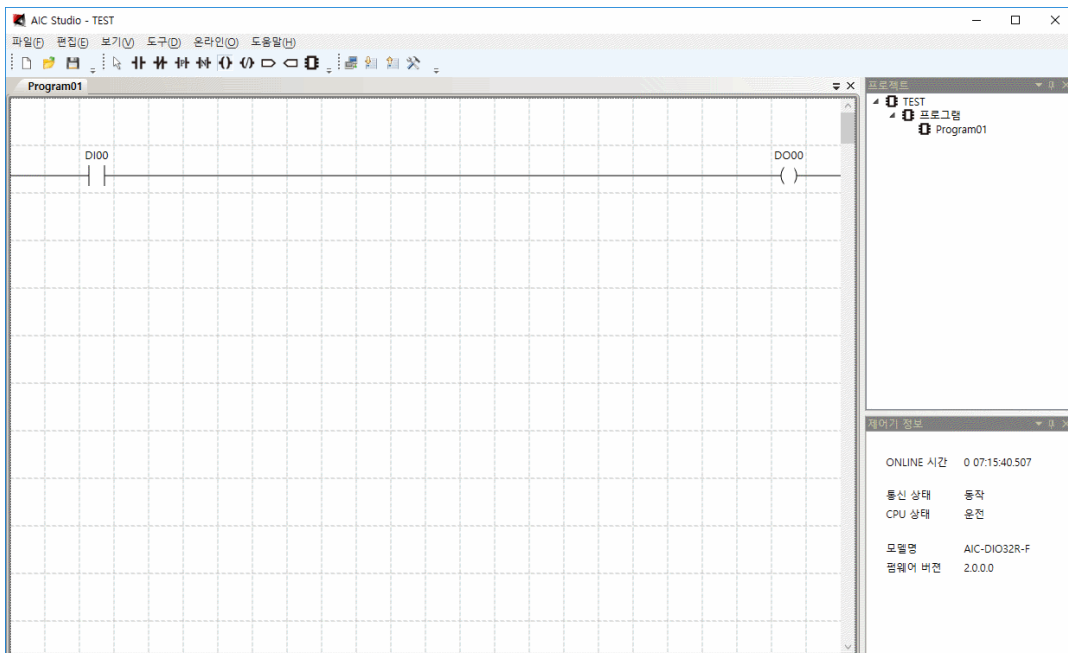
확인을 선택하면 화면에 다음과 같이 **⚡**가 삽입됩니다. **⚡** 좌측 선은 자동으로 연결이 됩니다.



다음으로  아이콘을 선택한 후 우측 위치에 마우스를 클릭하면 다음과 같이 데이터 설정 대화 상자가 생성되며, 스크롤을 내려 DO00 포인트를 선택합니다.

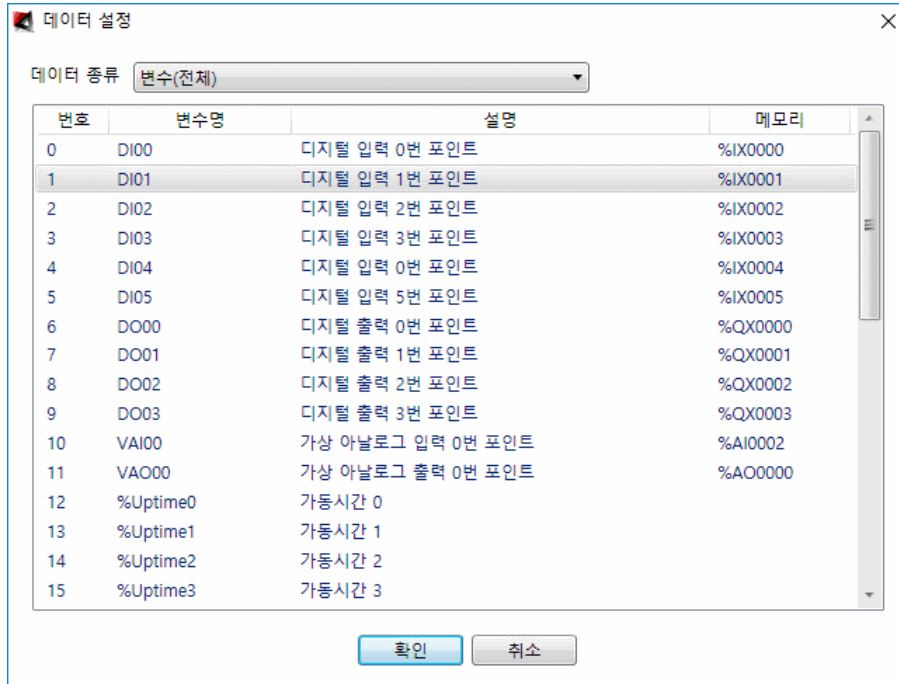


확인을 선택하면 화면에 다음과 같이  가 삽입됩니다.  좌우 선 또한 자동으로 연결이 됩니다.

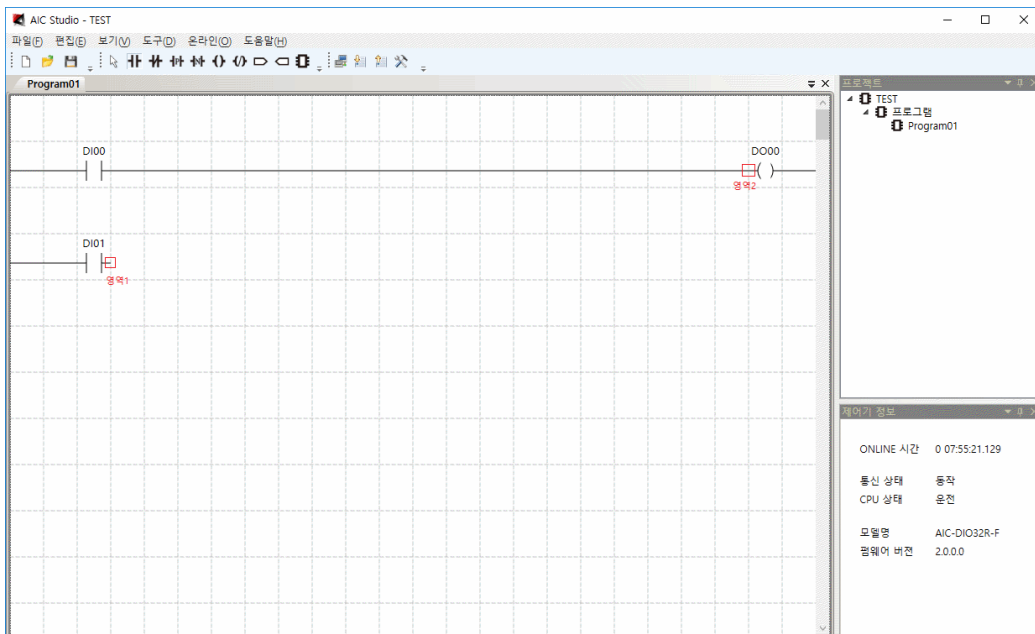


가장 기본적인 기능으로 DI00 포인트 값을 DO00 포인트에 쓰도록 로직 편집을 한 것입니다.

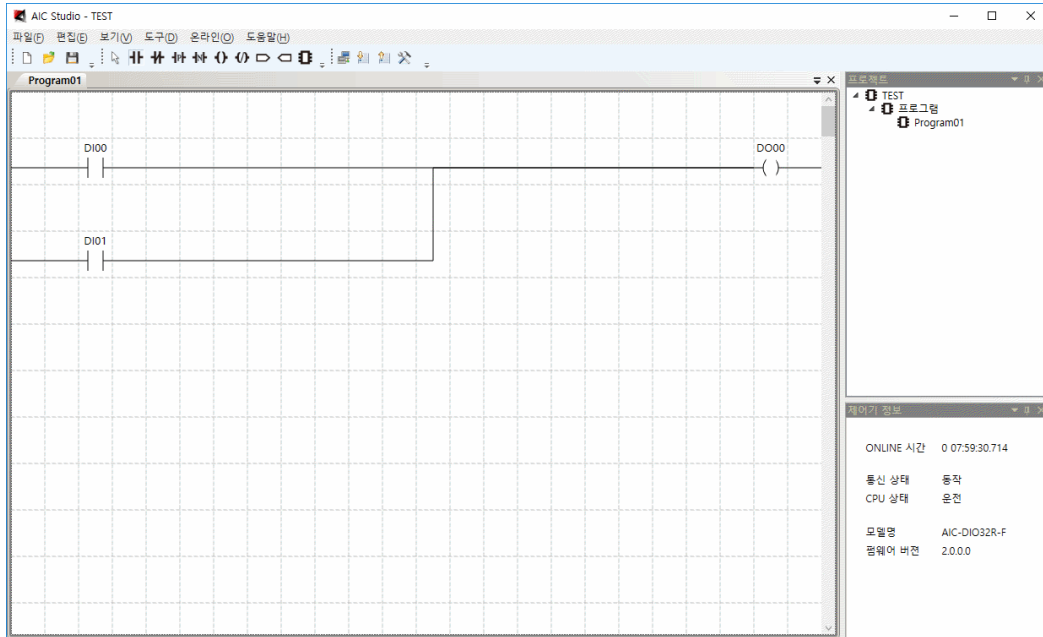
이제 **⇧** 를 하나더 추가해보겠습니다. **⇧** 아이콘을 선택한 후 DI00 아래에 마우스를 클릭하면 데이터 설정 대화 상자가 생성되며, 다음과 같이 DI01 을 선택합니다.



확인을 선택하면 화면에 다음과 같이 **⇧** 가 추가로 삽입됩니다. **⇧** 좌측 선은 자동으로 연결이 됩니다



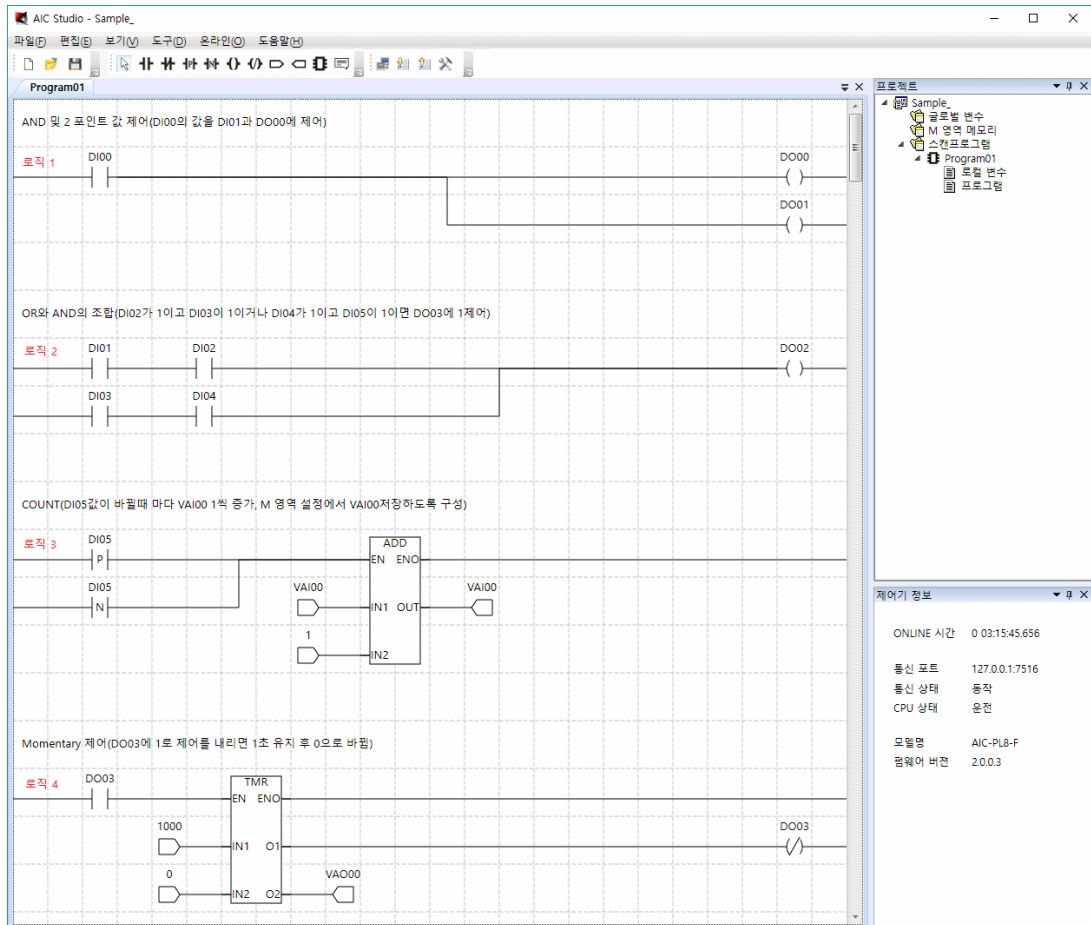
영역 1 에서 마우스 왼쪽 버튼을 눌러 영역 2 에서 마우스 왼쪽 버튼을 떼면 다음과 같이 선 연결이 됩니다.



이 로직은 DI00 포인트 값이 1 이거나 DI01 포인트 값이 1 인 경우 DO00 포인트에 1 을 그외에는 0 을 씁니다.

다음화면은 Sample 로 작성한 로직이며 앞에서 설명한 내용이면 충분히 작성할 수 있을 것이므로 작성된 로직에 대해서 설명하도록 하겠습니다.

파일 메뉴의 부메뉴 열기를 선택하여 Sample 프로젝트 열기를 수행하면 다음과 같이 화면에 나타납니다.



- 로직 1 : DI00 포인트 값을 DO00 포인트와 DO01 포인트에 씁니다.

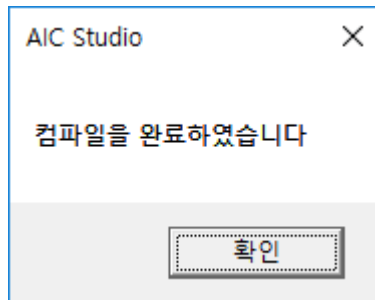
- 로직 2 : DI01 포인트 값이 1 이고 DI02 포인트 값이 1 이거나 DI03 포인트 값이 1 이고 DI04 포인트 값이 1 이면, DO02 포인트에 1 을 그외에는 0 을 씁니다.

- 로직 3 : DI05 포인트 값이 0 에서 1 로 바뀌거나(**↑**) DI05 포인트 값이 1 에서 0 로 바뀔(**↓**) 경우 VAI00 포인트의 값에 1 을 더하여 VAI00 에 넣습니다(DI05 값이 바뀔때 마다 VAI00 을 1 씩 증가시킵니다).

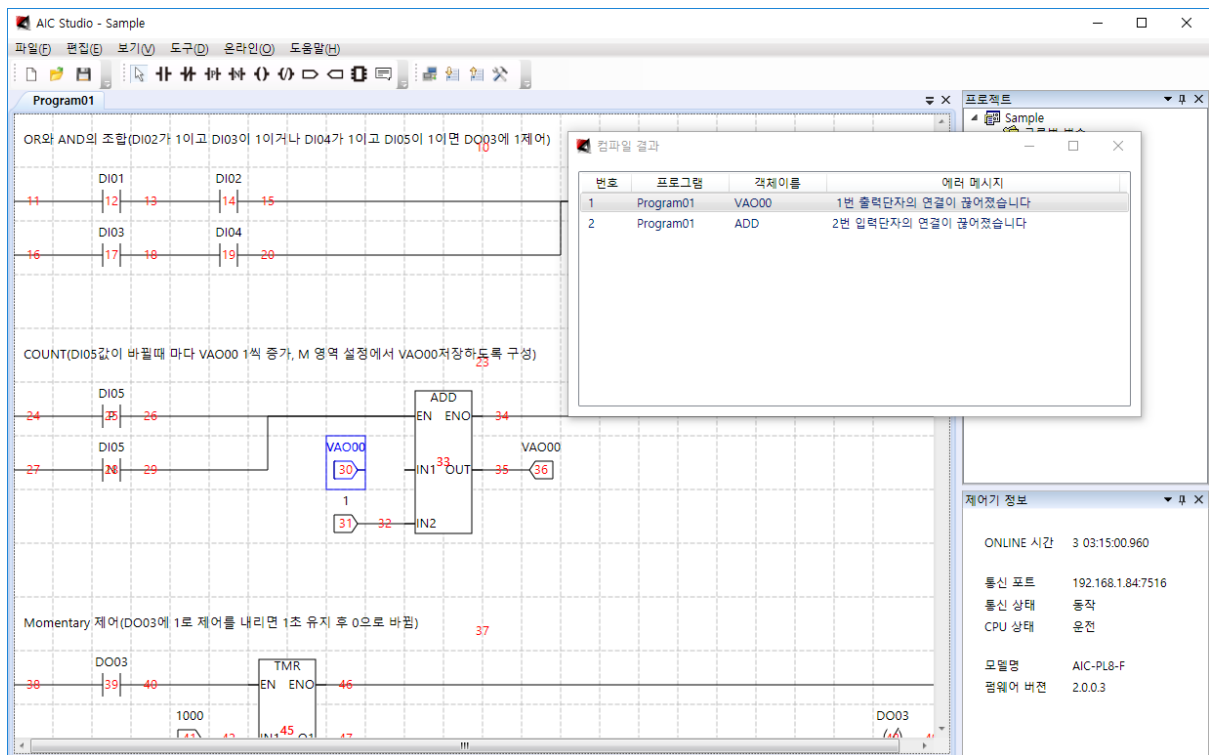
- 로직 4 : DO03 포인트 값을 1 로 제어 하면 1000ms 후 0 으로 씁니다(Monentry 제어). VAO00 에는 1000ms 가 될 때까지 경과시간을 씁니다.

2.3 컴파일

컴파일 메뉴를 선택하면 그림으로 그려진 로직을 해석하여 제어기에서 해석할 수 있는 기계어로 변환하여 파일로 저장한 후 다음과 같은 대화 상자를 띄워 완료를 알립니다.

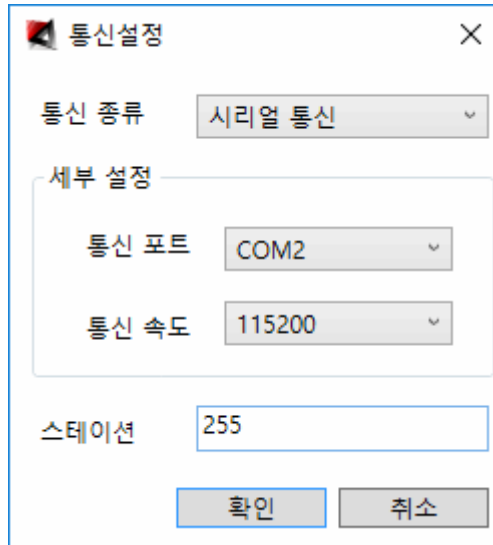


만약 작성한 로직에 문제가 있을 경우 다음과 같이 컴파일 결과 대화상자를 화면에 표시합니다. 컴파일 결과의 각 항목을 마우스로 더블 클릭하면, 해당 객체가 선택되며 화면 가운데 부분에 표시 됩니다.



2.4 통신 설정

온라인 메뉴의 부메뉴 통신 설정을 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.



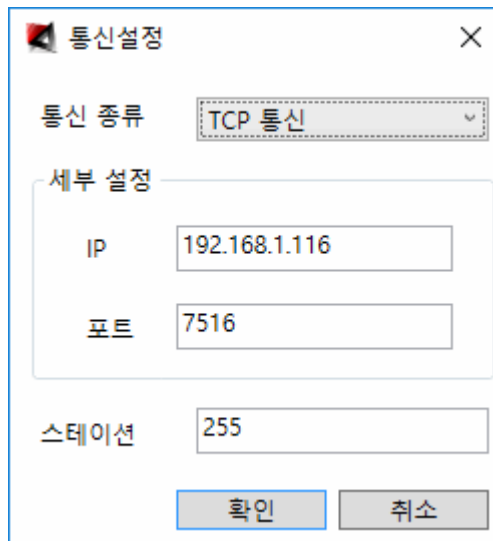
통신 종류 : 리얼 통신, TCP 통신, 시뮬레이터를 선택할 수 있습니다.

통신 포트 : 시리얼 통신 포트번호를 선택합니다.

통신 속도 : 통신 속도를 설정합니다.

스테이션 : 스테이션 번호를 입력합니다. 255 를 설정하는 것이 좋다(255 는 제어기의 번호를 어떤 번호를 설정하더라도 통신이 가능하게 하는 번호입니다. 255 외의 번호를 사용할 경우 제어기와 번호가 맞아야만 통신이 됩니다).

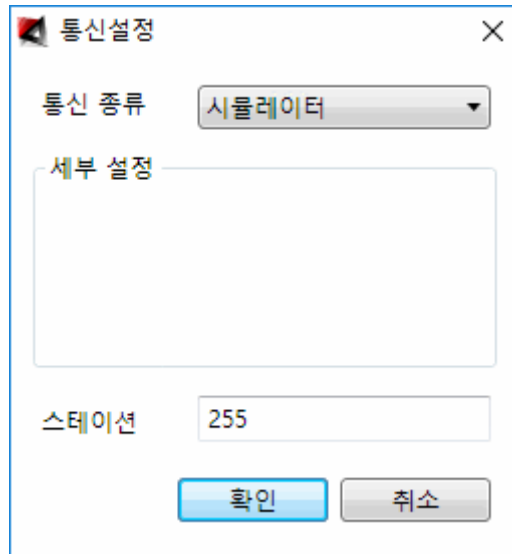
통신 종류를 TCP 로 변경하면 대화 상자는 다음과 같이 바뀝니다.



IP : IP 를 입력합니다. 만약 본 프로그램과 같이 제공되는 AIC_Simulator.exe(시뮬레이터)를 같은 컴퓨터에 실행하고 시뮬레이션을 하고자 합니다면, IP 를 127.0.0.1 로 입력하면 됩니다(시뮬레이터와 통신하면서 제어기가 있는 것처럼 모든 기능을 사용할 수 있습니다).

포트 : 통신 포트를 입력합니다. 특별한 경우가 아니면 수정하지 않는 것이 좋다. 수정하고자 하는 경우 경우 제어기의 포트를 먼저 바꾸고 이 포트를 바꾸어야 통신이 가능합니다.

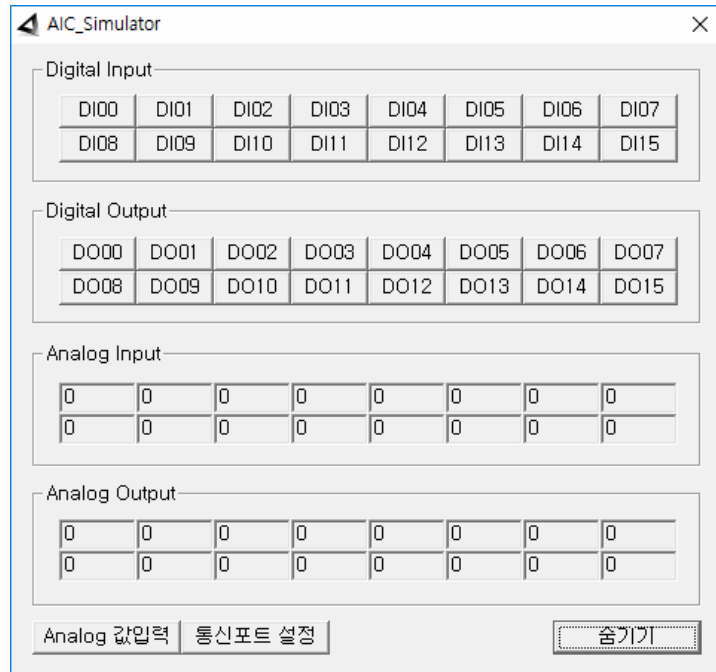
통신 종류를 시뮬레이터로 변경하면 대화 상자는 다음과 같이 바뀝니다.



시뮬레이터를 선택하고 확인을 실행하면 내부적으로 시뮬레이터가 실행되어 통신을 수행합니다.

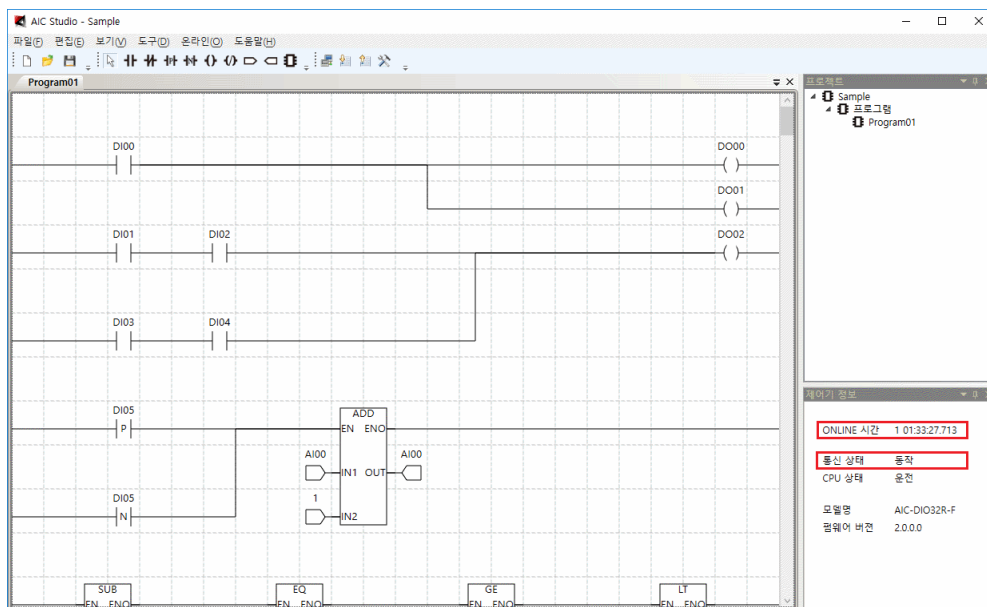
시뮬레이터는 제어기와 동일하게 동작하여 제어기의 모든 테스트를 PC 에서 수행이 가능하며, 테스트의 편의를 위해 상태값을 마음대로 수정이 가능하게 구성한 프로그램입니다.

시뮬레이터는 Tray 아이콘으로 실행되며 Tray 아이콘을 더블클릭하면 다음과 같이 화면에 나타납니다.



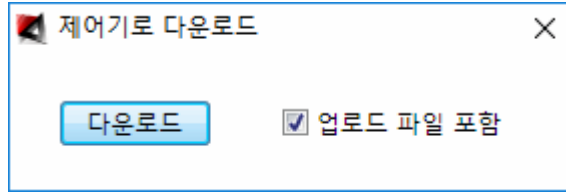
총 DI, DO, AI, AO 각각 16 포인트를 화면상에 표시하며 값을 관리할 수 있도록 구성 되어 있으며, Digital 값 변경은 원하는 포인터를 마우스로 클릭하면 되고, Analog 값 변경은 변경하고자 하는 포인터를 한번 마우스로 클릭한 후 Analog 값 입력 버튼을 실행하면 됩니다.

통신 설정을 제대로 하여 제어기 또는 시뮬레이터와 통신을 수행할 경우 우측하단의 제어기 정보창에 통신 상태가 동작으로 표시되며 ONLINE 시간이 계속 올라갑니다(ONLINE 시간은 제어기가 켜진 후 흐른 시간을 표시합니다).

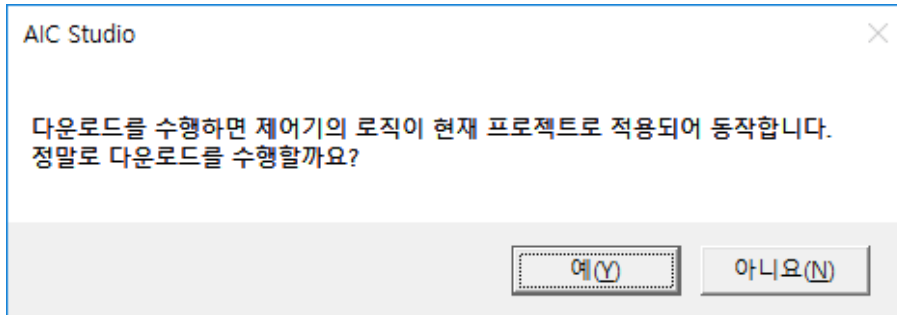


2.5 제어기로 다운로드

온라인 메뉴의 부메뉴 제어기로 다운로드를 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.

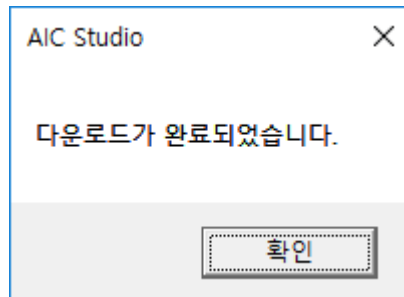


다운로드 : [컴파일](#)된 결과물을 제어기로 다운로드합니다. 다운로드 버튼을 선택하면 다음과 같은 대화 상자가 생성되어 다운로드할 것인지를 묻습니다.

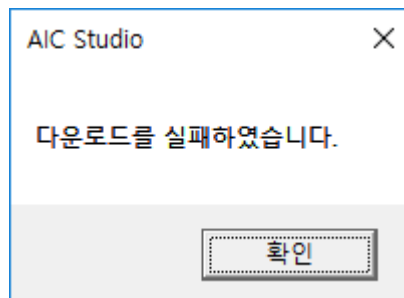


예를 선택하면 실제 다운로드가 수행이 됩니다.

다운로드가 정상적으로 수행이 되면 다음과 같은 대화상자가 생성되어 정상적으로 수행되었음을 알립니다.



만약 통신이상 등으로 다운로드가 실패하면 다음과 같은 대화상자가 생성되어 실패하였음을 알립니다.



업로드 파일 포함 : 업로드 파일 포함을 체크해야만 향후 제어기로부터 업로드 기능을 사용하여 업로드가 가능합니다. 관리를 위하여 가능하면 업로드 파일 포함을 체크해두기 바랍니다.

2.6 제어기 설정

온라인 메뉴의 부메뉴 제어기 설정을 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.

The screenshot shows a software window titled "제어기 설정" (Controller Settings). It features a tabbed interface with the following tabs: "기본설정" (Basic Settings), "Serial", "LAN", "아날로그 교정" (Analog Calibration), "네트워크 블럭" (Network Block), and "암호화" (Encryption). The "기본설정" tab is currently selected. The main content area includes a "스테이션" (Station) field containing the number "0". Below this is a "포인트 갯수" (Number of Points) section, which is a container for four input fields: "디지털 입력" (Digital Input) set to 32, "디지털 출력" (Digital Output) set to 32, "아날로그 입력" (Analog Input) set to 16, and "아날로그 출력" (Analog Output) set to 8. A "설정" (Apply) button is located at the bottom right of the dialog box.

스테이션 : 스테이션 번호(제어기 번호)를 입력합니다.

포인트 갯수 : 포인트 갯수를 입력합니다. 여기에서 입력하는 포인트 수 만큼 메모리가 할당되며, 제어기의 실 포인트 이상으로 설정된 포인트는 가상 포인트로 사용됩니다.

디지털 입력 : 디지털 입력 포인트의 갯수를 입력합니다.

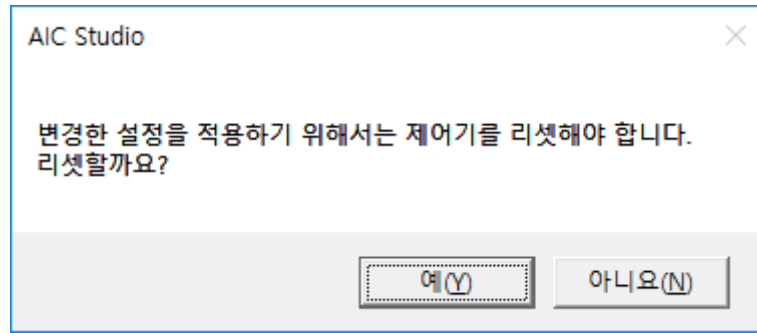
디지털 출력 : 디지털 출력 포인트의 갯수를 입력합니다.

아날로그 입력 : 아날로그 입력 포인트의 갯수를 입력합니다.

아날로그 출력 : 아날로그 출력 포인트의 갯수를 입력합니다.

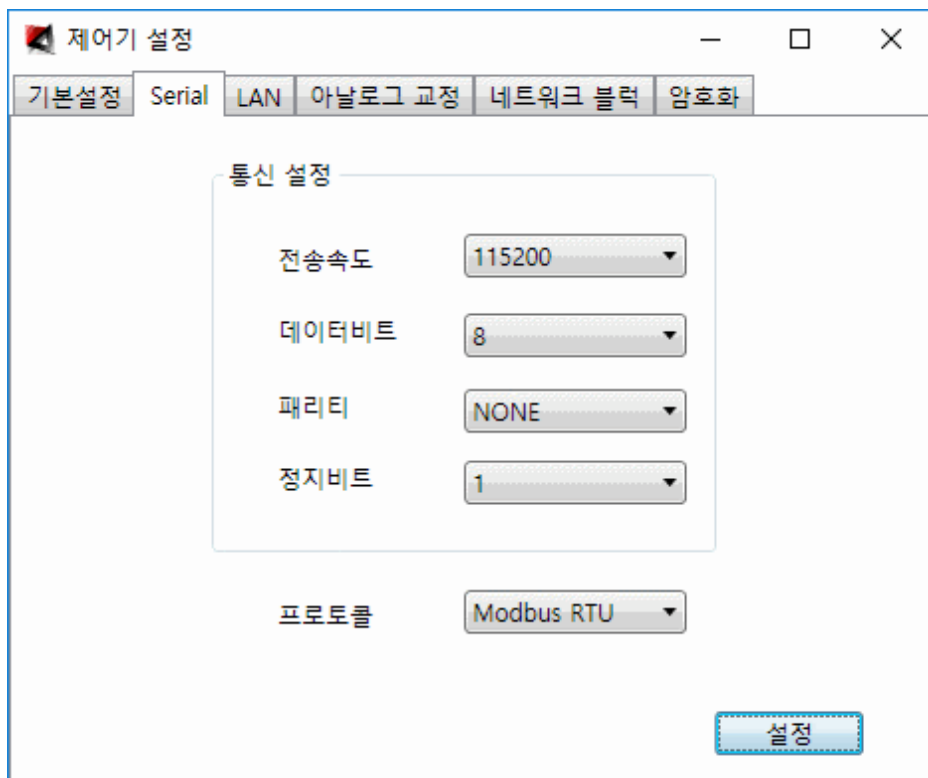
설정 : 설정한 데이터를 통신을 통해 제어기로 보내 제어기의 설정을 변경합니다.

설정을 하고 우측 상단의 **X** 를 선택하면 다음과 같이 제어기를 리셋할 것인지 묻는 대화 상자가 화면에 나타납니다.

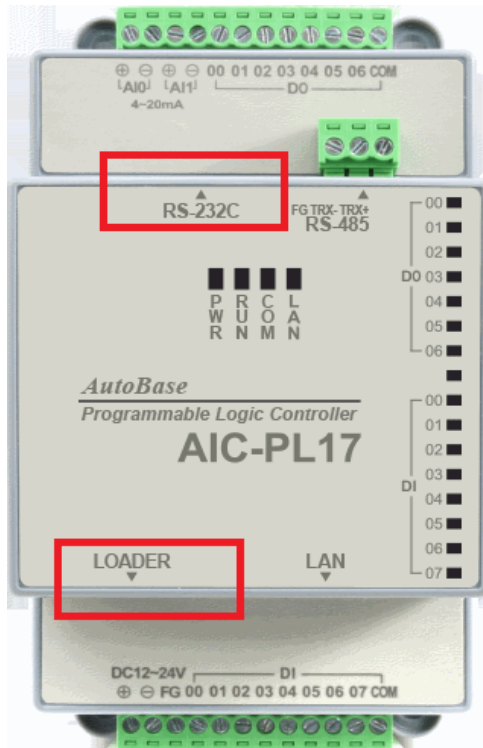


예를 선택하면 제어기에 적용이 됩니다.

Serial 탭을 선택하면 대화 상자는 다음과 같이 바뀝니다.



오토베이스에서 제작한 제어기는 아래 예시 사진과 같이 두 개의 Serial 포트가 있습니다. 그 중 사진 하단의 설정용 포트(LOADER)는 통신이 고정되어 있고(115200, 8, NONE, 1), 본 대화상자는 사진 상단의 통신용 포트(RS-232C)의 정보를 설정하기 위해 사용합니다.



전송속도 : 전송속도를 설정합니다.

데이터비트 : 데이터비트를 설정합니다. 8Bit 와 7Bit 설정이 가능합니다.

패리티 : 패리티를 를 설정합니다. NONE, EVEN, ODD 설정이 가능합니다.

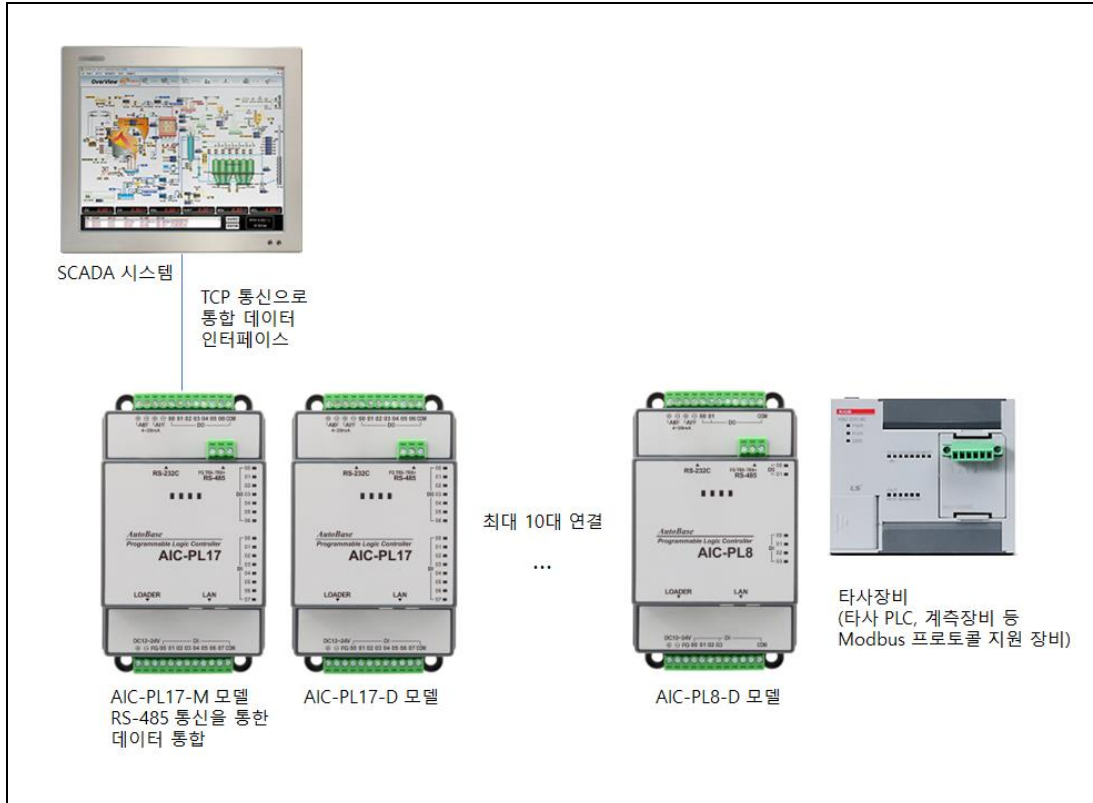
정지비트 : 정지비트를 설정합니다. 1Bit 와 2Bit 설정이 가능합니다.

프로토콜 : 통신 프로토콜을 설정합니다. AOB Auto, AOB ASCII, AOB Binary, Modbus RTU 설정이 가능합니다.

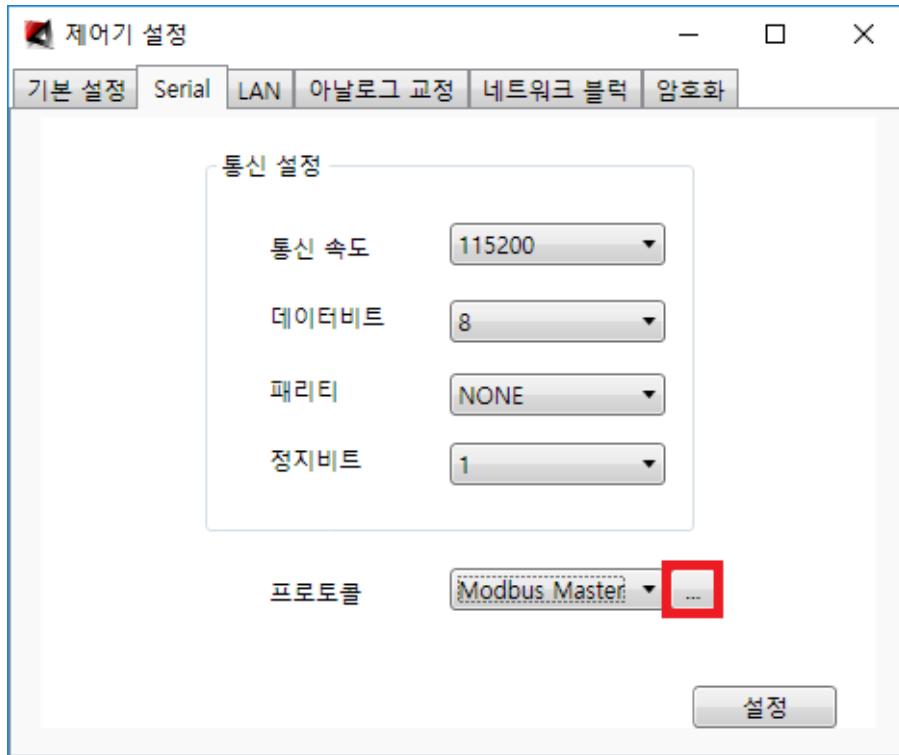
AIC-PL17-M 장비를 구매하여 연결하면 프로토콜에 Modbus Master 가 추가됩니다.

