

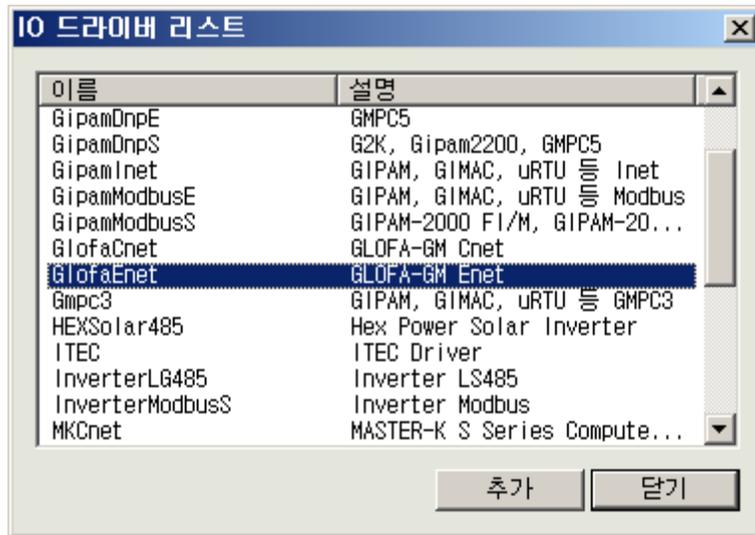
부록.1. I/O 드라이버 설정방법	1-3
1. GLOFAENET 드라이버	1-3
2. GLOFACNET 드라이버	1-7
3. XGKENET 드라이버	1-11
4. XGKCNET 드라이버	1-14
5. MASTER-K S SERIES COMPUTER LINK 드라이버	1-17
6. MASTER-K S SERIES LOADER PORT 드라이버	1-20
7. PMU 드라이버	1-23
8. INVERTERLG485 드라이버	1-26
9. INVERTERMODBUS 드라이버	1-33
10. OPC CLIENT 드라이버	1-40
11. MELSECQENET 드라이버	1-51
12. MELSECQSERIAL 드라이버	1-56
13. HEXSOLAR 드라이버	1-61
14. GLOFA LOADER 드라이버	1-69
15. MKENET 드라이버	1-72
16. P3000DCS 드라이버	1-75
17. GLOFAENET20 드라이버	1-81
18. GLOFACNET20 드라이버	1-89
19. XGIENET 드라이버	1-96
20. XGICNET 드라이버	1-102
21. XGKENET20 드라이버	1-109
22. XGKCNET20 드라이버	1-116
23. XGKLOADER 드라이버	1-123

24.	XGILOADER 드라이버.....	1-130
25.	MODBUSTCP 드라이버.....	1-137
26.	MODBUSRTU 드라이버.....	1-143
27.	OMRONE 드라이버.....	1-150
28.	OMRONS 드라이버.....	1-157
29.	NAISS 드라이버.....	1-164
30.	MELSECFXSERIAL 드라이버.....	1-171
31.	ABETHERNETIP 드라이버.....	1-178
32.	IDL1000S 드라이버.....	1-187

부록.1. I / O 드라이버 설정방법

1. GLOFAenet 드라이버

(1) 통신채널 추가



I/O 드라이버 리스트에서 “GLOFAenet” 을 선택한 후 “추가” 버튼을 누릅니다.

- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- (d) 서버 IP Address #2 : 라인 이중화를 사용할 경우 사용하게 될 두번째 IP Address를 입력합니다.
- (e) 최대 응답 시간 : 장비로부터 데이터를 한번 주고 받는데 걸리는 시간입니다.
- (f) 통신 Time Out : 장비에게 데이터를 요청한 후 타임 아웃(Time Out)으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (g) 통신 주기 : 데이터를 요청하는 주기입니다.
- (h) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.

(2) Station 추가

- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (d) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (e) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다.
- (f) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
- (g) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.
- (h) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
- (i) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.
- (j) 통신 방식 : TCP/UDP 중의 한가지를 선택합니다.
- (k) 포트 번호 : 통신 방식 선택에 따라 자동으로 입력됩니다.
- (l) 구조태그 타입 : 사용할 구조태그 타입을 선택합니다.
- (m) M영역 메모리 크기 : PLC의 M영역 메모리 크기를 선택합니다.
- (n) 블록 크기 : PLC 통신 블록의 크기를 입력합니다.

(3) 입출력 주소

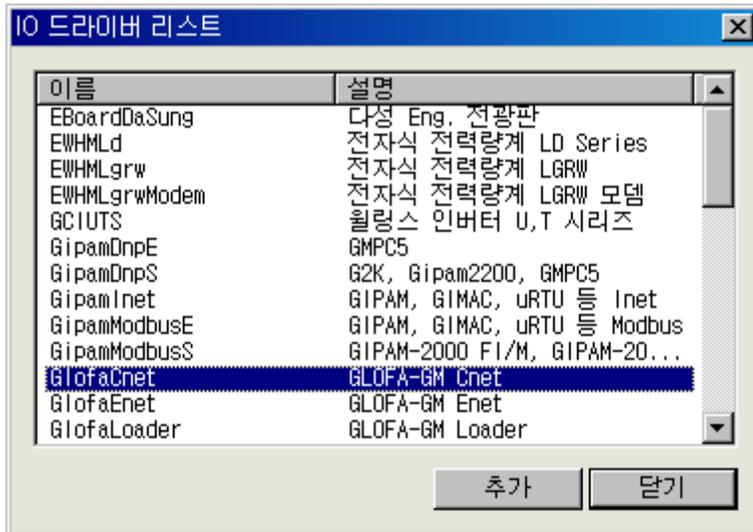
(a) I/O Address 형식

: %MXO, %MBO, %MWO, %MDO, %MLO

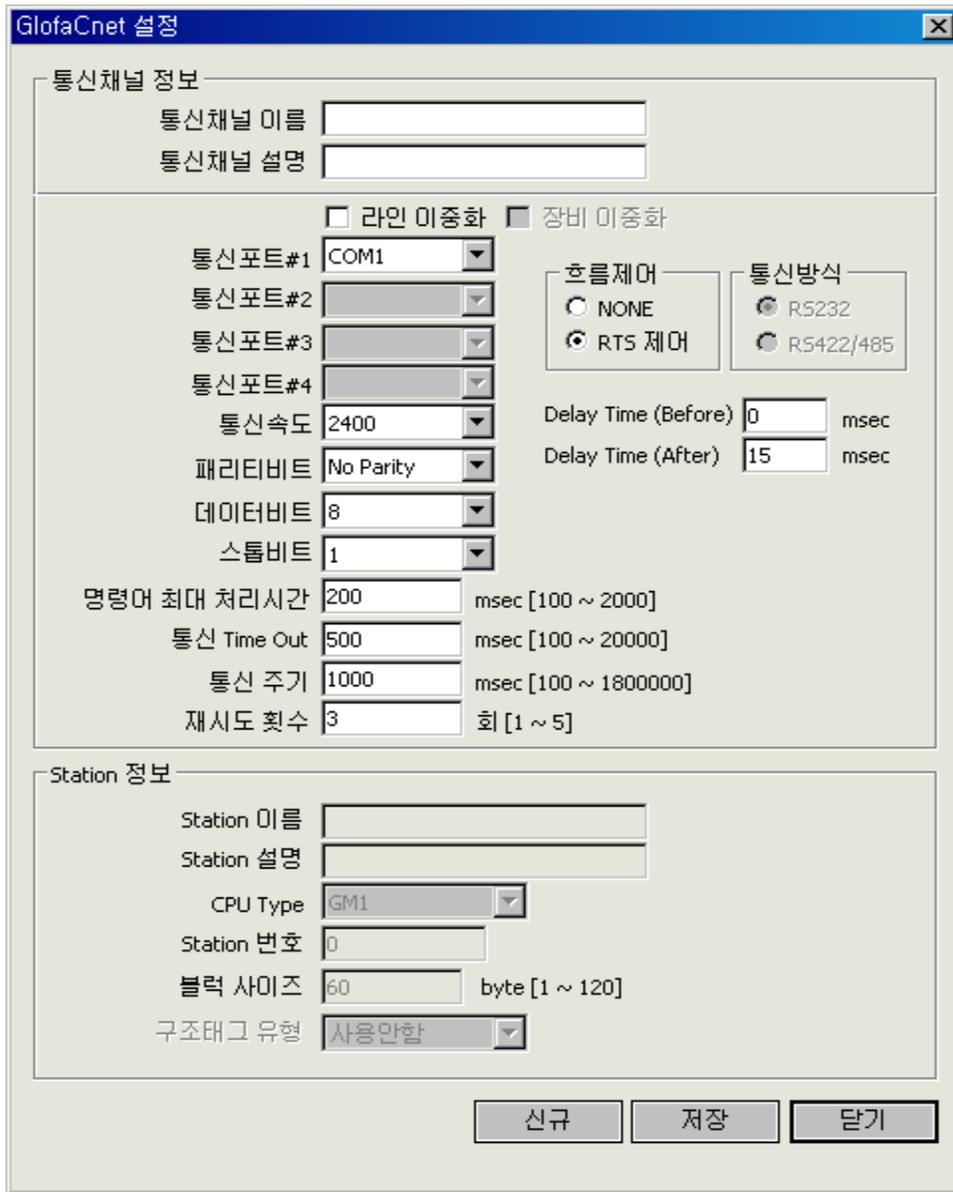
: %MBO.0, %MWO.0, %MDO.0, %MLO.0

2. GLOFACnet 드라이버

(1) 통신채널 추가



I/O 드라이버 리스트에서 “GLOFACnet” 을 선택한 후 “추가” 버튼을 누릅니다.



- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (d) 통신포트#1 : 통신포트를 선택합니다.
- (e) 통신포트#2 : 통신포트를 선택합니다.
- (f) 통신포트#3 : 통신포트를 선택합니다.
- (g) 통신포트#4 : 통신포트를 선택합니다.
- (h) 통신속도 : 통신 속도를 선택합니다.
- (i) 패리티비트 : 패리티비트를 선택합니다.
- (j) 데이터비트 : 데이터비트를 선택합니다.
- (k) 스톱비트 : 스톱비트를 선택합니다.
- (l) 명령어 최대 처리시간 : 한번 데이터를 요청한뒤 응답을 받을때까지 걸리는 시간을 말합니다.
- (m) 통신 Time Out : 장비에게 데이터를 요청한 후 타임 아웃으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (n) 통신 주기 : 데이터를 요청하는 주기입니다.