AIC-PL17-M 장비는 485 통신을 통해 타 장비 데이터를 통합하여 관리할 수 있는 장비입니다.

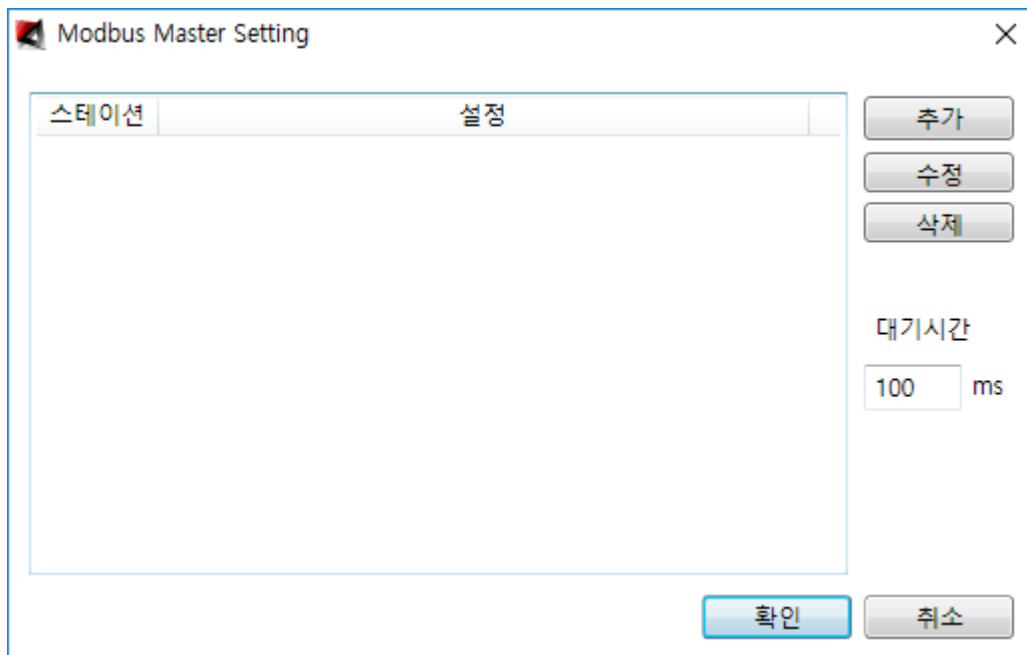
다음 그림은 AIC-PL17-M 을 사용하여 구성한 시스템 예입니다.



Modbus Master 를 선택하면 다음과 같이 버튼이 추가되어 화면에 나타납니다.

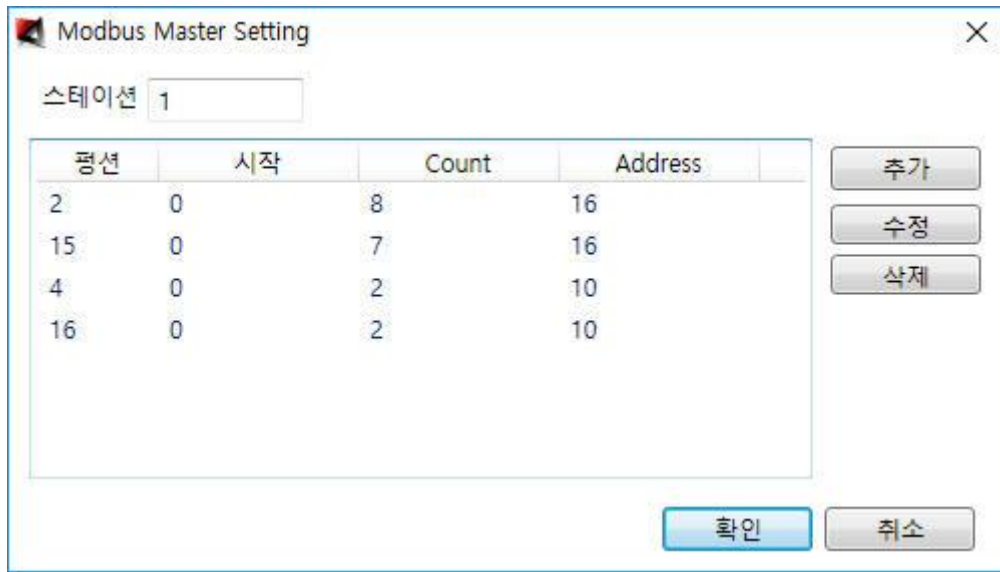


버튼을 선택하면 다음과 같이 화면에 나타납니다.



대기 시간 : 485 통신 수행 시 요청 데이터를 보내고 응답이 오지 않을 경우 대기하는 시간입니다.

추가를 선택하면 자사 장비인 AIC-PL Series 를 연결하기위한 기본 정보를 자동으로 추가하여 다음과 같이 화면에 나타납니다.



- 스테이션 : RS-485 통신을 통하여 데이터를 취득할 장비의 스테이션 번호를 입력합니다. 추가를 선택하면 다음과 같이 화면에 나타납니다.



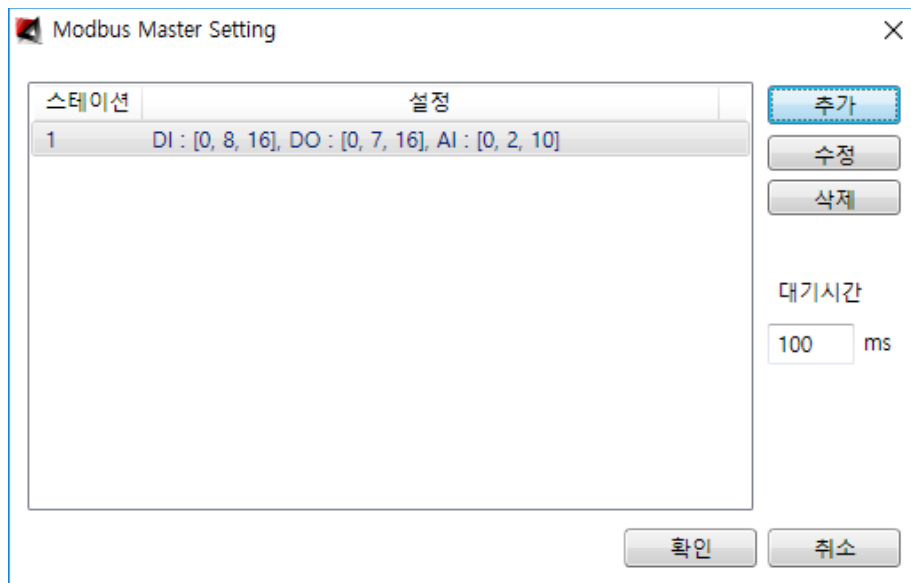
- 종류 : AIC-PL Master 장비로 데이터를 읽을지(Read), AIC-PL Master 장비의 데이터를 타장비로 보낼지(Write) 선택합니다.
 - 평선 : Modbus 평선 코드를 선택합니다. 종류가 Read 일 경우 1(DO 읽기), 2(DI 읽기), 3(AO 읽기), 4(AI 읽기)를 선택할 수 있으며, 종류가 Write 일 경우 15(DO 쓰기), 16(AO 쓰기)를 선택할 수 있습니다. 시작 : 요청시 사용할 시작주소를 입력합니다.
 크기 : 요청시 사용할 크기를 입력합니다.
 주소 : 데이터를 취득하여 통합 관리할 AIC-PL17-M 장비의 주소를 입력합니다. 평선 1 과 2 는 AIC-PL17M 장비의 DI 영역에 평선 3 과 4 는 AI 영역에 들어가고, 평선 15 는 DO 영역의

데이터를 평선 16은 AO 영역의 데이터를 전송하며 평선을 바꿀때 마다 해당 주소가 표시되므로 참고하면 됩니다.

주소는 AIC-PL17-M 장비에서 사용하는 포인트와 중복되지 않도록 입력해야만 합니다.

<그림 1>에서 자동으로 추가된 주소는 처음 추가 시 Master 장비와 충돌나지 않도록 설정하고 나면, 이후 부터는 이전에 입력한 주소를 참고하여 중복되지 않도록 자동으로 입력되어 나타나므로 편리하게 사용할 수 있습니다.

다음화면은 스테이션 하나를 추가한 화면입니다.



같은 방식으로 최대 10 대까지 추가가 가능합니다.

LAN 탭을 선택하면 대화 상자는 다음과 같이 바뀝니다.



IP 주소 : 제어기의 IP 주소를 입력합니다.

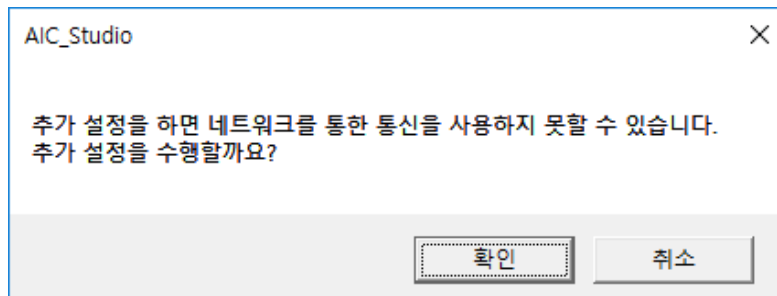
서브넷 마스크 : 제어기의 서브넷 마스크를 입력합니다.

게이트웨이 : 제어기의 게이트웨이 주소를 입력합니다.

포트 : TCP 통신을 수행할 포트를 입력합니다.

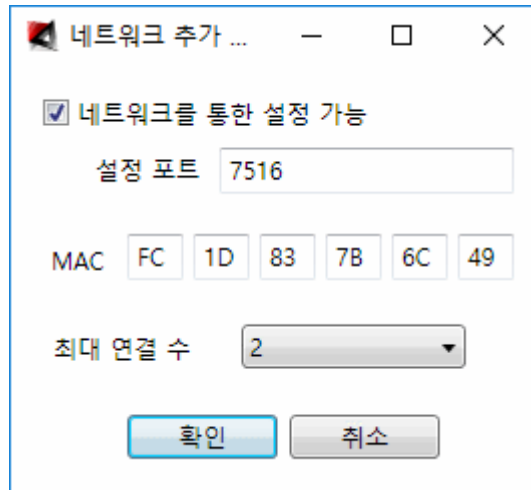
프로토콜 : 통신 프로토콜을 설정합니다. AOB Auto, AOB ASCII, AOB Binary, Modbus TCP 설정이 가능합니다.

추가설정 : 추가설정을 선택하면 다음과 같은 대화 상자가 생성되어 변경에 대한 경고를 합니다.



문구에서 경고한 바대로 네트워크를 통한 통신을 사용하지 못할 수 있으므로, 정확한 내용을 알기 전에는 설정하지 않길 바랍니다.

확인을 선택하면 다음과 같은 대화 상자가 생성됩니다.



네트워크를 통한 설정 가능 : 네트워크를 통한 설정을 할 것인지 설정합니다. 오토베이스에서 제작한 제어기의 설정을 위한 통신은 AES 암호화를 통해 안전하게 수행하나, 보안을 위해 근본적으로 네트워크를 통한 설정이 불가능하게 설정할 수 있는 기능입니다.

설정 포트 : 네트워크를 통한 설정 기능을 수행하기 위한 통신 포트를 설정합니다.

MAC : 제어기의 MAC 주소를 입력합니다.

최대 연결 수 : TCP 최대 연결 수를 설정합니다. 최대 지원 수는 2 개이며, 0 으로 설정하면 TCP 통신이 되지 않습니다.

아날로그 교정 탭을 선택하면 대화 상자는 다음과 같이 바뀝니다.



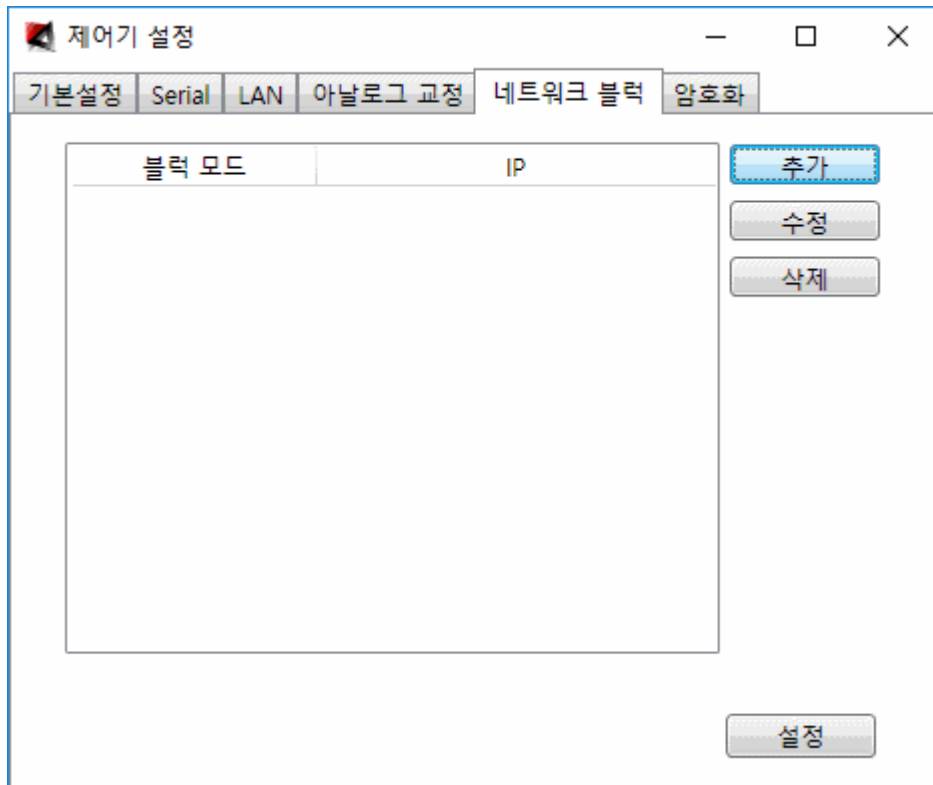
제어기의 아날로그 실 포인트 수만큼 교정할 수 있도록 인터페이스를 제공합니다. 오โต베이스에서 제공하는 제어기는 4mA 일때 0 값을 20mA 일때 4095 값을 전송하도록 정확히 교정되어 제공되므로 정확한 내용을 알기 전에는 손대지 않는 것이 좋습니다. 만약 연결하는 계측 장비가 0~20mA 장비일 경우 Calibrator(전류발생기)가 있으면 아래 내용을 이해하여 설정을 하면 되고, 그렇지 않으면 최소값을 0으로 설정하여 사용하면 대략적으로 0mA 일때 0 값을 20mA 일때 4095 값을 전송하게 됩니다.

구분 : 교정 정보를 사용할 것인지를 선택합니다. 체크를 제외하면 교정 없이 아날로그 값을 그대로 전송합니다. 체크를 빼면 대략적으로 0mA 일때 0 값을 24mA 일때 4095 값을 전송합니다.

최소 : 최소값 일때 보정값을 입력합니다. Calibrator 를 제어기의 해당 포인트에 연결하고 최소값(연결 할 계측 장비에 따라 다르나 일반적으로 4mA)을 맞춘 후, 입력 버튼을 선택하여 자동으로 입력되는 값을 설정합니다.

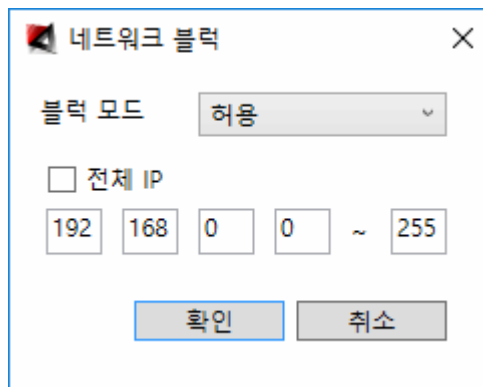
최대 : 최대값 일때 보정값을 입력합니다. Calibrator 를 제어기의 해당 포인트에 연결하고 최대값(연결 할 계측 장비에 따라 다르나 일반적으로 20mA)을 맞춘 후, 입력 버튼을 선택하여 자동으로 입력되는 값을 설정합니다.

네트워크 블록 탭을 선택하면 대화 상자는 다음과 같이 바뀝니다.



네트워크 블록은 제어기에 접속하는 시스템의 IP 로 그 접근을 제한하는 기능입니다.

추가 : 추가를 선택하면 다음과 같은 대화 상자가 생성됩니다.

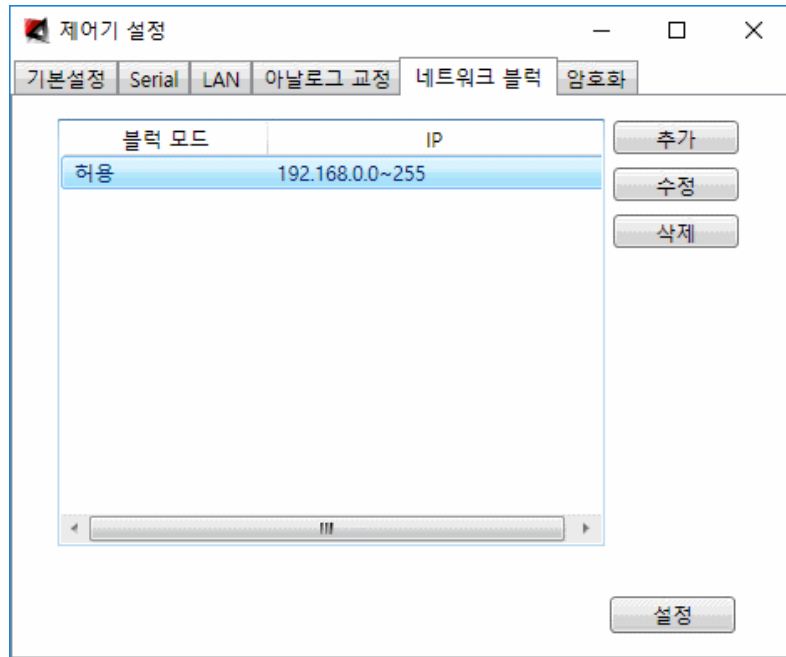


블록 모드 : 허용할 것인지 거부할 것인지를 선택합니다

전체 IP : 전체 IP 에 설정한 블록 모드를 적용할 것인지를 선택합니다.

IP : 전체 IP 가 설정되지 않았을 때 적용할 IP 의 범위를 지정합니다.

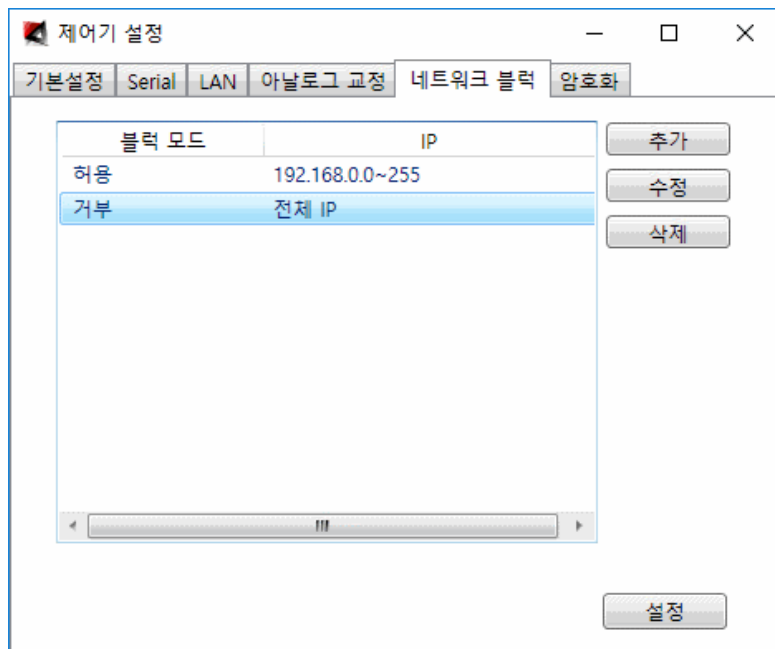
확인을 선택하면 다음과 같이 추가가 됩니다.



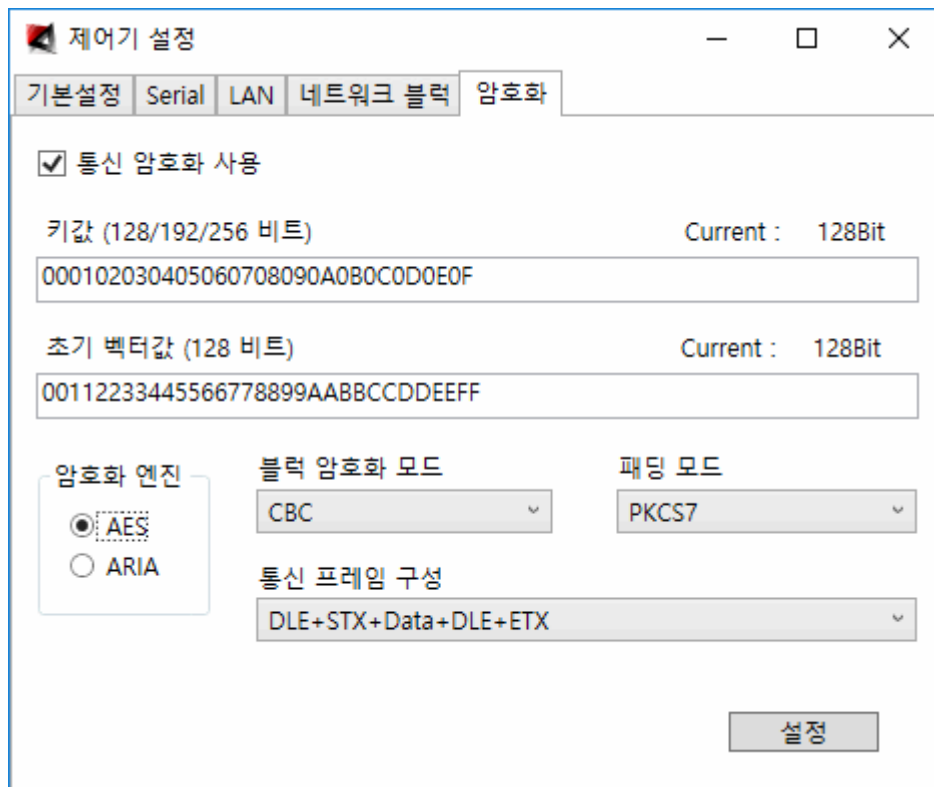
수정 : 수정을 선택하면 추가와 같은 대화 상자가 생성되어 선택한 정보를 수정할 수 있도록 합니다.

삭제 : 선택한 정보를 삭제합니다.

네트워크 블럭은 일반적으로 원하는 대역만 허용하고 나머지는 모두 차단하는 것이 일반적입니다. 네트워크 블럭은 리스트에 추가한 순서대로 체크하여 허용 거부를 결정하도록 되어 있고 전체를 다 체크해도 조건에 맞는 것이 없으면 허용하도록 되어 있습니다. 따라서 위의 설정만 하면 설정하지 않은 것과 동일하게 동작합니다. 일반적 설정을 하기 위해 아래와 같이 허용한 리스트 다음에 전체를 거부하게 설정하면 됩니다.



암호화 탭을 선택하면 대화 상자는 다음과 같이 바뀝니다.



제어기와 통신은 설정을 위한 통신과 감시 제어를 위한 통신이 있습니다. 여기서 설정하는 암호화는 감시 제어를 위한 통신에 적용이 되며, 설정을 위한 통신은 이 설정과 상관없이 AES 암호화 알고리즘으로 무조건 암호화하여 통신합니다.

통신 암호화 사용 : 통신 암호화를 사용할 것인지 설정합니다.

키 : 키값을 입력합니다. 키값은 0~F 의 값을 입력하며 1 문자당 4Bit 로 계산하여 128(32 문자), 192(48 문자), 258(64 문자) Bit 를 입력할 수 있습니다(계산하기 어려우므로 우측 Current 에 계산하여 Bit 수를 표시합니다).

초기 벡터값 : 초기 벡터값을 입력합니다. 초기 벡터값은 0~F 의 값을 입력하며 128(32 문자) Bit 를 입력할 수 있습니다.

암호화 엔진 : AES(미국 표준 기술 연구소(NIST)에 의해 제정된 암호화 방식), ARIA(인터넷진흥원과 국정원에서 함께 만든 알고리즘으로 국가표준(KS))를 선택할 수 있습니다.

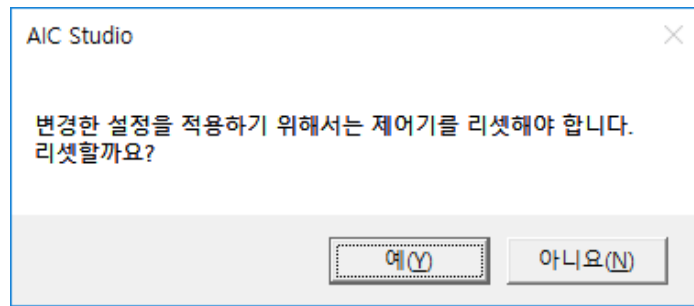
블럭 암호화 모드 : 암호화 시 어떤 방식으로 연산할 것인지를 선택합니다. CBC, ECB, OFB, CFB, CTS 를 선택할 수 있습니다.

패딩 모드 : 암호화는 16BYTE 단위로 수행합니다. 전송하는 데이터가 16BYTE 에 맞지 않을 경우 모자라는 데이터를 어떻게 채워 넣을 것인지를 선택합니다. PKCS7, ZERO, ANSIX923, ISO10126 을 선택할 수 있습니다.

통신 프레임 구성 : 통신 프레임 구성을 선택합니다.


DLE+STX+Data+DLE+ETX : DLE(0x10) + STX(0x03) + 암호화된 데이터(데이터에 DLE 가 있을 경우 DLE 를 한번더 넣어 전송) + DLE + ETX(0x02)의 형태로 전송합니다.

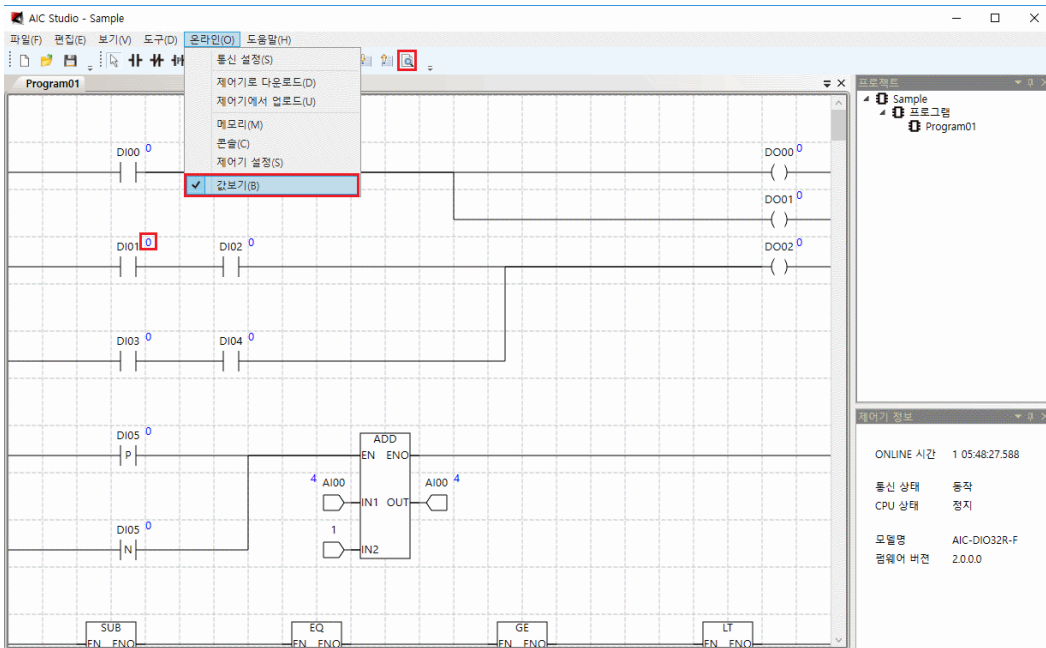
설정을 하고 우측 상단의 X 를 선택하면 다음과 같이 제어기를 리셋할것인지 묻는 대화 상자가 화면에 나타납니다.





예를 선택하면 제어기에 적용이 됩니다.

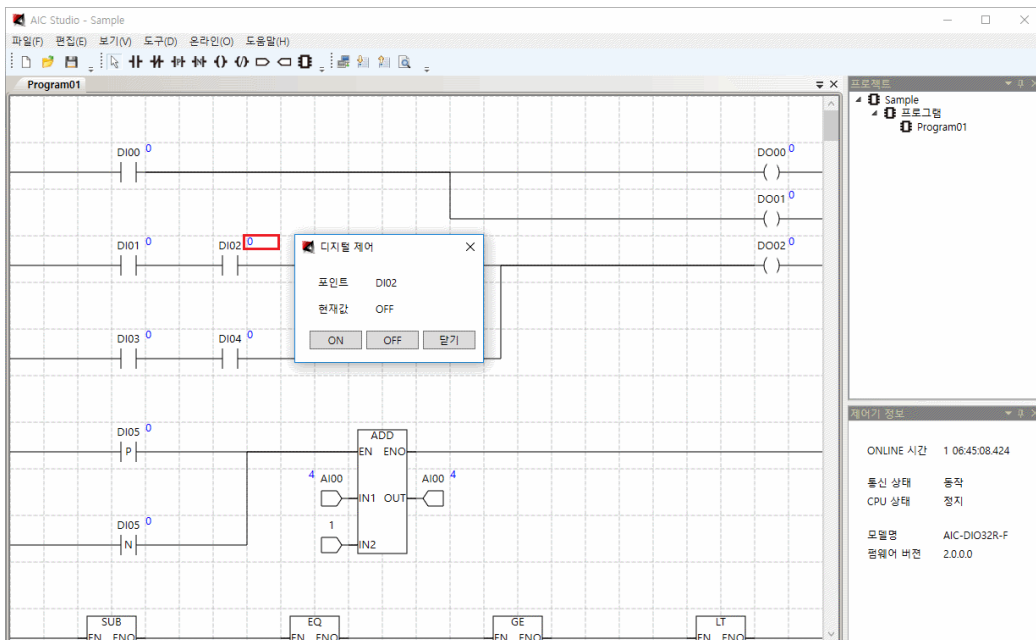
2.7 값보기(디버깅)

온라인 메뉴의 부메뉴 값 보기를 선택하거나 툴바 메뉴의  을 선택하면 다음과 같이 값 보기 모드로 변경됩니다.



값보기 모드가 되면 온라인 메뉴의 부메뉴 값보기에 체크표시가 되고 툴바의  아이콘은  아이콘으로 변경되며, 각 심볼 옆에 값이 표시됩니다. 심볼 옆의 값은 각 심볼에 정의한 포인트의 현재 값으로 직관적으로 바로 감시가 가능합니다.

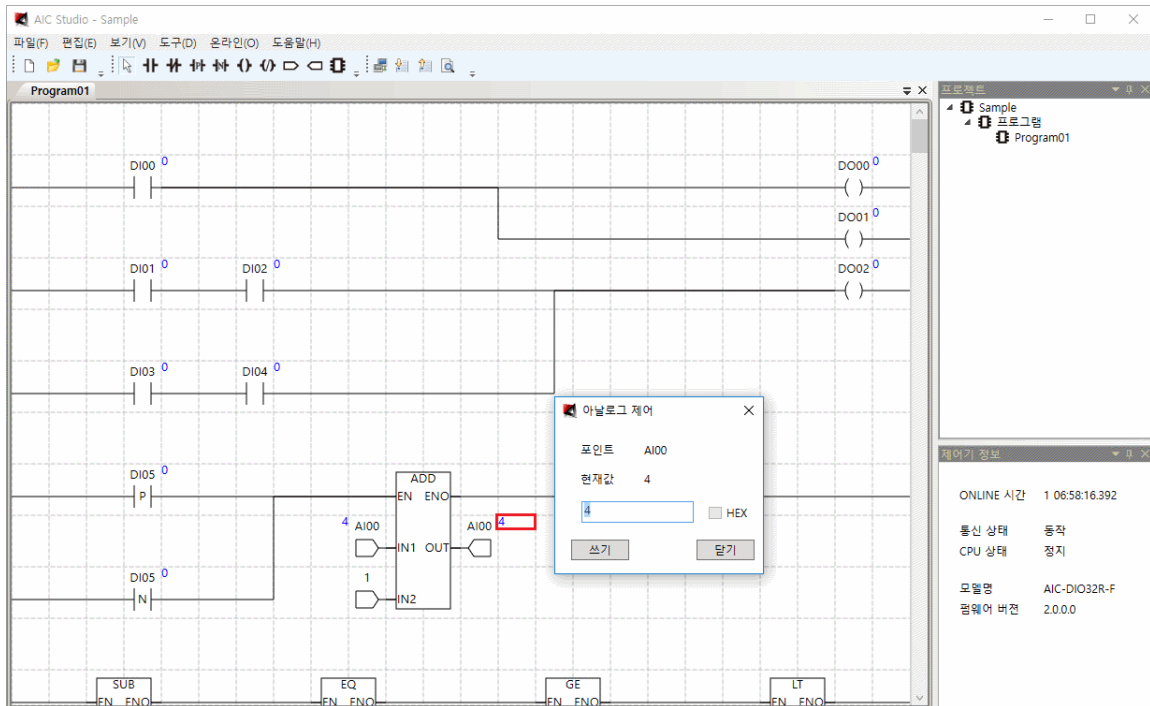
값보기 모드에서 심볼 옆의 값에서 마우스 왼쪽 버튼을 선택하면 다음과 같이 해당 포인트의 값을 제어할 수 있는 대화 상자가 생성됩니다.



제어 포인트가 디지털 입력인 경우 실 포인트는 제어가 되지 않으며(현장의 실 포인트가 바뀌면 그에 따라 바뀌게 됩니다), 가상포인트는 제어가 수행되어 값이 바뀌게 됩니다(시뮬레이터에서는 모든 값이 제어가 됩니다).

제어 포인트가 디지털 출력인 경우 실 포인트는 현장으로 실제 제어가 나가며, 가상포인트는 제어가 수행되어 값이 바뀌게 됩니다.

제어하는 값이 아날로그 포인트일 경우 다음과 같은 대화 상자가 화면에 나타납니다.



제어 포인트가 아날로그 입력인 경우 실 포인트는 제어가 되지 않으며(현장의 실 포인트가 바뀌면 그에 따라 바뀌게 됩니다), 가상포인트는 제어가 수행되어 값이 바뀌게 됩니다(시뮬레이터에서는 모든 값이 제어가 됩니다).

제어 포인트가 아날로그 출력인 경우 실 포인트는 현장으로 실제 제어가 나가며, 가상포인트는 제어가 수행되어 값이 바뀌게 됩니다.

값보기 기능을 이용하여 로직이 제대로 동작하는지 직관적으로 디버깅을 하면 됩니다.

3. 메뉴별 도움말

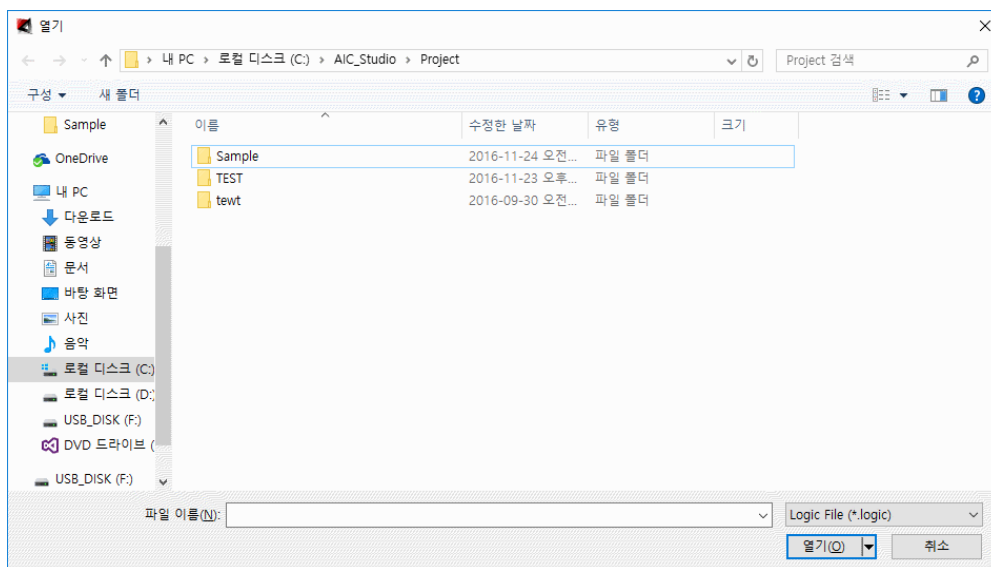
3.1 파일 메뉴

3.1.1 새파일 메뉴

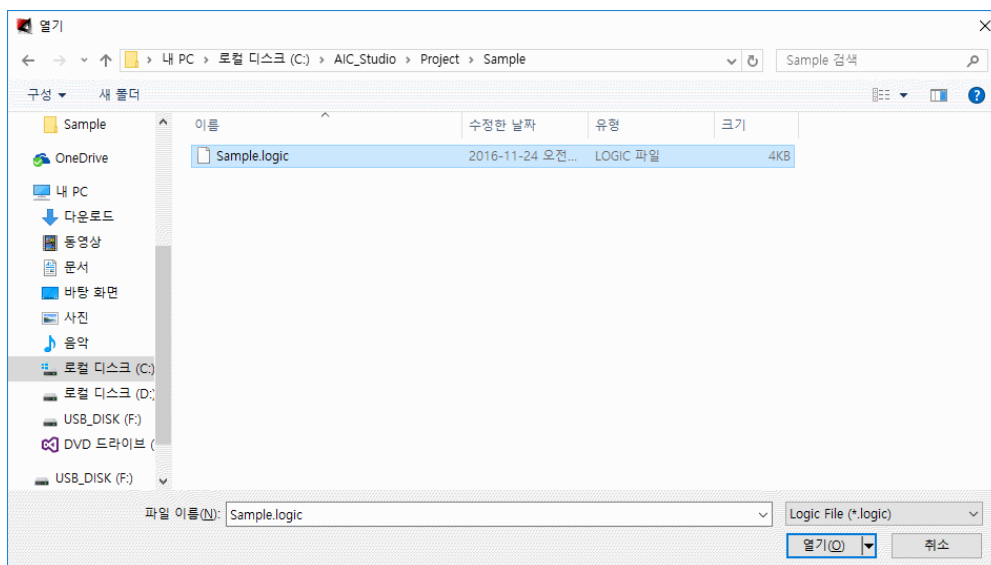
새 파일을 생성할 때 사용합니다. 사용법은 2.1 새파일을 참조하기 바랍니다.

3.1.2 열기 메뉴

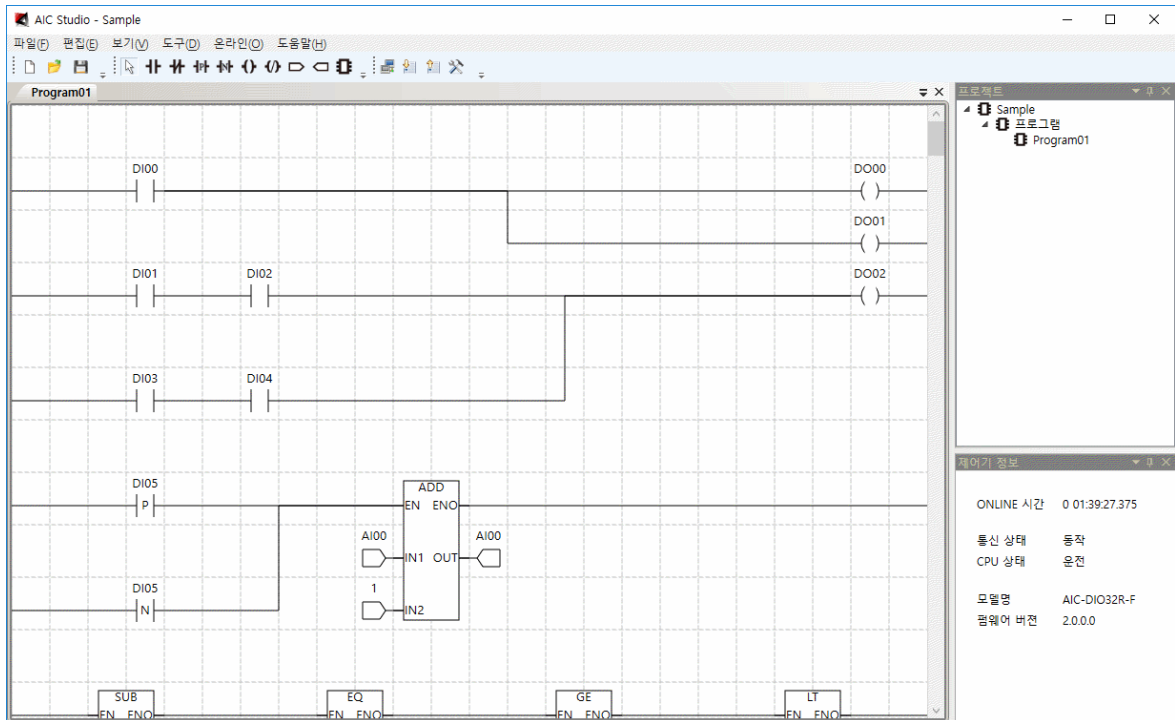
열기 메뉴를 선택하면 다음과 같은 대화 상자가 생성됩니다.