- (o) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (p) 흐름제어 : NONE 과 RTS제어 중 하나를 선택합니다.
- (q) Delay Time (Before) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.
- (r) Delay Time (After) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.

(2) Station 추가

- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) CPU Type : CPU 타입을 선택합니다.
- (d) Station 번호 : Station 번호를 입력합니다.
- (e) 블록 사이즈 : PLC 통신 블록의 크기를 입력합니다.

(3) 입출력 주소

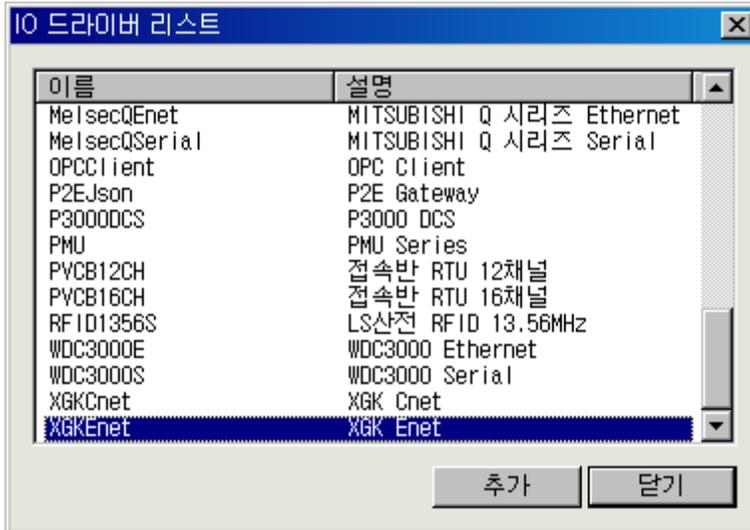
(a) I/O Address 형식

: %MXO, %MBO, %MMO, %MDO, %MLO

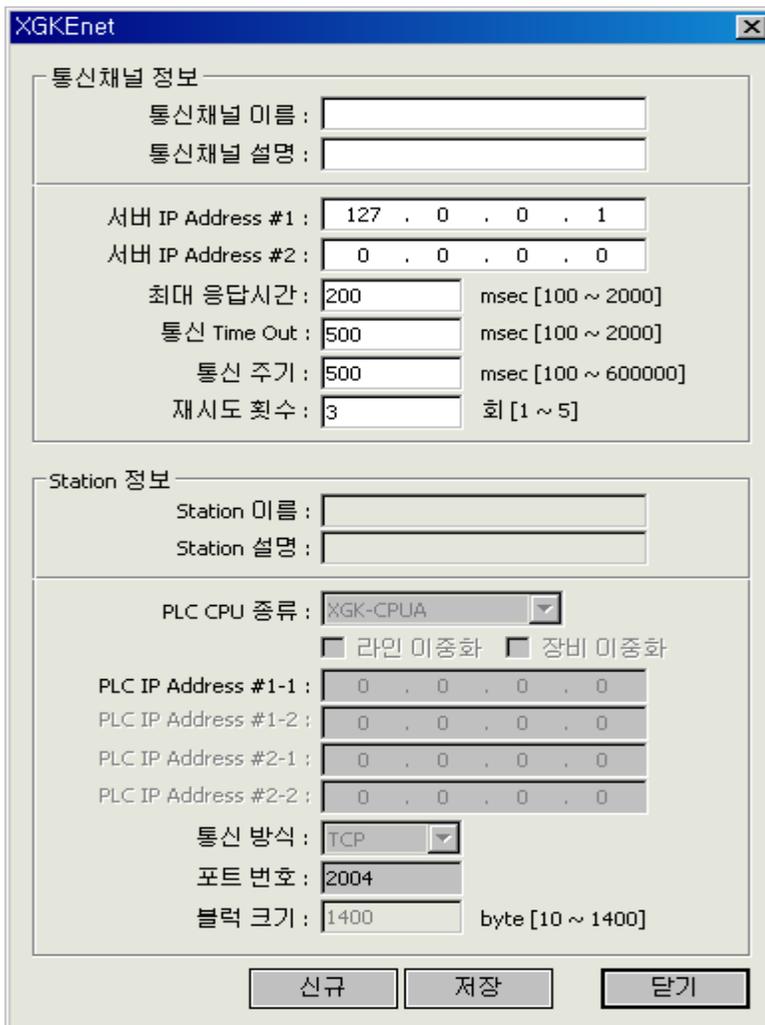
: %MBO.0, %MMO.0, %MDO.0, %MLO.0

3. XGEnet 드라이버

(1) 통신채널 추가



I/O 드라이버 리스트에서 “XGEnet” 을 선택한 후 “추가” 버튼을 누릅니다.



부록1 I/O 드라이버 설정방법

- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- (d) 서버 IP Address #2 : 라인 이중화를 사용할 경우 사용하게 될 두번째 IP Address를 입력합니다.
- (f) 최대 응답 시간 : 장비로부터 데이터를 한번 주고 받는데 걸리는 시간입니다.
- (g) 통신 Time Out : 장비에게 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (h) 통신 주기 : 데이터를 요청하는 주기입니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.

(2) Station 추가

The screenshot shows the XGKEnet configuration window with the following fields and values:

- 통신채널 정보**
 - 통신채널 이름 : XGKEnet
 - 통신채널 설명 : (empty)
 - 서버 IP Address #1 : 127 . 0 . 0 . 1
 - 서버 IP Address #2 : 0 . 0 . 0 . 0
 - 최대 응답시간 : 200 msec [100 ~ 2000]
 - 통신 Time Out : 500 msec [100 ~ 2000]
 - 통신 주기 : 500 msec [100 ~ 600000]
 - 재시도 횟수 : 3 회 [1 ~ 5]
- Station 정보**
 - Station 이름 : (empty)
 - Station 설명 : (empty)
 - PLC CPU 종류 : XGK-CPUA (dropdown)
 - 라인 이중화 장비 이중화
 - PLC IP Address #1-1 : 0 . 0 . 0 . 0
 - PLC IP Address #1-2 : 0 . 0 . 0 . 0
 - PLC IP Address #2-1 : 0 . 0 . 0 . 0
 - PLC IP Address #2-2 : 0 . 0 . 0 . 0
 - 통신 방식 : TCP (dropdown)
 - 포트 번호 : 2004
 - 블럭 크기 : 1400 byte [10 ~ 1400]

Buttons at the bottom: 신규, 저장, 닫기

- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (d) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (e) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다.
- (f) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
- (g) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.

- (h) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
- (i) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.
- (j) 통신 방식 : TCP/UDP 중의 한가지를 선택합니다.
- (k) 포트 번호 : 통신 방식 선택에 따라 자동으로 입력됩니다.
- (l) 블록 크기 : PLC 통신 블록의 크기를 입력합니다.

(3) 입출력 주소

- (a) 지원하는 디바이스
: P, M, K, F, L, N, D, R (T, C, Z, S, ZR 디바이스는 지원하지 않습니다.)
- (b) I/O 어드레스 형식
: P000(아날로그 태그: 디바이스 뒤의 숫자는 워드영역을 표시합니다.), P00F(디지털 태그: 맨뒤의 한자리가 비트 영역을 나타냅니다. 나머지 앞의 영역은 워드영역을 표시합니다.)

4. XGKnet 드라이버

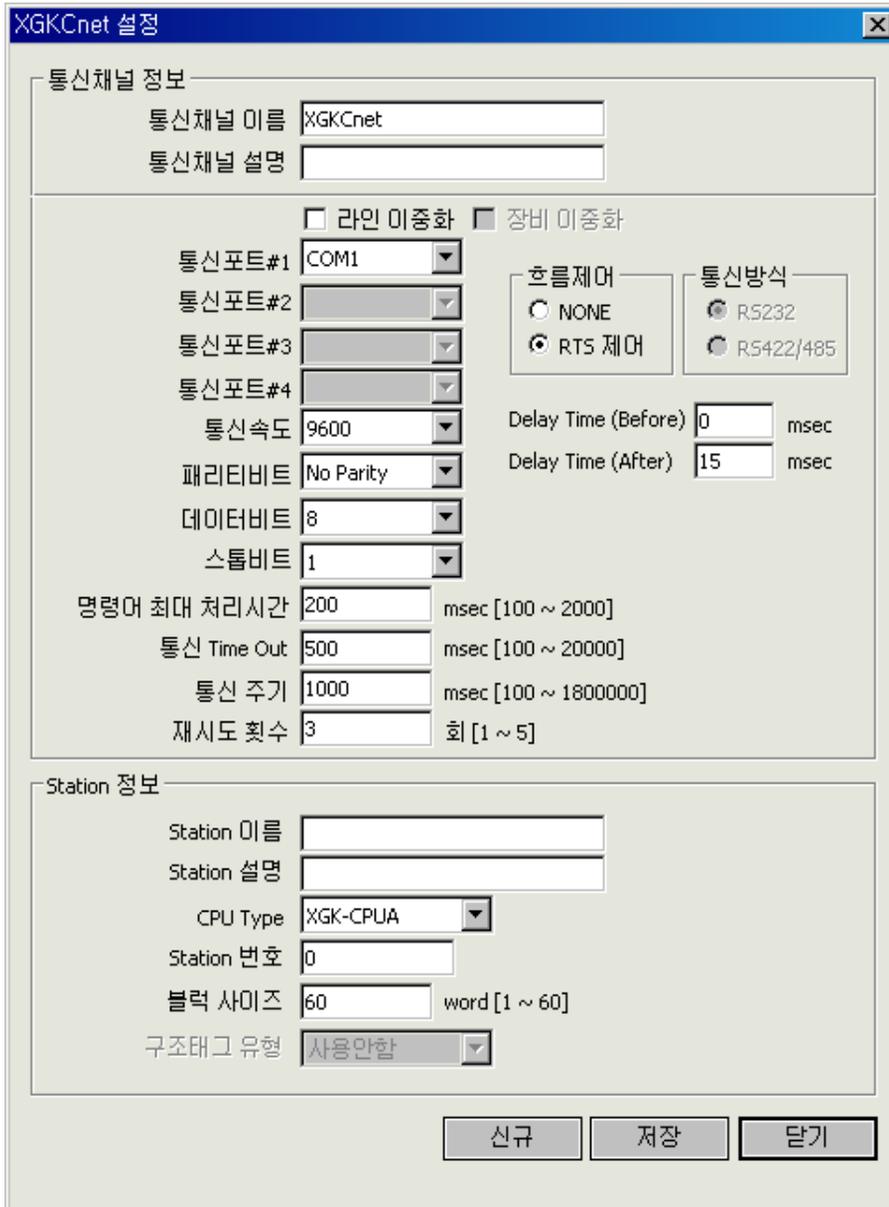
(1) 통신채널 추가



I/O 드라이버 리스트에서 “XGKnet” 을 선택한 후 “추가” 버튼을 누릅니다.

- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (d) 통신포트#1 : 통신포트를 선택합니다.
- (e) 통신포트#2 : 통신포트를 선택합니다.
- (f) 통신포트#3 : 통신포트를 선택합니다.
- (g) 통신포트#4 : 통신포트를 선택합니다.
- (h) 통신속도 : 통신 속도를 선택합니다.
- (i) 패리티비트 : 패리티비트를 선택합니다.
- (j) 데이터비트 : 데이터비트를 선택합니다.
- (k) 스톱비트 : 스톱비트를 선택합니다.
- (l) 명령어 최대 처리시간 : 한번 데이터를 요청한뒤 응답을 받을때까지 걸리는 시간을 말합니다.
- (m) 통신 Time Out : 장비에게 데이터를 요청한 후 타임아웃으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (n) 통신 주기 : 데이터를 요청하는 주기입니다.
- (o) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (p) 흐름제어 : NONE 과 RTS제어 중 하나를 선택합니다.
- (q) Delay Time (Before) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.
- (r) Delay Time (After) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.

(2) Station 추가



- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) CPU Type : CPU 타입을 선택합니다.
- (d) Station 번호 : Station 번호를 입력합니다.
- (e) 블록 사이즈 : PLC 통신 블록의 크기를 입력합니다.

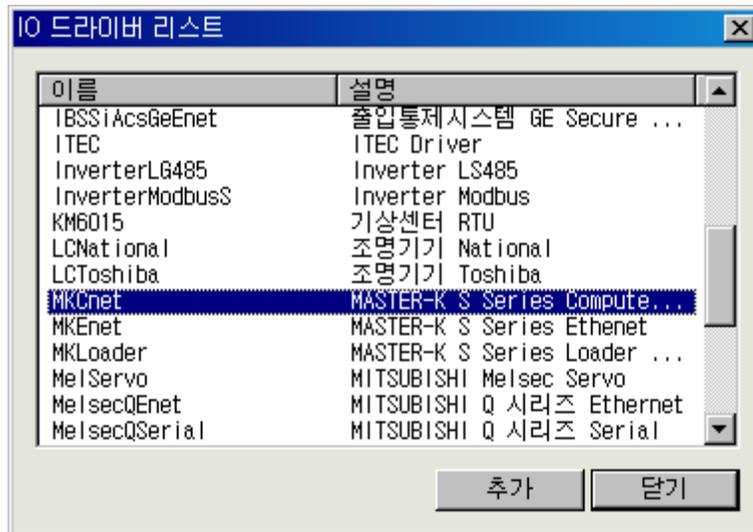
(3) 입출력 주소

- (a) 지원하는 디바이스
: P, M, K, F, L, N, D, R (T, C, Z, S, ZR 디바이스는 지원하지 않습니다.)
- (b) I/O 어드레스 형식
: P000 (아날로그 태그: 디바이스 뒤의 숫자는 워드영역을 표시합니다.), P00F (디지털태그: 맨뒤의 한자리가

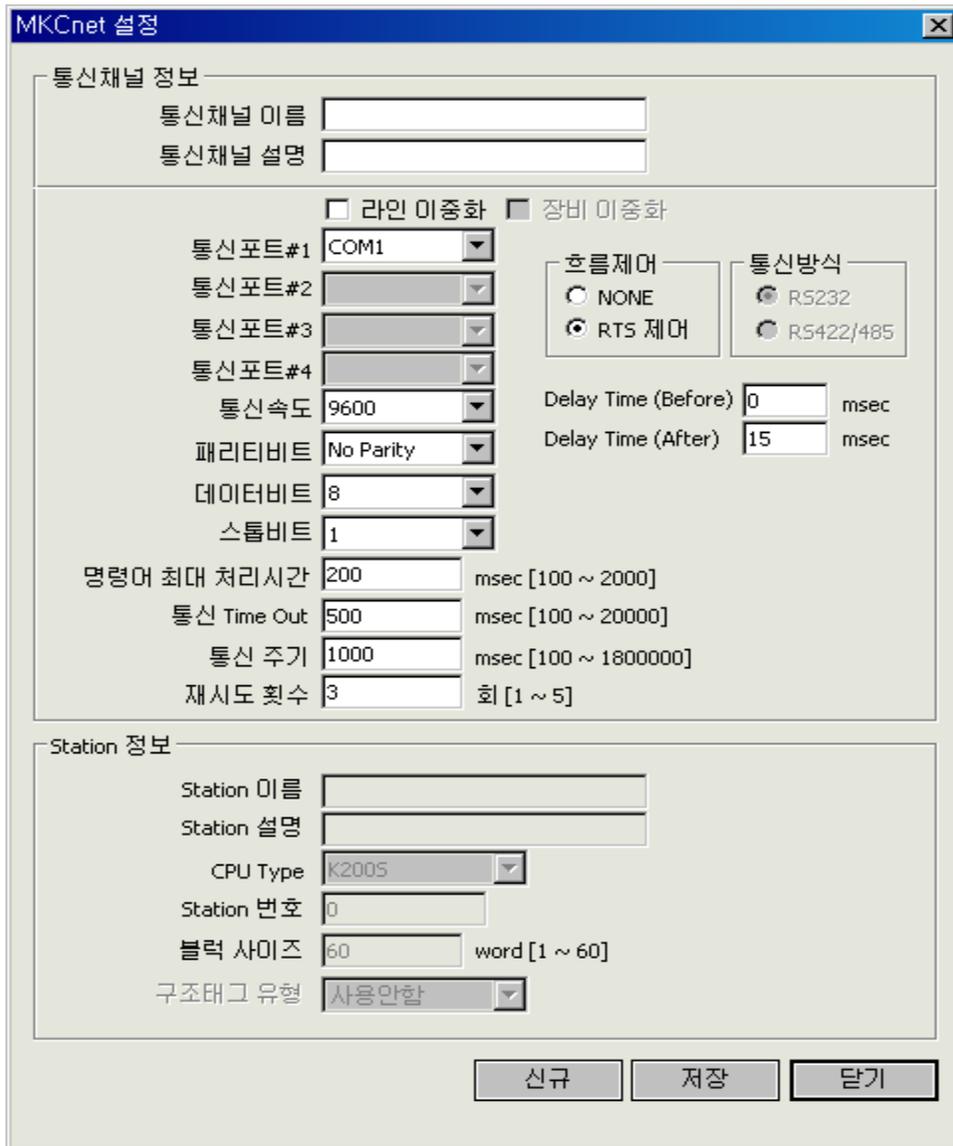
비트영역을 나타냅니다.) 나머지 앞의 영역은 워드영역을 표시합니다.

5. MASTER-K S Series Computer Link 드라이버

(1) 통신채널 추가



I/O 드라이버 리스트에서 “MKCnet” 을 선택한 후 “추가” 버튼을 누릅니다.



- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (d) 통신포트#1 : 통신포트를 선택합니다.
- (e) 통신포트#2 : 통신포트를 선택합니다.
- (f) 통신포트#3 : 통신포트를 선택합니다.
- (g) 통신포트#4 : 통신포트를 선택합니다.
- (h) 통신속도 : 통신 속도를 선택합니다.
- (i) 패리티비트 : 패리티비트를 선택합니다.
- (j) 데이터비트 : 데이터비트를 선택합니다.
- (k) 스톱비트 : 스톱비트를 선택합니다.
- (l) 명령어 최대 처리시간 : 한번 데이터를 요청한뒤 응답을 받을때까지 걸리는 시간을 말합니다.
- (m) 통신 Time Out : 장비에게 데이터를 요청한 후 타임아웃으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (n) 통신 주기 : 데이터를 요청하는 주기입니다.
- (o) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (p) 흐름제어 : NONE 과 RTS제어 중 하나를 선택합니다.

- (q) Delay Time (Before) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.
- (r) Delay Time (After) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.

(2) Station 추가

The image shows the 'MKCnet 설정' (MKCnet Settings) dialog box. It is divided into two main sections: '통신채널 정보' (Communication Channel Information) and 'Station 정보' (Station Information).

통신채널 정보 (Communication Channel Information):

- 통신채널 이름: MKCnet
- 통신채널 설명: (empty)
- 라인 이중화: 라인 이중화, 장비 이중화
- 통신포트#1: COM1
- 통신포트#2: (empty)
- 통신포트#3: (empty)
- 통신포트#4: (empty)
- 통신속도: 9600
- 패리티비트: No Parity
- 데이터비트: 8
- 스톱비트: 1
- 흐름제어: NONE, RTS 제어
- 통신방식: R5232, R5422/485
- Delay Time (Before): 0 msec
- Delay Time (After): 15 msec
- 명령어 최대 처리시간: 200 msec [100 ~ 2000]
- 통신 Time Out: 500 msec [100 ~ 20000]
- 통신 주기: 1000 msec [100 ~ 1800000]
- 재시도 횟수: 3 회 [1 ~ 5]

Station 정보 (Station Information):

- Station 이름: (empty)
- Station 설명: (empty)
- CPU Type: K2005
- Station 번호: 0
- 블록 사이즈: 60 word [1 ~ 60]
- 구조태그 유형: 사용안함

Buttons at the bottom: 신규 (New), 저장 (Save), 닫기 (Close).

- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) CPU Type : CPU 타입을 선택합니다.
- (d) Station 번호 : Station 번호를 입력합니다.
- (e) 블록 사이즈 : PLC 통신 블록의 크기를 입력합니다.

(3) 입출력 주소

- (a) 지원하는 디바이스 : P, M, K, F, L, D (T, C 디바이스는 지원하지 않습니다.)
- (b) I/O 어드레스 형식 : P000(아날로그 태그: 디바이스 뒤의 숫자는 워드영역을 표시합니다.), P00F(디지털 태그 : 맨뒤의 한자리가 비트 영역을 나타냅니다.) 나머지 앞의 영역은 워드영역을 표시합니다.