열고자 하는 프로젝트 폴더를 더블클릭하면 다음과 같이 대화 상자가 바뀝니다(여기서는 Sample 폴더 더블클릭).



프로젝트 파일을 선택하고 열기를 선택하면 다음과 같이 선택한 프로젝트 파일이 열려 작성된 로직을 화면에 표시합니다.

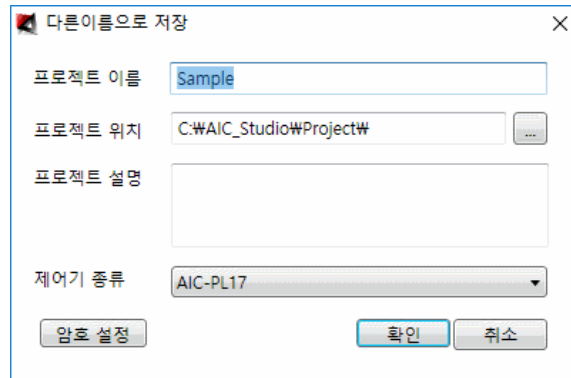


3.1.3 저장 메뉴

파일로 저장할 때 사용합니다. 저장을 수행하면 이미 저장할 환경이 모두 설정되어 있으므로 바로 저장이 됩니다.

3.1.4 다른이름으로 저장 메뉴

다른이름으로 저장을 선택하면 다음과 같은 대화 상자가 생성됩니다.



프로젝트를 복사하여 다른이름으로 저장하거나 암호 설정을 할 경우 사용합니다.

3.1.5 닫기 메뉴

편집중인 화면을 닫기 위해 사용합니다.

3.1.6 모두 닫기 메뉴

모든 화면을 닫기 위해 사용합니다.

3.1.6 종료 메뉴

프로그램을 종료하기 위해 사용합니다.

3.2 편집 메뉴

3.2.1 실행 취소 메뉴

바로 전에 편집한 상태로 돌아가기 위해 사용됩니다.

3.2.2 다시 실행 메뉴

실행 취소를 한 경우 다시 원래대로 돌아가기 위해 사용됩니다.

3.2.3 복사 메뉴

선택한 객체들을 붙여 넣기를 할 수 있도록 메모리에 저장하기 위해 사용됩니다.

3.2.4 잘라내기 메뉴

선택한 객체들을 붙여 넣기를 할 수 있도록 메모리에 저장하고 화면에서 삭제하기 위해 사용됩니다.

3.2.5 붙여넣기 메뉴

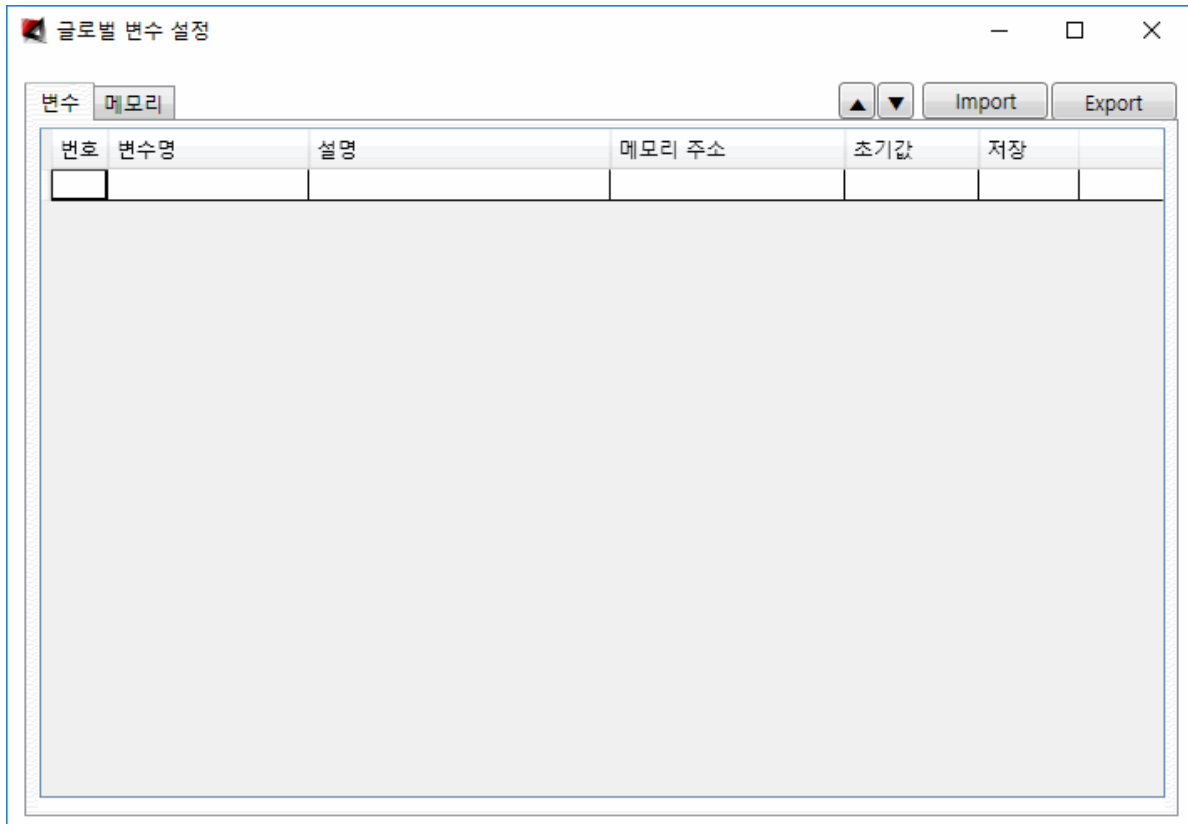
복사하거나 잘라내기를 하여 메모리에 저장된 객체들을 화면에 붙여 넣기를 하기 위해 사용됩니다.

3.2.6 지우기 메뉴

선택한 객체들을 화면에서 지우기 위해 사용됩니다.

3.2.7 글로벌 변수 설정 메뉴

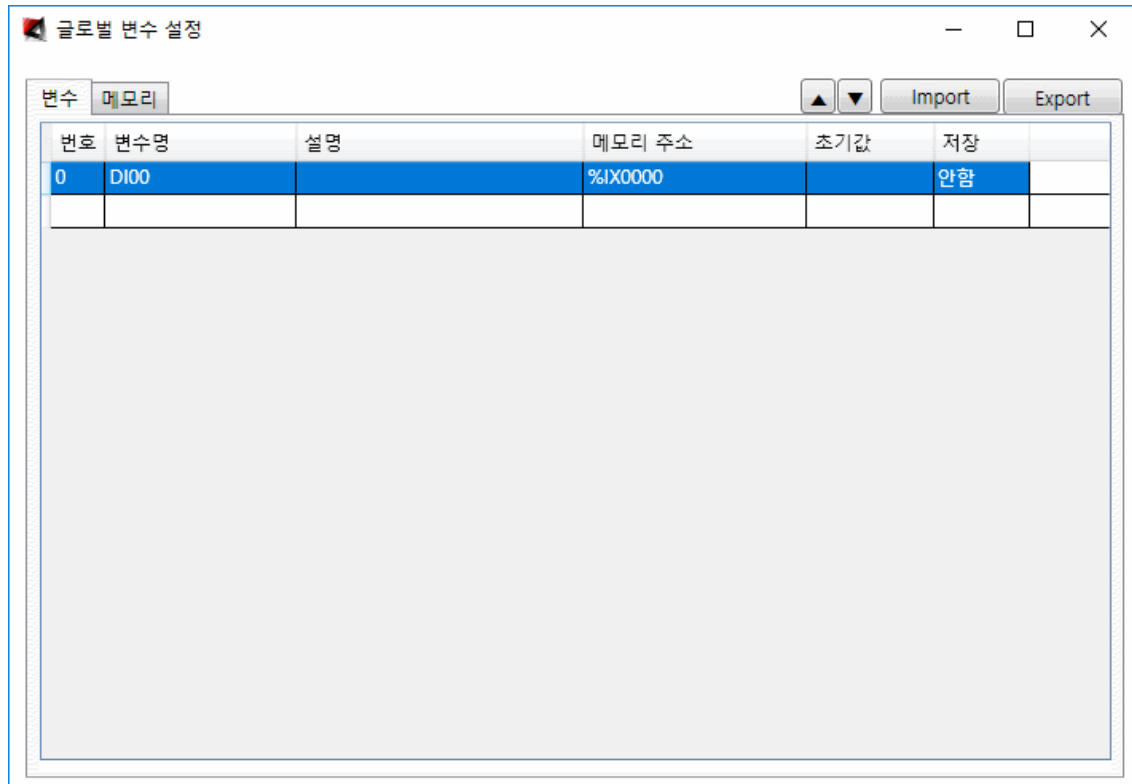
글로벌 변수 설정을 선택하면 다음과 같은 대화 상자가 생성됩니다.



글로벌 변수 설정은 제어기 내부적으로 가지고 있는 메모리 데이터를 변수로 설정하여 제어로직 작성시 그 변수를 이용하여 해당 메모리를 사용 하도록 합니다.

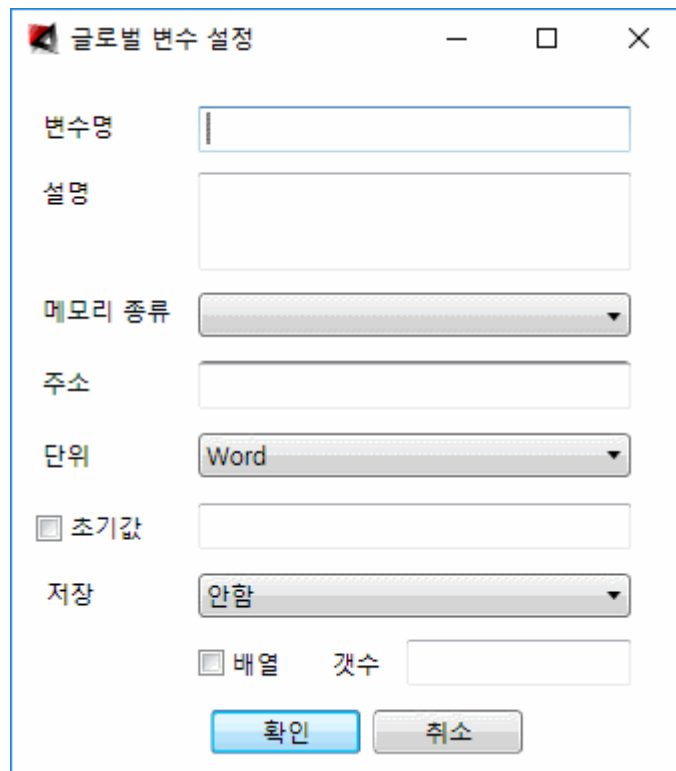
변수명과 메모리주소를 입력하면 번호가 자동으로 매겨지며 추가가 됩니다.

다음은 변수명에 DI00 을 메모리 주소에 %IX0000 을 입력하여 추가한 모양입니다.



메모리 주소를 직접 입력하기 위해서는 그 의미를 정확히 알아야 합니다. 따라서 대화 상자를 통해 입력할 수 있는 다른 방법도 제공합니다.

추가 : 제일 아래 공백으로 된 한줄에서 마우스로 더블클릭을 하면 다음과 같은 대화 상자가 생성됩니다.



변수명 : 변수명을 입력합니다. 일반적으로 변수명은 변수 이름만 보아도 그 포인트의 의미를 알 수 있도록 설정하며 로컬 변수([3.2.9 로컬 변수 설정](#)에서 설명)와 구분하기 위하여 대문자로 설정하는 것을 권장합니다.

설명 : 설명을 입력합니다.

메모리 종류 : 설정가능한 메모리 종류는 다음과 같습니다.

I : 디지털 입력 포인트의 값을 나타냅니다.

Q : 디지털 출력 포인트의 값을 나타냅니다.

AI : 아날로그 입력 포인트의 값을 나타냅니다.

AO : 아날로그 출력 포인트의 값을 나타냅니다.

SYSTEM : 시스템 변수의 값을 나타냅니다. 설정 가능한 System 변수는 다음과 같습니다.

변수	설명
Uptime0	가동시간 0 의 값을 표시합니다.
Uptime1	가동시간 1 의 값을 표시합니다.
Uptime2	가동시간 2 의 값을 표시합니다.
Uptime3	가동시간 3 의 값을 표시합니다.
TCPCount	TCP 연결수를 표시합니다.
CPUStatus	CPU 상태를 표시합니다.
DebugChannel	Debugging 중인 채널을 표시합니다(0 : 디버깅 안함, 1 : Serial 설정, 2 : Serial 통신, 3: LAN 설정, 4 : LAN 통신).
Year	현재 시각 년도를 표시합니다.
Month	현재 시각 월을 표시합니다.
Day	현재 시각 일을 표시합니다.
Hour	현재 시각 시를 표시합니다.
Minute	현재 시각 분을 표시합니다.
Second	현재 시각 초를 표시합니다.
SyncYear	시각 동기를 위한 년도를 표시합니다.
SyncMonth	시각 동기를 위한 월을 표시합니다.
SyncDay	시각 동기를 위한 일을 표시합니다.

SyncHour	시각 동기를 위한 시를 표시합니다.
SyncMinute	시각 동기를 위한 분을 표시합니다.
SyncSecond	시각 동기를 위한 초을 표시합니다.
TimeSync	1 로 제어하면 설정한 시각으로 동기를 수행합니다.
TimeSynced	제어 동작 후 1 회라도 시각동기가 되면 1 이됩니다.
Reset	1 로 제어하면 제어를 리셋합니다.
DiCount	디지털 입력 실 포인트 수를 표시합니다.
DoCoun	디지털 출력 실 포인트 수를 표시합니다.
AiCount	아날로그 입력 실 포인트 수를 표시합니다.
AoCount	아날로그 출력 실 포인트 수를 표시합니다.

M : M 영역 메모리의 값을 나타냅니다. M 영역 메모리의 의미는 [3.2.8 M 영역 메모리 설정](#)에서 설명합니다.

주소 : 주소를 설정합니다. 프로젝트 생성시 설정한 제어기 종류에 따라 설정가능한 주소는 달라집니다.

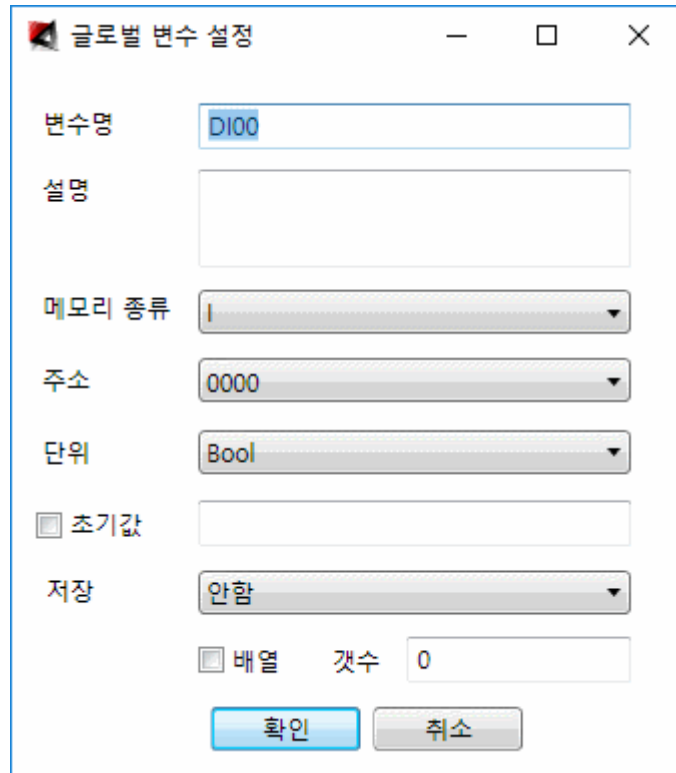
단위 : 단위를 설정합니다. 설정가능한 단위는 Bool, Byte, Word, Short, DWord, Int, Float 가 있습니다.

초기값 : 초기값을 입력합니다. 체크박스를 체크하면 초기값을 입력할 수 있습니다. 만약 배열로 설정한 경우 각 초기값의 사이에 ,나 ;를 넣어 배열 개수와 동일하게 입력하면 됩니다.

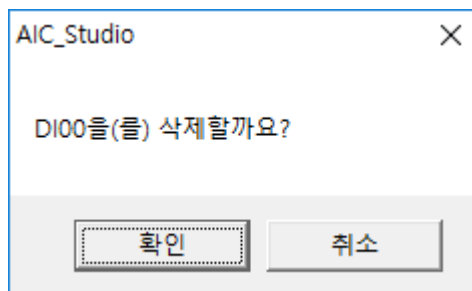
저장 : 저장을 설정하면 그 값이 바뀌었을때 Flash 메모리에 저장하여 재 부팅 시에도 그 값을 유지하도록 합니다. Flash 메모리는 10 만회 쓰기까지 보장하고 있습니다. 따라서 꼭 필요한 데이터만 저장해야 하며, 자주 변하는 아날로그 값을 저장하도록 설정하면 제어기의 수명이 현격히 떨어지는 결과를 초래하므로 주의가 필요합니다. 설정 종류는 안함, 저장, 즉시 저장이 있으며 저장을 설정하면 값이 변하더라도 저장 주기(60 초 간격)가 되면 저장을 수행하고, 즉시저장을 설정하면 저장 주기와 상관없이 바로 저장합니다.

배열 : 배열을 체크하면 변수는 입력하는 갯수 만큼의 배열로 사용이 됩니다.

수정 : 설정한 위치에서 마우스로 더블클릭하며 다음과 같이 선택한 글로벌 변수에 대한 수정 대화 상자가 화면에 나타납니다. 참고로 제어로직에 이미 수정하는 변수가 사용된 변수에 대해 수정을 하면 자동으로 찾아 변경하도록 되어있습니다.



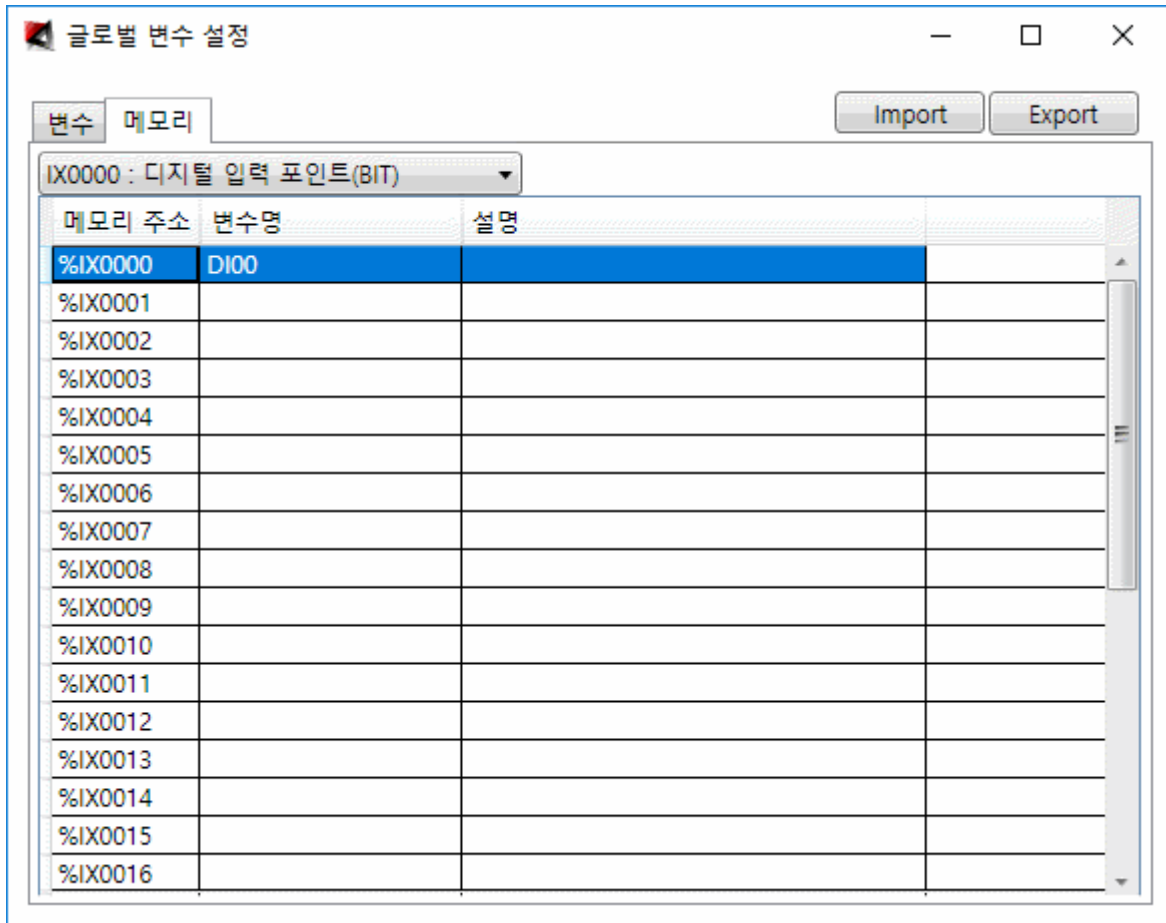
삭제 : 삭제하고자하는 항목을 선택(다중 선택 가능)하고 Delete 키를 누르면 다음과 같은 대화 상자가 생성되어 삭제할 것인지를 확인합니다.



확인을 선택하면 선택한 항목이 삭제됩니다.

메모리탭을 선택하면 다음과 같이 화면이 바뀝니다.

메모리탭은 같은 내용을 메모리를 기준으로 변수를 입력할 수 있는 인터페이스를 제공합니다.



DI00 은 변수탭 설명하며 입력한 글로벌 변수로 동일하게 표시됩니다.

메모리 종류를 선택하면 해당메모리의 전체 주소가 나열이 되고 할당하고자 하는 메모리 위치에 변수명과 설명을 입력하면 됩니다(사용법이 쉬워 일반적으로 이 인터페이스를 이용하여 입력).

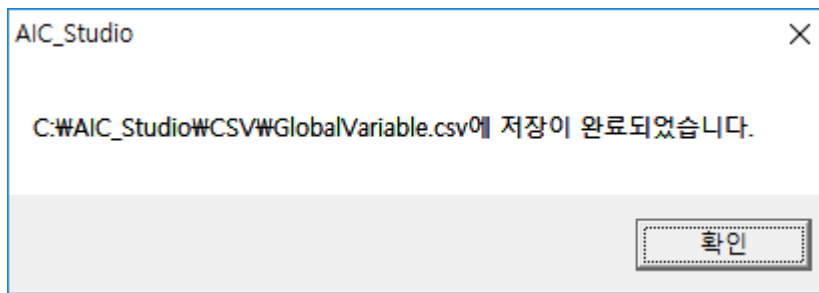
.선택할 수 있는 메모리 종류는 다음과 같습니다.

메모리 종류	구분	단위	비고
IX0000	디지털 입력 포인트	BIT	일반적으로 사용
IW0000	디지털 입력 포인트	WORD	16BIT 의 데이터를 한꺼번에 처리할 때 사용
QX0000	디지털 출력 포인트	BIT	일반적으로 사용
QW0000	디지털 출력 포인트	WORD	16BIT 의 데이터를 한꺼번에 처리할 때 사용
AI0000	아날로그 입력 포인트	WORD	
AO0000	아날로그 출력 포인트	WORD	
SYSTEM	시스템 데이터	WORD	

MX0000	M 영역 메모리	BIT	
MW0000	M 영역 메모리	WORD	
MD0000	M 영역 메모리	DWORD	
MF0000	M 영역 메모리	FLOAT	

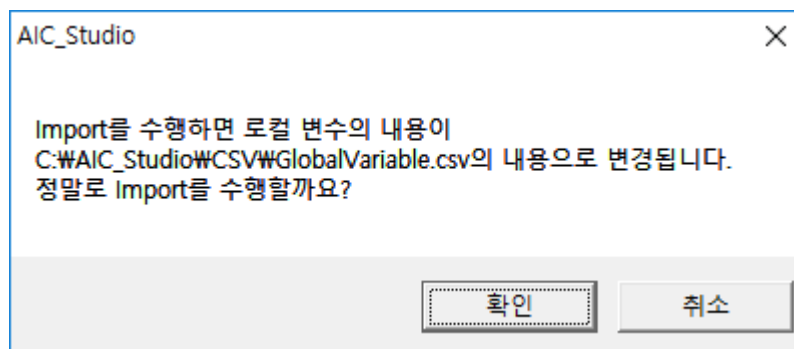
변수명을 입력한 후 설명까지 입력하고자 한다면 Tab 키를 눌러 설명을 입력하고, 변수명만 입력하고자 한다면 Enter 키를 눌러 다음 변수명을 입력하면 됩니다.

Export : 글로벌 변수 설정에서 입력한 내용을 .csv 파일로 저장합니다. 저장 위치는 C:\AIC_Studio\CSV\GlobalVariable.csv 에 저장되며 Export 를 실행하면 다음과 같은 대화 상자가 생성되어 저장 위치를 알려줍니다.



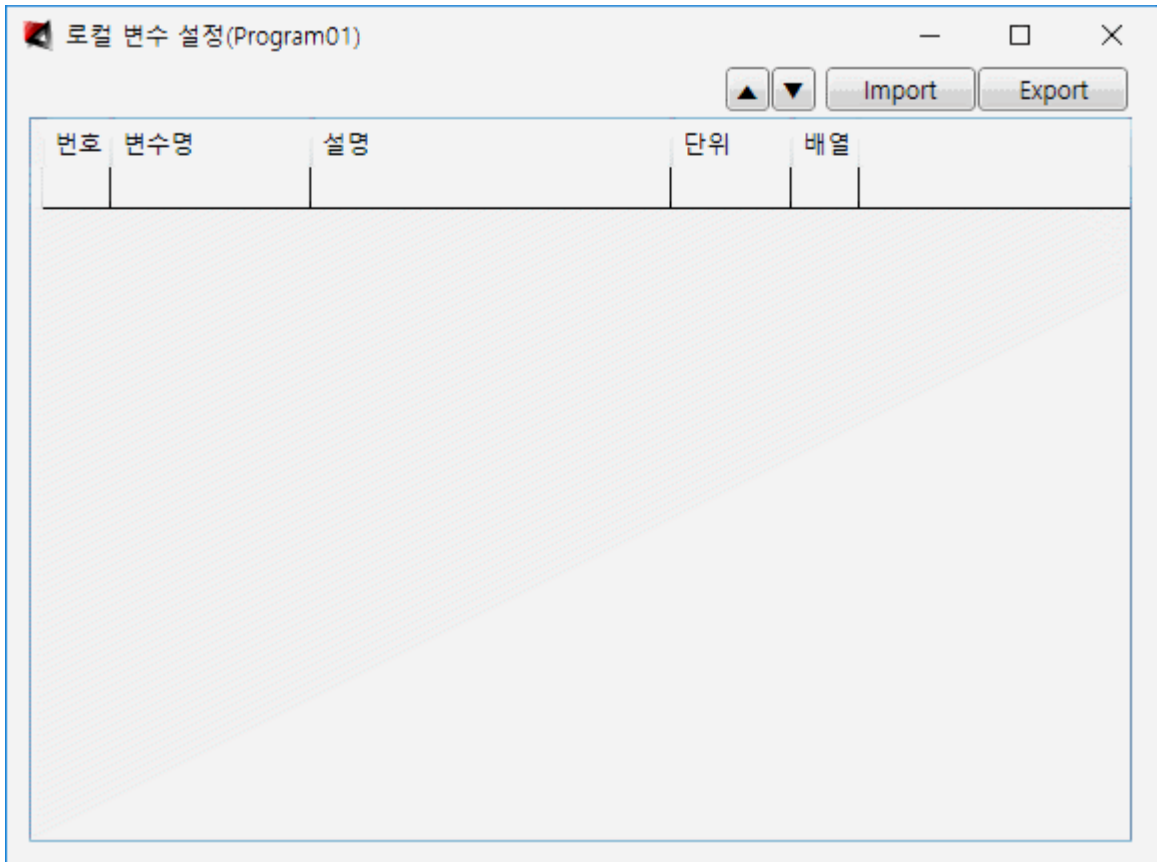
Import : C:\AIC_Studio\CSV\GlobalVariable.csv 로부터 데이터를 읽어 적용합니다. 일반적으로 Export 를 이용하여 저장한 파일을 Excel 등을 이용하여 편집한 후 Import 를 이용하여 적용하는 방식으로 일괄작업을 수행합니다.

Import 기능은 글로벌 변수 전체가 변경되므로 실행하게 되면 다음과 같은 대화 상자를 띄워 경고를 합니다.



3.2.8 로컬 변수 설정 메뉴

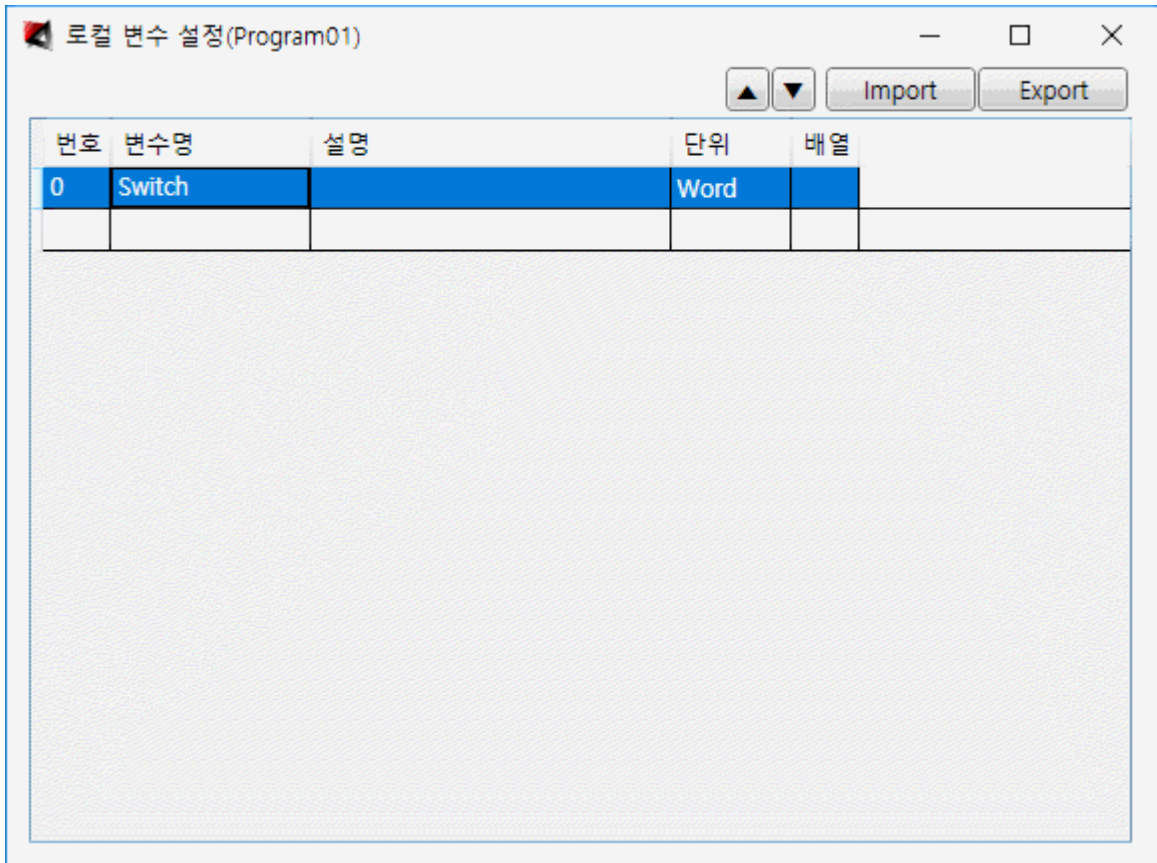
로컬 변수 설정을 선택하면 다음과 같은 대화 상자가 생성됩니다.



로컬 변수 설정은 제어기의 로직 Program 단위로 데이터를 저장할 필요가 있을 때 사용합니다.

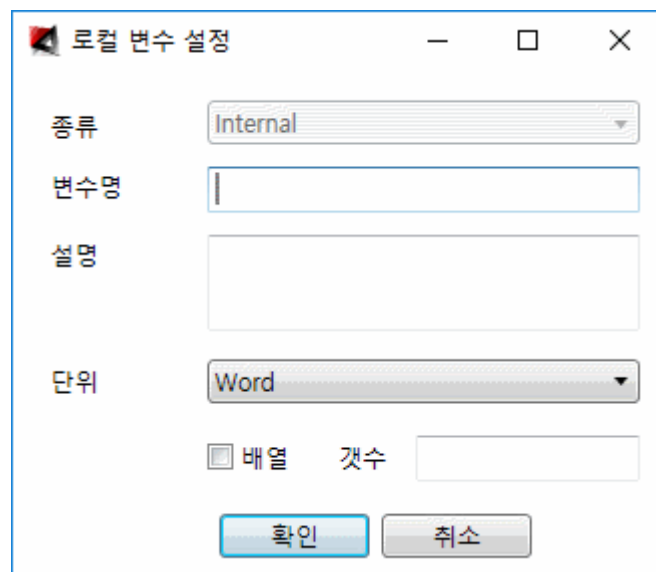
변수명만 번호가 자동으로 매겨지며 추가가 됩니다(단위는 마지막에 추가한 단위로 자동 설정).

다음은 변수명에 Switch 를 입력하여 추가한 모양입니다.



로컬 변수 설정에서도 글로벌 변수 설정과 같이 대화 상자를 통해 입력할 수 있는 다른 방법을 제공합니다.

추가 : 제일 아래 공백으로 된 한줄에서 마우스로 더블클릭을 하면 다음과 같은 대화 상자가 생성됩니다.



종류 : 여기서는 사용하지 않습니다.

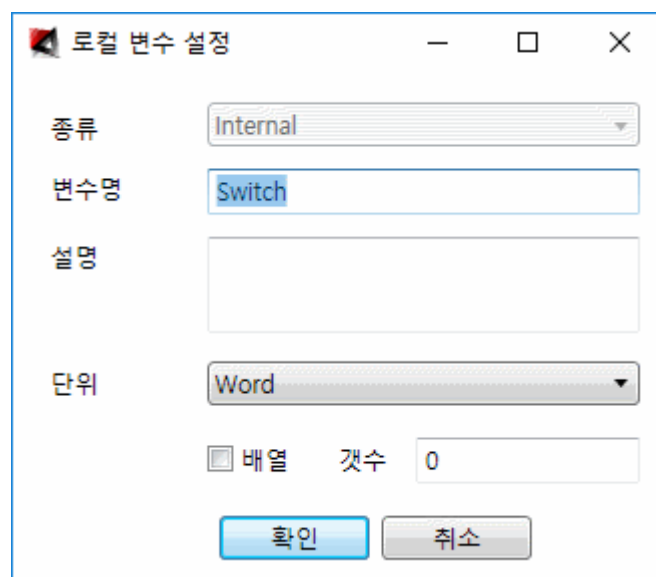
변수명 : 변수명을 입력합니다. 일반적으로 변수명은 변수 이름만 보아도 그 의미를 알 수 있도록 설정하며 글로벌 변수(3.1.7 글로벌 변수 설정에서 설명)와 구분하기 위하여 첫글자만 대문자로 나머지는 소문자로 설정합니다.

설명 : 설명을 입력합니다.

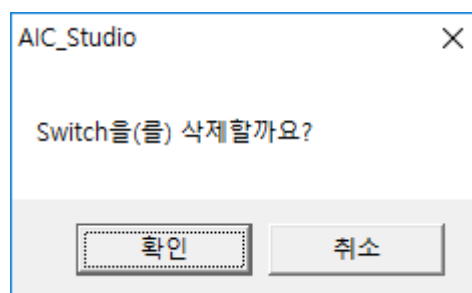
단위 : 단위를 설정합니다. 설정가능한 단위는 Bool, Byte, Word, Short, DWord, Int, Float 가 있습니다.

배열 : 배열을 체크하면 변수는 입력하는 갯수 만큼의 배열로 사용이 됩니다.

수정 : 설정한 위치에서 마우스로 더블클릭하며 다음과 같이 선택한 로컬 변수에 대한 수정 대화 상자가 화면에 나타납니다.

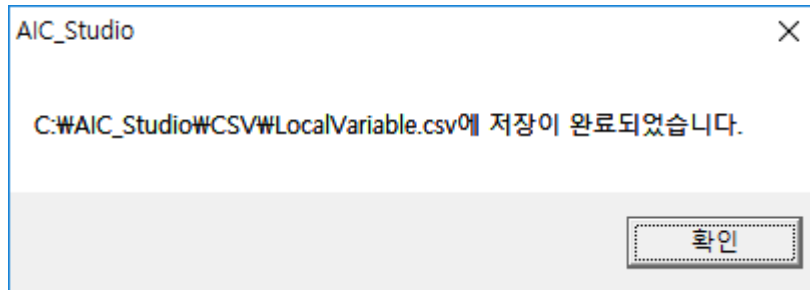


삭제 : 삭제하고자하는 항목을 선택(다중 선택 가능)하고 Delete 키를 누르면 다음과 같은 대화 상자가 생성되어 삭제할 것인지를 확인합니다.



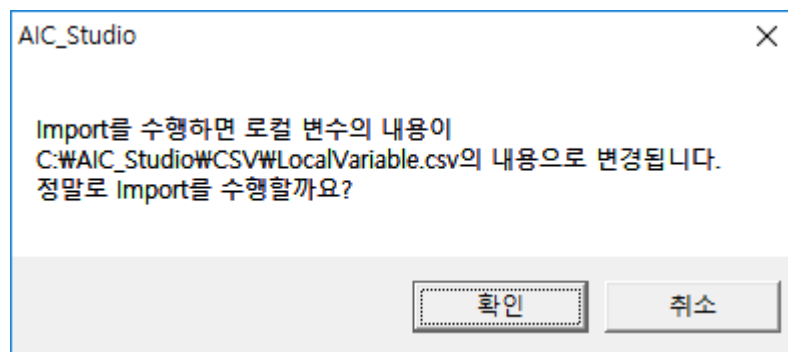
확인을 선택하면 선택한 항목이 삭제됩니다.

Export : 로컬 변수 설정에서 입력한 내용을 .csv 파일로 저장합니다. 저장 위치는 C:\AIC_Studio\CSV\LocalVariable.csv 에 저장되며 Export 를 실행하면 다음과 같은 대화 상자가 생성되어 저장 위치를 알려줍니다.



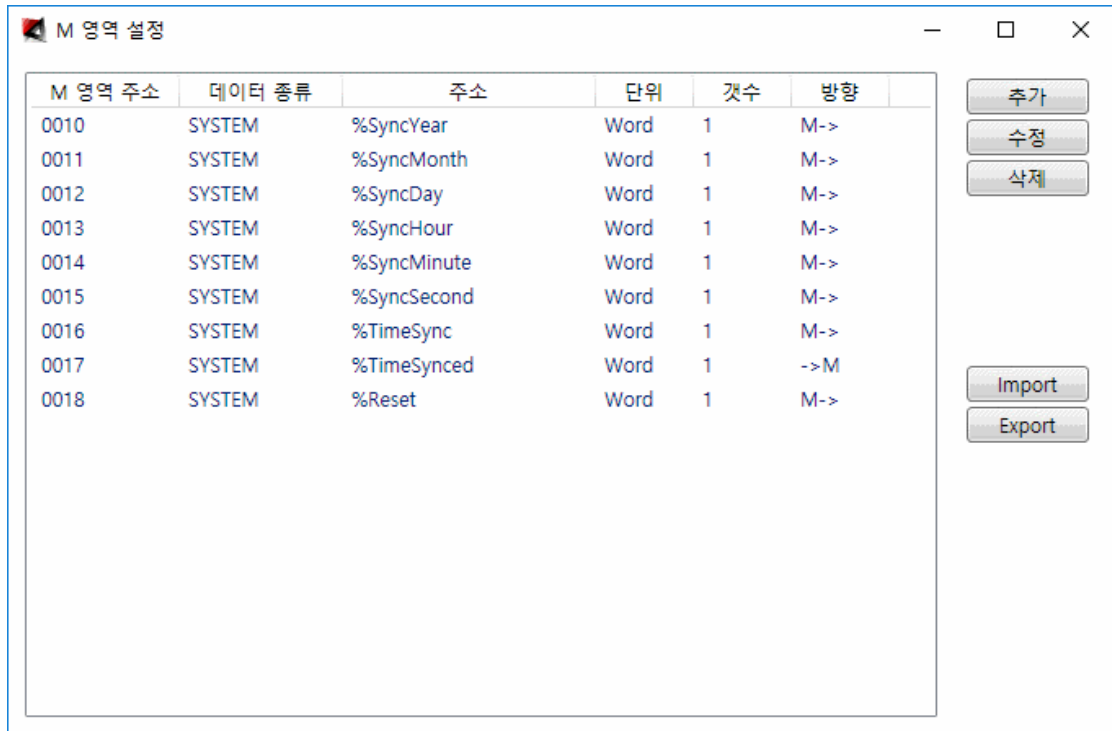
Import : C:\AIC_Studio\CSV\LocalVariable.csv 로부터 데이터를 읽어 적용합니다. 일반적으로 Export 를 이용하여 저장한 파일을 Excel 등을 이용하여 편집한 후 Import 를 이용하여 적용하는 방식으로 일괄작업을 수행합니다.

Import 기능은 로컬 변수 전체가 변경되므로 실행하게 되면 다음과 같은 대화 상자를 띄워 경고를 합니다.



3.2.9 M 영역 설정 메뉴

M 영역 설정을 선택하면 다음과 같은 대화 상자가 생성됩니다.



M 영역은 외부 HMI S/W 와 데이터를 공유하기 위한 역할과, Flash 메모리로 저장할 것인지를 설정하기 위해 사용됩니다.

M 영역 설정 대화 상자를 열면 위 그림과 같이 기본적으로 데이터가 설정 되어있습니다.

1 번과 2 번은 AO 메모리를 설정한 것으로 %MW0(메모리 주소 표기법에 대해서는 부록 2 참고)에 제어를 내리면 AO 0 번에 제어가 내려가고 %MW1 에 제어를 내리면 AO 1 번에 제어가 내려가도록 설정한 것입니다. 이 두 설정은 프로젝트 생성시 아날로그 출력 포인트가 없는 제어를 설정할 경우 이 항목은 생성되지 않습니다.

오토베이스에서 제작한 제어기는 외부 HMI S/W 와 데이터를 공유하기 위해 기본적으로 Modbus 프로토콜과 AOB(Autobase Open Bus)을 지원하고 있습니다.

Modbus 프로토콜을 예로 M 영역의 필요성을 간략히 설명하도록 하겠습니다.

Function Code	Name	Address	Remark	제어기와의 매핑
1	Read Coil Status	0xxxx(bit-출력)	Bit Read	Q 영역(디지털 출력 상태)
2	Read Input Status	1xxxx(bit-입력)	Bit Read	I 영역(디지털 입력)
3	Read Holding Reg.	4xxxx(word-출력)	Word Read	M 영역(입력)
4	Read Input Reg.	3xxxx(word-입력)	Word Read	AI 영역(아날로그 입력)
5	Force Single Coil	0xxxx(bit-출력)	Bit Write	Q 영역(디지털 출력 제어)
6	Preset Single Reg	4xxxx(word-출력)	Word Write	M 영역(제어)
15	Force Multi Coil	0xxxx(bit-출력)	Bit Write	Q 영역(디지털 출력 제어)
16	Preset Multi Reg.	4xxxx(word-출력)	Word Write	M 영역(제어)

기본적으로 Function Code 2 번과 DI 메모리(I 영역)가, 1 번과 DO 메모리(Q 영역)가, 4 번과 AI 메모리(AI 영역)가 매핑되어 있으며, 3 번 영역과 AO 메모리(AO 영역)을 바로 매핑할 경우 제어로직에서 사용하는 다양한 변수를 HMI 에서는 감시 제어를 수행할 수 없게 됩니다. 이를 해결하기 위해 Modbus Function Code 3 번과 M 영역을 매핑하고, M 영역 설정을 이용하여 원하는 메모리 데이터를 외부 HMI S/W 와 공유 가능하도록 구성했습니다.

추가를 선택하면 다음과 같은 대화 상자가 생성됩니다.

M 영역 주소 : M 영역 주소를 입력합니다. 설정의 편의를 위해 기존에 설정한 M 영역 주소의 마지막 다음 주소를 자동으로 찾아 초기값으로 표시합니다.

메모리 종류 : 메모리 종류를 설정합니다. 설정가능한 메모리 종류는 I, Q, AI, AO, SYSTEM 등이 있습니다. 메모리 종류에 대한 자세한 내용은 [3.2.9 글로벌 변수 설정](#)을 참고하기 바랍니다.

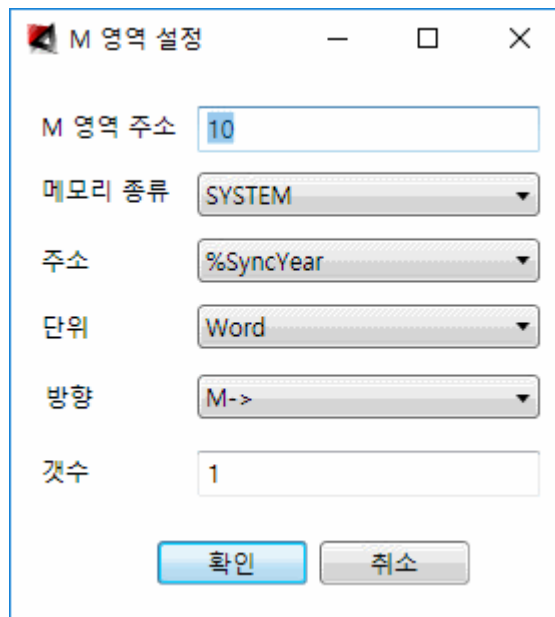
주소 : 주소를 설정합니다. 프로젝트 생성시 설정한 제어기 종류에 따라 설정가능한 주소는 달라집니다.

단위 : 단위를 설정합니다. 설정가능한 단위는 Bool, Byte, Word, Short, DWord, Int, Float 가 있습니다.

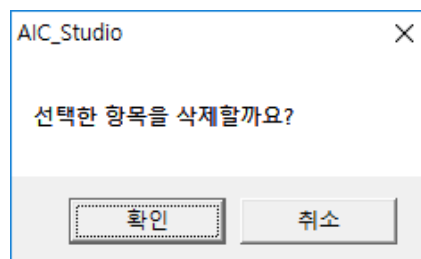
방향 : 데이터 복사 방향을 설정합니다. 설정가능한 방향은 ->M 과 M->이 있으며, ->M 을 설정하면 설정한 메모리의 값을 M 영역으로 주기적으로 전송하게 되고, M->을 설정하면 외부 HMI S/W 에서 M 영역에 제어를 내리면 그 값을 설정한 메모리에 출력을 하게 됩니다.

갯수 : 데이터 처리를 할 갯수를 입력합니다.

수정 : 수정을 선택하면 다음과 같이 선택한 M 영역 설정에 대한 수정 대화 상자가 화면에 나타납니다.

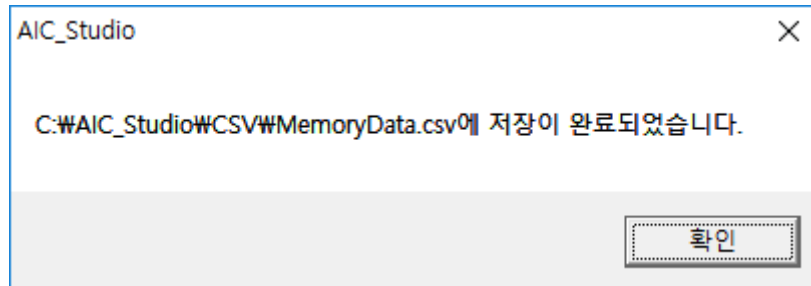


삭제 : 삭제를 선택하면 다음과 같은 대화 상자가 생성되어 삭제할 것인지를 확인합니다.



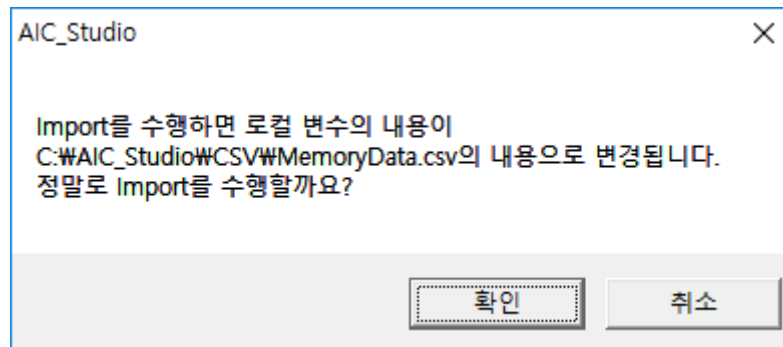
확인을 선택하면 선택한 항목이 삭제됩니다.

Export : 메모리 데이터 설정에서 입력한 내용을 .csv 파일로 저장합니다. 저장 위치는 C:\AIC_Studio\CSV\MemoryData.csv 에 저장되며 Export 를 실행하면 다음과 같은 대화 상자가 생성되어 저장 위치를 알려줍니다.



Import : C:\AIC_Studio\CSV\MemoryData.csv 로부터 데이터를 읽어 적용합니다. 일반적으로 Export 를 이용하여 저장한 파일을 Excel 등을 이용하여 편집한 후 Import 를 이용하여 적용하는 방식으로 일괄작업을 수행합니다.

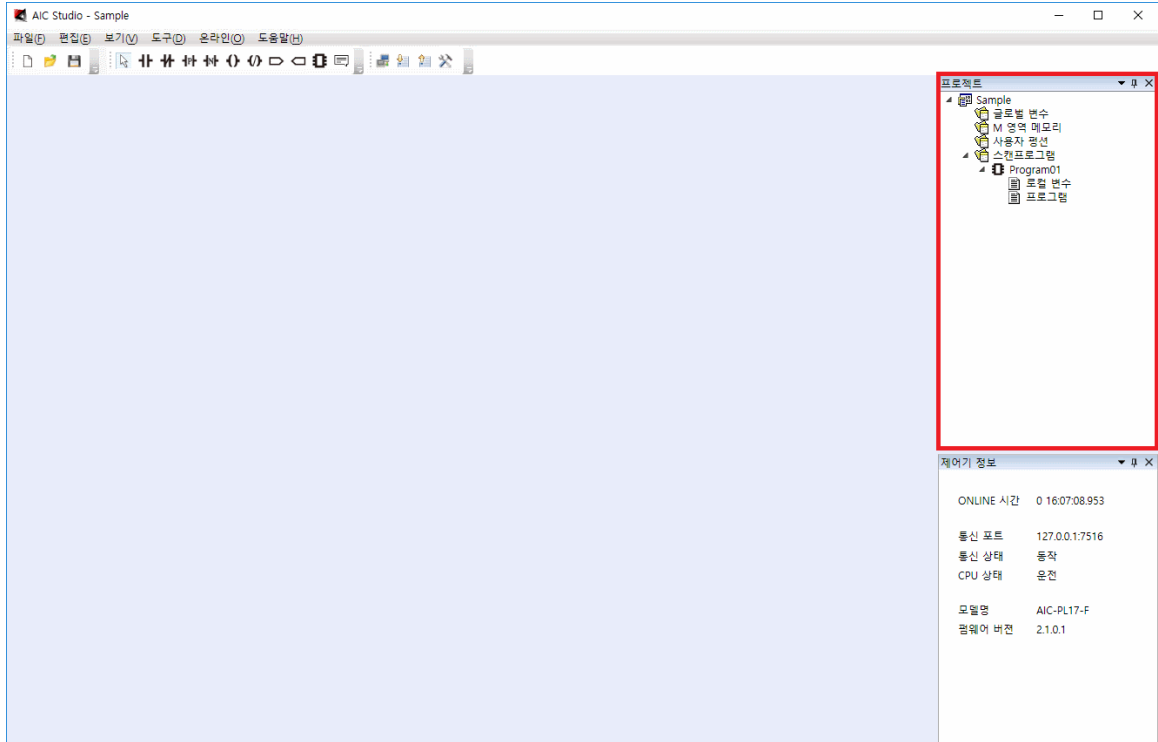
Import 기능은 메모리 데이터 전체가 변경되므로 실행하게 되면 다음과 같은 대화 상자를 띄워 경고를 합니다.



3.3 보기 메뉴

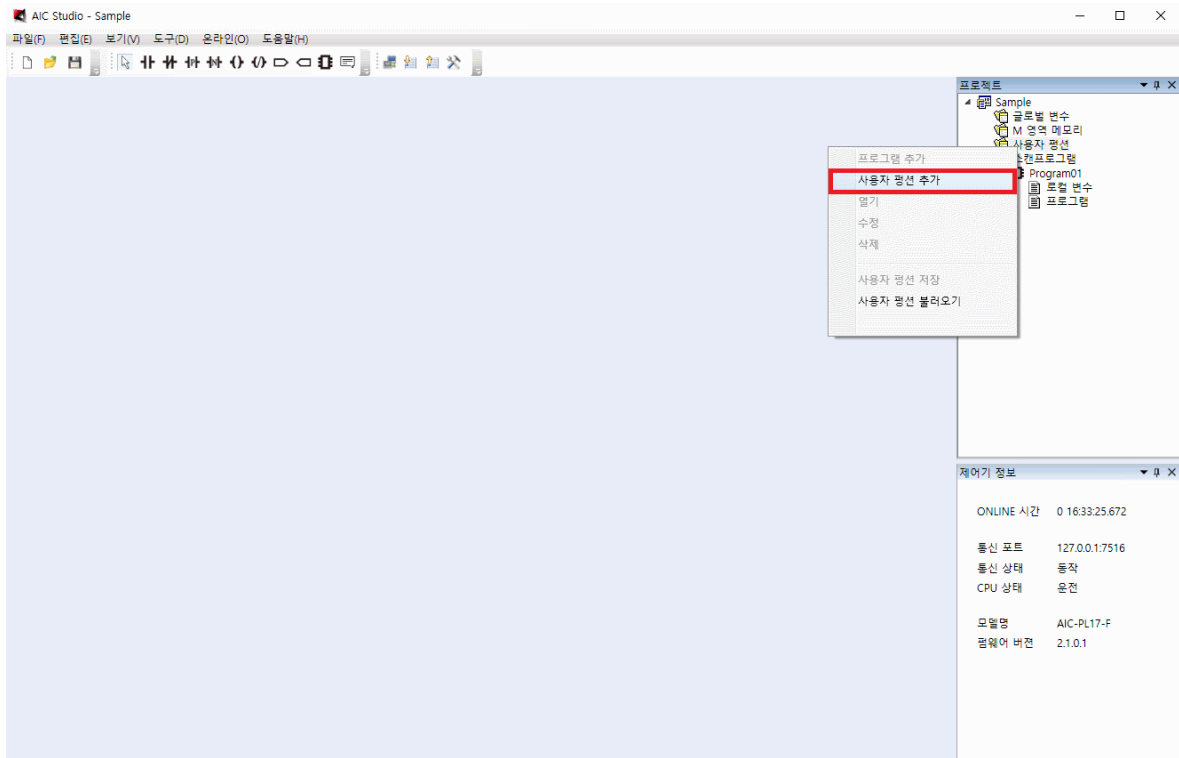
3.3.1 프로젝트 메뉴

화면 우측의 프로젝트 바를 보일건지 숨길건지 결정합니다.



프로젝트 바의 기능은 다음과 같습니다.

- 글로벌 변수 : 편집메뉴의 글로벌 변수와 같은 기능을 수행합니다.
- M 영역 메모리 : 편집메뉴의 M 영역 메모리와 같은 기능을 수행합니다.
- 사용자 평선 : 사용자 평선을 관리합니다. 사용자 평선은 자주 사용하는 기능을 Ladder 나 C 언어로 작성하여 Function 과 같이 사용할 수 있도록 하는 기능입니다.



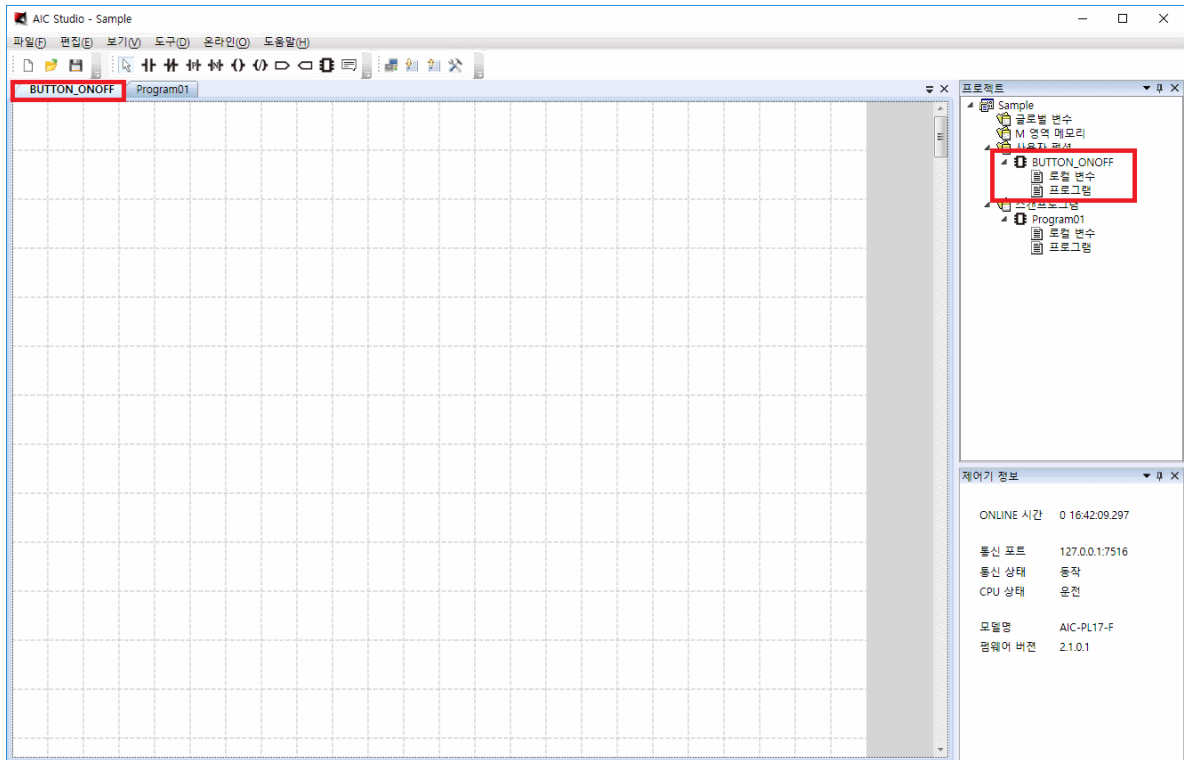
사용자 평선에서 마우스 우측 버튼을 클릭하여 생성된 메뉴에서 사용자 평선 추가를 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.

평선이름 : 평선 이름을 입력합니다(여기서는 BUTTON_ONOFF).

설명 : 사용자 평선의 설명을 입력합니다.

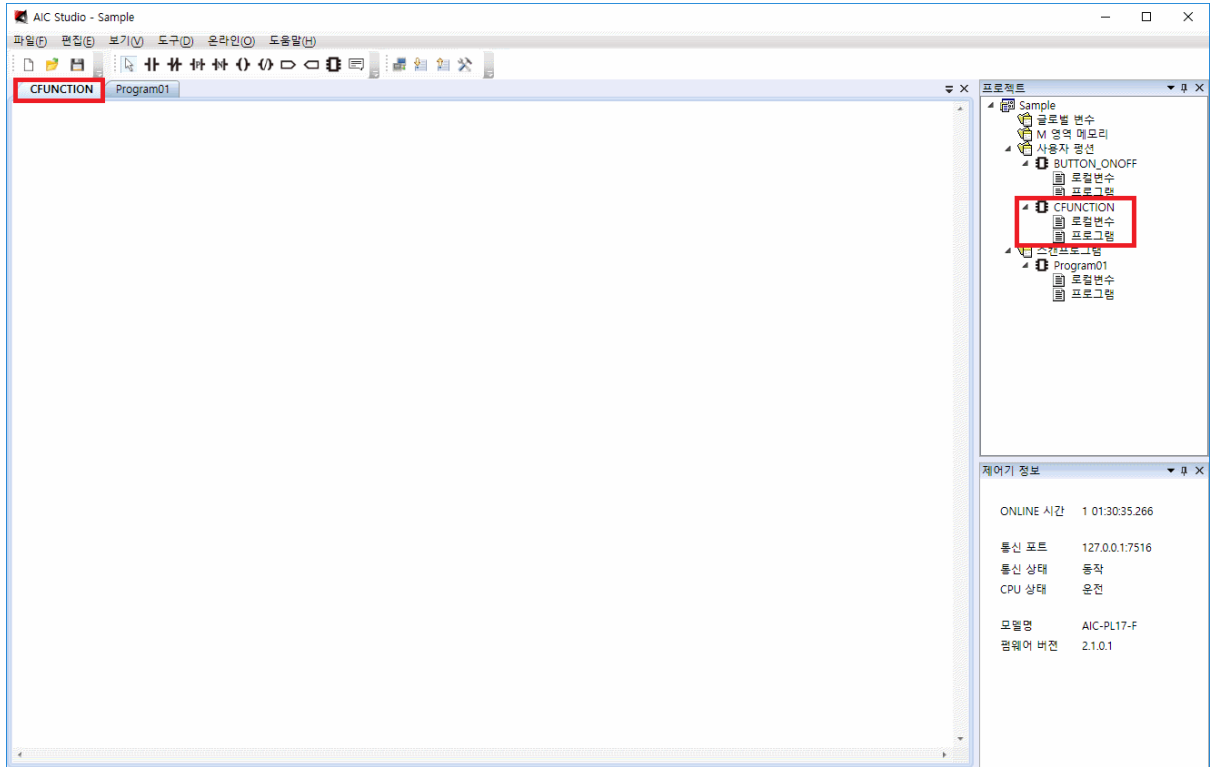
언어 : 언어를 선택합니다. 현재는 LD 만 지원하며 향후 C 언어를 추가로 지원할 예정입니다.

확인을 선택하면 다음과 같이 입력한 평선을 생성한 후 편집이 가능하도록 화면에 표시합니다.

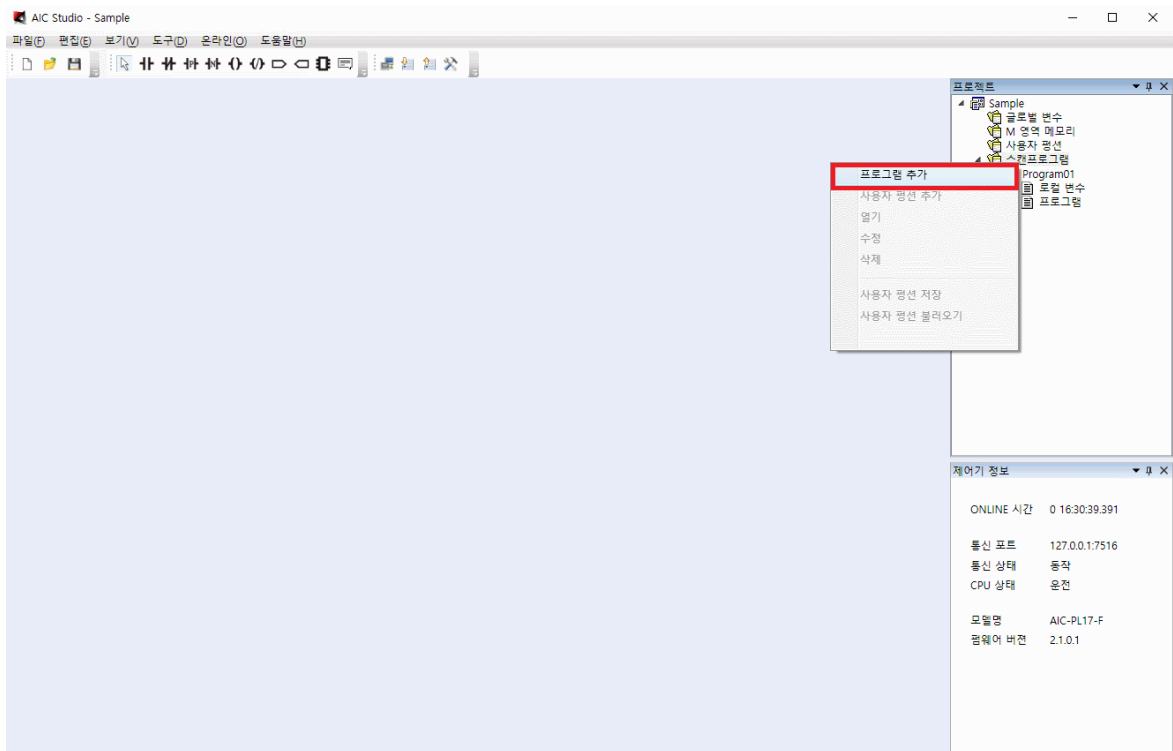


평선은 기존 프로그램과 같이 작성하면 되며 로컬변수 사용법만 약간 다릅니다. 자세한 사용법은 [부록 4 사용자 평선 작성법](#)을 참조하기 바랍니다.

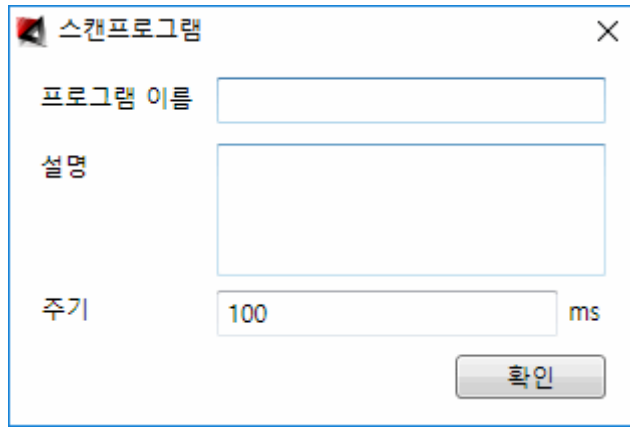
언어를 C 언어로 선택한 후 확인을 선택하면 다음음과 같이 입력한 평선을 생성한 후 C 언어를 입력 가능하도록 화면에 표시합니다. 자세한 사용법은 [부록 5 C 언어 사용법](#)을 참조하기 바랍니다.



- 스캔 프로그램 : 제어로직 프로그램을 관리합니다.



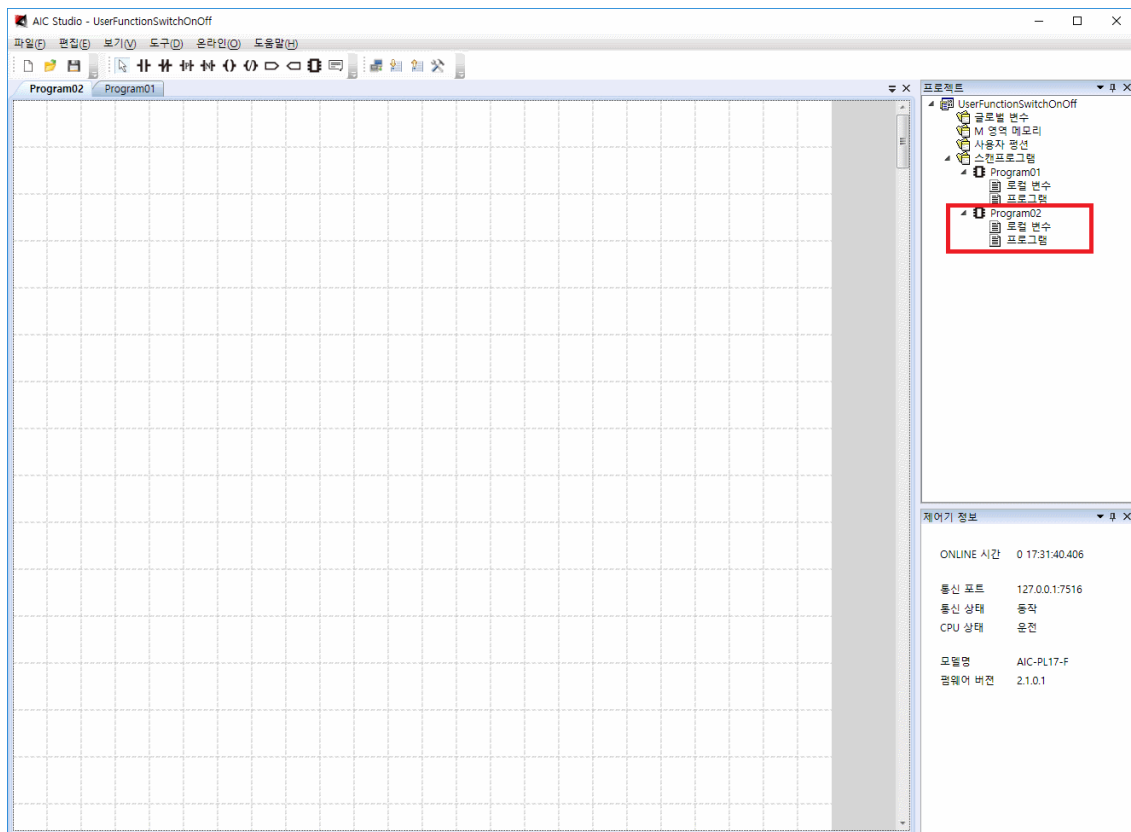
사용자 평선에서 마우스 우측 버튼을 클릭하여 생성된 메뉴에서 사용자 평선 추가를 선택하면 다음과 같이 화면에 나타납니다.



프로그램 이름 : 프로그램 이름을 입력합니다(여기서는 Program02).

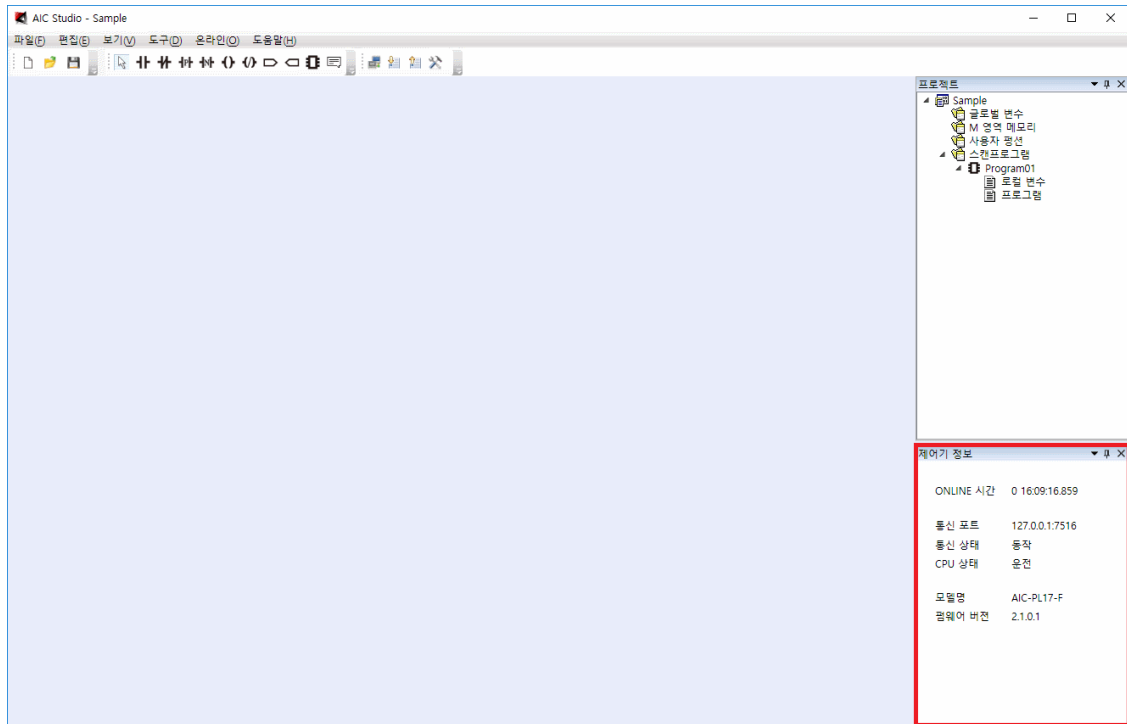
설명 : 설명을 입력합니다.

주기 : 프로그램이 실행 되는 주기를 설정합니다. 만약 0 을 입력하면 제어가 지원하는 최고 속도로 수행이 됩니다.



3.3.2 제어기 정보 메뉴

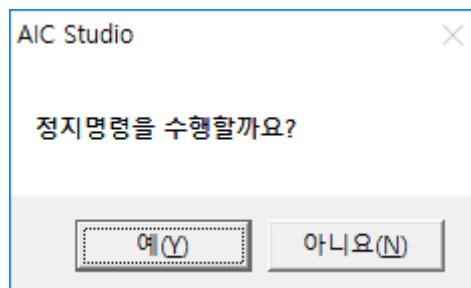
화면 우측의 제어기 정보 바를 보일건지 숨길건지 결정합니다.



제어기 정보에 표시되는 내용은 다음과 같습니다.

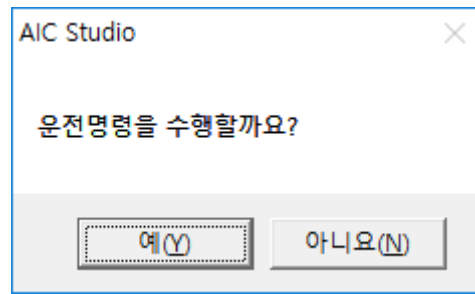
- ONLINE 시간 : 제어기가 켜진 후 현재까지의 시간을 표시합니다.
- 통신 포트 : 제어기와 통신하기 위해 설정한 통신 정보를 표시합니다.
- 통신 상태 : 제어기와의 통신 상태를 표시합니다.
- CPU 상태 : CPU 상태를 표시합니다.

CPU 상태가 운전일때 마우스로 클릭하면 다음과 같은 대화 상자가 생성됩니다.



예를 선택하면 CPU 를 정지 상태로 만듭니다. CPU 가 정지 상태가 되면 다운로드된 제어로직의 동작이 멈추게 됩니다.

CPU 상태가 정지일때 마우스로 클릭하면 다음과 같은 대화 상자가 생성됩니다.



예를 선택하면 CPU 를 운전 상태로 만듭니다.

- 모델명 : 제어기의 모델을 표시합니다.
- 펌웨어 버전 : 펌웨어의 버전을 표시합니다.

3.3.3 모두보기 메뉴

프로젝트 바와 제어기 정보 바를 모두 보이게 합니다.

3.3.3 모두숨기기 메뉴

프로젝트 바와 제어기 정보 바를 모두 숨깁니다.

3.4 도구 메뉴

3.4.1 컴파일 메뉴

그림으로 그려진 로직을 해석하여 제어기에서 해석할 수 있는 기계어로 변환하여 파일로 저장하기 위해 사용합니다. 사용법은 2.3 컴파일을 참조하기 바랍니다.

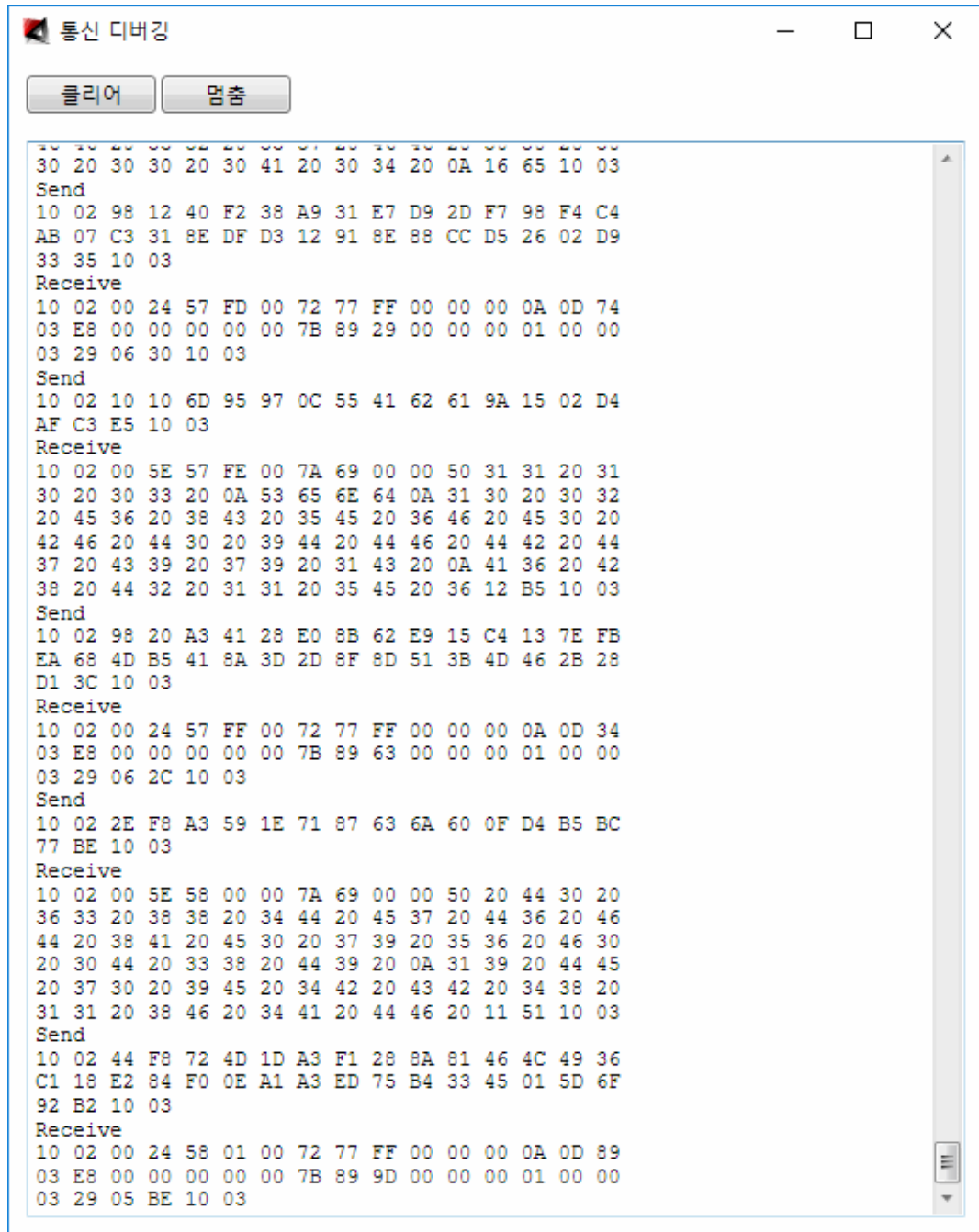
3.5 온라인 메뉴

3.5.1 통신 설정 메뉴

제어기와의 통신을 설정하기 위해 사용합니다. 사용법은 [2.4 통신 설정](#)을 참조하기 바랍니다.

3.5.2 통신 디버깅 메뉴

제어기와의 통신을 디버깅하기 위해 사용합니다. 통신 디버깅 메뉴를 선택하면 다음과 같이 화면에 나타납니다.



본 화면은 통신이 원활 할 때의 화면으로 제어기와 통신하는 데이터를 Send 와 Receive 로 나누어 빠른 속도로 번갈아 가며 표시를 합니다.

다음 예는 통신 연결은 되나 제어기에서 응답이 없는 경우에 표시되는 메시지입니다.

```

Send

10 02 4E 43 E0 C7 CB 22 50 87 24 65 B7 38 63 B1

28 93 10 03

```

Disconnect : CommPort = COM7, Baud = 115200

Connect OK : CommPort = COM7, Baud = 115200

Send

10 02 78 89 C9 7D 7E F0 25 DE 59 69 E5 51 DC 99

AF 73 10 03

Send

10 02 CC 90 59 15 82 B0 7A AF 59 E3 96 DD 39 8A

FA 8B 10 03

Send

10 02 7B 91 83 35 80 1C 3E A6 59 B9 C0 8D D3 CF

84 3D 10 03

Send

10 02 3B 42 9D 32 64 3C CA D2 F4 E0 C4 53 5C 8A

D1 BA 10 03

Send

10 02 F0 BB FD DE 8E 69 95 86 BE 86 B7 13 D6 DB

82 3E 10 03

Disconnect : CommPort = COM7, Baud = 115200

Connect OK : CommPort = COM7, Baud = 115200

데이터를 Send 하다가 응답이 없어 통신을 끊고 다시 연결하는 것을 반복하고 있습니다. 이 경우 제어기 쪽 이상이 가능성이 더 높으며 콘솔의 통신 디버깅 (제어기 입장에서 통신 디버깅)과 함께 분석해야만 합니다.

다음 예는 통신 연결이 되지 않을 경우입니다. 이 경우 통신 연결이 되지 않는 이유를 컴퓨터를 중심으로 찾아야만 합니다(연결이 되지 않아 연결 시도를 하여 Connect Fail 만 반복하고 있습니다).

```
Connect Fail : CommPort = COM7, Baud = 115200

Connect Fail : CommPort = COM7, Baud = 115200

Connect Fail : CommPort = COM7, Baud = 115200

Connect Fail : CommPort = COM7, Baud = 115200

Connect Fail : CommPort = COM7, Baud = 115200

Connect Fail : CommPort = COM7, Baud = 115200

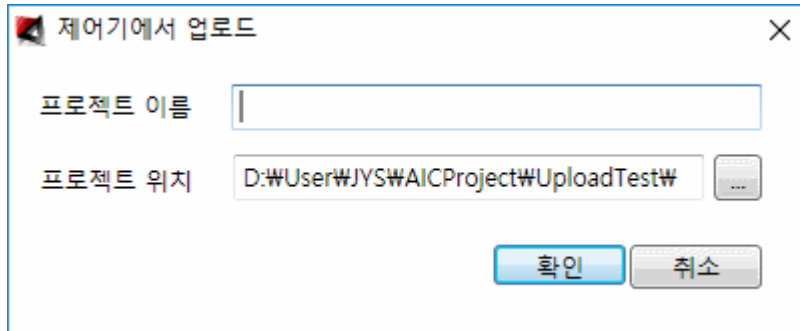
Connect Fail : CommPort = COM7, Baud = 115200
```


3.5.3 제어기로 다운로드 메뉴

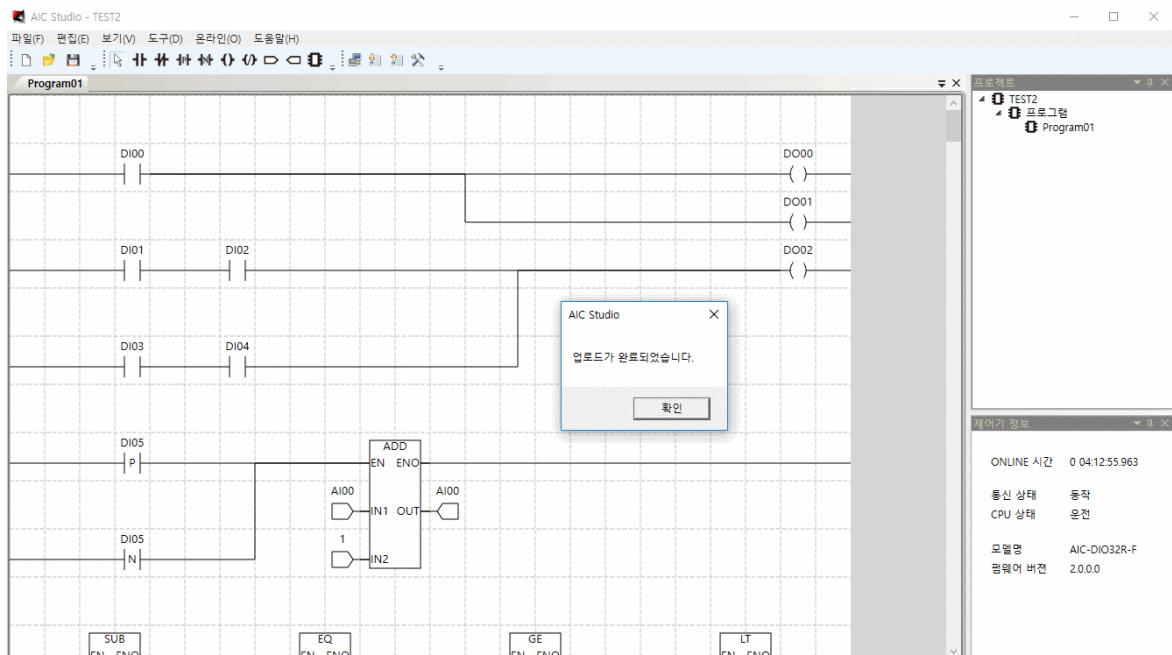
제어기에 로직 데이터를 다운로드 하기 위해 사용합니다. 사용법은 2.5 제어기로 다운로드를 참조하기 바랍니다.