6. MASTER-K S Series Loader Port 드라이버

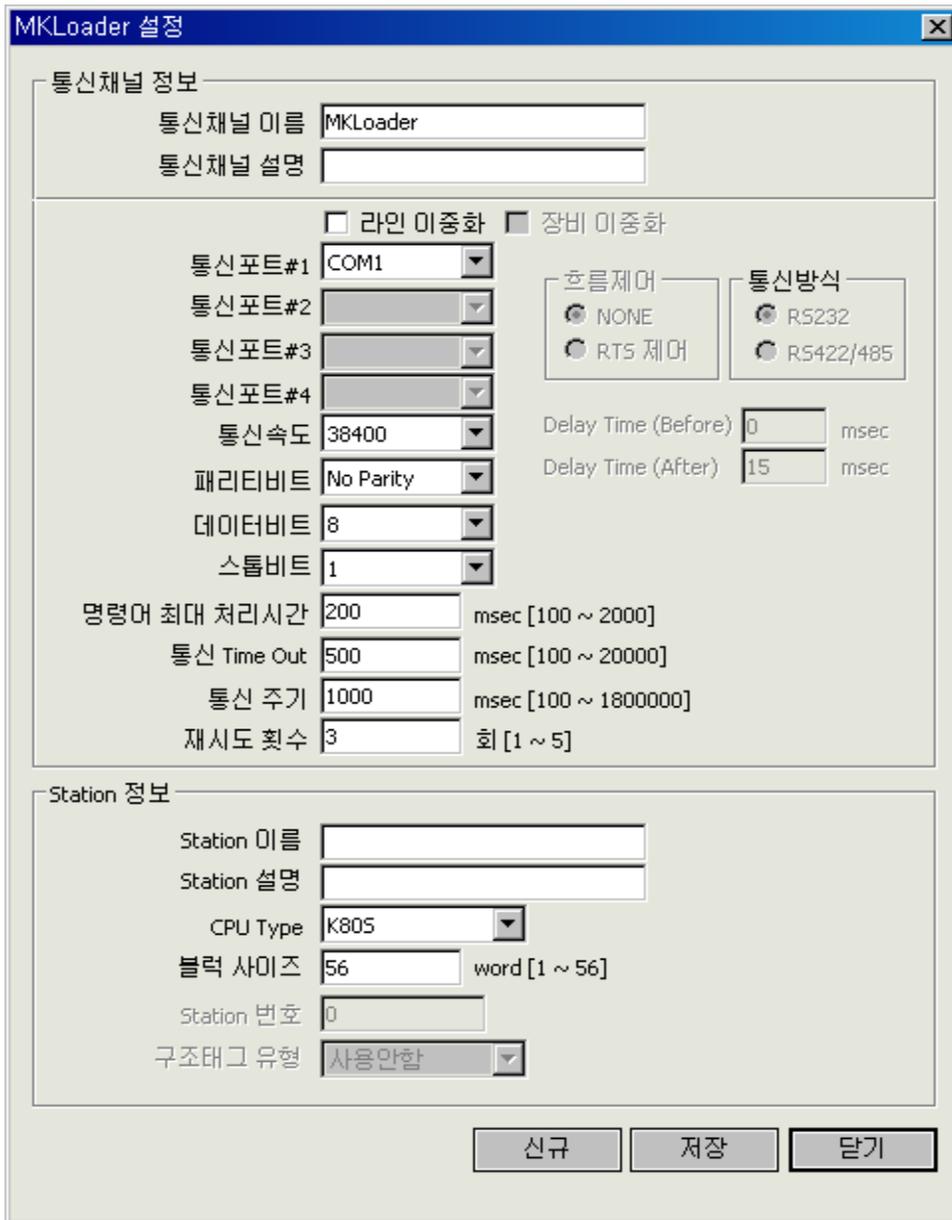
(1) 통신채널 추가



I/O 드라이버 리스트에서 “MKLoader” 을 선택한 후 “추가” 버튼을 누릅니다.

- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (d) 통신포트#1 : 통신포트를 선택합니다.
- (e) 통신포트#2 : 통신포트를 선택합니다.
- (f) 통신포트#3 : 통신포트를 선택합니다.
- (g) 통신포트#4 : 통신포트를 선택합니다.
- (h) 통신속도 : 통신 속도를 선택합니다.
- (i) 패리티비트 : 패리티비트를 선택합니다.
- (j) 데이터비트 : 데이터비트를 선택합니다.
- (k) 스톱비트 : 스톱비트를 선택합니다.
- (l) 명령어 최대 처리시간 : 한번 데이터를 요청한뒤 응답을 받을때까지 걸리는 시간을 말합니다.
- (m) 통신 Time Out : 장비에게 데이터를 요청한 후 타임아웃으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (n) 통신 주기 : 데이터를 요청하는 주기입니다.
- (o) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.

(2) Station 추가



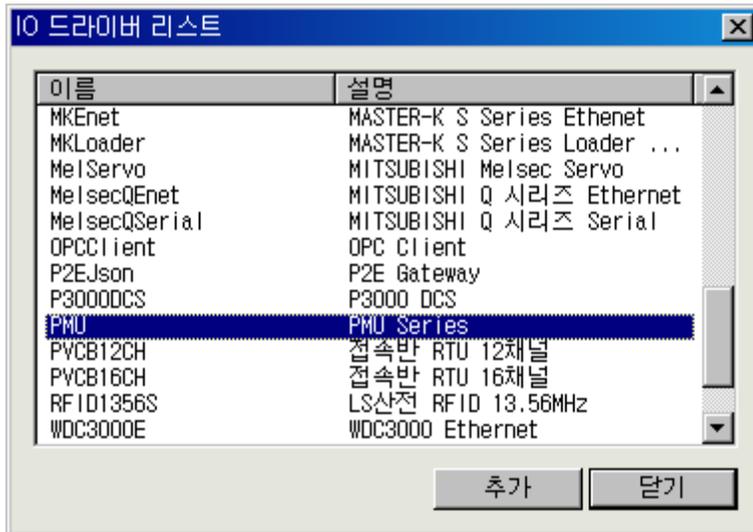
- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) CPU Type : CPU 타입을 선택합니다.
- (d) 블록 사이즈 : PLC 통신 블록의 크기를 입력합니다.

(3) 입출력 주소

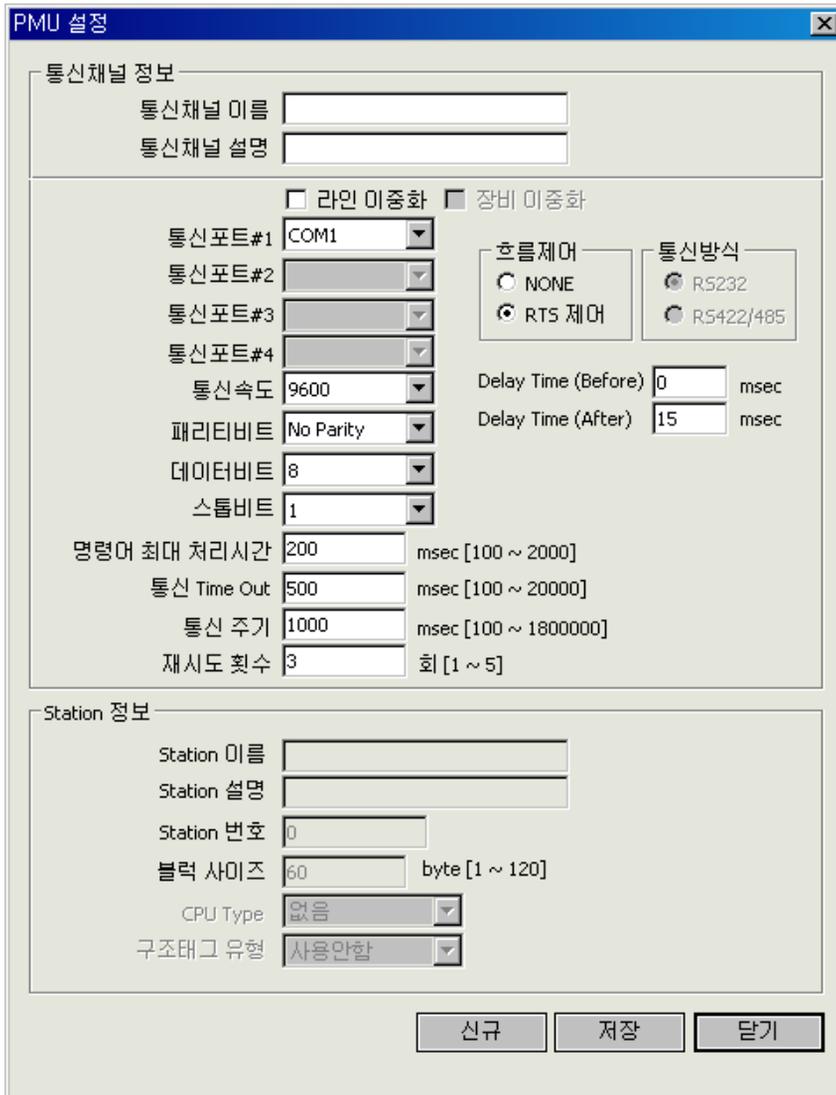
- (a) 지원하는 디바이스 : P, M, K, F, L, D (T, C 디바이스는 지원하지 않습니다.)
- (b) I/O 어드레스 형식 : P000(아날로그 태그: 디바이스 뒤의 숫자는 워드영역을 표시합니다.), P00F(디지털 태그 : 맨뒤의 한자리가 비트영역을 나타냅니다.) 나머지 앞의 영역은 워드영역을 표시합니다.

7. PMU 드라이버

(1) 통신채널 추가



I/O 드라이버 리스트에서 “PMU” 를 선택한 후 “추가” 버튼을 누릅니다.



- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (d) 통신포트#1 : 통신포트를 선택합니다.
- (e) 통신포트#2 : 통신포트를 선택합니다.
- (f) 통신포트#3 : 통신포트를 선택합니다.
- (g) 통신포트#4 : 통신포트를 선택합니다.
- (h) 통신속도 : 통신 속도를 선택합니다.
- (i) 패리티비트 : 패리티비트를 선택합니다.
- (j) 데이터비트 : 데이터비트를 선택합니다.
- (k) 스톱비트 : 스톱비트를 선택합니다.
- (l) 명령어 최대 처리시간 : 한번 데이터를 요청한뒤 응답을 받을때까지 걸리는 시간을 말합니다.
- (m) 통신 Time Out : 장비에게 데이터를 요청한 후 타임아웃으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (n) 통신 주기 : 데이터를 요청하는 주기입니다.
- (o) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (p) 흐름제어 : NONE 과 RTS제어 중 하나를 선택합니다.
- (q) Delay Time (Before) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.