3.5.4 제어기에서 업로드 메뉴

제어기에 다운로드 된 로직 데이터를 컴퓨터로 업로드하기 위해 사용합니다. 제어기에서 업로드 메뉴를 선택하면 다음과 같은 대화 상자가 생성됩니다.

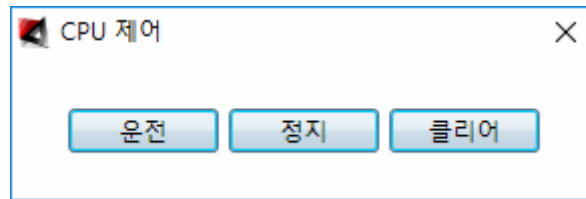


프로젝트 이름과 프로젝트 위치를 입력하고 확인을 선택하면 제어기로부터 프로젝트 데이터를 수신하여 다음과 같이 화면에 표시한 후 업로드가 완료되었다는 대화 상자를 표시합니다.

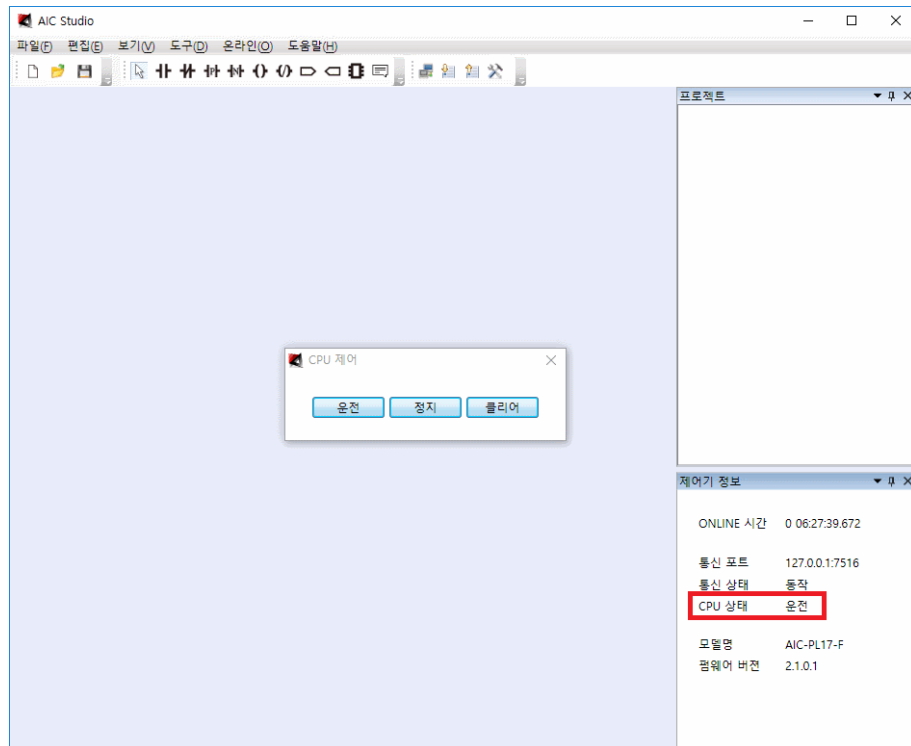


3.5.5 CPU 제어 메뉴

CPU 상태 제어 및 클리어를 위해 사용합니다. CPU 제어 메뉴를 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.



운전 : CPU 상태를 운전 상태로 만듭니다. 운전을 수행하면 성공 여부를 대화 상자로 표시합니다. 운전 상태가 되면 제어로직 동작을 수행하게 되고 우측 하단 제어기 정보 창의 CPU 상태에 운전이라고 표시됩니다.

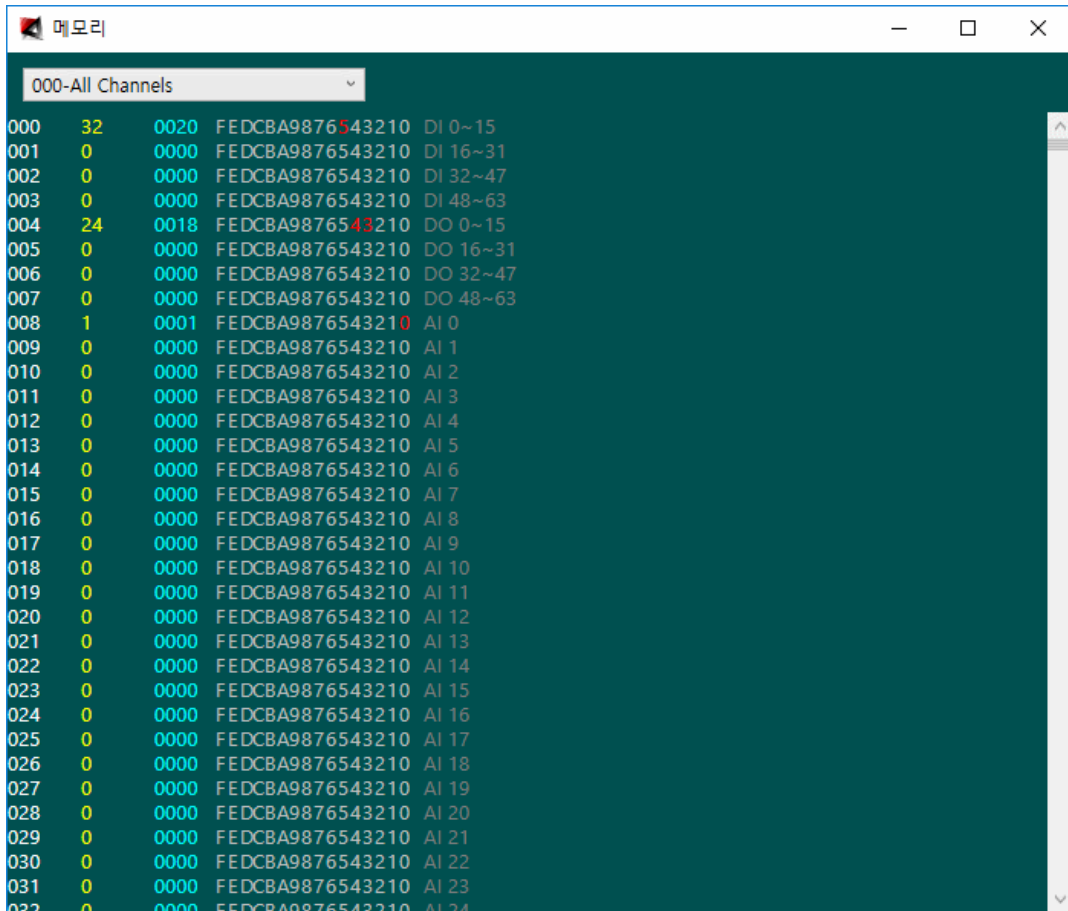


정지 : CPU 상태를 정지 상태로 만듭니다. 정지를 수행하면 성공 여부를 대화 상자로 표시합니다. 정지 상태가 되면 제어로직 동작을 멈추게 됩니다

클리어 : 다운로드된 제어 로직을 포함하여 제어기의 모든 데이터를 클리어 시킵니다. 클리어를 수행하면 성공 여부를 대화 상자로 표시합니다.

3.5.6 메모리 메뉴

제어기의 메모리를 감시제어하기 위해 사용합니다. 메모리 메뉴를 선택하면 다음과 같은 창이 화면에 나타납니다.

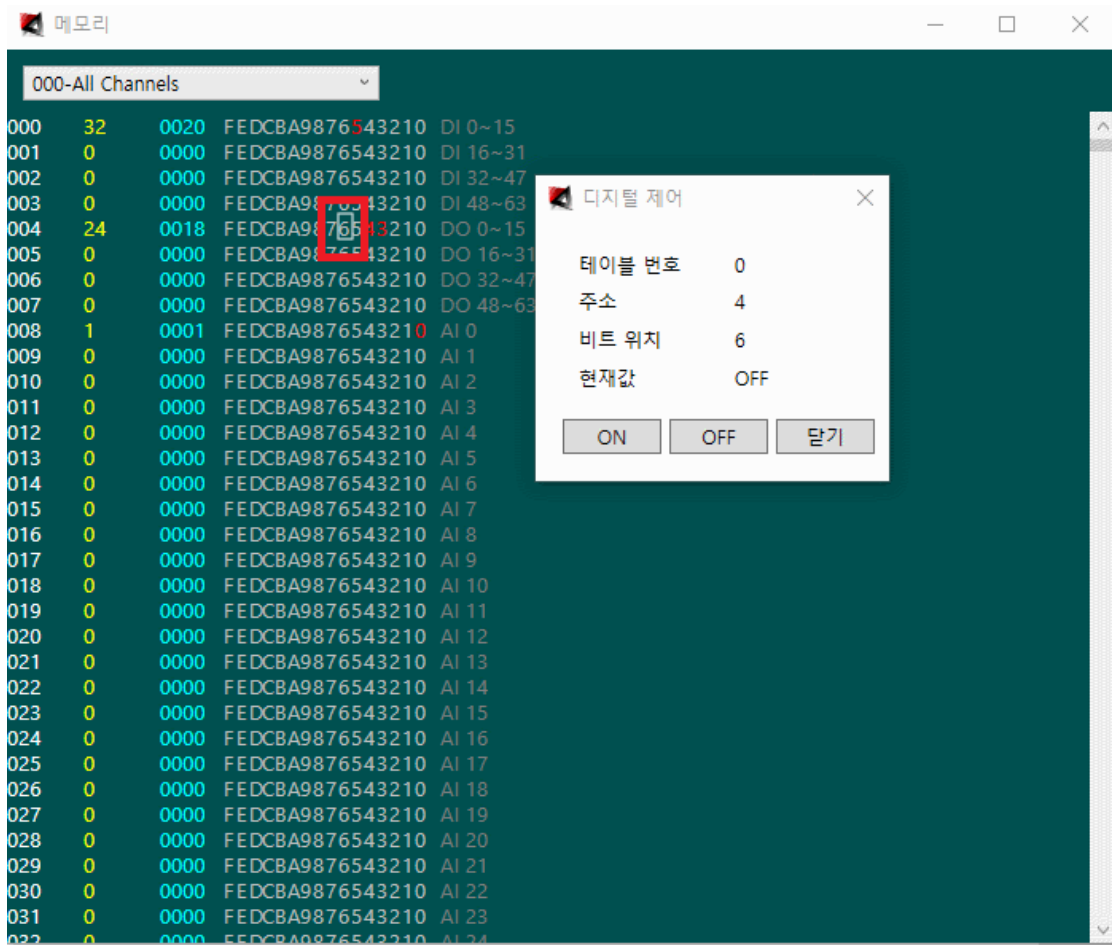


제어기의 내부메모리를 표시합니다.

- 000-All Channels : 포인트 메모리 전체를 표시합니다. DI, DO, AI, AO 메모리 순서대로 제어기 설정에서 할당된 포인트 수만큼 씩 표시합니다.
- 001-DI : DI 포인트 메모리를 표시합니다.
- 002-DO : DO 포인트 메모리를 표시합니다.
- 003-AI : AI 포인트 메모리를 표시합니다.
- 004-AO : AO 포인트 메모리를 표시합니다.
- 005-System Memory : System 메모리를 표시합니다.

2 진 제어

2 진 값 표시 위치에서 마우스 좌측버튼을 누르면 다음과 같은 대화 상자가 화면에 나타납니다.



ON(1)이나 OFF(0)를 누르면 해당 값으로 제어가 됩니다.

제어 포인트가 디지털 입력인 경우 실 포인트는 제어가 되지 않으며(현장의 실 포인트가 바뀌면 그에 따라 바뀌게 됩니다), 가상포인트는 제어가 수행되어 값이 바뀌게 됩니다(시뮬레이터에서는 모든 값이 제어가 됩니다).

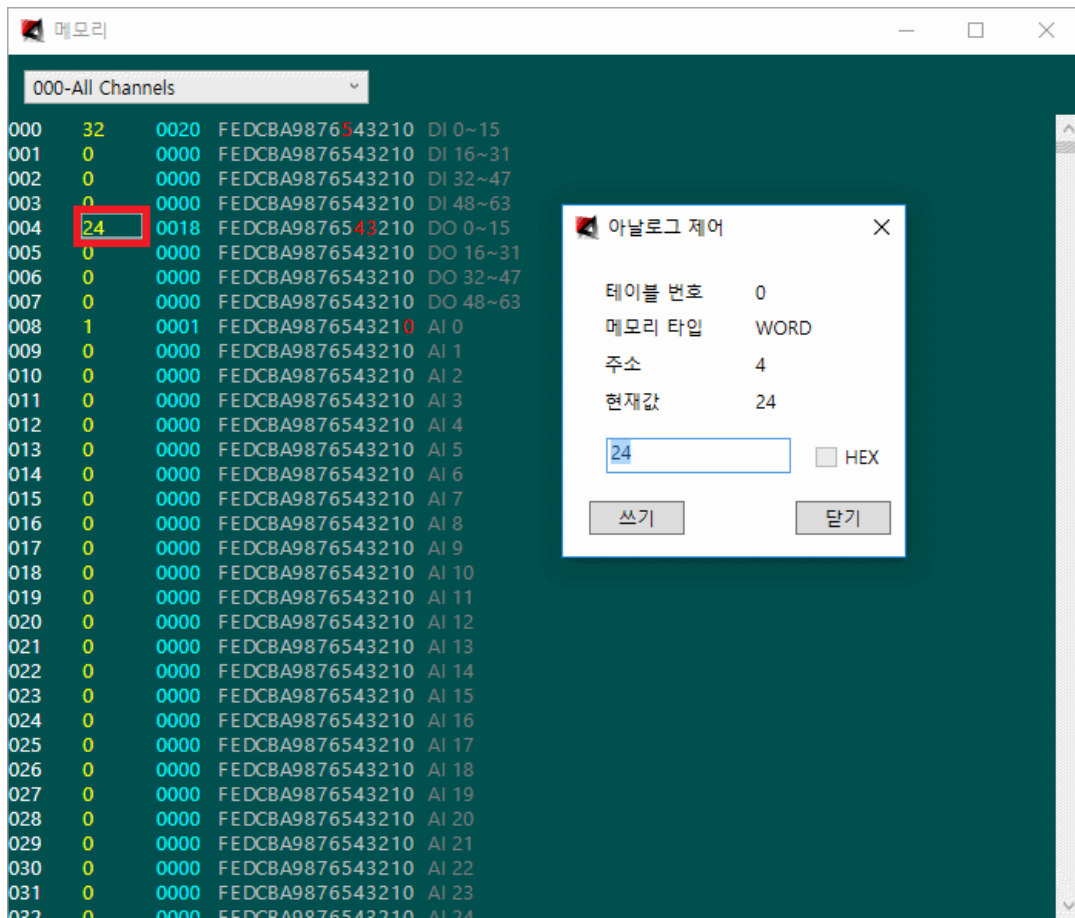
제어 포인트가 디지털 출력인 경우 실 포인트는 현장으로 실제 제어가 나가며, 가상포인트는 제어가 수행되어 값이 바뀌게 됩니다.

제어 포인트가 아날로그 입력인 경우 실 포인트는 제어가 되지 않으며(현장의 실 포인트가 바뀌면 그에 따라 바뀌게 됩니다), 가상포인트는 제어가 수행되어 값이 바뀌게 됩니다(시뮬레이터에서는 모든 값이 제어가 됩니다).

제어 포인트가 아날로그 출력인 경우 실 포인트는 현장으로 실제 제어가 나가며, 가상포인트는 제어가 수행되어 값이 바뀌게 됩니다.

10 진 제어

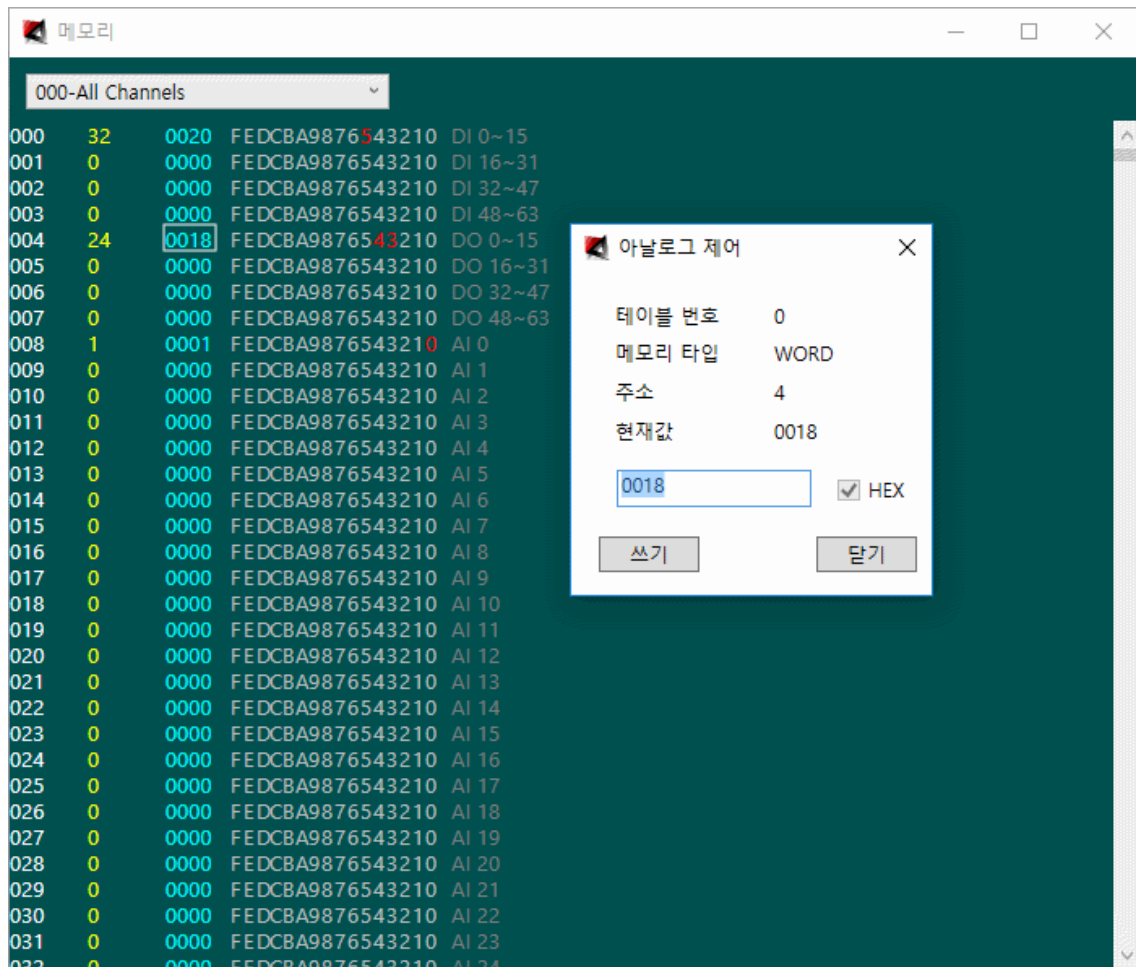
10 진 값 표시 위치에서 마우스 좌측버튼을 누르면 다음과 같은 대화 상자가 화면에 나타납니다.



값을 변경하고 쓰기를 누르면 해당 값으로 제어가 됩니다.

16 진 제어

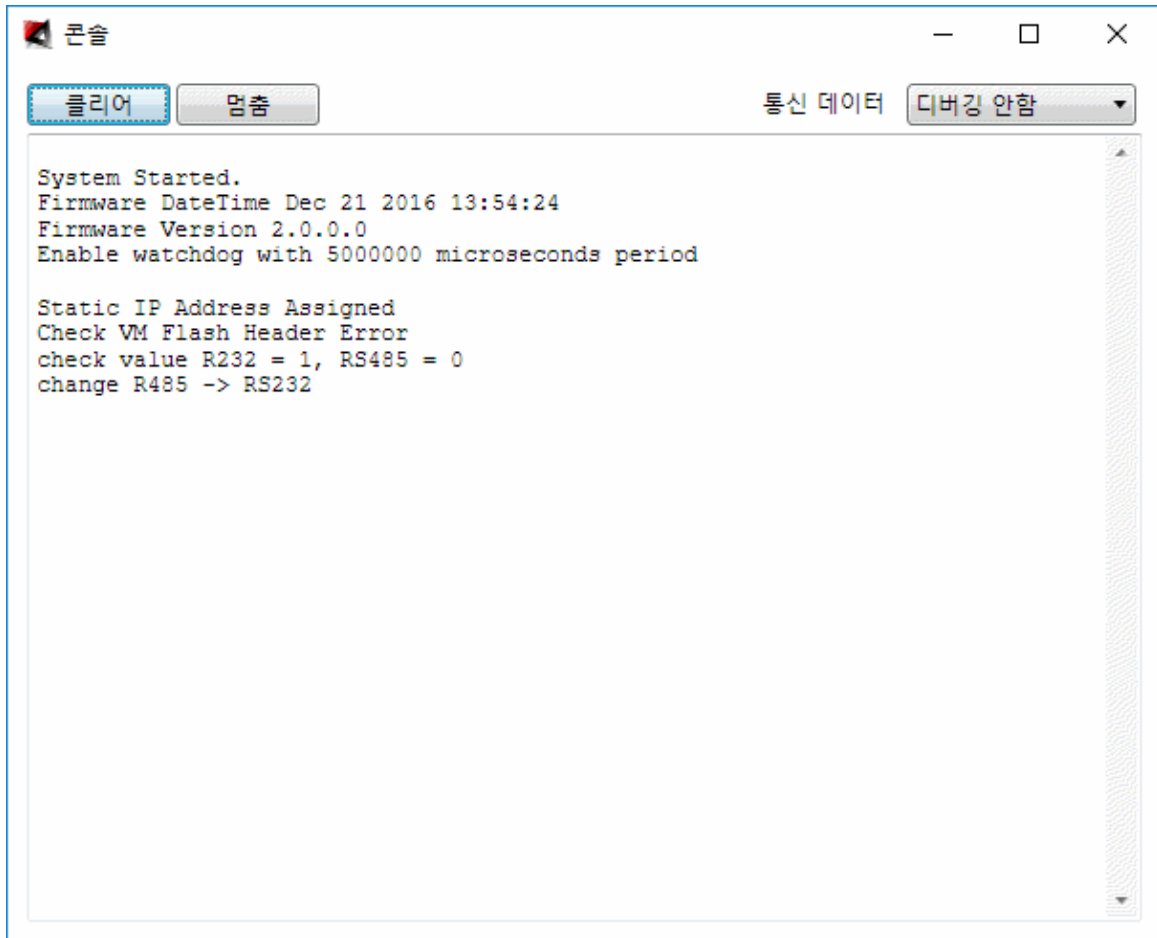
16 진 값 표시 위치에서 마우스 좌측버튼을 누르면 다음과 같은 대화 상자가 화면에 나타납니다.



값을 변경하고 쓰기를 누르면 해당 값으로 제어가 됩니다.

3.5.7 콘솔 메뉴

제어기에서 발생하는 주요 메시지를 실시간으로 보기 위해 사용합니다. 콘솔 메뉴를 선택하면 다음과 같은 창이 화면에 나타납니다.



클리어 : 클리어를 선택하면 발생한 메시지를 화면상에서 지웁니다.

통신 데이터 : 통신 데이터의 디버깅 여부를 설정합니다.

디버깅 안함 : 통신데이터를 디버깅하지 않습니다.

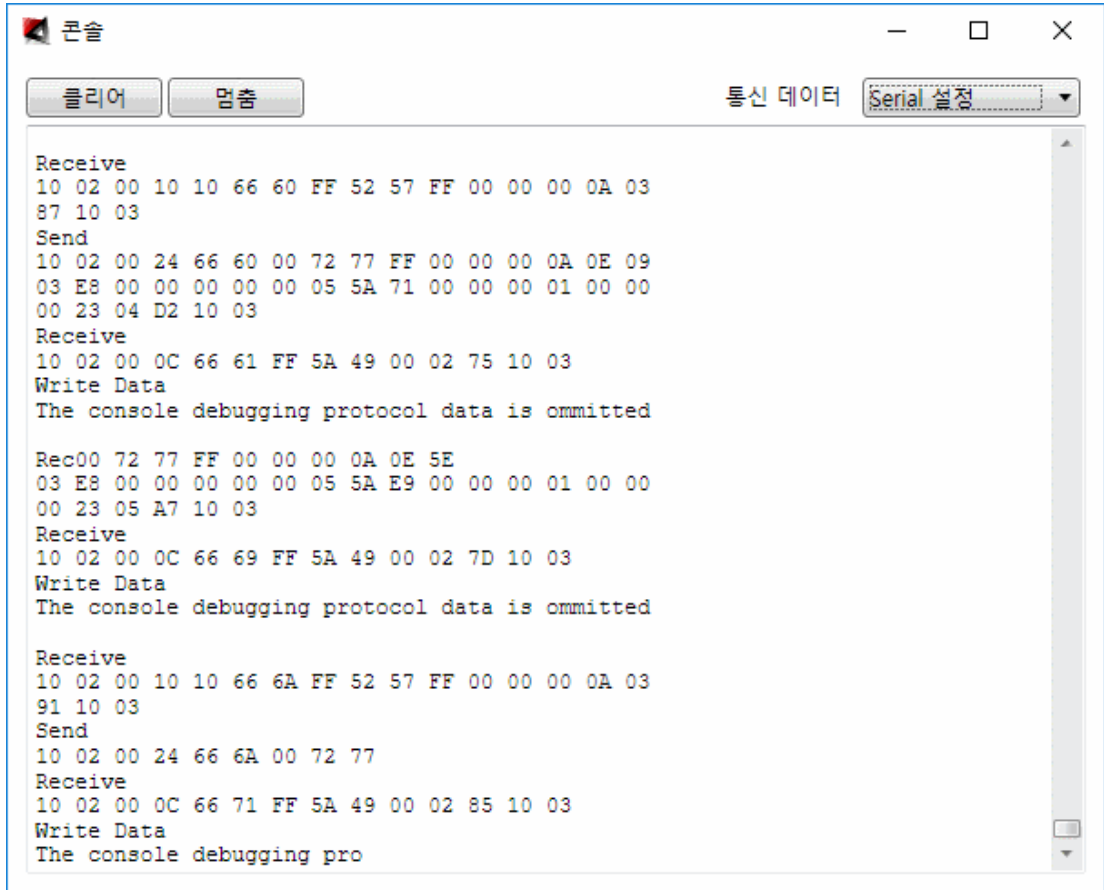
Serial 설정 : Serial 설정 데이터를 디버깅 합니다.

Serial 통신 : Serial 통신 데이터를 디버깅 합니다.

LAN 설정 : LAN 설정 데이터를 디버깅 합니다.

LAN 통신 : LAN 통신 데이터를 디버깅 합니다.

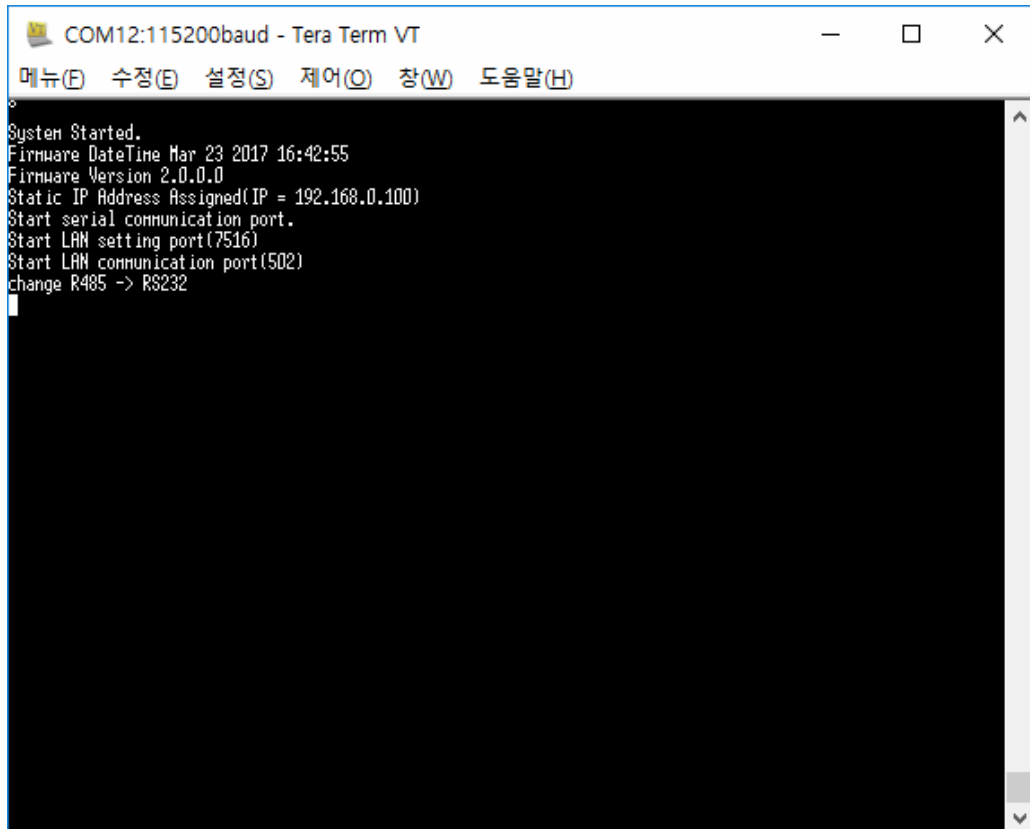
다음 화면은 통신 데이터를 Serial 설정하여 디버깅하는 모양입니다.



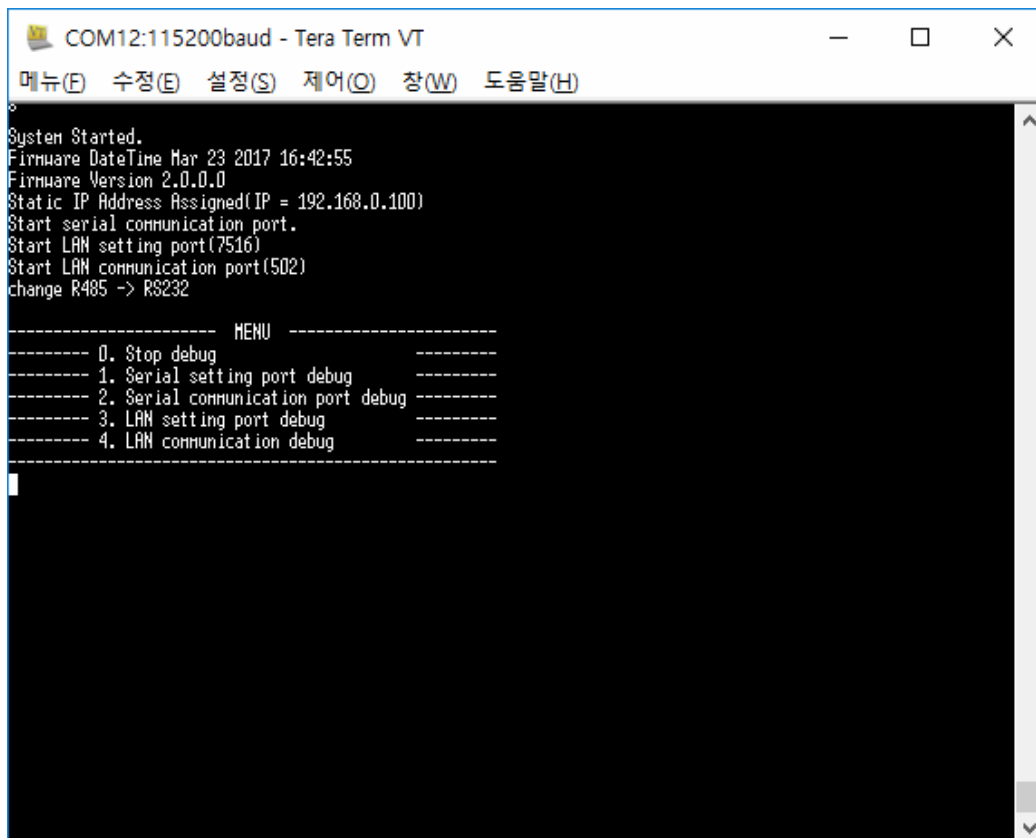
멈춤 : 멈춤을 선택하면 데이터 표출을 멈춥니다. 통신을 수행한 후 분석하기 위해 데이터를 캡처하거나 스크롤을 위로 올려 이전 데이터를 보기 위해 사용합니다. 멈춤을 선택하면 멈춤은 동작으로 바뀌고 데이터 표출을 멈춥니다. 다시 동작 버튼을 클릭하면 동작이 됩니다.

본 콘솔 기능은 AIC_Studio 에서 데이터 감시 등 다른 통신을 수행하며 중간중간에 제어기와 통신을 통해 받아온 데이터를 표출하므로 통신 분석을 정확히 하기에는 간헐적으로 데이터가 빠질 수 있습니다. 정확한 분석을 하기 위해서는 Tera term 등 Serial 프로그램으로 연결하여 분석해야만 합니다. 이를 위해 Autobase 에서 제작한 제어기는 초기에 Serial 프로그램을 이용하여 디버깅이 가능한 콘솔 모드로 동작하고, AIC_Studio 를 실행하여 통신을 시작하면 자동으로 인식하여 AIC_Studio 와 통신하기 위한 모드로 변환이 됩니다(한번 AIC_Studio 와 통신을 시작한 제어기는 콘솔 모드로 동작하기 위해서는 재부팅을 해야 합니다).

다음 화면은 Tera Term 을 이용하여 콘솔 모드에서 접속한 예입니다.

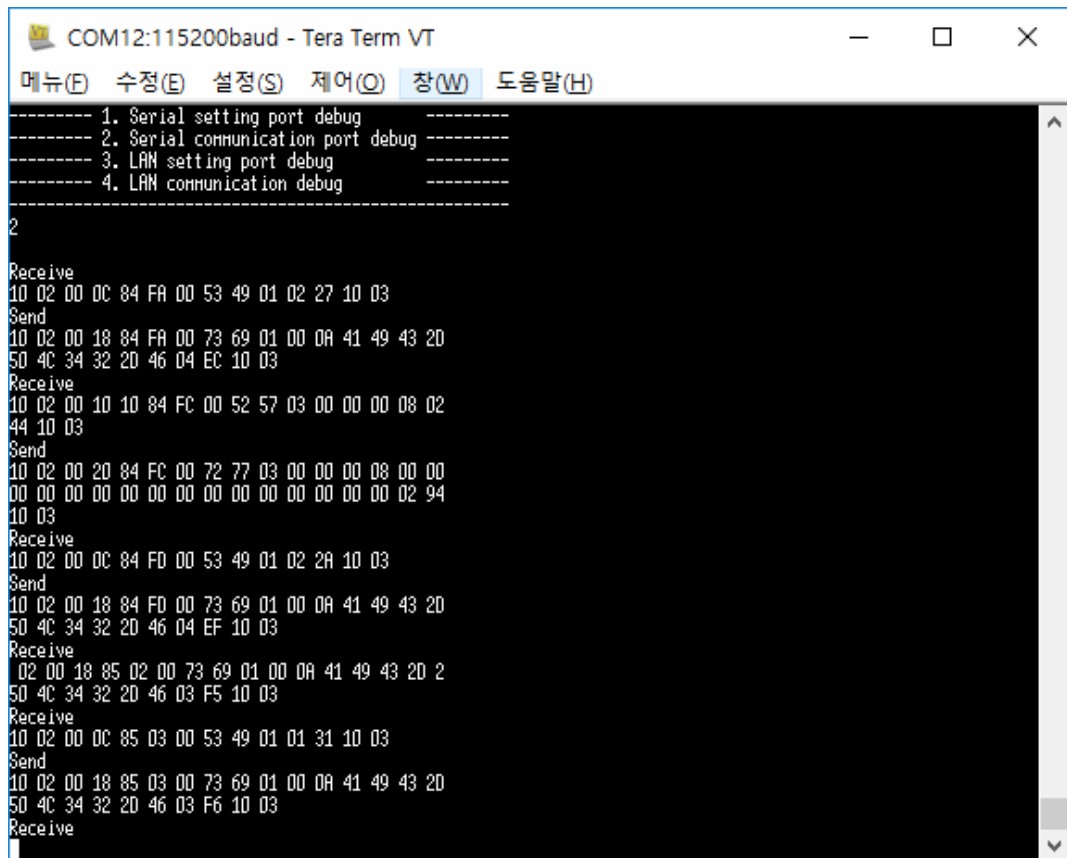


Booting 을 하며 중요 메시지를 표시한 상태이며 m 이나 h 또는 ?를 입력하면 다음과 같이 Menu 가 화면에 나타납니다.



1~4 를 입력하면 해당 포트의 통신이 디버깅 되며 0 을 입력하면 디버깅이 멈춥니다.

다음은 통신포트를 디버깅한 화면입니다.



```
COM12:115200baud - Tera Term VT
메뉴(E) 수정(E) 설정(S) 제어(O) 창(W) 도움말(H)
----- 1. Serial setting port debug -----
----- 2. Serial communication port debug -----
----- 3. LAN setting port debug -----
----- 4. LAN communication debug -----
-----
2
Receive
10 02 00 0C 84 FA 00 53 49 01 02 27 10 03
Send
10 02 00 18 84 FA 00 73 69 01 00 0A 41 49 43 20
50 4C 34 32 2D 46 04 EC 10 03
Receive
10 02 00 10 10 84 FC 00 52 57 03 00 00 00 08 02
44 10 03
Send
10 02 00 2D 84 FC 00 72 77 03 00 00 00 08 00 00
00 00 00 00 00 00 00 00 00 00 00 00 02 94
10 03
Receive
10 02 00 0C 84 FD 00 53 49 01 02 2A 10 03
Send
10 02 00 18 84 FD 00 73 69 01 00 0A 41 49 43 20
50 4C 34 32 2D 46 04 EF 10 03
Receive
10 02 00 18 85 02 00 73 69 01 00 0A 41 49 43 20 2
50 4C 34 32 2D 46 03 F5 10 03
Receive
10 02 00 0C 85 03 00 53 49 01 01 31 10 03
Send
10 02 00 18 85 03 00 73 69 01 00 0A 41 49 43 20
50 4C 34 32 2D 46 03 F6 10 03
Receive
```

콘솔 기능은 기본적으로 LOADER 포트와 연결 시 동작하며, LOADER 포트와 통신을 시작한 후에는 RS-232C 포트가 동일하게 동작합니다.

3.5.8 제어기 설정 메뉴

제어기를 설정하기 위해 사용합니다. 사용법은 2.6 제어기 설정을 참조하기 바랍니다.

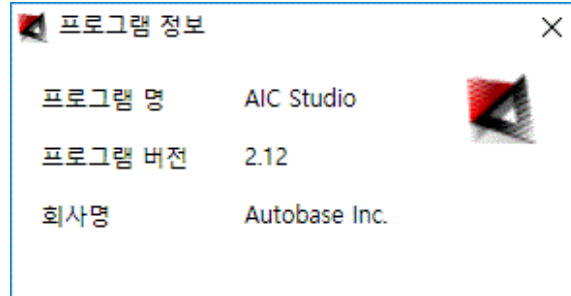
3.5.9 값보기 메뉴

값보기 모드/편집기 모드를 결정하기 위해 사용합니다. 사용법은 2.7 값보기(디버깅)을 참조하기 바랍니다.