(r) Delay Time (After) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.

(2) Station 추가

- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) Station 번호 : Station 번호를 입력합니다.
- (d) 블록 사이즈 : 통신 블록의 크기를 입력합니다.

(3) 입출력 주소

- (a) I/O 어드레스 형식
 - : PMU는 최대 워드 사이즈 0~4095워드까지 사용할 수 있습니다.
 - : 아날로그 어드레스 : 0~4095
 - : 디지털 어드레스 : (0.0~0.15) ~ (4095.0~4095.15)
 - : 사용가능한 디바이스 형식 : INT_16, UNSIGNED INT_16, INT_32, UNSIGNED INT_32

8. InverterLG485 드라이버

(1) 통신채널 추가



I/O 드라이버 리스트에서 “InverterLG485” 을 선택한 후 “추가” 버튼을 누릅니다.

InverterLG485 Serial 설정

통신채널 정보

통신채널 이름

통신채널 설명

라인 이중화 장비 이중화

통신포트#1

통신포트#2

통신포트#3

통신포트#4

통신속도

패리티비트

데이터비트

스톱비트

명령어 최대 처리시간 msec [100 ~ 2000]

통신 Time Out msec [100 ~ 20000]

통신 주기 msec [100 ~ 1800000]

재시도 횟수 회 [1 ~ 5]

흐름제어 NONE RTS 제어

통신방식 RS232 RS422/485

Delay Time (Before) msec

Delay Time (After) msec

Station 정보

Station 이름

Station 설명

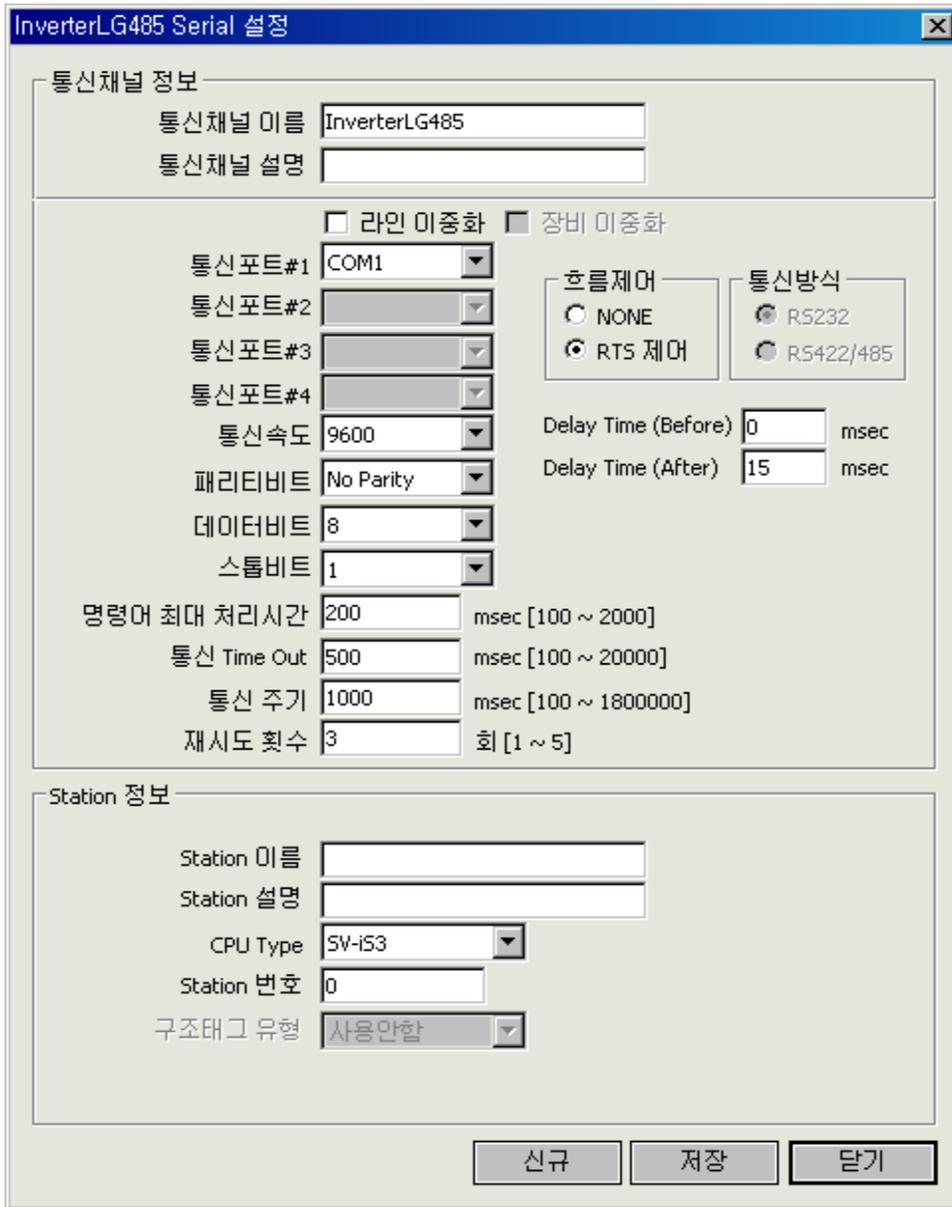
CPU Type

Station 번호

구조태그 유형

- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (d) 통신포트#1 : 통신포트를 선택합니다.
- (e) 통신포트#2 : 통신포트를 선택합니다.
- (f) 통신포트#3 : 통신포트를 선택합니다.
- (g) 통신포트#4 : 통신포트를 선택합니다.
- (h) 통신속도 : 통신 속도를 선택합니다.
- (i) 패리티비트 : 패리티비트를 선택합니다.
- (j) 데이터비트 : 데이터비트를 선택합니다.
- (k) 스톱비트 : 스톱비트를 선택합니다.
- (l) 명령어 최대 처리시간 : 한번 데이터를 요청한뒤 응답을 받을때까지 걸리는 시간을 말합니다.
- (m) 통신 Time Out : 장비에게 데이터를 요청한 후 타임아웃으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (n) 통신 주기 : 데이터를 요청하는 주기입니다.
- (o) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (p) 흐름제어 : NONE 과 RTS제어 중 하나를 선택합니다.
- (q) Delay Time (Before) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.
- (r) Delay Time (After) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.

(2) Station 추가



- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) CPU Type : CPU 타입을 선택합니다.
- (d) Station 번호 : Station 번호를 입력합니다.

(3) 입출력 주소

(a) I/O 어드레스 맵

1) 아날로그 입력

어드레스	레지스터 명	범위	단위
AI000	파라미터 설정허용 [0:파라미터 설정금지, 1:파라미터 설정허용]	-	-
AI001	지령주파수	0.01	Hz
AI002	운전지령정보 [0:단자대, 1:로더, 2: Reserved, 3: 통신운전]	-	-
AI003	주파수지령정보 [0:DRV-00, 1:사용안함, 2:다단속 1,3 : 다단속 2, 4:다단속 3, 5:다단속 4, 6:다단속 5, 7:다단속 6, 8:다단속 7, 9:업, 10:다운, 11:업다운제로, 12:V0, 13:V1, 14:I, 15: V0+I 16: V1+I, 17:JOG, 18:PID, 19:통신운전, 20~31:예약	-	-
AI004	가속 시간	0.1	sec
AI005	감속 시간	0.1	sec
AI006	출력전류	0.1	A
AI007	출력 주파수	0.01	Hz
AI008	출력 전압	0.1	V
AI009	DC Link 전압	0.1	V
AI010	출력 파워	0.1	kW
AI011	V1	0~3FF	-
AI012	V2	0~3FF	-
AI013	I	0~3FF	-
AI014	RPM	-	-

2) 디지털 입력

어드레스	레지스터 명	범위	단위
DI000	운전지령 정지	-	-
DI001	운전지령 정방향운전	-	-
DI002	운전지령 역방향운전	-	-
DI003	운전지령 폴트리셋	-	-
DI004	운전지령 비상정지	-	-
DI005	운전지령 Not Used	-	-
DI006	운전지령 Not Used	-	-
DI007	운전지령 Not Used	-	-
DI008	운전지령 Not Used	-	-
DI009	운전지령 Not Used	-	-
DI010	운전지령 Not Used	-	-
DI011	운전지령 Not Used	-	-
DI012	운전지령 Not Used	-	-
DI013	운전지령 Not Used	-	-
DI014	운전지령 Not Used	-	-

부록1 I/O 드라이버 설정방법

어드레스	레지스터 명	범위	단위
DI015	운전지령 Not Used	-	-
DI016	운전상태 정지	-	-
DI017	운전상태 정방향 운전중	-	-
DI018	운전상태 역방향 운전중	-	-
DI019	운전상태 Fault (Trip)	-	-
DI020	운전상태 가속중	-	-
DI021	운전상태 감속중	-	-
DI022	운전상태 속도도달	-	-
DI023	운전상태 DC Braking 중	-	-
DI024	운전상태 정지중	-	-
DI025	운전상태 Not Used	-	-
DI026	운전상태 Brake 개방신호	-	-
DI027	운전상태 정방향 지령(지령이면 1)	-	-
DI028	운전상태 역방향 지령(지령이면 1)	-	-
DI029	운전상태 REM. R/S	-	-
DI030	운전상태 REM. Freq.	-	-
DI031	운전상태 Not Used	-	-
DI032	트립정보 OCT	-	-
DI033	트립정보 OVT	-	-
DI034	트립정보 EXT-A	-	-
DI035	트립정보 EST(BX)	-	-
DI036	트립정보 COL	-	-
DI037	트립정보 GFT(Ground Fault)	-	-
DI038	트립정보 OHT(인버터 과열)	-	-
DI039	트립정보 ETH(전동기 과열)	-	-
DI040	트립정보 OLT(과부하)	-	-
DI041	트립정보 HW-Diag	-	-
DI042	트립정보 EXT-B	-	-
DI043	트립정보 EEP(Parameter Write Error)	-	-
DI044	트립정보 FAN(LOCK & Open Error)	-	-
DI045	트립정보 PO(Phase Open)	-	-
DI046	트립정보 IOLT	-	-
DI047	트립정보 LVT	-	-
DI048	입력단자정보 P1	-	-
DI049	입력단자정보 P2	-	-
DI050	입력단자정보 P3	-	-
DI051	입력단자정보 P4	-	-
DI052	입력단자정보 P5	-	-
DI053	입력단자정보 P6	-	-
DI054	입력단자정보 P7	-	-
DI055	입력단자정보 P8	-	-
DI056	입력단자정보 Not Used	-	-
DI057	입력단자정보 Not Used	-	-
DI058	입력단자정보 Not Used	-	-
DI059	입력단자정보 Not Used	-	-

어드레스	레지스터 명	범위	단위
DI060	입력단자정보 Not Used	-	-
DI061	입력단자정보 Not Used	-	-
DI062	입력단자정보 Not Used	-	-
DI063	입력단자정보 Not Used	-	-
DI064	출력단자정보 Not Used	-	-
DI065	출력단자정보 Not Used	-	-
DI066	출력단자정보 Not Used	-	-
DI067	출력단자정보 Not Used	-	-
DI068	출력단자정보 MO(Multi-Output With OC)	-	-
DI069	출력단자정보 Not Used	-	-
DI070	출력단자정보 Not Used	-	-
DI071	출력단자정보 3ABC	-	-
DI072	출력단자정보 Not Used	-	-
DI073	출력단자정보 Not Used	-	-
DI074	출력단자정보 Not Used	-	-
DI075	출력단자정보 Not Used	-	-
DI076	출력단자정보 Not Used	-	-
DI077	출력단자정보 Not Used	-	-
DI078	출력단자정보 Not Used	-	-
DI079	출력단자정보 Not Used	-	-
DI080	트립정보-B COM(I/O Board Reset)	-	-
DI081	트립정보-B FLTL	-	-
DI082	트립정보-B NTC	-	-
DI083	트립정보-B REEP	-	-
DI084	트립정보-B Not Used	-	-
DI085	트립정보-B Not Used	-	-
DI086	트립정보-B Not Used	-	-
DI087	트립정보-B Not Used	-	-
DI088	트립정보-B Not Used	-	-
DI089	트립정보-B Not Used	-	-
DI090	트립정보-B Not Used	-	-
DI091	트립정보-B Not Used	-	-
DI092	트립정보-B Not Used	-	-
DI093	트립정보-B Not Used	-	-
DI094	트립정보-B Not Used	-	-
DI095	트립정보-B Not Used	-	-

3) 문자열 입력

어드레스	레지스터 명	범위	단위
SI000	인버터 모델	-	-
SI001	인버터 용량	-	-
SI002	인버터 입력 전압	-	-
SI003	버전		

4) 아날로그 출력

어드레스	레지스터 명	범위	단위
A0000	파라 미터 설정허용 [0:파라 미터 설정금지, 1:파라 미터 설정 허용]	-	-
A0001	지령주파수	0.1	Hz
A0002	가속 시간	0.1	sec
A0003	감속 시간	0.1	sec

5) 디지털 출력

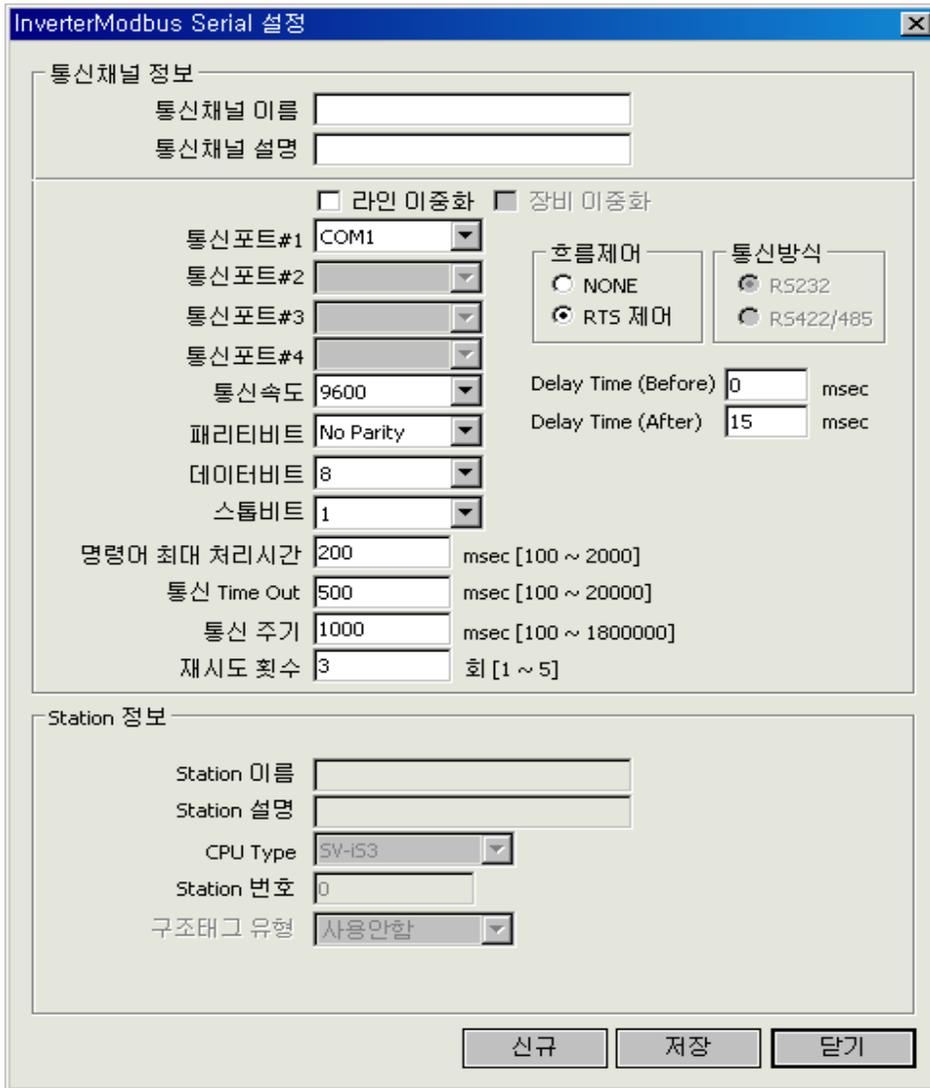
어드레스	레지스터 명	범위	단위
D0000	운전지령 정지	-	-
D0001	운전지령 정방향 운전	-	-
D0002	운전지령 역방향 운전	-	-
D0003	운전지령 폴트 Reset	-	-
D0004	운전지령 비상정지	-	-

9. InverterModbusS 드라이버

(1) 통신채널 추가



I/O 드라이버 리스트에서 “InverterModbusS” 을 선택한 후 “추가” 버튼을 누릅니다.



- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (d) 통신포트#1 : 통신포트를 선택합니다.
- (e) 통신포트#2 : 통신포트를 선택합니다.
- (f) 통신포트#3 : 통신포트를 선택합니다.
- (g) 통신포트#4 : 통신포트를 선택합니다.
- (h) 통신속도 : 통신 속도를 선택합니다.
- (i) 패리티비트 : 패리티비트를 선택합니다.
- (j) 데이터비트 : 데이터비트를 선택합니다.
- (k) 스톱비트 : 스톱비트를 선택합니다.
- (l) 명령어 최대 처리시간 : 한번 데이터를 요청한뒤 응답을 받을때까지 걸리는 시간을 말합니다.
- (m) 통신 Time Out : 장비에게 데이터를 요청한 후 타임아웃으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (n) 통신 주기 : 데이터를 요청하는 주기입니다.
- (o) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (p) 흐름제어 : NONE 과 RTS제어 중 하나를 선택합니다.
- (q) Delay Time (Before) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.
- (r) Delay Time (After) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.

(2) Station 추가

InverterModbus Serial 설정

통신채널 정보

통신채널 이름: InverterModbus5
통신채널 설명:

라인 이중화 장비 이중화

통신포트#1: COM1
통신포트#2:
통신포트#3:
통신포트#4:
통신속도: 9600
패리티비트: No Parity
데이터비트: 8
스톱비트: 1

흐름제어: NONE RTS 제어
통신방식: R5232 R5422/485

Delay Time (Before): 0 msec
Delay Time (After): 15 msec

명령어 최대 처리시간: 200 msec [100 ~ 2000]
통신 Time Out: 500 msec [100 ~ 20000]
통신 주기: 1000 msec [100 ~ 1800000]
재시도 횟수: 3 회 [1 ~ 5]

Station 정보

Station 이름:
Station 설명:
CPU Type: SV-IS3
Station 번호: 0
구조태그 유형: 사용안함

신규 저장 닫기

- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) CPU Type : CPU 타입을 선택합니다.
- (d) Station 번호 : Station 번호를 입력합니다.

부록1 I/O 드라이버 설정방법

(3) 입출력 주소

(a) I/O 어드레스 맵

1) 아날로그 입력

어드레스	레지스터 명	범위	단위
AI000	파라미터 설정허용 [0:파라미터 설정금지, 1:파라미터 설정허용]	-	-
AI001	지령주파수	0.01	Hz
AI002	운전지령정보 [0: 단자대, 1: 로더, 2:Reserved, 3: 통신운전]	-	-
AI003	주파수지령정보 [0:DRV-00, 1:사용안함, 2:다단속 1,3:다단속 2, 4:다단속 3, 5:다단속 4, 6:다단속 5, 7:다단속 6, 8:다단속 7, 9:업, 10:다운, 11:업다운제로, 12:V0, 13:V1, 14:I, 15: V0+I 16: V1+I, 17:JOG, 18:PID, 19:통신운전, 20~31:Reserved]	-	-
AI004	가속 시간	0.1	sec
AI005	감속 시간	0.1	sec
AI006	출력전류	0.1	A
AI007	출력 주파수	0.01	Hz
AI008	출력 전압	0.1	V
AI009	DC Link 전압	0.1	V
AI010	출력 파워	0.1	kW
AI011	V1	0~3FF	-
AI012	V2	0~3FF	-
AI013	I	0~3FF	-
AI014	RPM	-	-

2) 디지털 입력

어드레스	레지스터 명	범위	단위
D1000	운전지령 정지	-	-
D1001	운전지령 정방향운전	-	-
D1002	운전지령 역방향운전	-	-
D1003	운전지령 플트리셋	-	-
D1004	운전지령 비상정지	-	-
D1005	운전지령 Not Used	-	-
D1006	운전지령 Not Used	-	-
D1007	운전지령 Not Used	-	-
D1008	운전지령 Not Used	-	-
D1009	운전지령 Not Used	-	-
D1010	운전지령 Not Used	-	-
D1011	운전지령 Not Used	-	-
D1012	운전지령 Not Used	-	-