3.6 도움말 메뉴

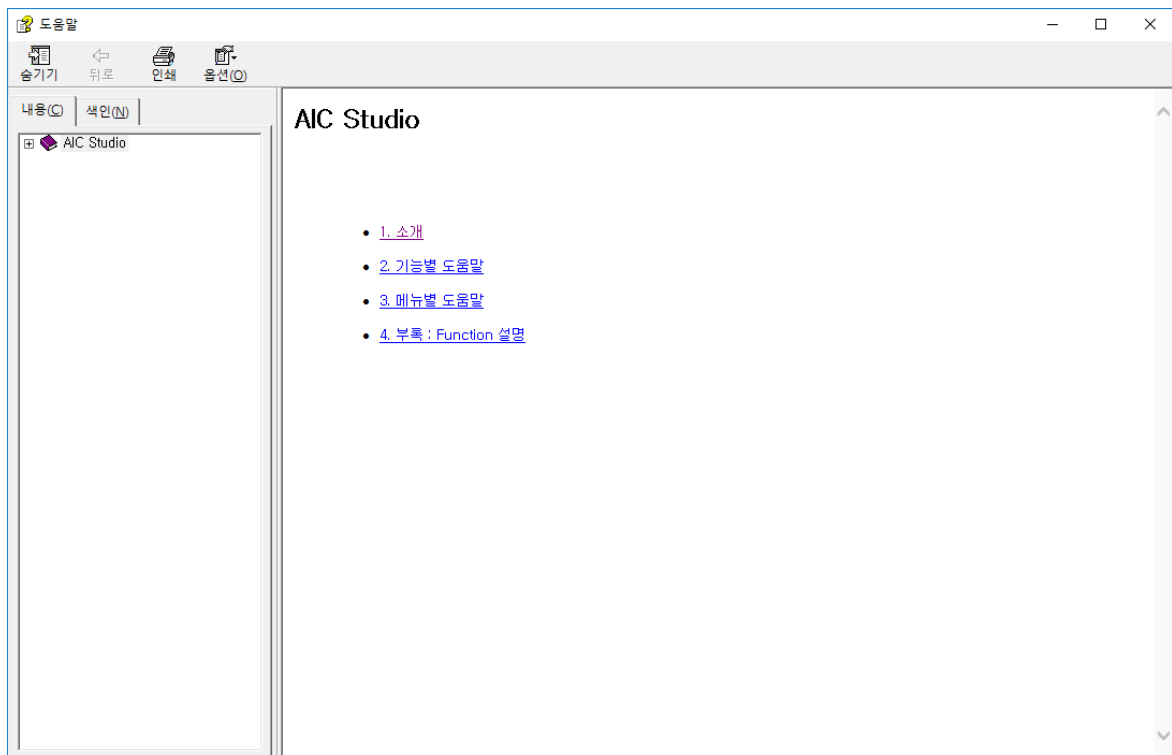
3.6.1 프로그램 정보 메뉴

프로그램 정보 메뉴를 선택하면 다음과 같이 프로그램 정보 대화 상자가 화면에 나타납니다.



3.6.2 도움말 메뉴

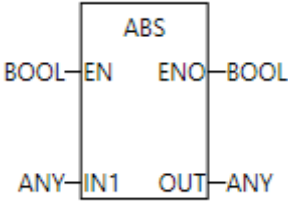
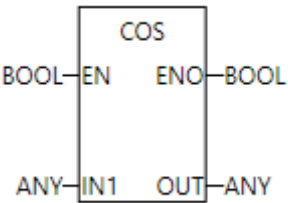
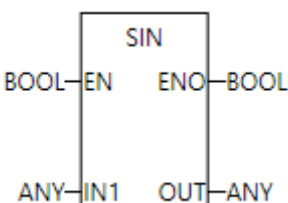
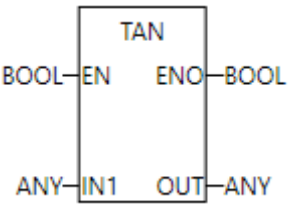
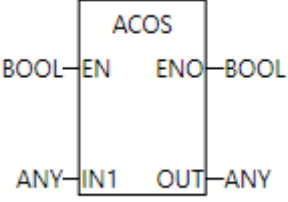
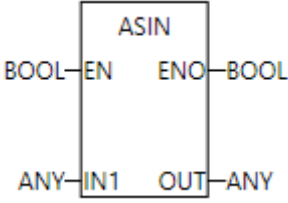
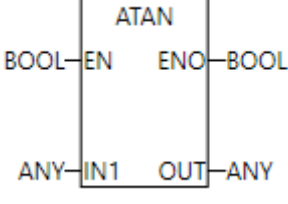
도움말 메뉴를 선택하면 다음과 같이 도움말이 화면에 나타납니다.

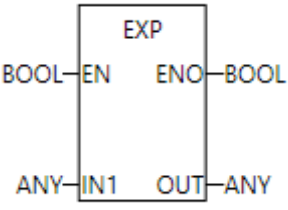
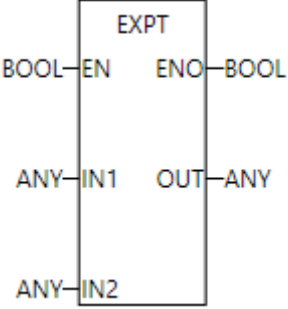
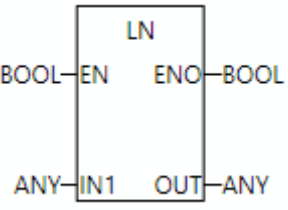
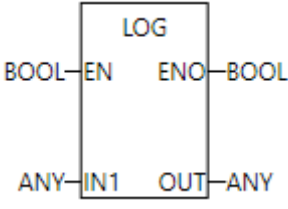
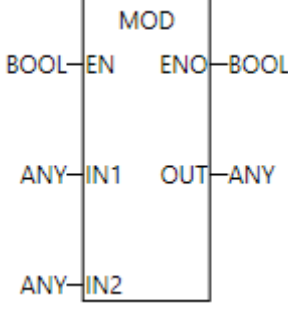
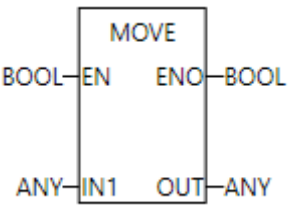


부록 1 : Function 설명

4.1 수치연산

이름	Function	설명
ADD		<p>IN1 과 IN2 를 더하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
SUB		<p>IN1 에서 IN2 를 빼서 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
MUL		<p>IN1 과 IN2 를 곱하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
DIV		<p>IN1 에서 IN2 를 나누어 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

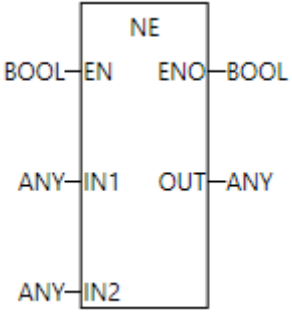
ABS		<p>IN1의 절대값을 구하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>
COS		<p>IN1의 Cosine 값을 구하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>
SIN		<p>IN1의 Sine 값을 구하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>
TAN		<p>IN1의 Tangent 값을 구하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>
ACOS		<p>IN1의 Arc Cosine 값을 구하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>
ASIN		<p>IN1의 Arc Sine 값을 구하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>
ATAN		<p>IN1의 Arc Tangent 값을 구하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>

EXP		<p>IN1의 EXP 값을 구하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>
EXPT		<p>IN1에 IN2를 지수승하여 OUT에 넣습니다($OUT = IN1^{IN2}$).</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>
LN		<p>IN1의 LN 연산을 수행하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>
LOG		<p>IN1의 LOG 연산을 수행하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>
MOD		<p>IN1을 IN2로 나눈 나머지를 구하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>
MOVE		<p>IN1의 값을 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>

SQRT		<p>IN1 의 SQRT 연산을 수행하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
------	--	---

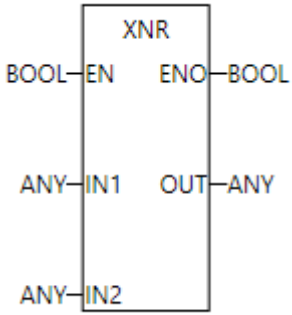
4.2 비교

이름	Function	설명
EQ		<p>IN1 과 IN2 의 값이 같으면 OUT 에 1 을 아니면 0 을 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
GE		<p>IN1 의 값이 IN2 의 값 보다 크거나 같으며 OUT 에 1 을 아니면 0 을 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
GT		<p>IN1 의 값이 IN2 의 값 보다 크면 OUT 에 1 을 아니면 0 을 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

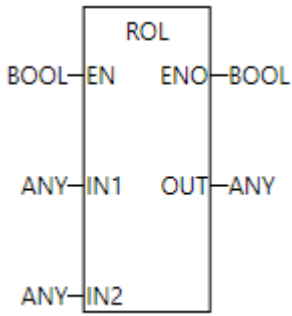
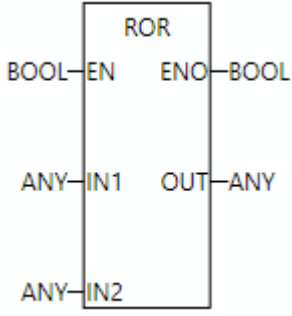
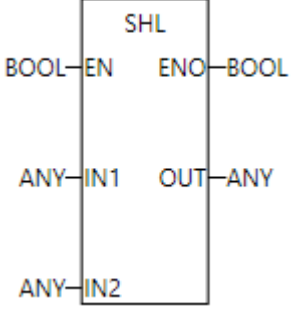
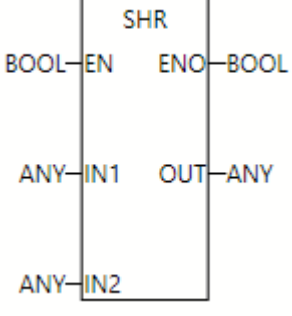
LE		<p>IN1의 값이 IN2의 값 보다 작거나 같으면 OUT에 1을 아니면 0을 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>
LT		<p>IN1의 값이 IN2의 값 보다 작으면 OUT에 1을 아니면 0을 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>
NE		<p>IN1과 IN2의 값이 다르면 OUT에 1을 아니면 0을 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>

4.3 비트연산

이름	Function	설명
AND		<p>IN1 과 IN2 의 AND 값을 구하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
NOT		<p>IN1 의 NOT 값을 구하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
OR		<p>IN1 과 IN2 의 OR 값을 구하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
NOR		<p>IN1 과 IN2 의 NOR 값을 구하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
XOR		<p>IN1 과 IN2 의 XOR 값을 구하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

XNR		<p>IN1 과 IN2 의 XNR 값을 구하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
-----	---	--

4.4 비트이동

이름	Function	설명
ROL		<p>IN1의 값을 IN2만큼 왼쪽으로 회전하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>
ROR		<p>IN1의 값을 IN2만큼 오른쪽으로 회전하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>
SHL		<p>IN1의 값을 IN2만큼 왼쪽으로 이동하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>
SHR		<p>IN1의 값을 IN2만큼 오른쪽으로 이동하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN값은 그대로 ENO에 전달됩니다.</p>

4.5 선택

이름	Function	설명
MAX		<p>IN1 과 IN2 중 큰값을 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
MIN		<p>IN1 과 IN2 중 작은값을 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
SEL		<p>SEL 이 0 이면 IN1 의 값을 1 이면 IN2 의 값을 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
LIMIT		<p>IN 이 MIN 과 MAX 사이에 있으면 IN 의 값을 OUT 에 넣습니다.</p> <p>IN 이 MIN 보다 작으면 MIN 값을 OUT 에 넣고, IN 이 MAX 보다 크면 MAX 값을 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

4.7 각도 변환

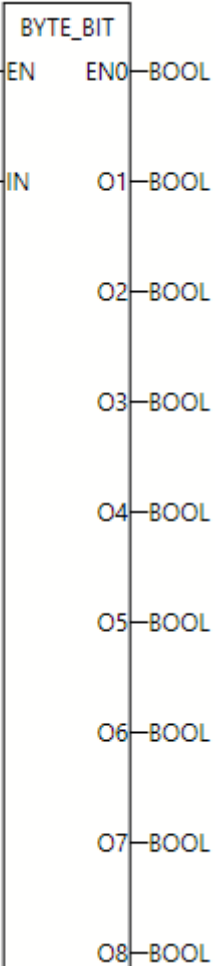
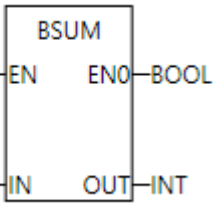
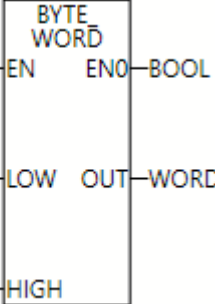
이름	Function	설명
DEG		<p>IN1의 값(Radian 값)을 각도로 변환하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>
RAD		<p>IN1의 값(각도)을 Radian 값으로 변환하여 OUT에 넣습니다.</p> <p>EN이 1일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>

4.8 데이터 교환

이름	Function	설명
SWAP		<p>IN 의 값의 상위 하위를 바꾸어 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
XCHG		<p>2 개의 입력 값을 서로 교환하여 OUT 에 넣습니다(IN1->OUT2, IN2->OUT1).</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

4.9 데이터 처리

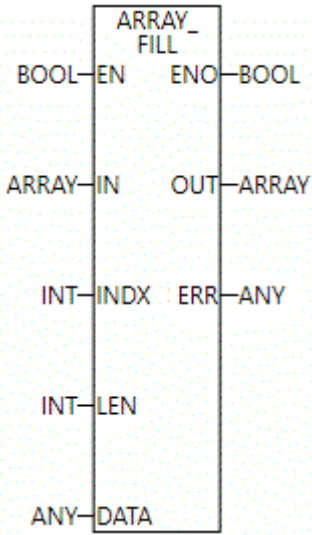
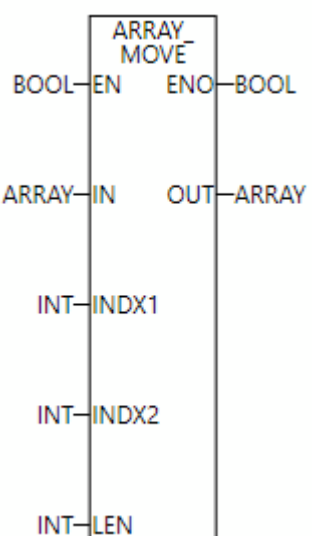
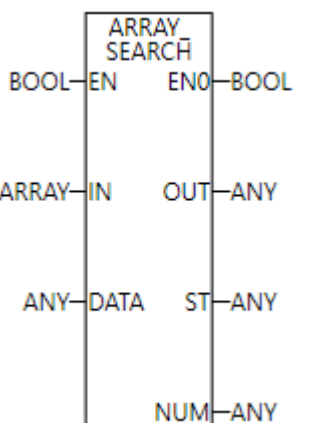
이름	Function	설명																								
BIT_BYTE	<div style="border: 1px solid black; padding: 5px; margin: 5px;"> <p style="text-align: center;">BIT_BYTE</p> <p>BOOL-EN ENO-BOOL</p> <p>BOOL-IN1 OUT-BYTE</p> <p>BOOL-IN2</p> <p>BOOL-IN3</p> <p>BOOL-IN4</p> <p>BOOL-IN5</p> <p>BOOL-IN6</p> <p>BOOL-IN7</p> <p>BOOL-IN8</p> </div>	<p>8 개의 BIT(IN1~IN8)을 BYTE 로 묶어서 OUT 에 넣습니다.</p> <table border="1" style="margin: 10px auto; text-align: center;"> <thead> <tr> <th colspan="8">OUT</th> </tr> <tr> <th>IN</th> <th>IN</th> <th>IN</th> <th>IN</th> <th>IN</th> <th>IN</th> <th>IN</th> <th>IN</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> </tr> </tbody> </table> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>	OUT								IN	IN	IN	IN	IN	IN	IN	IN	8	7	6	5	4	3	2	1
OUT																										
IN	IN	IN	IN	IN	IN	IN	IN																			
8	7	6	5	4	3	2	1																			

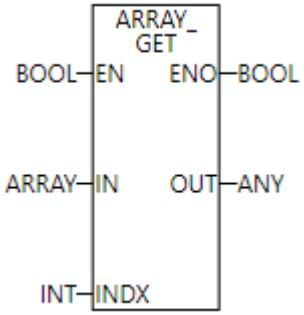
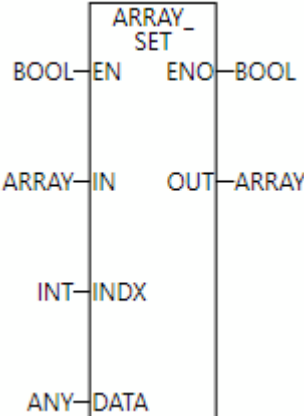
<p>BYTE_BIT</p>		<p>BYTE 를 8 개의 BIT 로 분리하여 O1~O8 에 넣습니다.</p> <table border="1" data-bbox="794 629 1401 734"> <tr> <th colspan="8">IN</th> </tr> <tr> <td>O8</td> <td>O7</td> <td>O6</td> <td>O5</td> <td>O4</td> <td>O3</td> <td>O2</td> <td>O1</td> </tr> </table> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>	IN								O8	O7	O6	O5	O4	O3	O2	O1
IN																		
O8	O7	O6	O5	O4	O3	O2	O1											
<p>BSUM</p>		<p>IN 값에 ON 이된 비트 갯수를 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>																
<p>BYTE_WORD</p>		<p>2 개의 BYTE 를 WORD 로 묶어 OUT 에 넣습니다.</p> <table border="1" data-bbox="794 1536 1401 1653"> <tr> <th colspan="2">OUT</th> </tr> <tr> <td>HIGH</td> <td>LOW</td> </tr> </table> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>	OUT		HIGH	LOW												
OUT																		
HIGH	LOW																	

WORD_BYTE		<p>WORD 를 2 개의 BYTE 로 분리하여 LOW, HIGH 에 넣습니다.</p> <table border="1" data-bbox="794 315 1399 432"> <tr> <td colspan="2" style="text-align: center;">IN</td> </tr> <tr> <td style="text-align: center;">HIGH</td> <td style="text-align: center;">LOW</td> </tr> </table> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>	IN		HIGH	LOW
IN						
HIGH	LOW					
WORD_DWORD		<p>2 개의 WORD 를 DWORD 로 묶어 OUT 에 넣습니다.</p> <table border="1" data-bbox="794 710 1399 826"> <tr> <td colspan="2" style="text-align: center;">OUT</td> </tr> <tr> <td style="text-align: center;">HIGH</td> <td style="text-align: center;">LOW</td> </tr> </table> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>	OUT		HIGH	LOW
OUT						
HIGH	LOW					
DWORD_WORD		<p>DWORD 를 2 개의 WORD 로 분리하여 LOW, HIGH 에 넣습니다.</p> <table border="1" data-bbox="794 1104 1399 1220"> <tr> <td colspan="2" style="text-align: center;">IN</td> </tr> <tr> <td style="text-align: center;">HIGH</td> <td style="text-align: center;">LOW</td> </tr> </table> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>	IN		HIGH	LOW
IN						
HIGH	LOW					
DEC		<p>IN 값에 1 만큼 감소하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>				
INC		<p>IN 값에 1 만큼 증가하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>				

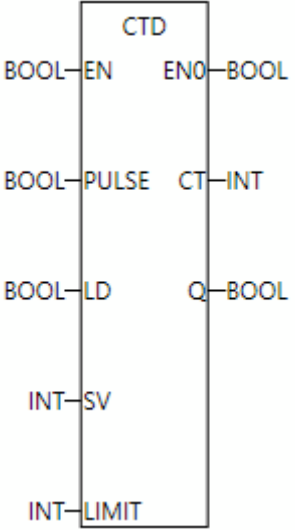
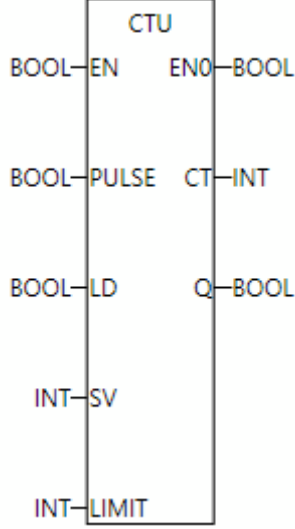
4.10 배열연산

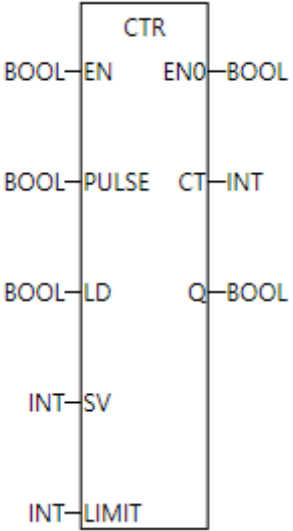
이름	Function	설명
ARRAY_AVERAGE		<p>입력 ARRAY 의 INDEX 부터 LEN 만큼의 평균값을 구하여 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
ARRAY_SWAP		<p>입력 ARRAY 의 INDEX 부터 LEN 만큼의 데이터를 상위와 하위를 바꾸어 OUT 에 넣습니다. 나머지 데이터는 모두 IN 값과 동일하게 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
ARRAY_ISEQUAL		<p>IN1 ARRAY 의 INDEX1 부터 LEN 만큼의 데이터와 IN2 ARRAY 의 INDEX2 부터 LEN 만큼의 데이터를 비교하여 같으면 1 을 다르면 0 을 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

<p>ARRAY_FILL</p>		<p>입력 ARRAY 의 INDEX 부터 LEN 만큼을 DATA 로 채워 OUT 에 넣습니다. 나머지 데이터는 모두 IN 값과 동일하게 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
<p>ARRAY_MOVE</p>		<p>입력 ARRAY 의 INDEX1 부터 LEN 만큼의 데이터를 INDEX2 의 위치로 이동하여 OUT 에 넣습니다. 나머지 데이터는 모두 IN 값과 동일하게 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
<p>ARRAY_SEARCH</p>		<p>입력 ARRAY 에 DATA 의 값이 있는지를 탐색하여 있으면 OUT 에 1 을 없으면 0 을 넣습니다.</p> <p>ST : 발견된 DATA 의 첫 번째 위치</p> <p>NUM : 발견된 DATA 의 갯수</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

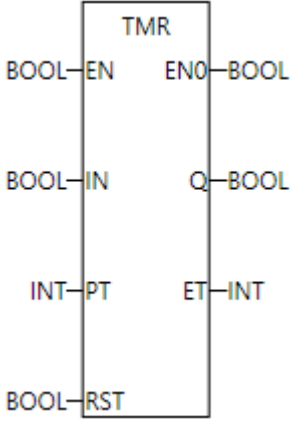
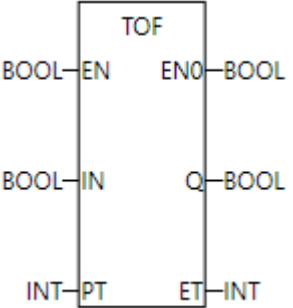
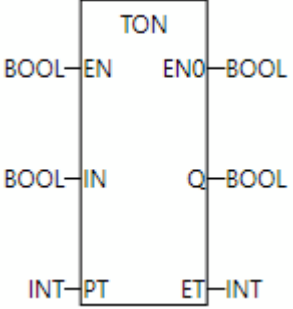
<p>ARRAY_GET</p>		<p>입력 ARRAY 데이터 INDEX 위치의 값을 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
<p>ARRAY_SET</p>		<p>입력 ARRAY 데이터 INDEX 위치의 값을 DATA 값으로 채워 OUT 에 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

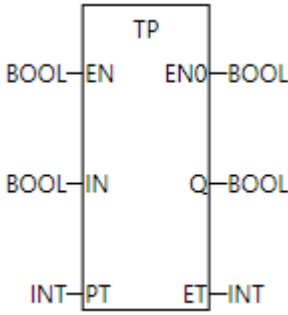
4.11 카운터

이름	Function	설명
CTD	 <p>The diagram shows a vertical rectangular block labeled 'CTD'. On the left side, there are four input ports: 'EN' (Boolean), 'PULSE' (Boolean), 'LD' (Boolean), and 'SV' (Integer). On the right side, there are three output ports: 'ENO' (Boolean), 'CT' (Integer), and 'Q' (Boolean). At the bottom, there is an input port 'LIMIT' (Integer).</p>	<p>PULSE 가 0 에서 1 로 바뀌면 Counter 의 값을 1 감소하여 CT 에 넣습니다.</p> <p>LD 가 1 이면 SV 값을 Counter 값으로 넣고, LIMIT 가 되면 Counter 는 더 이상 감소하기 않고 고정되며 Q 에 1 을 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
CTU	 <p>The diagram shows a vertical rectangular block labeled 'CTU'. On the left side, there are four input ports: 'EN' (Boolean), 'PULSE' (Boolean), 'LD' (Boolean), and 'SV' (Integer). On the right side, there are three output ports: 'ENO' (Boolean), 'CT' (Integer), and 'Q' (Boolean). At the bottom, there is an input port 'LIMIT' (Integer).</p>	<p>PULSE 가 0 에서 1 로 바뀌면 Counter 의 값을 1 증가하여 CT 에 넣습니다.</p> <p>LD 가 1 이면 SV 값을 Counter 값으로 넣고, LIMIT 가 되면 Counter 는 더 이상 증가하기 않고 고정되며 Q 에 1 을 넣습니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

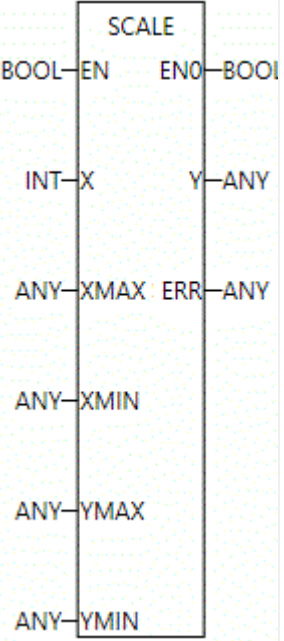
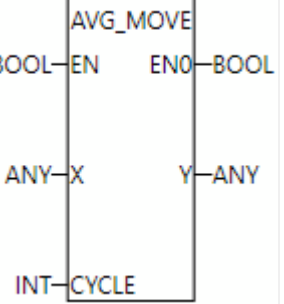
CTR	 <p>The diagram shows a vertical rectangular block labeled 'CTR'. On the left side, there are four input labels: 'EN' (with 'BOOL' to its left), 'PULSE' (with 'BOOL' to its left), 'LD' (with 'BOOL' to its left), and 'SV' (with 'INT' to its left). On the right side, there are four output labels: 'ENO' (with 'BOOL' to its right), 'CT' (with 'INT' to its right), 'Q' (with 'BOOL' to its right), and 'LIMIT' (with 'INT' to its right).</p>	<p>PULSE 가 0 에서 1 로 바뀌면 Counter 의 값을 1 증가하여 CT 에 넣습니다.</p> <p>LD 가 1 이면 SV 값을 Counter 값으로 넣고, LIMIT 가 되면 Q 에 1 을 넣고 Counter 는 여전히 변수가 지원하는 최대값까지 증가 합니다.</p> <p>Counter 가 최대값이 된 후 1 회 더 증가 하게 되면 최소값이 되어 계속 증가 동작을 수행합니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
-----	---	---

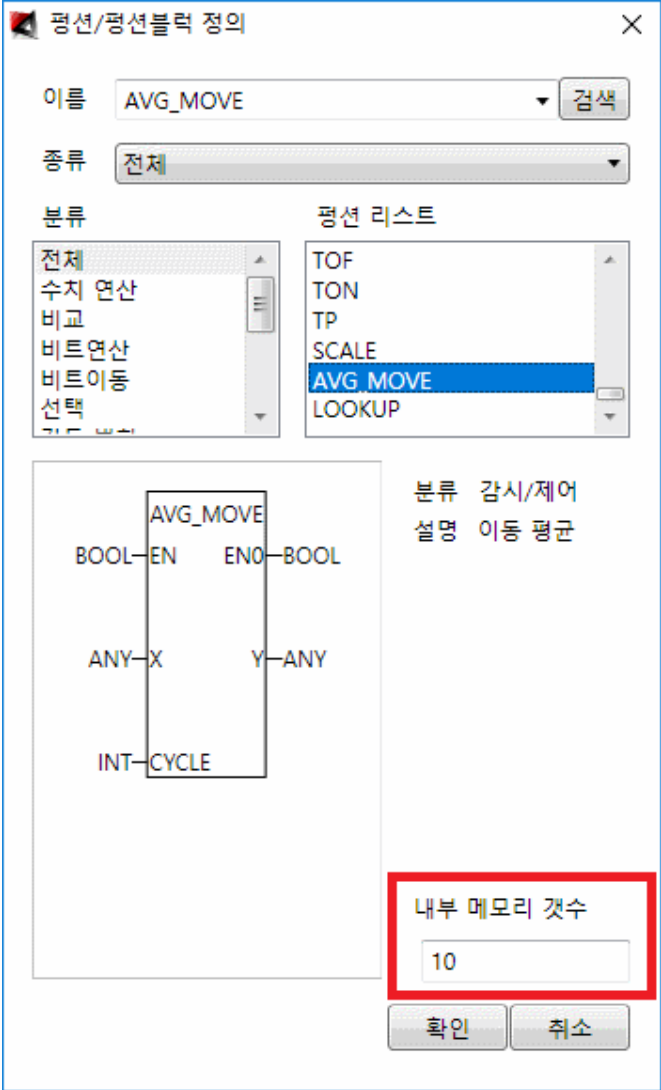
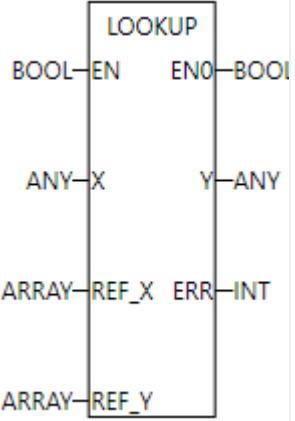
4.12 타이머

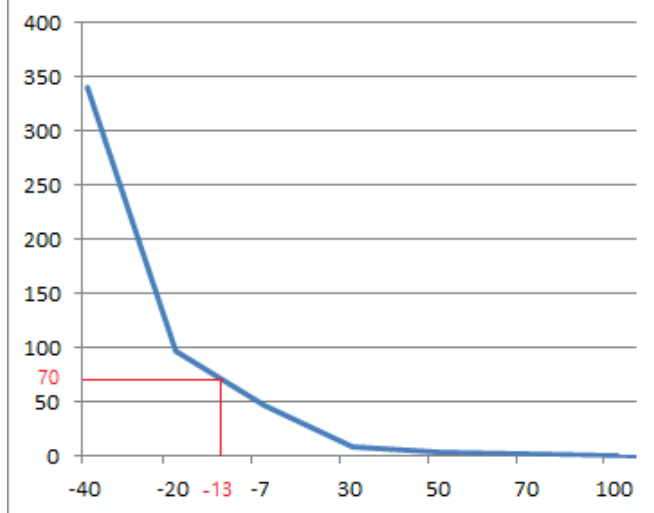
이름	Function	설명
TMR	 <p>The diagram shows a rectangular block labeled 'TMR'. On the left side, there are four inputs: 'BOOL-EN', 'BOOL-IN', 'INT-PT', and 'BOOL-RST'. On the right side, there are four outputs: 'ENO-BOOL', 'Q-BOOL', 'ET-INT', and 'ET-INT'.</p>	<p>적산 타이머</p> <p>IN 에 1 이 들어오는 시간을 누적하여 ET 에 넣습니다.</p> <p>경과 시간 ET 가 설정시간 PT(단위 ms)에 도착하기 전에 IN 이 0 이 되어도 현재의 경과 시간을 유지하다가 IN 이 다시 1 이 되면 경과 시간이 다시 증가 됩니다.</p> <p>경과 시간이 설정 시간에 도달하면 Q 가 1 이 됩니다.</p> <p>RST 에 1 을 넣으면 Q 와 ET 가 모두 0 이 됩니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
TOF	 <p>The diagram shows a rectangular block labeled 'TOF'. On the left side, there are three inputs: 'BOOL-EN', 'BOOL-IN', and 'INT-PT'. On the right side, there are three outputs: 'ENO-BOOL', 'Q-BOOL', and 'ET-INT'.</p>	<p>OFF 타이머</p> <p>IN 에 1 이 들어오는 순간 Q 는 1 이되고, IN 이 0 이 된 후부터 ET 가 증가합니다.</p> <p>경과 시간 ET 가 설정시간 PT(단위 ms)에 도달하면 Q 가 0 이 됩니다.</p> <p>만약 경과시간 ET 가 설정 시간에 도달하기 전에 IN 이 1 이되면, 경과 시간은 다시 0 으로 됩니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
TON	 <p>The diagram shows a rectangular block labeled 'TON'. On the left side, there are three inputs: 'BOOL-EN', 'BOOL-IN', and 'INT-PT'. On the right side, there are three outputs: 'ENO-BOOL', 'Q-BOOL', and 'ET-INT'.</p>	<p>ON 타이머</p> <p>IN 에 0 이 들어오는 순간 Q 는 0 이되고, IN 이 1 이 된 후부터 ET 가 증가합니다.</p> <p>경과 시간 ET 가 설정시간 PT(단위 ms)에 도달하면 Q 가 1 이 됩니다.</p> <p>만약 경과시간 ET 가 설정 시간에 도달하기 전에 IN 이 0 이되면, 경과 시간은 다시 0 으로 됩니다.</p>

		<p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>
<p>TP</p>	 <p>The diagram shows a rectangular block labeled 'TP' at the top. On the left side, there are three input ports: 'EN' (Boolean), 'IN' (Boolean), and 'PT' (Integer). On the right side, there are three output ports: 'ENO' (Boolean), 'Q' (Boolean), and 'ET' (Integer).</p>	<p>펄스 타이머</p> <p>IN 에 1 이 들어오는 순간 Q 는 1 이되고, ET 가 증가합니다.</p> <p>경과 시간 ET 가 설정시간 PT(단위 ms)에 도달하면 Q 가 0 이 됩니다.</p> <p>만약 경과시간 ET 가 설정 시간에 도달하기 전에 IN 의 값이 변화해도 무시합니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.</p>

4.13 감시/제어

이름	Function	설명
SCALE		<p>입력 X의 범위를 XMAX, XMIN 에, 출력 Y의 범위를 YMAX, YMIN 에 설정하면 해당 설정된 스케일로 변경하여 Y에 출력합니다.</p> <p>출력 방정식</p> $Y = (X - XMIN) \frac{YMAX - YMIN}{XMAX - XMIN} + YMIN$ <p>Y 값이 YMIN 보다 작으면 YMIN 값으로, YMAX 보다 크면 YMAX 값을 출력합니다.</p> <p>XMAX와 XMIN 값이 같을 경우 출력방정식의 분모가 0이기 때문에 STAT 에 8을 출력합니다.</p> <p>EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO에 전달됩니다.</p>
AVG-MOVE		<p>입력 X의 이동 평균을 구하여 출력 Y에 넣습니다. 이동 평균을 위한 데이터 취득은 CYCLE에 입력하는 주기(단위 : ms)마다 수행을 하며, 이동 평균을 위해 사용하는 메모리의 갯수는 평선/평선블럭 정의의 내부 메모리 갯수에서 설정합니다.</p>

		 <p>이름: AVG_MOVE 검색</p> <p>종류: 전체</p> <p>분류: 전체, 수치 연산, 비교, 비트 연산, 비트 이동, 선택, 기타 블록</p> <p>평선 리스트: TOF, TON, TP, SCALE, AVG MOVE, LOOKUP</p> <p>분류: 감시/제어 설명: 이동 평균</p> <p>내부 메모리 갯수: 10</p> <p>확인 취소</p>
	<p>LOOKUP</p> 	<p>입력 배열 REF_X 및 출력 배열 REF_Y를 이용해 구간 별로 선형 LOOKUP 테이블을 생성하고, 입력 X 를 적용하여 출력을 얻어냅니다.</p> <p>다음은 REF_X 에 -40, -20, -7, 30, 50, 70, 100 을 넣고, REF_Y 에 341, 97, 42, 8, 3.6, 1.7, 0.7 을 넣었을 때의 예시를 그래프로 표시한 것입니다.</p>



여기에서 만약 X에 -13 이 들어 오면 범위 계산에 의해 Y에 70 을 출력합니다.

REF_X 와 REF_Y 의 배열 원소 개수가 다를 경우 ERR 에 1 을 넣습니다.

입력 배열 REF_X 는 오름차순으로 정렬되어야 하며 배열의 원소가 같을 경우 에러 처리를 하고 ERR 에 2 를 넣습니다.

입력 X 의 값이 입력 배열 REF_X 의 범위를 벗어날 경우 최소를 벗어나면 ERR 에 3 을 최대를 벗어나면 ERR 에 4 를 넣습니다.

EN 이 1 일 경우에만 동작하며 EN 값은 그대로 ENO 에 전달됩니다.

부록 2 : 메모리 주소 입력법

제어기에서 제공하는 메모리는 I, Q, AI, AO, SYSTEM, M 등이 존재합니다. 각 메모리의 역할은 다음과 같습니다.

메모리는 각각 다음과 같은 역할을 수행합니다.

메모리 종류	역할
I	디지털 입력 포인트의 값을 입력으로 받음.
Q	디지털 출력 포인트를 감시 제어하기 위해 사용.
AI	아날로그 입력 포인트의 값을 입력으로 받음.
AO	아날로그 출력 포인트를 감시 제어하기 위해 사용.
SYSTEM	시스템 메모리를 감시 제어하기 위해 사용. 3.2.6 글로벌 변수 설정의 메모리 종류 참조
M	3.2.8 M 영역 설정 참조

AI, AO, SYSTEM 메모리는 WORD 로 사용하며 그 외 모든 메모리는 Bool, Byte, Word, Short, DWord, Int, Float 등 다양하게 표현하여 사용할 수 있습니다. 메모리 주소는 다음과 같은 형태로 표현합니다.

%메모리종류 단위 주소

- 메모리 종류 : I, Q, AI, AO, M 등 입력합니다. 단 AI 와 AO 는 WORD 로 사용하므로 단위를 생략합니다.
- 단위 : X(Bit), B(Byte), W(Word), S(Short), D(Dword), I(Int), F(Float)를 입력합니다.
- 주소 : 주소를 입력합니다.

주소 입력 예시

- %MX0016 : M 영역의 16 번째 Bit 주소의 데이터를 얻습니다. 본 주소는 M 영역의 메모리를 Bit 순으로 나타내어 16 번째 데이터로, M 영역의 Word 주소 1 번(1Word 는 16Bit)의 0 번째 Bit 를 나타냅니다. 즉 %MW01.0 과 같은 주소입니다.
- %IX0005 : I 영역의 5 번째 Bit 주소의 데이터를 얻습니다.
- %QX0003 : Q 영역의 3 번째 Bit 주소의 데이터를 얻습니다.
- %AI0004 : AI 영역의 4 번째 Word 주소의 데이터를 얻습니다.
- %MD0034 : M 영역의 34 번째와 35 번째 주소의 데이터를 DWord 로 표현합니다.
- %MF0044 : M 영역의 44 번째와 45 번째 주소의 데이터를 Float 로 표현합니다.

부록 3 : Modbus 통신을 위한 메모리 맵

Modbus 통신은 Modbus TCP(502 번 포트)와 Serial 통신을 모두 지원하며 통신을 위한 메모리 맵은 다음과 같습니다

Function Code	제어기 메모리	Modbus Address	메모리 맵
1	Q 영역	00001~00016	DO 0 번 ~ 15 번 입력
2	I 영역	10001~10016	DI 0 번 ~ 15 번 입력
3	M 영역 (설정에 따라 다름, 설정방법 3.2.8 M 영역 설정 참조)	40001	AO 0 번
		40002	AO 1 번
		40003	AO 2 번
		40004	AO 3 번
		40005	AO 4 번
		40006	AO 5 번
		40007	AO 6 번
		40008	AO 7 번
		40009	Reserved
		40010	Reserved
		40011	SyncYear
		40012	SyncMonth
		40013	SyncDay
		40014	SyncHour
		40015	SyncMinute
40016	SyncSecond		
40017	TimeSync		

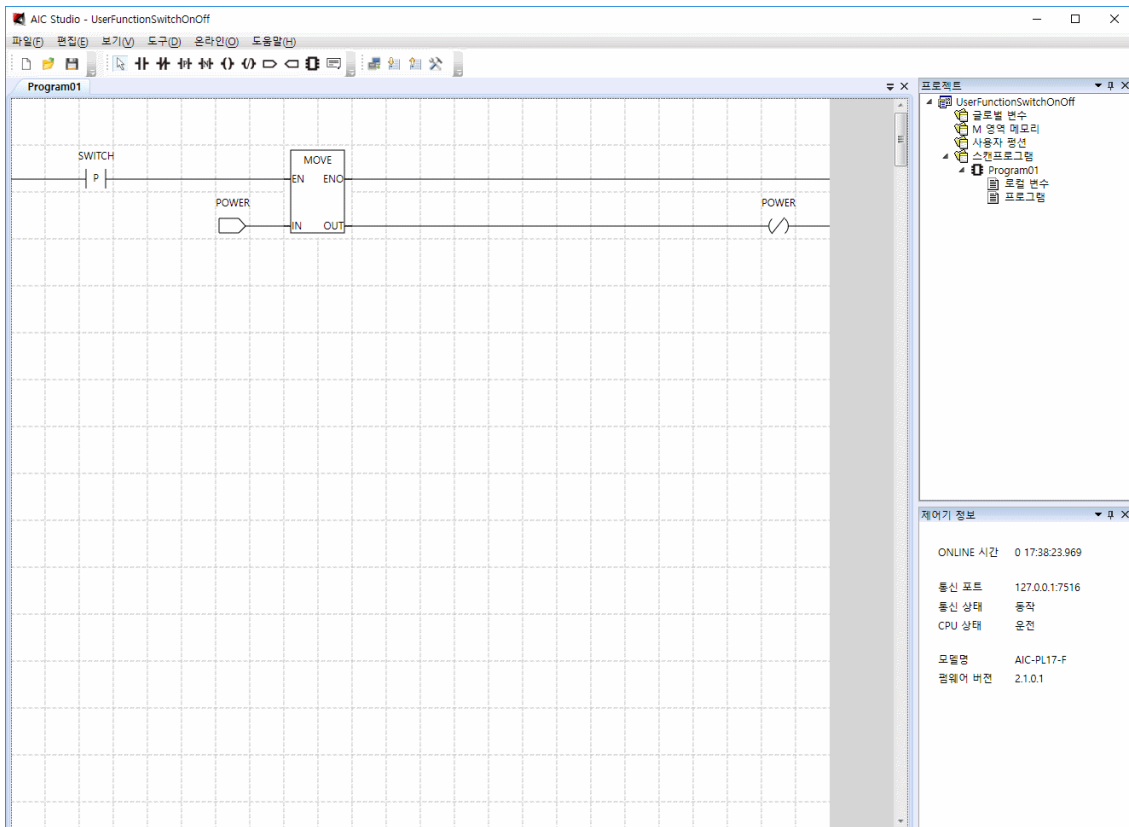
		40018	TimeSynced
4	AI 영역	30001	AI 0 번
		30002	AI 1 번
		30003	AI 2 번
		30004	AI 3 번
		30005	AI 4 번
		30006	AI 5 번
		30007	AI 6 번
		30008	AI 7 번
5	Q 영역	00001~00016	DO 0 번 ~ 15 번 출력
6	M 영역	40001~40018	M 영역 출력(Function Code 3 과 동일 구성)
15	Q 영역	00001~00016	DO 0 번 ~ 15 번 일괄 출력
16	M 영역	40001~40018	M 영역 일괄 출력(Function Code 3 과 동일 구성)

부록 4 : 사용자 평선 작성법

여기서는 사용자 평선 작성법에 대해 설명합니다.