어드레스	레지스터 명	범위	단위
D1013	운전지령 Not Used	-	-
D1014	운전지령 Not Used	-	-
D1015	운전지령 Not Used	-	-
D1016	운전상태 정지	-	-
D1017	운전상태 정방향 운전중	-	-
D1018	운전상태 역방향 운전중	-	-
D1019	운전상태 Fault (Trip)	-	-
D1020	운전상태 가속중	-	-
D1021	운전상태 감속중	-	-
D1022	운전상태 속도도달	-	-
D1023	운전상태 DC Braking 중	-	-
D1024	운전상태 정지중	-	-
D1025	운전상태 Not Used	-	-
D1026	운전상태 Brake 개방신호	-	-
D1027	운전상태 정방향 지령(지령이면 1)	-	-
D1028	운전상태 역방향 지령(지령이면 1)	-	-
D1029	운전상태 REM. R/S	-	-
D1030	운전상태 REM. Freq.	-	-
D1031	운전상태 Not Used	-	-
D1032	트립정보 OCT	-	-
D1033	트립정보 OVT	-	-
D1034	트립정보 EXT-A	-	-
D1035	트립정보 EST(BX)	-	-
D1036	트립정보 COL	-	-
D1037	트립정보 GFT(Ground Fault)	-	-
D1038	트립정보 OHT(인버터 과열)	-	-
D1039	트립정보 ETH(전동기 과열)	-	-
D1040	트립정보 OLT(과부하)	-	-
D1041	트립정보 HW-Diag	-	-
D1042	트립정보 EXT-B	-	-
D1043	트립정보 EEP(Parameter Write Error)	-	-
D1044	트립정보 FAN(LOCK & Open Error)	-	-
D1045	트립정보 PO(Phase Open)	-	-
D1046	트립정보 IOLT	-	-
D1047	트립정보 LVT	-	-
D1048	입력단자정보 P1	-	-
D1049	입력단자정보 P2	-	-
D1050	입력단자정보 P3	-	-
D1051	입력단자정보 P4	-	-
D1052	입력단자정보 P5	-	-
D1053	입력단자정보 P6	-	-
D1054	입력단자정보 P7	-	-
D1055	입력단자정보 P8	-	-
D1056	입력단자정보 Not Used	-	-
D1057	입력단자정보 Not Used	-	-

부록1 I/O 드라이버 설정방법

어드레스	레지스터 명	범위	단위
D1058	입력단자정보 Not Used	-	-
D1059	입력단자정보 Not Used	-	-
D1060	입력단자정보 Not Used	-	-
D1061	입력단자정보 Not Used	-	-
D1062	입력단자정보 Not Used	-	-
D1063	입력단자정보 Not Used	-	-
D1064	출력단자정보 Not Used	-	-
D1065	출력단자정보 Not Used	-	-
D1066	출력단자정보 Not Used	-	-
D1067	출력단자정보 Not Used	-	-
D1068	출력단자정보 MO(Multi-Output With OC)	-	-
D1069	출력단자정보 Not Used	-	-
D1070	출력단자정보 Not Used	-	-
D1071	출력단자정보 3ABC	-	-
D1072	출력단자정보 Not Used	-	-
D1073	출력단자정보 Not Used	-	-
D1074	출력단자정보 Not Used	-	-
D1075	출력단자정보 Not Used	-	-
D1076	출력단자정보 Not Used	-	-
D1077	출력단자정보 Not Used	-	-
D1078	출력단자정보 Not Used	-	-
D1079	출력단자정보 Not Used	-	-
D1080	트립정보-B COM(I/O Board Reset)	-	-
D1081	트립정보-B FLTL	-	-
D1082	트립정보-B NTC	-	-
D1083	트립정보-B REEP	-	-
D1084	트립정보-B Not Used	-	-
D1085	트립정보-B Not Used	-	-
D1086	트립정보-B Not Used	-	-
D1087	트립정보-B Not Used	-	-
D1088	트립정보-B Not Used	-	-
D1089	트립정보-B Not Used	-	-
D1090	트립정보-B Not Used	-	-
D1091	트립정보-B Not Used	-	-
D1092	트립정보-B Not Used	-	-
D1093	트립정보-B Not Used	-	-
D1094	트립정보-B Not Used	-	-
D1095	트립정보-B Not Used	-	-

3) 문자열 입력

어드레스	레지스터 명	범위	단위
SI000	인버터 모델	-	-
SI001	인버터 용량	-	-
SI002	인버터 입력 전압	-	-
SI003	버전	-	-

4) 아날로그 출력

어드레스	레지스터 명	범위	단위
A0000	파라미터 설정허용 [0:파라미터 설정금지, 1:파라미터 설정허용]	-	-
A0001	지령주파수	0.1	Hz
A0002	가속 시간	0.1	sec
A0003	감속 시간	0.1	sec

5) 디지털 출력

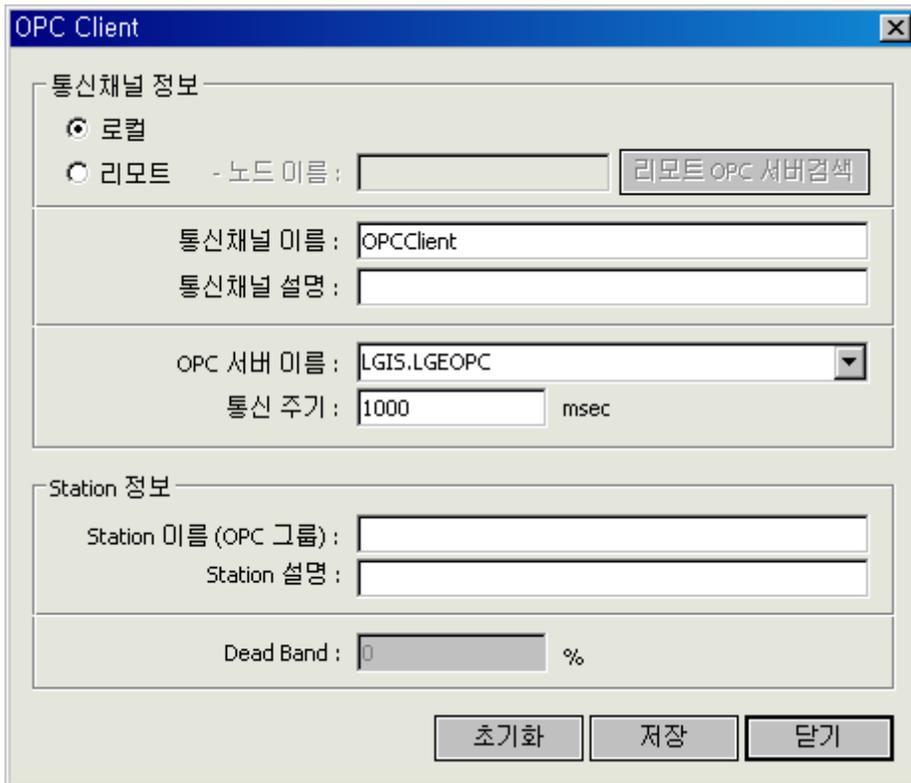
어드레스	레지스터 명	범위	단위
D0000	운전지령 정지	-	-
D0001	운전지령 정방향 운전	-	-
D0002	운전지령 역방향 운전	-	-
D0003	운전지령 폴트 리셋	-	-
D0004	운전지령 비상정지	-	-

10.OPC Client 드라이버

(1) 통신채널 추가



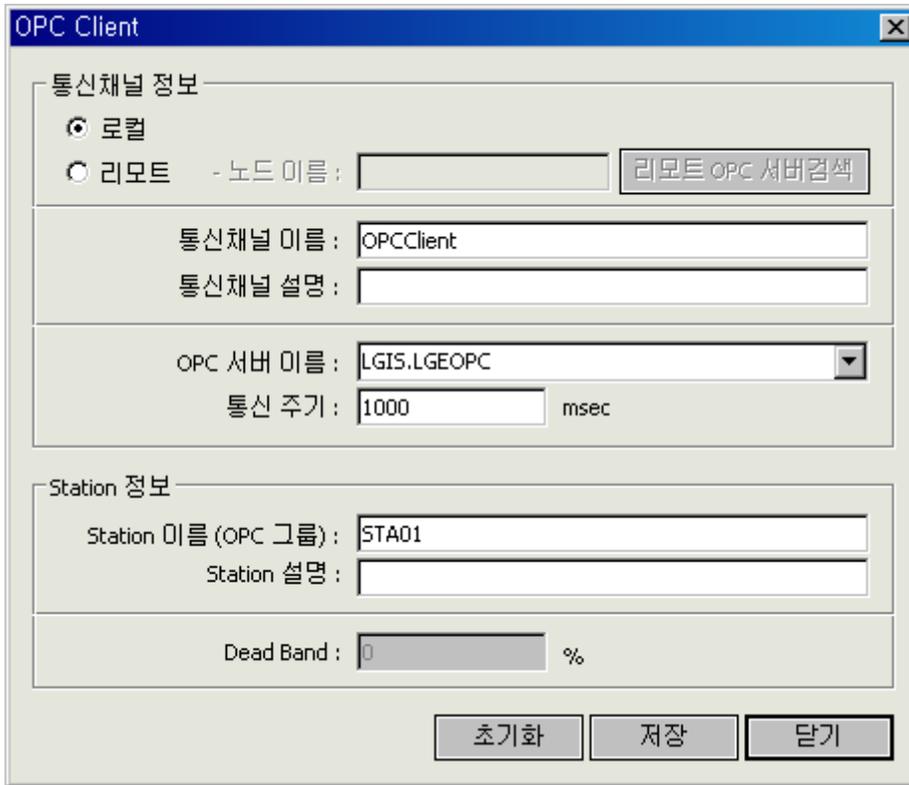
I/O 드라이버 리스트에서 “OPCClient” 를 선택한 후 “추가” 버튼을 누릅니다.



- (a) 통신채널 이름 : 통신채널 이름을 선택합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) OPC 서버 이름 : OPC서버 이름을 선택합니다. 로컬 라디오 버튼을 선택하면 로컬 PC상에 설치된 OPC 서버리스트 들이 콤보박스의 리스트에 표시됩니다. 리모트 라디오 버튼을 선택하고 노드 이름을 입력한 후 “리모트 OPC

- “서버검색” 버튼을 누르면 리모트 PC상에 설치된 OPC 서버리스트들이 콤보박스의 리스트에 표시됩니다.
- (d) 통신 주기 : 데이터 업데이트 주기를 설정합니다.
 - (e) 로컬/리모트 : 사용할 OPC 서버가 로컬 PC상에 있는지 리모트 PC 상에 있는지 선택합니다.
 - (f) 노드 이름 : 리모트 PC의 이름을 입력합니다. (“리모트” 를 선택 했을 때만 Enable 됩니다.)
 - (g) 리모트 OPC 서버검색 : 리모트 PC상에 설치된 OPC 서버 리스트를 검색합니다. 검색 결과는 “OPC 서버 이름” 의 콤보박스 리스트에 입력됩니다. (“리모트” 를 선택 했을 때만 Enable 됩니다.)