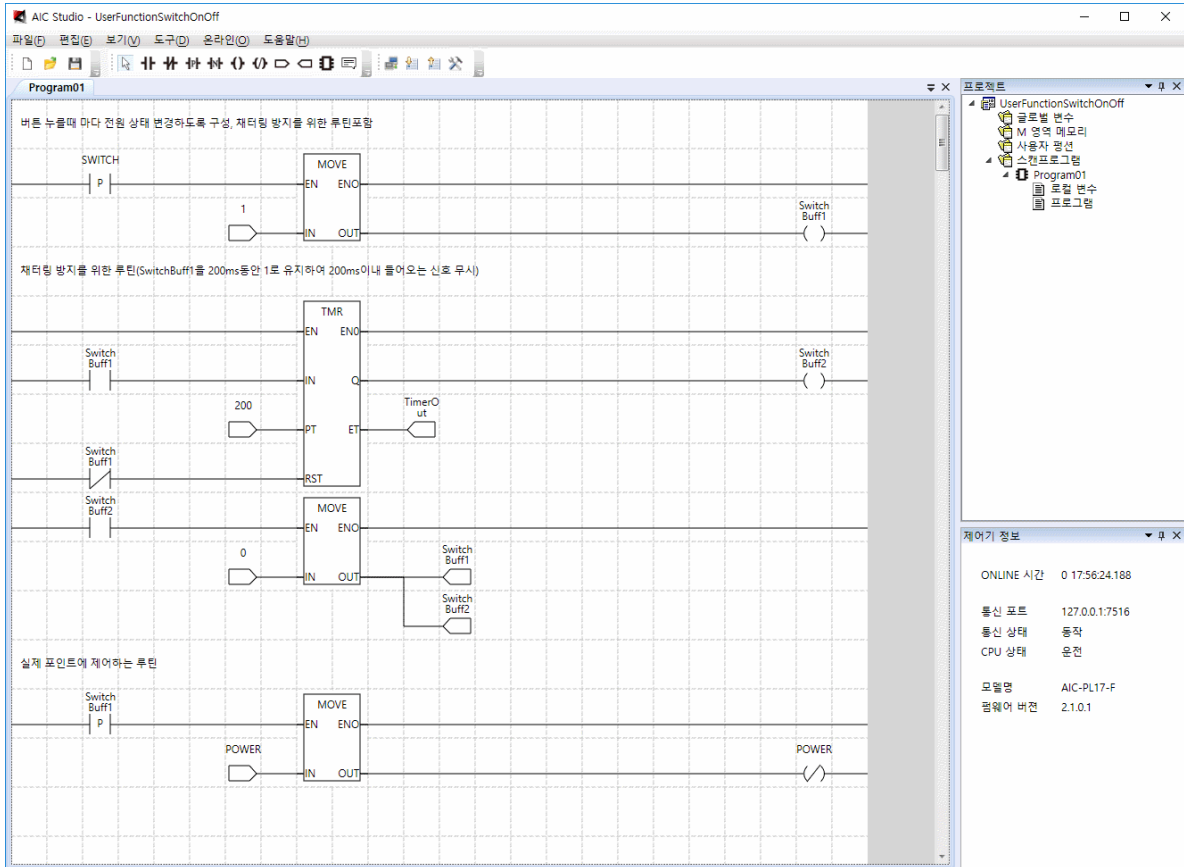
설명의 편의를 위해 Switch 를 눌렀다 떼면 디지털 출력 포인트의 상태를 바꾸는 루틴을 예시로 작성하여 사용자 평선으로 만들어 보도록 하겠습니다.

1. 로직 구현



SWICH 가 0 에서 1 로 변하면 POWER 의 상태를 바꾸어 그 기능을 구현하였습니다.

이렇게 구현을 하여 스위치를 누를때 간헐적으로 채터링이 발생하여 POWER 상태가 여러번 변하는 경우가 발생합니다. 이를 보완하기 위해 채터링 방지 루틴을 넣어 아래와 같이 구현했습니다.



SwitchBuff1 이라는 변수를 두고 TIMER 를 사용하여 200ms 동안 1로 유지 하도록 하여 200ms 이내에 들어오는 신호를 무시하도록 구성했습니다.

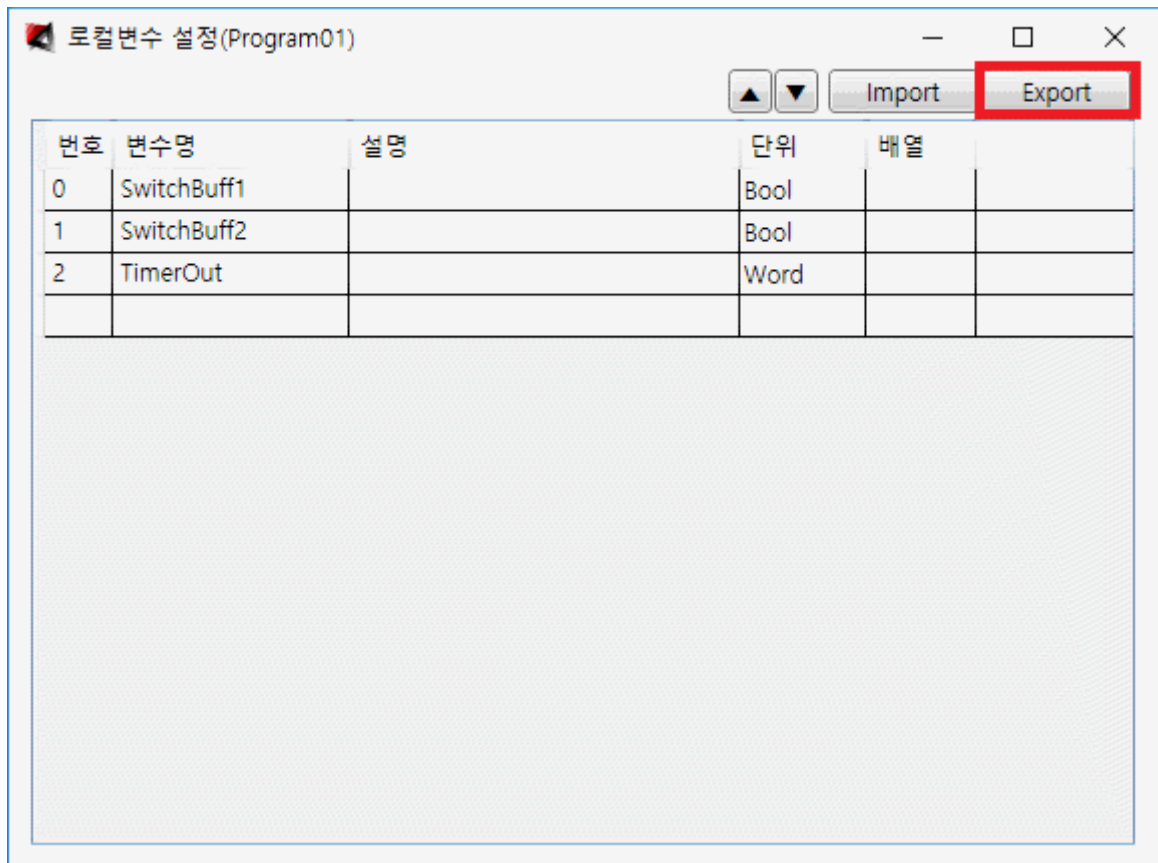
만약 이루틴을 2 개의 포인트에 적용하고자 한다면 위의 루틴을 복사하고 각 변수 이름을 다르게 만들어 적용해야 하면 됩니다. 그러나 만약 10 개 포인트에 적용하고자 한다면 복사하여 구성하기도 번거로울 뿐 아니라 향후 루틴을 수정해야 할 경우에도 많은 노력이 필요한 상황이 됩니다. 이를 편하게 하기 위해 위의 루틴을 사용자 평선으로 작성하고 그 평선을 불러와 프로그램을 간단히 작성하도록 합니다.

참고로 사용자 평선에서는 값보기 기능을 지원하지 않기 때문에 여기에서 값보기 기능을 이용하여 충분히 테스트 한 후에 사용자 평선 작성하는 순으로 작업을 진행하기 바랍니다.

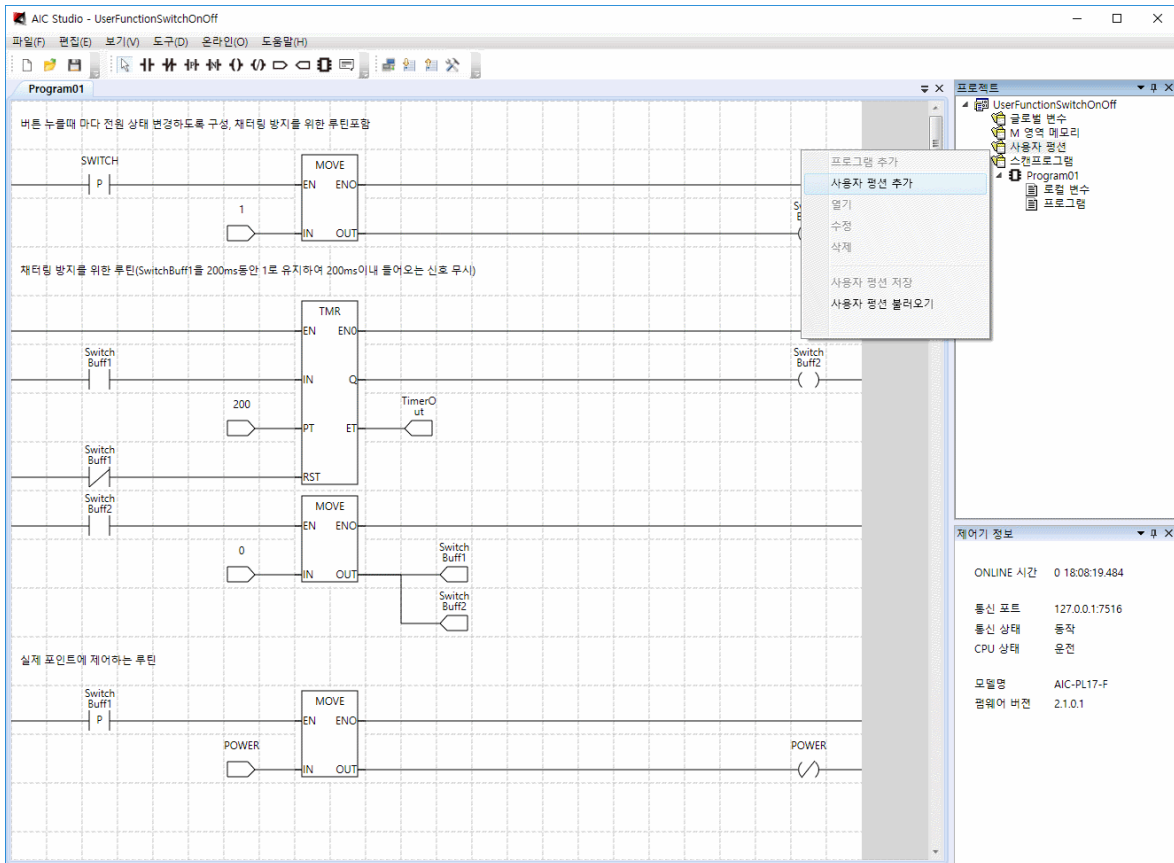
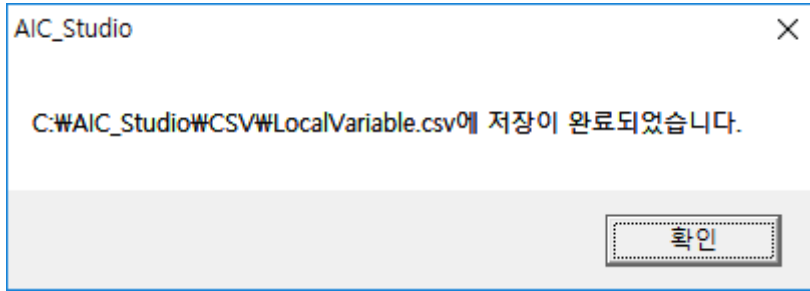
2. 사용자 평선 작성 및 사용

사용자 평선 작성하기 전에 프로그램에서 로컬 변수는 그대로 사용자 평선에서 사용해야 하므로 .csv 파일로 Export 를 해두도록 합니다.

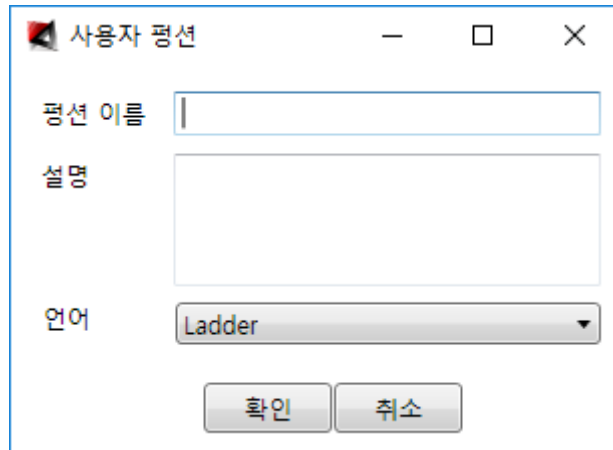
편집 메뉴의 부메뉴 로컬변수 설정을 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.



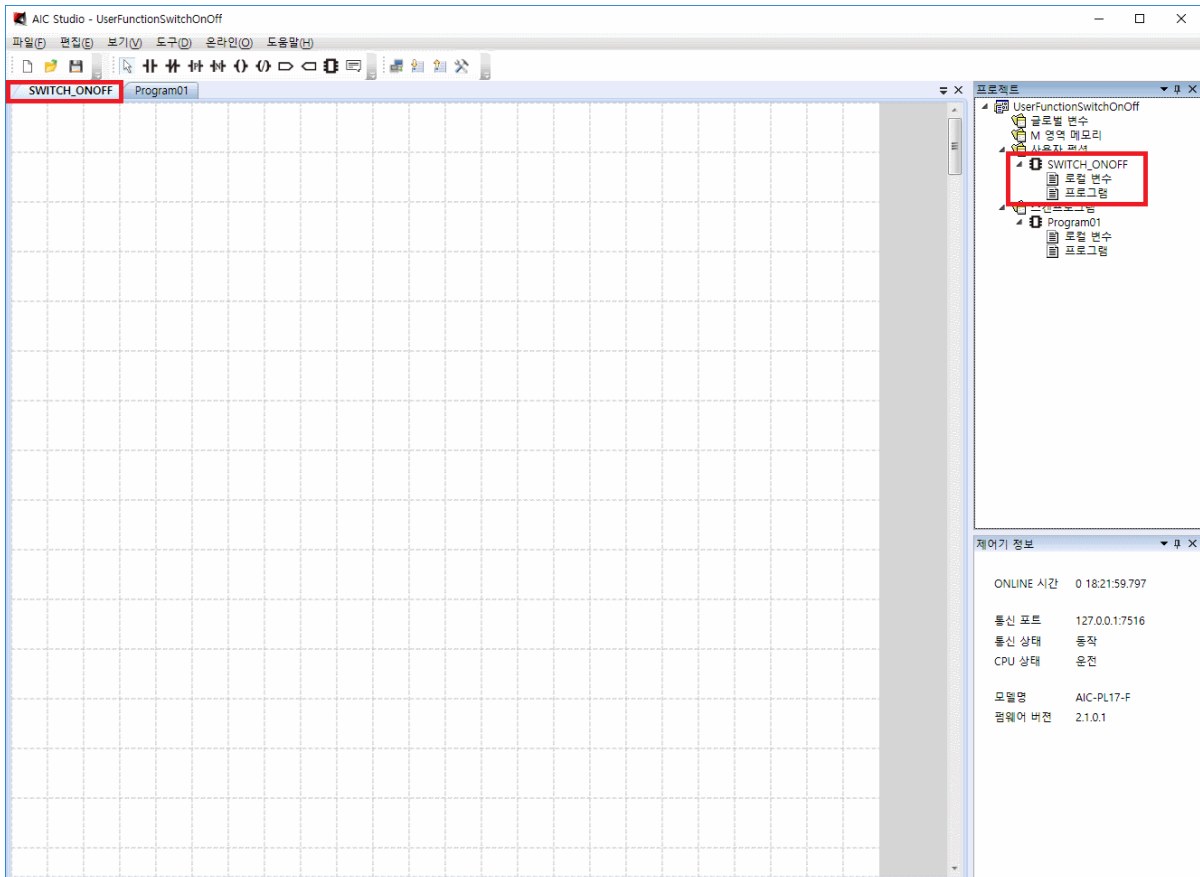
Export 를 클릭하면 다음과 같은 대화 상자가 화면에 나타나며 .csv 파일로 Export 가 됩니다.



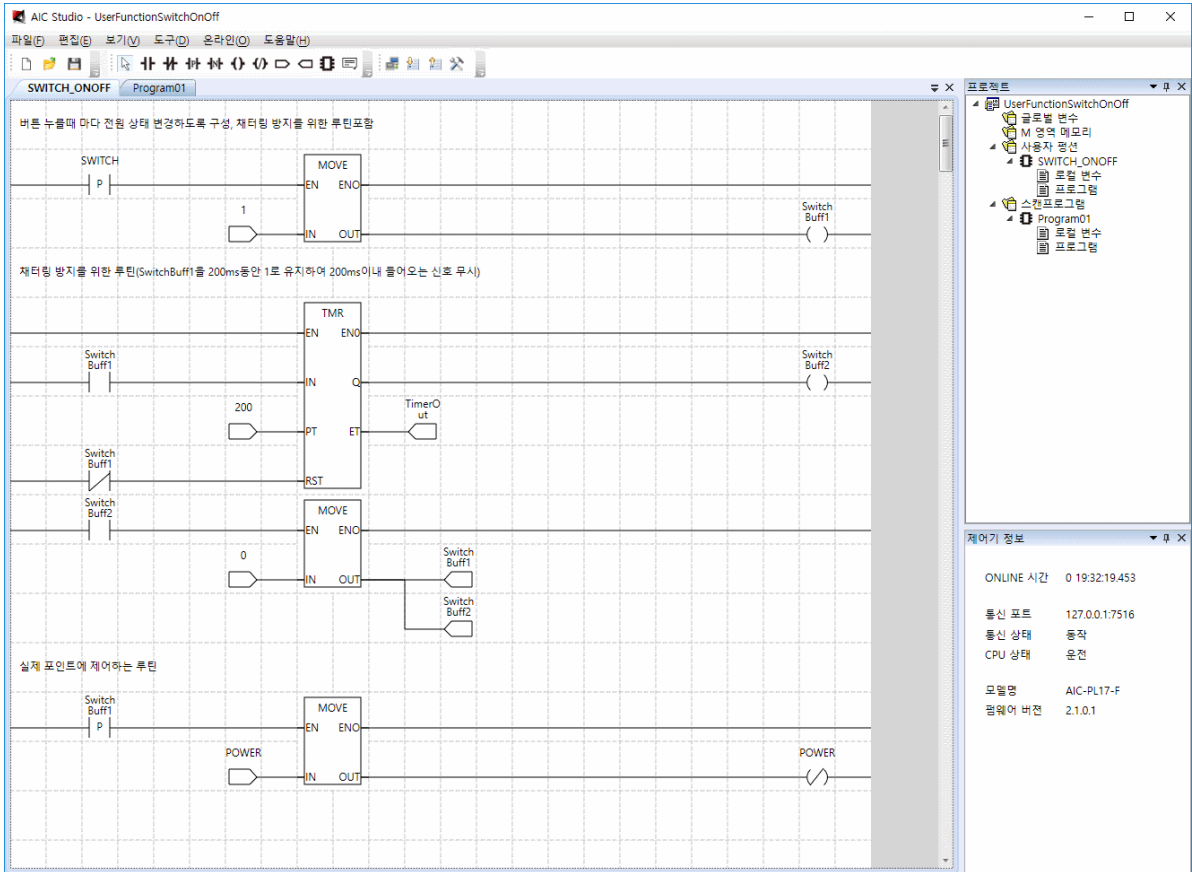
프로젝트의 사용자 평선에서 마우스 오른쪽 버튼을 클릭하여 생성된 메뉴에서 사용자 평선 추가를 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.



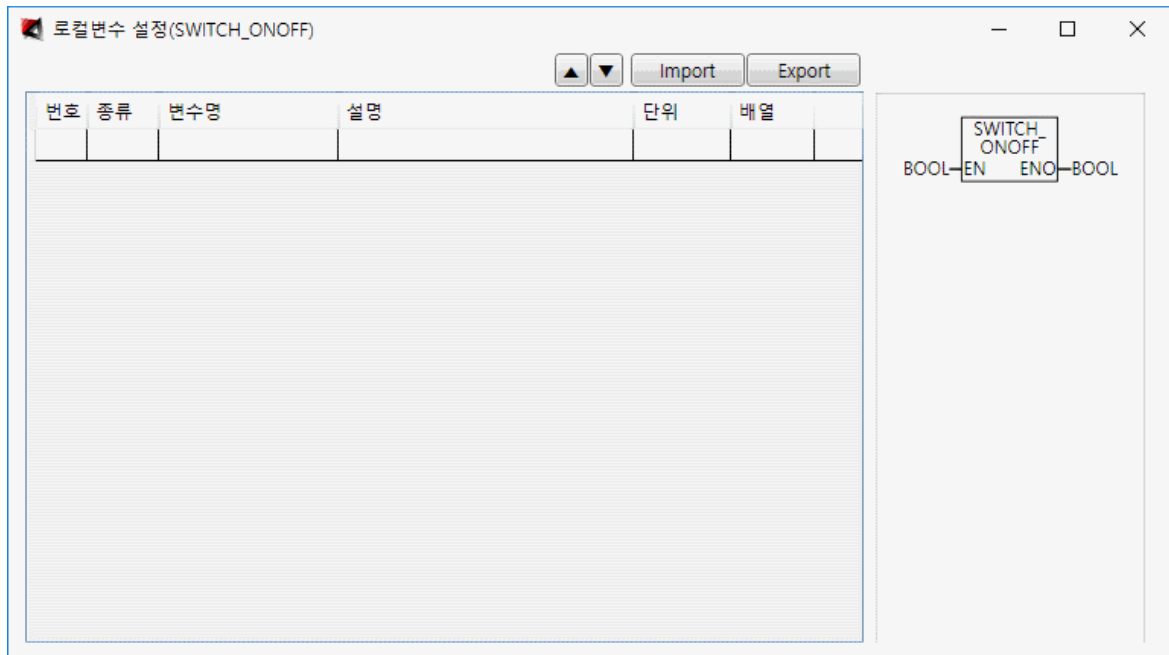
평선 이름에 SWITCH_ONOFF 라고 입력하고 설명에 버튼을 누를때 마다 전원 상태 변경하도록 구성, 채터링 방지를 위한 루틴포함이라고 입력한 후 확인을 선택하면 다음과 같이 SWITCH_ONOFF 라는 평선을 생성한 후 편집이 가능하도록 화면에 표시합니다.



프로그램에 있는 로직을 모두 복사하여 SWITCH_ONOFF 에 붙여 넣습니다.

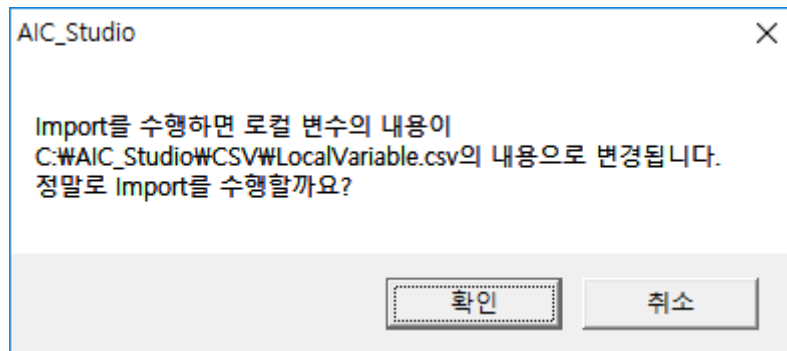


편집 메뉴의 부메뉴 로컬 변수 정의를 선택하면 다음과 같은 대화 상자가 생성됩니다.

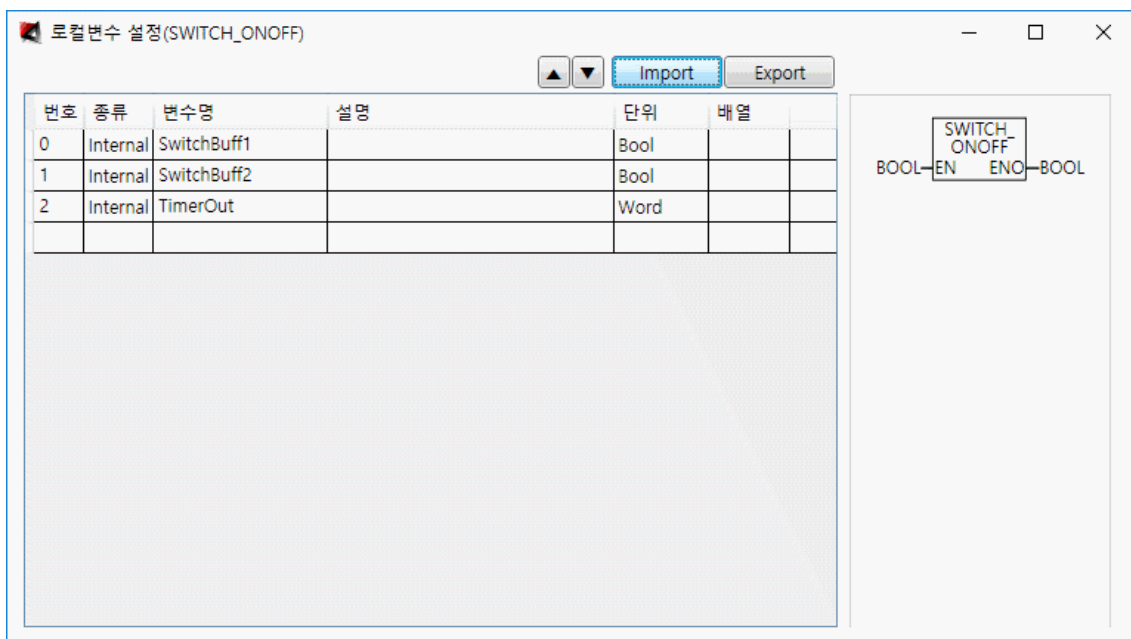


사용자 평선의 로컬변수는 프로그램의 로컬변수와 달이 리스트에 종류와 우측에 평선 그림이 추가되어 있습니다.

Import 를 선택하면 다음과 같은 대화 상자가 생성됩니다.

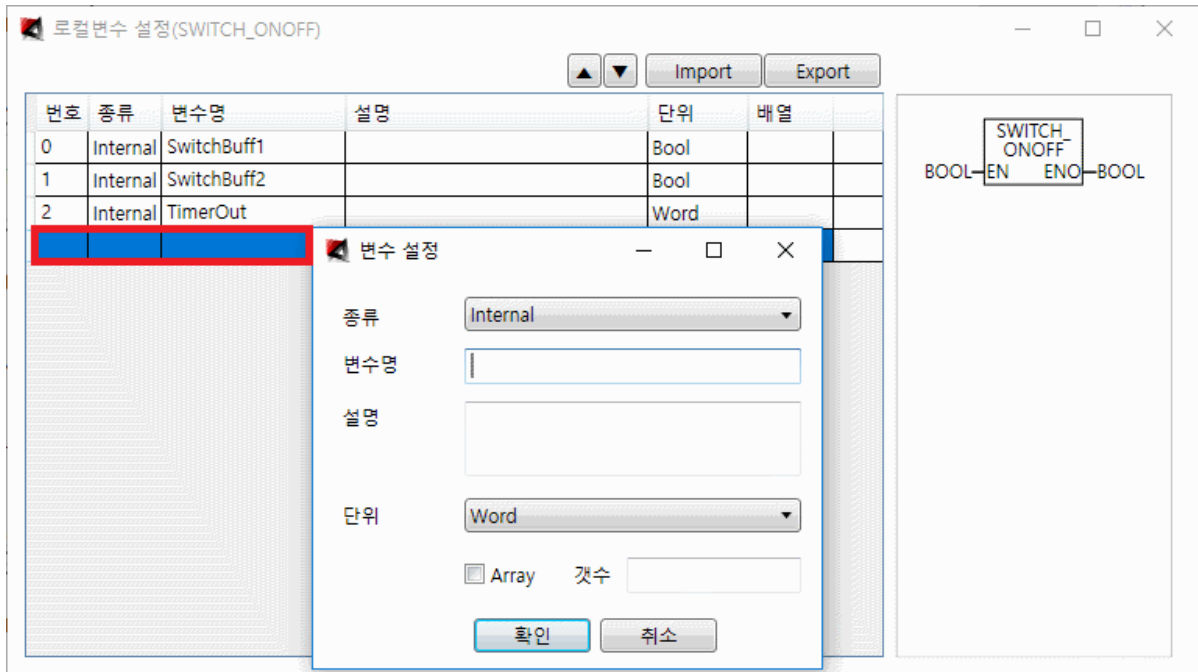


확인을 선택하면 다음과 같이 프로그램 로컬변수에서 Export 한 내용이 그대로 Import 됩니다.



이제 SWITCH 와 POWER 만 처리하면 됩니다.

리스트의 마지막 줄에서 마우스로 더블클릭을 하면 다음과 같은 대화 상자가 생성됩니다.

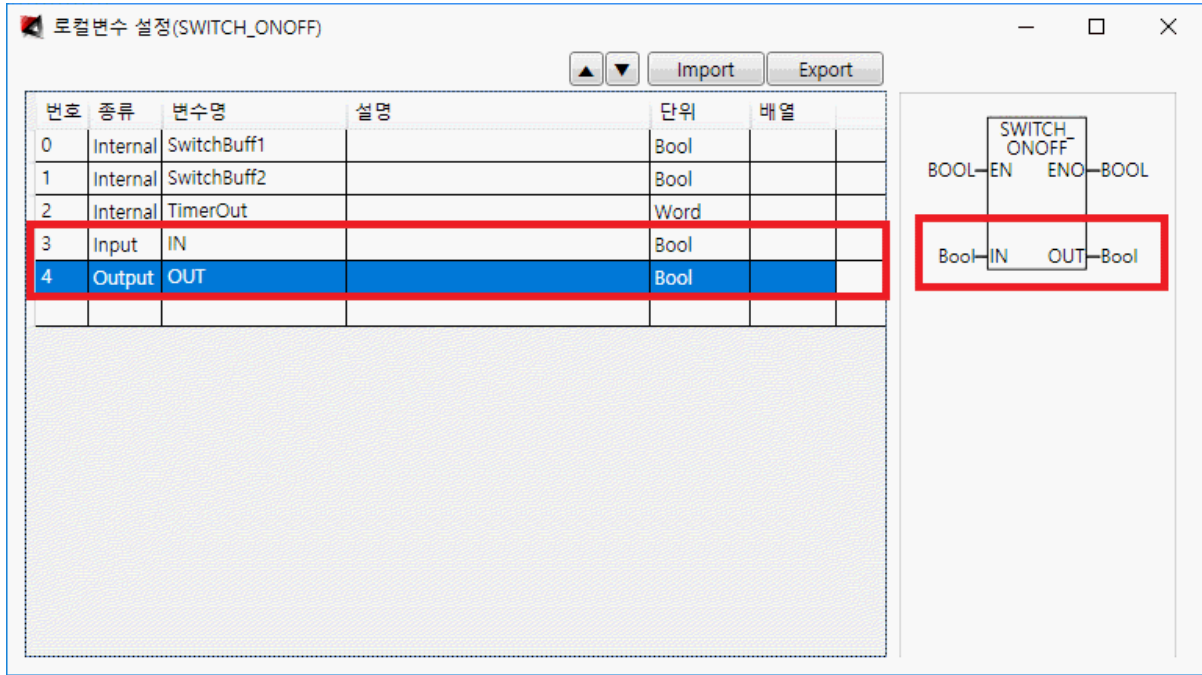


종류를 Input 으로 설정하고 변수명에 IN, 단위를 Bool 로 선택한 후 확인을 선택하며

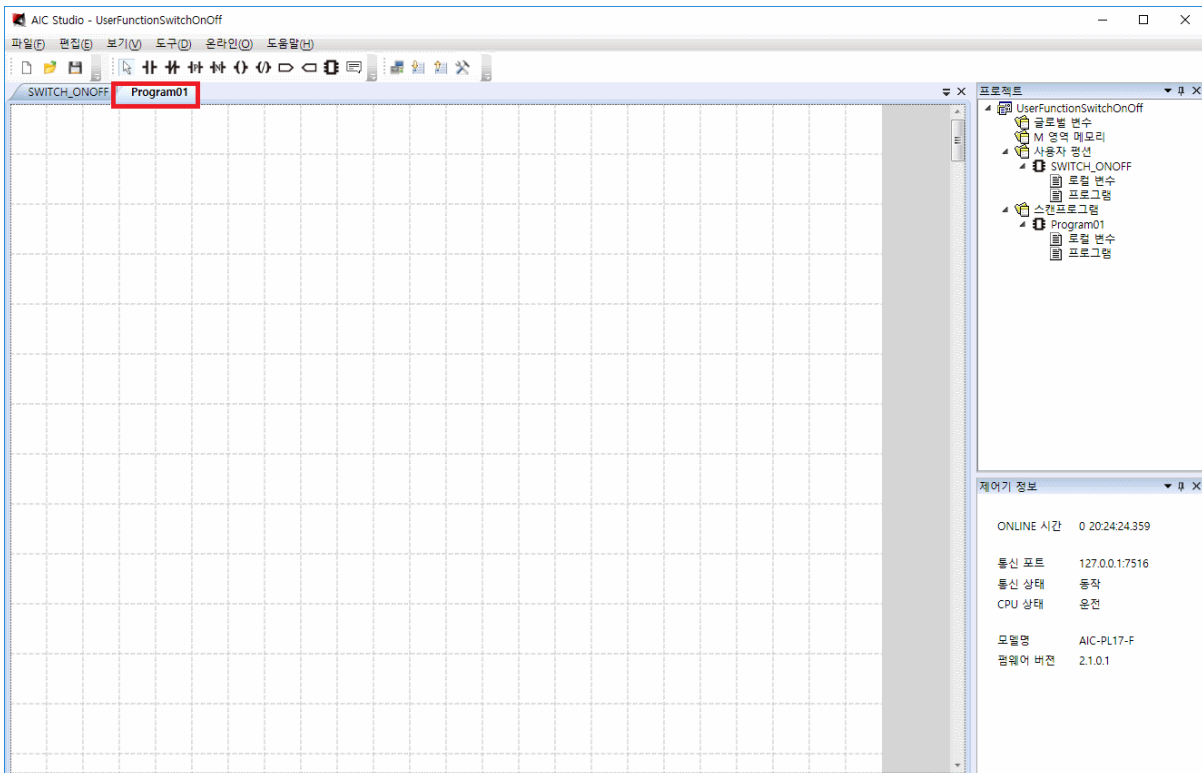
IN(SWITCH 용 입력인자, 내부변수는 일반적으로 소문자로 작성하나 인자의 경우는 대문자로 작성)이 추가 됩니다.

다시한번 리스트의 마지막 줄에서 마우스로 더블클릭을 하여 생성된 대화 상자에서 종류를 Output 으로 설정하고 변수명에 OUT, 단위를 Bool 로 선택한 후 확인을 선택하며 OUT(POWER 용 출력인자)이 추가 됩니다.

다음은 IN 변수와 OUT 변수를 추가한 모양입니다.

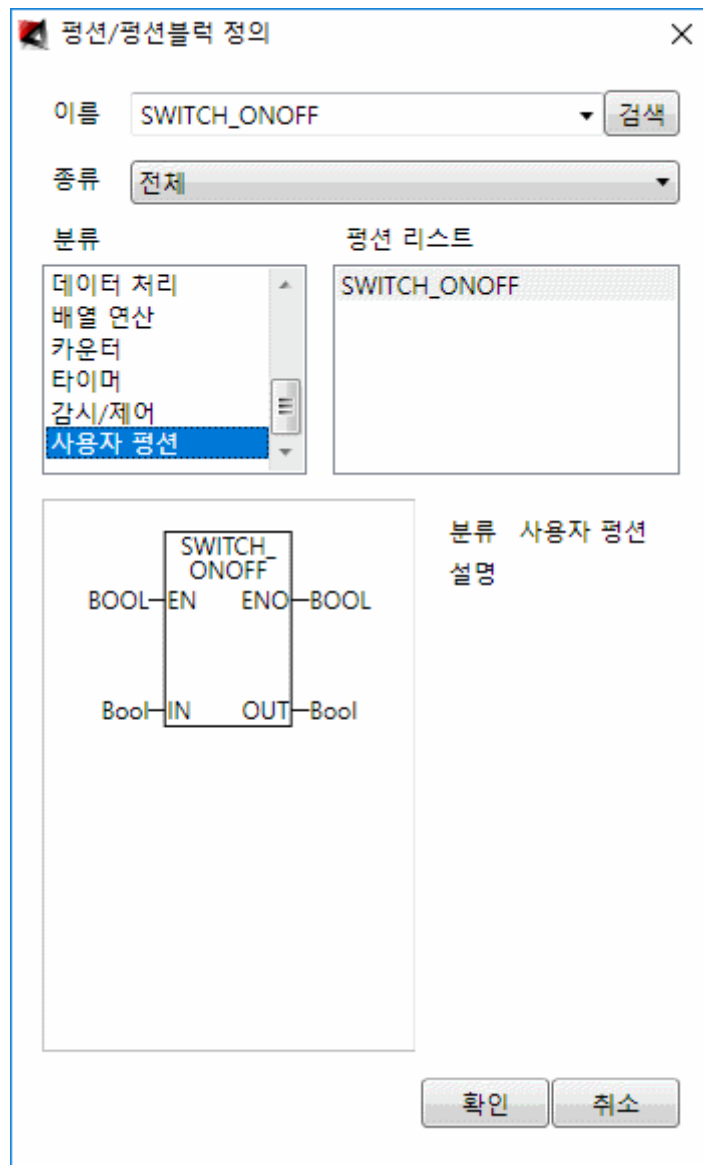


화면 우측의 평선 그림의 인자가 추가된 것을 알 수 있습니다.



이제 프로그램(Program01)을 선택하여 모두 선택한 후 삭제를 합니다.

평선을 선택하여 화면에 마우스로 클릭하면 다음과 같은 대화 상자가 화면에 나타납니다.



사용자 평선을 추가함에 의해 분류에 사용자 평선이 추가되었으며 평선 리스트에

SWITCH_ONOFF 가 추가되어 있습니다.

다음은 확인을 선택하여 SWITCH_ONOFF 를 넣고 SWITCH 와 POWER 를 이용하여 로직을 완성한

모양입니다.

AIC Studio - UserFunctionSwitchOnOff

파일(F) 편집(E) 보기(V) 도구(D) 온라인(O) 도움말(H)

Program01

SWITCH ONOFF

EN ENO

IN OUT

SWITCH POWER

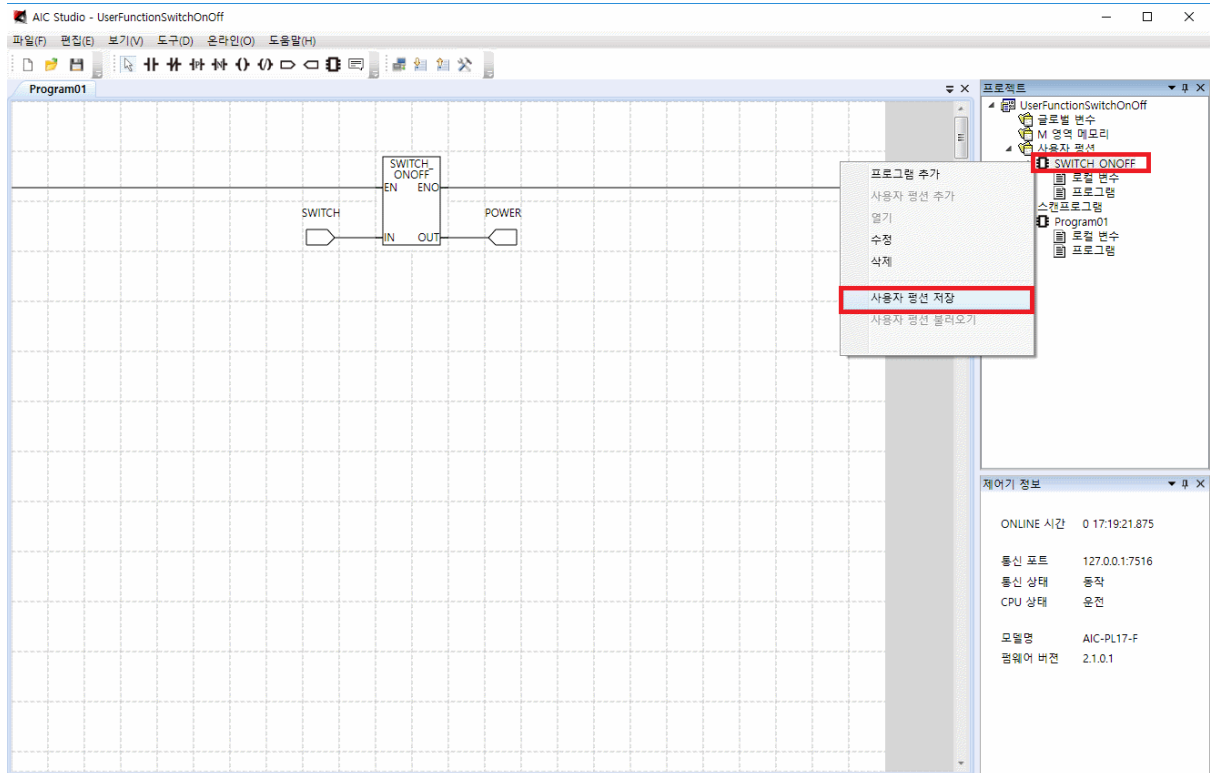
프로젝트

- UserFunctionSwitchOnOff
 - 클러블 변수
 - M 영역 메모리
 - 사용자 명선
 - SWITCH_ONOFF
 - 로컬 변수
 - 프로그램
 - 스칼라 프로그램
 - Program01
 - 로컬 변수
 - 프로그램

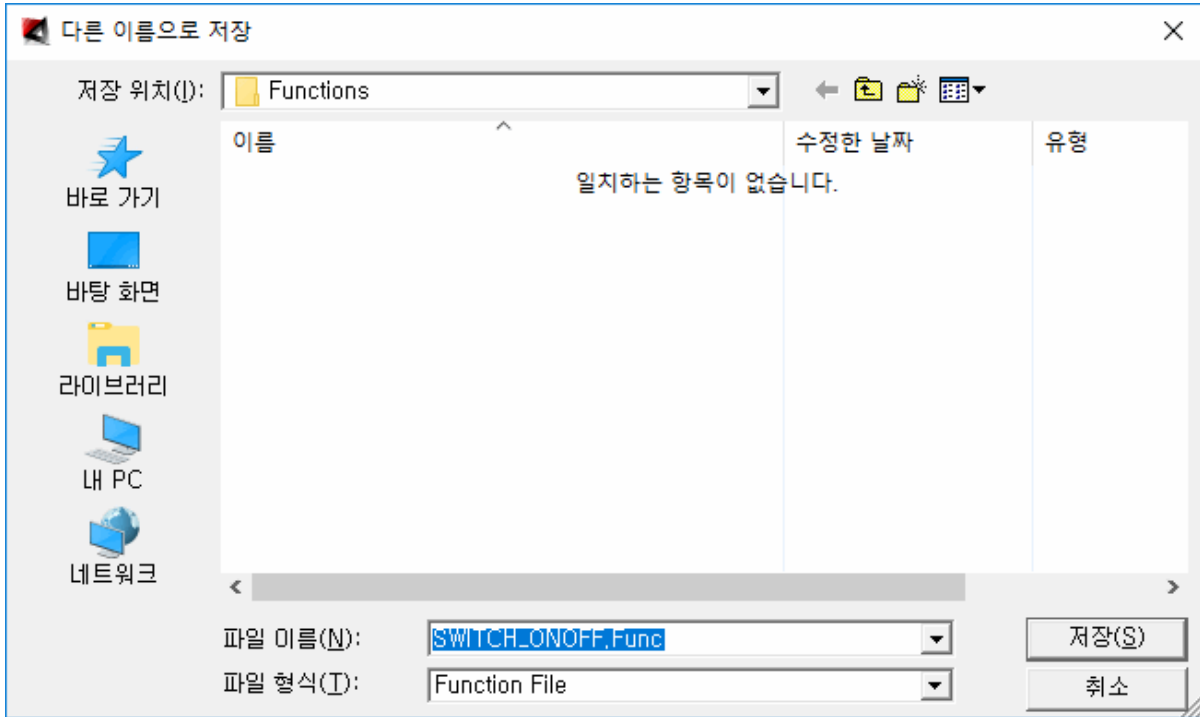
제어기 정보

ONLINE 시간	0 17:16:41.125
통신 포트	127.0.0.1:7516
통신 상태	동작
CPU 상태	운전
모델명	AIC-PL17-F
펌웨어 버전	2.1.0.1

3. 사용자 평선 재사용

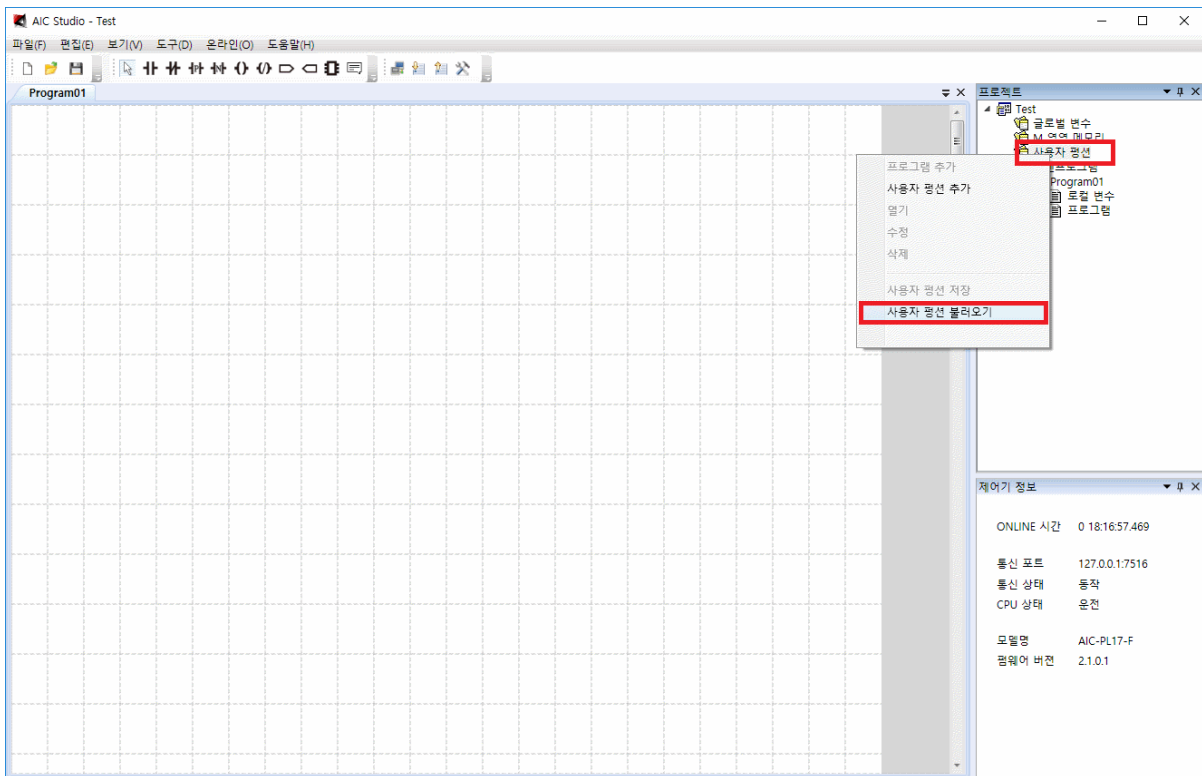


재사용하고자하는 사용자 평선에서 마우스 오른쪽 버튼을 클릭하여 생성된 팝업 메뉴에서 사용자 평선 저장을 선택하면 다음과 같은 대화 상자가 생성됩니다.

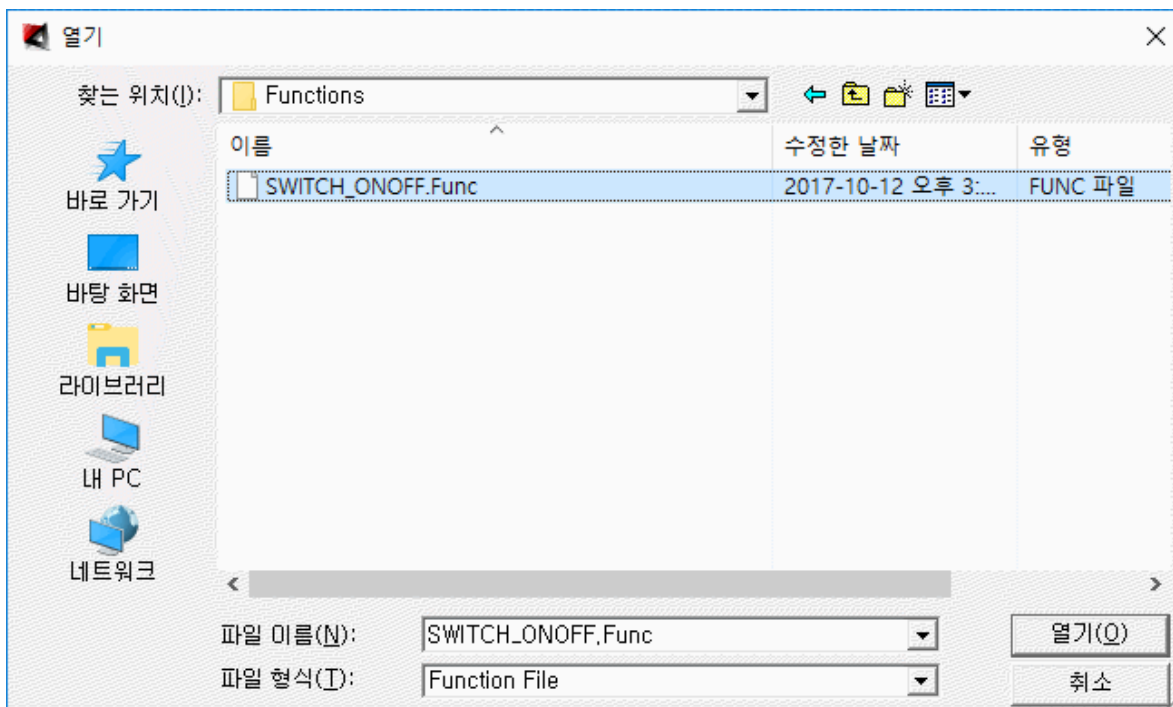


저장하고자 하는 위치를 선택한 후 저장 버튼을 누르면 저장이 됩니다(그대로 저장하면

C:\AIC_Studio\Functions 에 저장됩니다).



Test 라는 프로젝트를 새로 만들고 사용자 평선에서 마우스 오른쪽 버튼을 클릭하여 생성된 메뉴에서 사용자 평선 불러오기를 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.



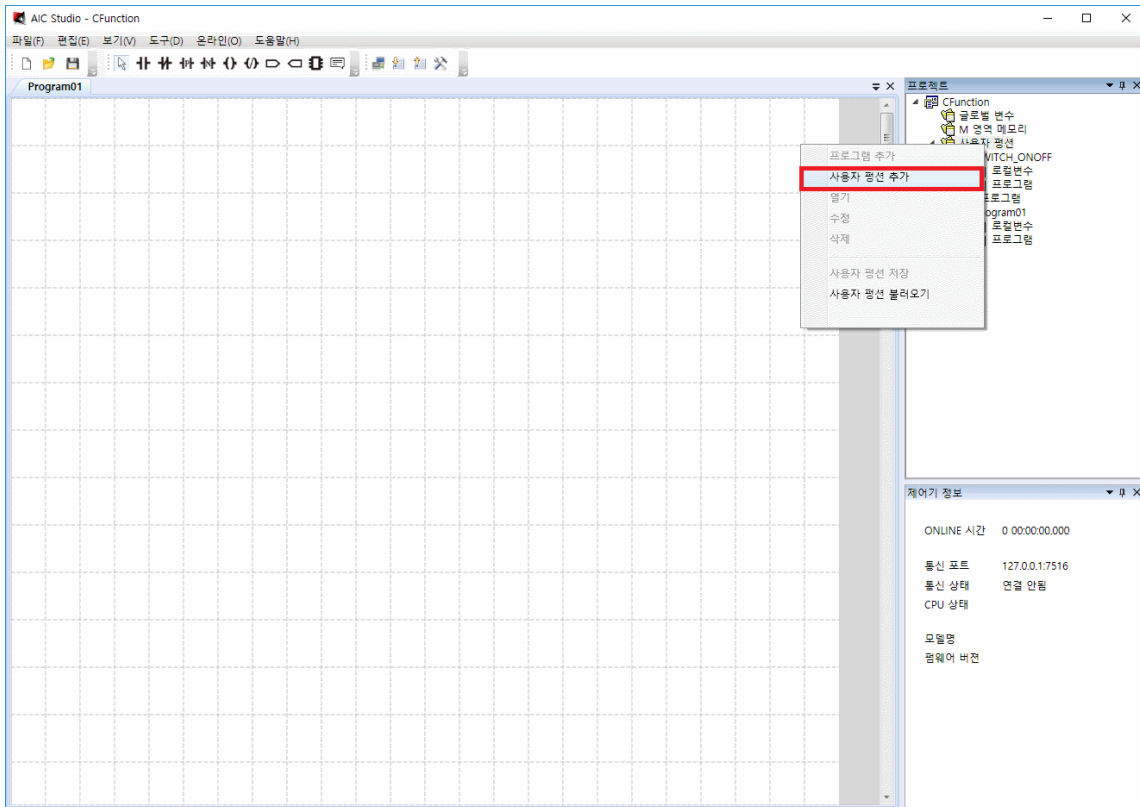
이전에 저장한 파일을 찾아 열기를 하면 사용이 가능합니다.

간헐적으로 사용하는 평선의 경우 이 방법을 사용하면 되나, 프로젝트 마다 사용할 수 있는 평선의 경우 프로젝트를 만들때마다 이 작업을 하기는 번거롭습니다.

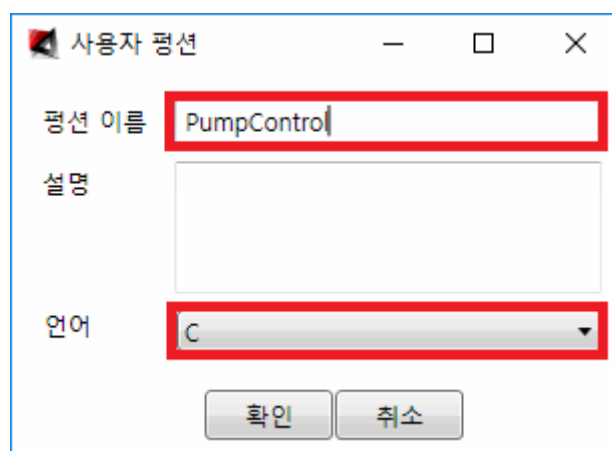
자주 사용하는 평선의 경우 저장한 Func 파일을 C:\WAIC_StudioWDefaultFunctions 에 복사를 해 두면 이후 새 프로젝트를 만들면 해당 평선은 자동으로 추가가 됩니다.

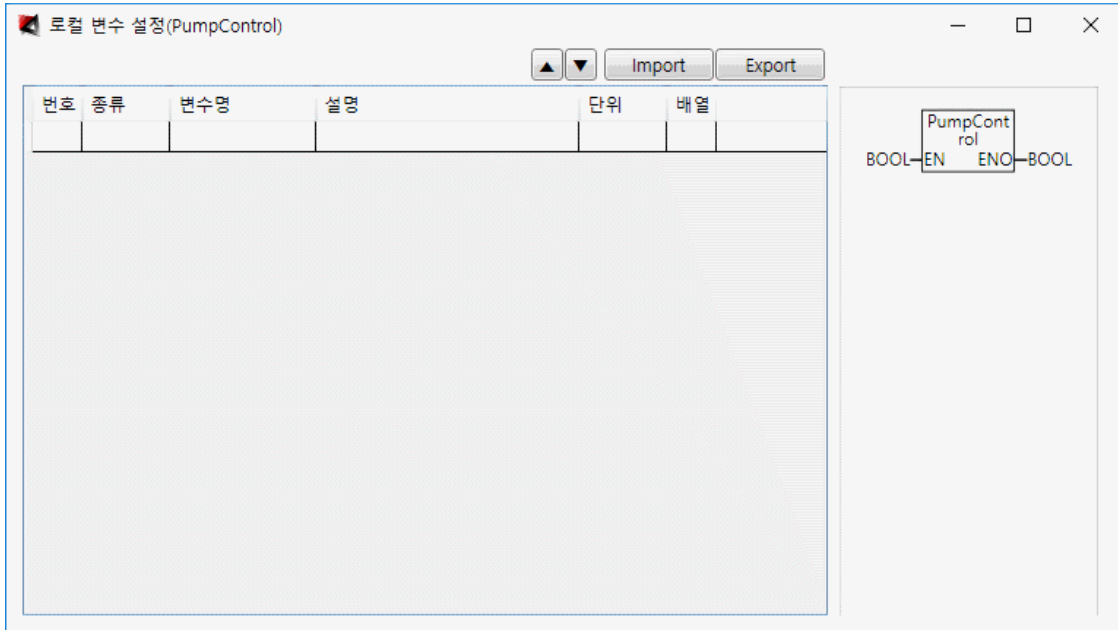
부록 5 : C 언어 사용법

여기서는 C 언어 사용에 대해 설명합니다.

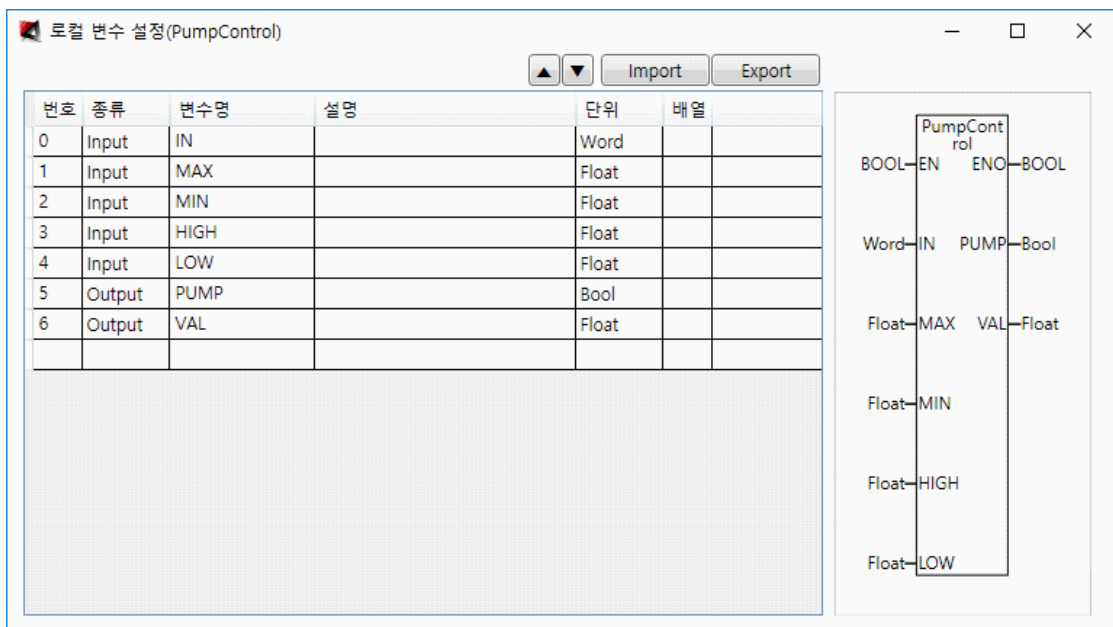


프로젝트의 사용자 평선에서 마우스 오른쪽 버튼을 클릭하여 생성된 메뉴에서 사용자 평선 추가를 선택하면 다음과 같은 대화 상자가 화면에 나타납니다.

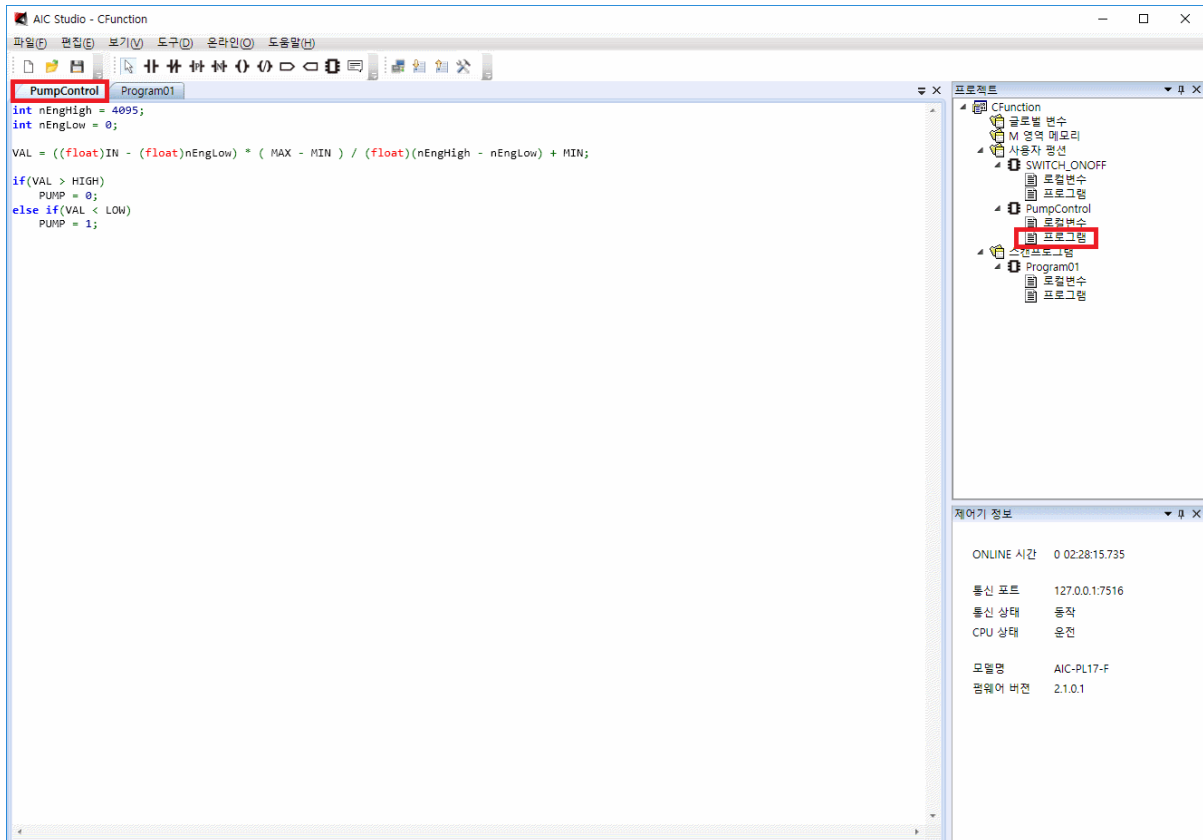




IN 을 MAX, MIN 을 이용하여 Range 변환한 후 HIGH 보다 크면은 PUMP 에 0 을 넣고(끄고), LOW 보다 작으면 PUMP 에 1 을 넣도록(켜도록) 구성할 예정이므로 아래와 같이 5 개의 입력 변수와 2 개(VAL 의 경우 디버깅 용으로 Scale 변환된 값이 어떻게 되는지 표시하기 위해 설정했습니다)의 출력 변수를 설정하였습니다.



C 언어 편집화면에서 다음과 같이 입력합니다.



```
int nEngHigh = 4095;
int nEngLow = 0;
```

nEngHigh 에 기계에서 들어오는 최대값(AIC-PL Series 의 경우 4095 로 고정)을 nEngLow 에 기계에서 들어오는 최소값(AIC-PL Series 의 경우 0 으로 고정)을 입력합니다.

$$VAL = ((float)IN - (float)nEngLow) * (MAX - MIN) / (float)(nEngHigh - nEngLow) + MIN;$$

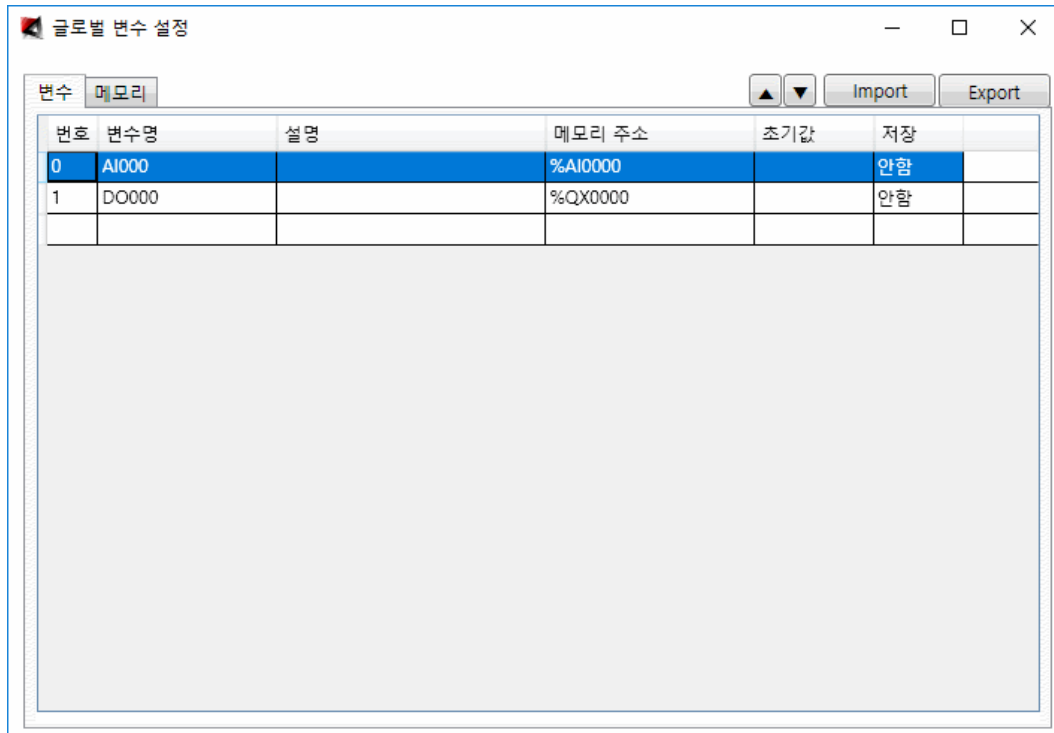
VAL 에 IN 의 Range 변환 값을 저장합니다.

```
if(VAL > HIGH)
    PUMP = 0;
else if(VAL < LOW)
    PUMP = 1;
```

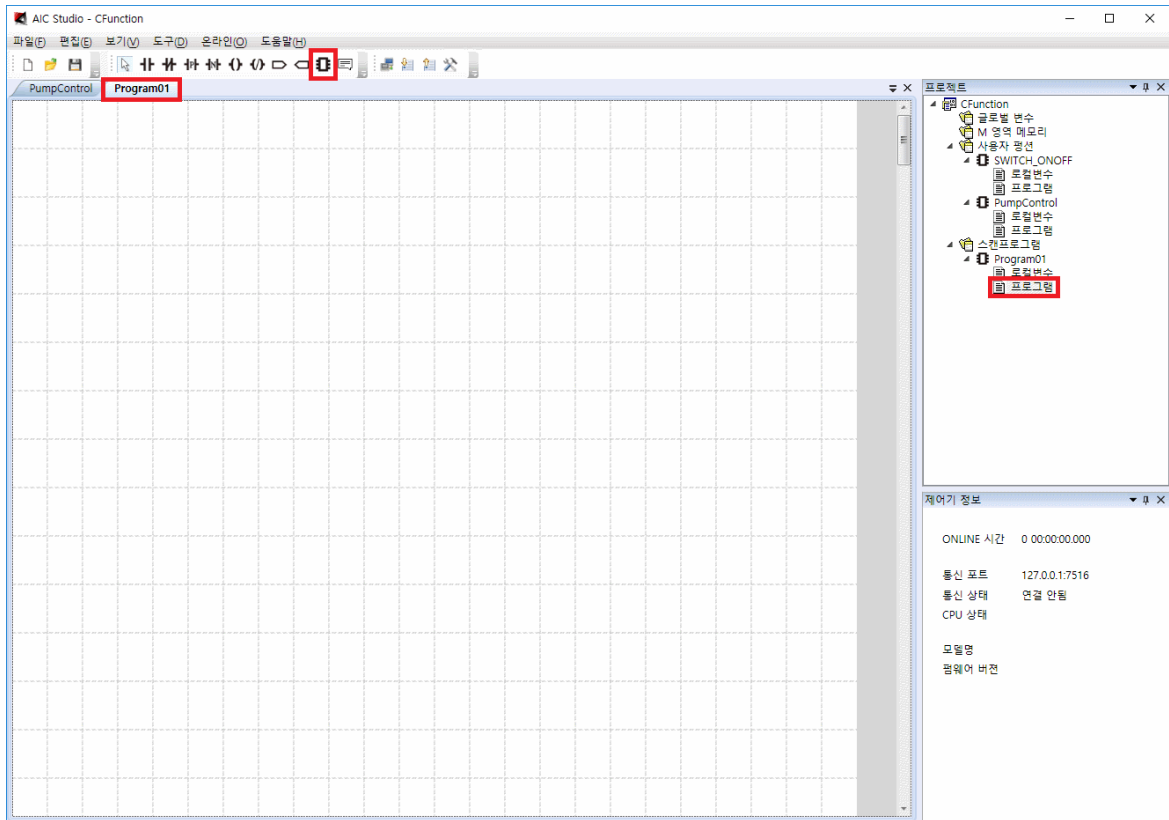
VAL 이 HIGH 보다 크면 PUMP 를 끄고 LOW 보다 작으면 PUMP 를 켜도록 합니다.


이제 작성한 C 평선 블록을 사용해 보도록 하겠습니다.

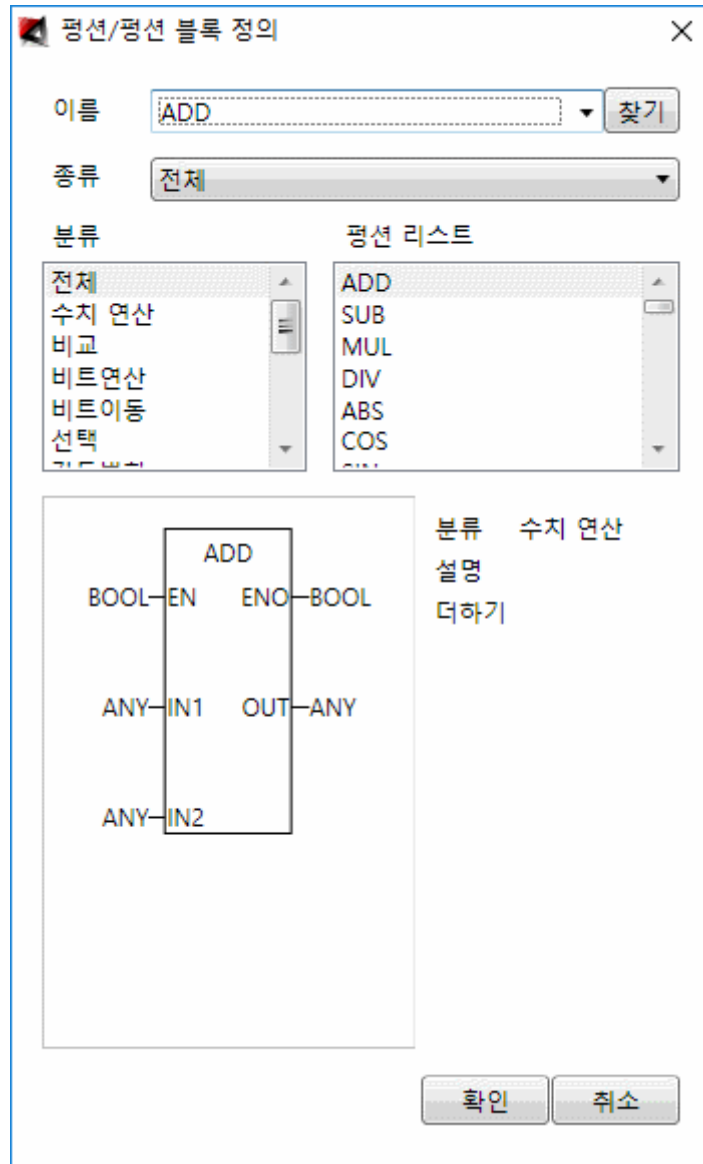
편집 메뉴의 부메뉴 글로벌 변수 설정을 선택하여 아날로그 입력 포인트 AI000 과 디지털 출력
 포인트 DO000 을 설정하면 다음과 같이 화면에 나타납니다.



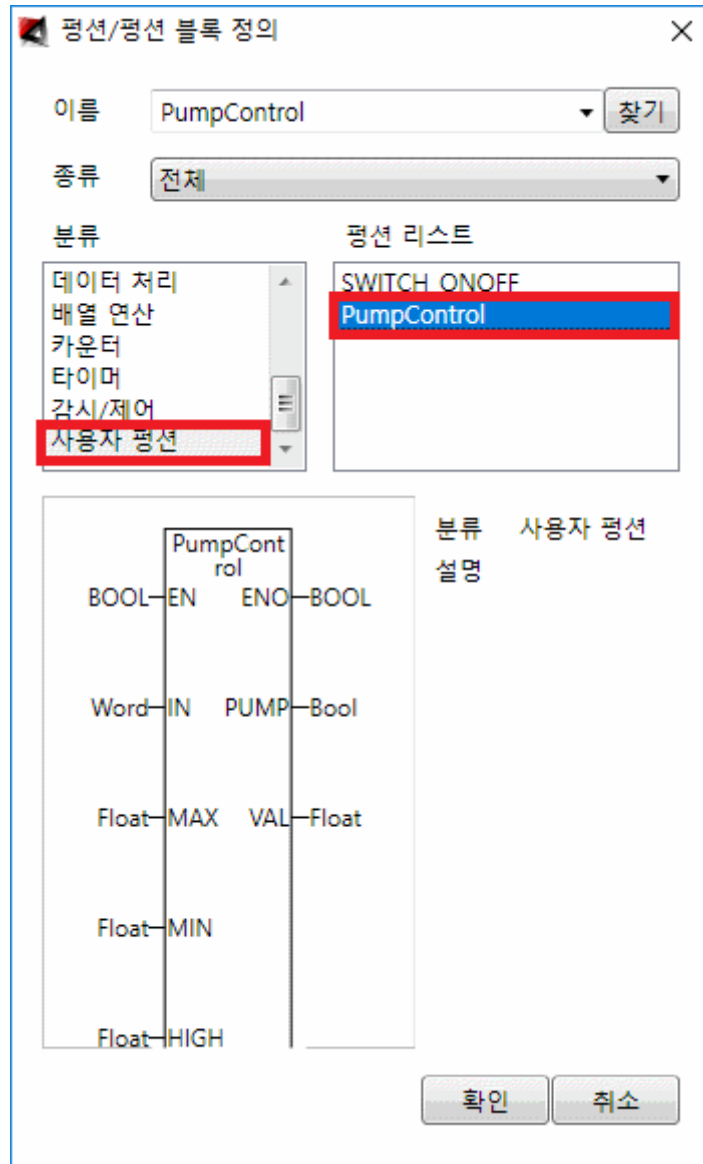
스캔프로그램의 Program01 을 선택한다음 편집메뉴의 부메뉴 로컬 변수 설정을 선택하여
 로컬변수 Value 를 float 로 설정하면 다음과 같이 화면에 나타납니다.



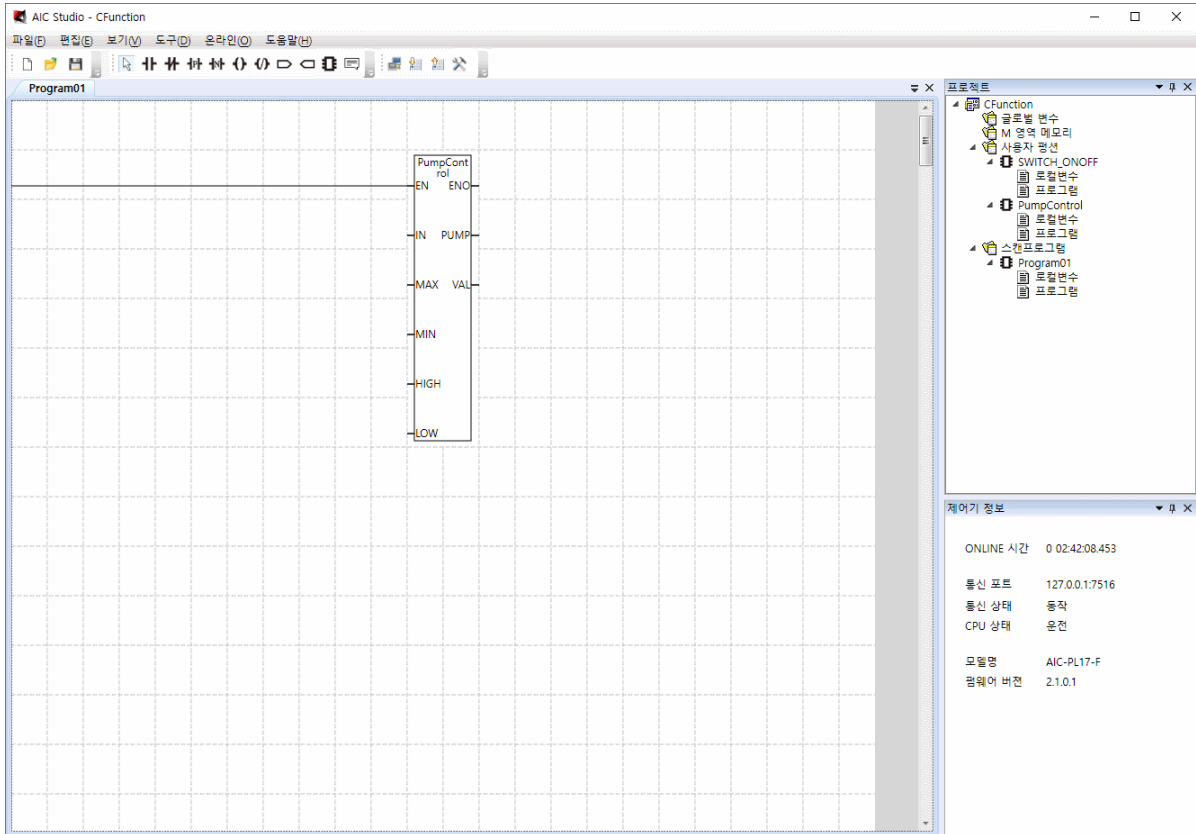
툴바 메뉴의  를 선택하고 삽입하고자 하는 위치에 마우스로 클릭하면 다음과 같이 화면에 나타납니다.



분류를 사용자 펄스로 선택하고 펄스 리스트를 PumpControl 로 선택합니다.



확인 버튼을 선택하면 다음과 같이 PumpControl 이 삽입됩니다.



다음 화면은 AI000 을 0~100 으로 Range 변환하여 80 보다 크면 DO000(펌프)을 끄고 20 보다 작으면 DO000(펌프)를 켜도록 설정한 모양입니다.

AIC Studio - CFunction

파일(F) 편집(E) 보기(V) 도구(D) 온라인(O) 도움말(H)

Program01

The diagram shows a function block named "PumpControl" with the following connections:

- Input: A1000 connected to EN
- Input: 100 connected to IN
- Input: 0 connected to MAX
- Input: 80 connected to MIN
- Input: 20 connected to HIGH
- Input: LOW connected to LOW
- Output: DC000 connected to ENO
- Output: Value connected to VAL
- Internal label: PUMP

프로그램 목록

- CFunction
 - 프로그램 변수
 - M 영역 메모리
 - 사용자 옵션
 - SWITCH_ONOFF
 - 로컬 변수 프로그램
 - PumpControl
 - 로컬 변수 프로그램
 - 스택 프로그램
 - Program01
 - 로컬 변수 프로그램

제어기 정보

ONLINE 시간 0 03:15:35.563

통신 포트 127.0.0.1:7516

통신 상태 동작

CPU 상태 운전

모델명 AIC-PL17-F

펌웨어 버전 2.1.0.1

C 언어는 C 기본 문법에 아래 함수들을 추가로 지원하고 있습니다.

이름	설명
Time	int Time(); 시간값(1970년 1월 1일 자정부터 1초마다 증가시킨 값)을 넘겨줍니다.
Min	float Min(float in1, float in2); in1, in2 중 작은값을 넘겨줍니다.
Max	float Max(float in1, float in2); in1, in2 중 큰값을 넘겨줍니다.
Abs	float Abs(float value); value의 절대값을 넘겨줍니다.
Cos	float Cos(float value); value의 Cosine 값을 넘겨줍니다.
Sin	float Sin(float value); value의 Sine 값을 넘겨줍니다.
Tan	float Tan(float value); value의 Tangent 값을 넘겨줍니다.
ACos	float ACos(float value); value의 Cosine 값을 넘겨줍니다.
ASin	float ASin(float value); value의 Sine 값을 넘겨줍니다.
ATan	float ATan(float value); value의 Tangent 값을 넘겨줍니다.
Exp	float Exp(float value); value의 Exp 값을 넘겨줍니다.
Expt	float Exp(float in1, float in2); in1에 in2를 지수승하여 넘겨줍니다.($e^{in1 \times in2}$).
LN	float LN(float value); value의 LN 값을 넘겨줍니다.
Log	float Log(float value); value의 Log 값을 넘겨줍니다.
PrintValue	void PrintValue(float value); value 값을 콘솔(온라인 메뉴의 부메뉴 콘솔)에 표시하도록 하는 함수로 디버깅을 하기 위해 사용합니다.