(2) Station 추가



- (a) Station 이름(OPC 그룹) : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.

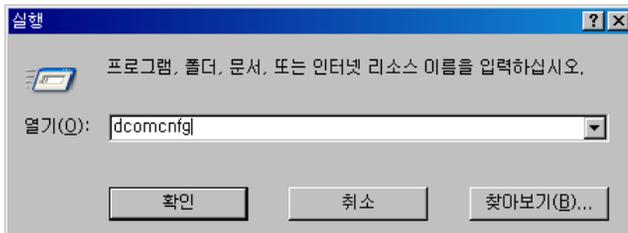
(3) DCOM 설정

(a) 기본

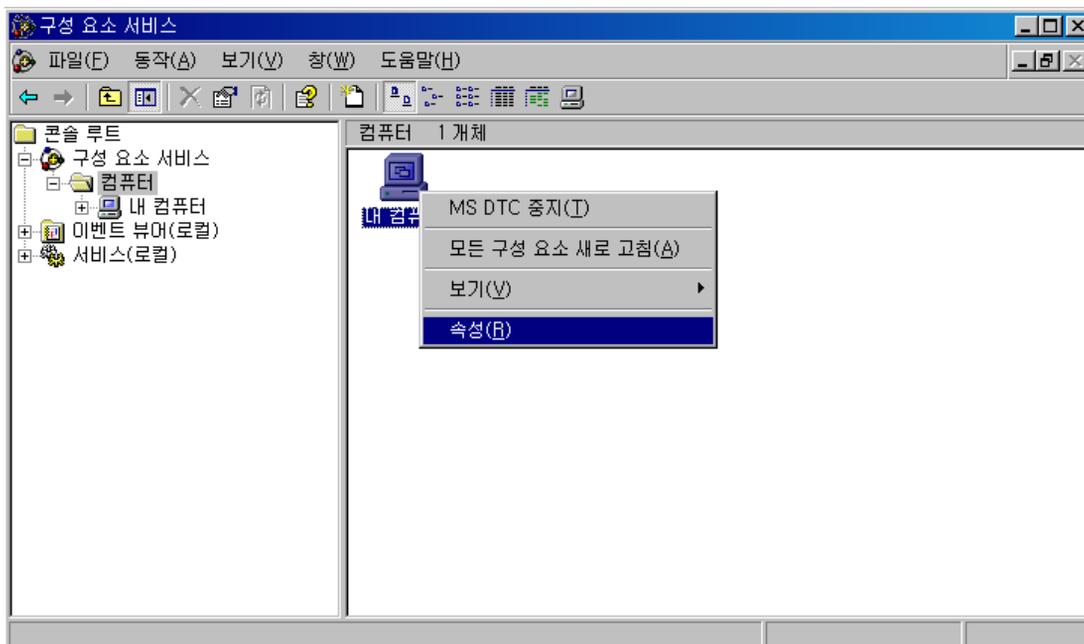
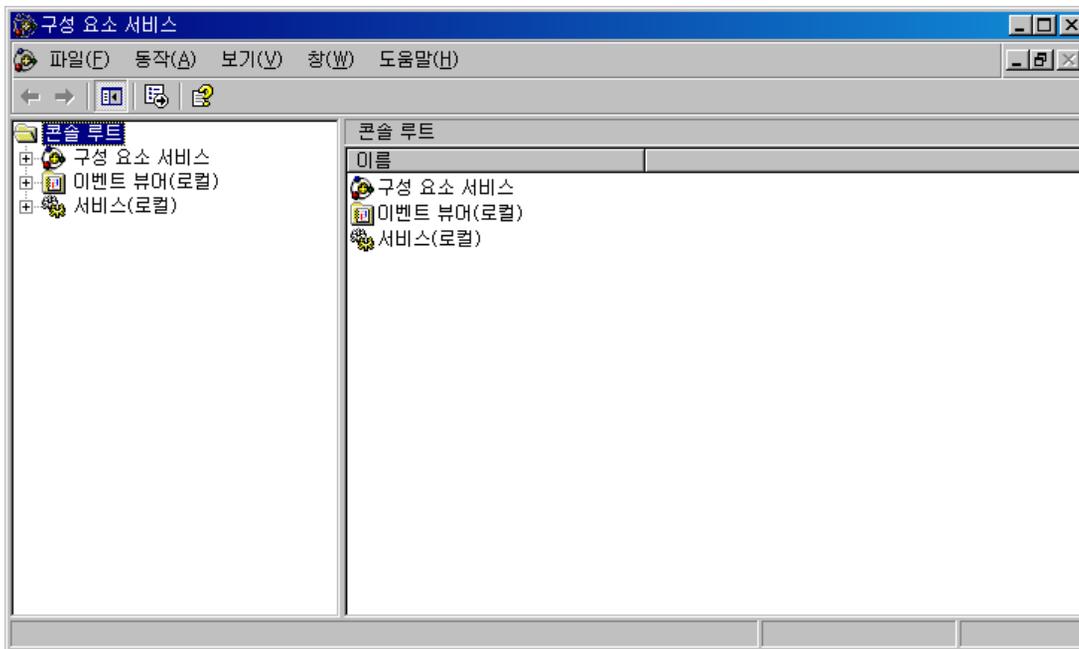
- 리모트 PC상에 설치된 OPC서버를 사용하기 위해서는 DCOM 설정을 해 주어야 합니다.
- DCOM 설정은 아래의 절차를 따라서 설정해야 하며 로컬 PC와 리모트 PC에 동일한 설정을 해 주어야 합니다.
- 아래 설명은 윈도우 XP에서 DCOM 설정을 예로 들어 설명한 것이며 다른 OS에서는 이와 유사한 형태의 DCOM 설정을 해 주어야 합니다.
- 로컬 PC와 리모트 PC에는 동일한 계정이 등록 되어있어야 하며 동일한 암호를 사용해야 합니다.
- 로컬 PC와 리모트 PC에는 동일한 계정으로 로그인 되어있어야 합니다.
- 리모트 PC의 방화벽 설정이 “사용안함” 으로 설정되어 있어야 합니다.
- 리모트 PC상에 설치된 OPC서버를 사용하기 위해서는 로컬 PC상에 기본적인 OPC 파일들이 설치되어 있어야 합니다. (OPC Core Componets Redistributable 프로그램을 설치합니다. www.opcfoundation.org 참조)

(b) 구성 요소 서비스 설정

부록1 I/O 드라이버 설정방법

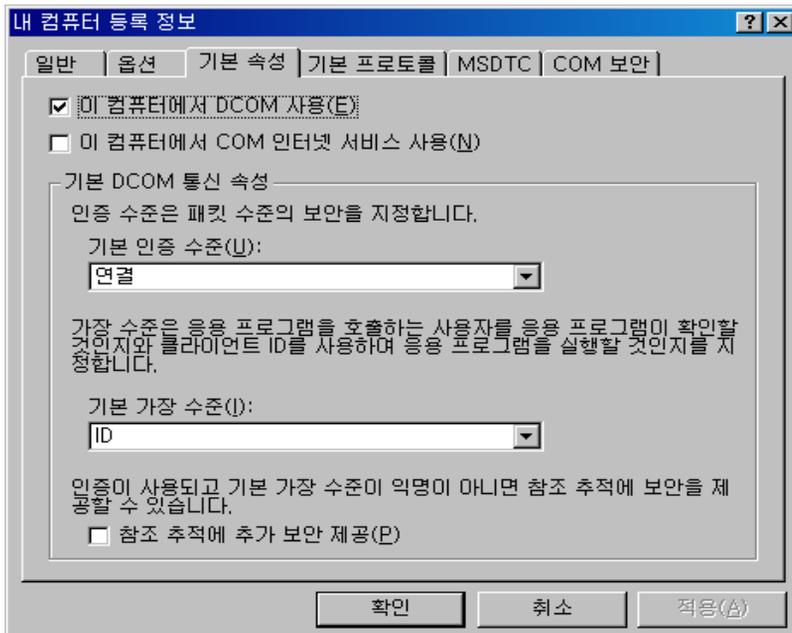


- 입력 창에 “dcomcnfg” 를 입력하고 확인 버튼을 누릅니다.
- 아래와 같은 “구성 요소 서비스” 창이 표시됩니다.

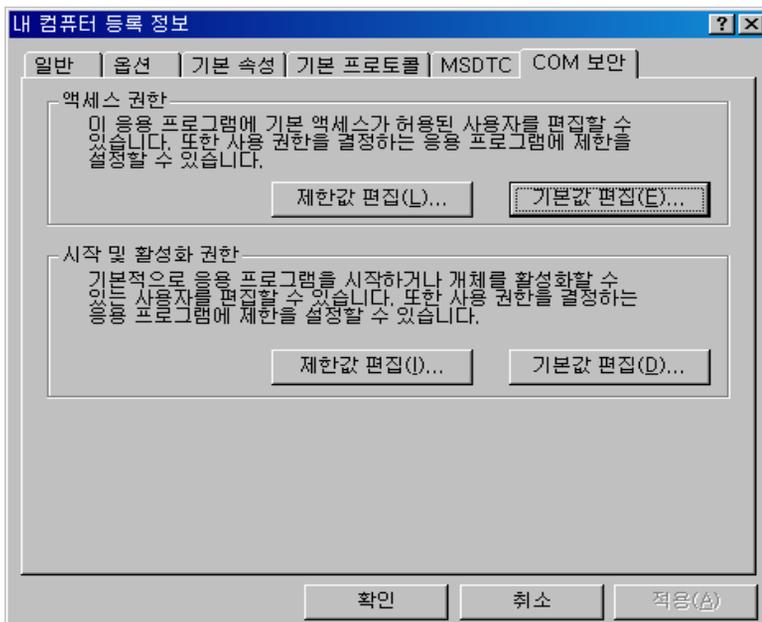


- “내 컴퓨터” 항목을 찾아서 “속성” 메뉴를 실행해 줍니다.

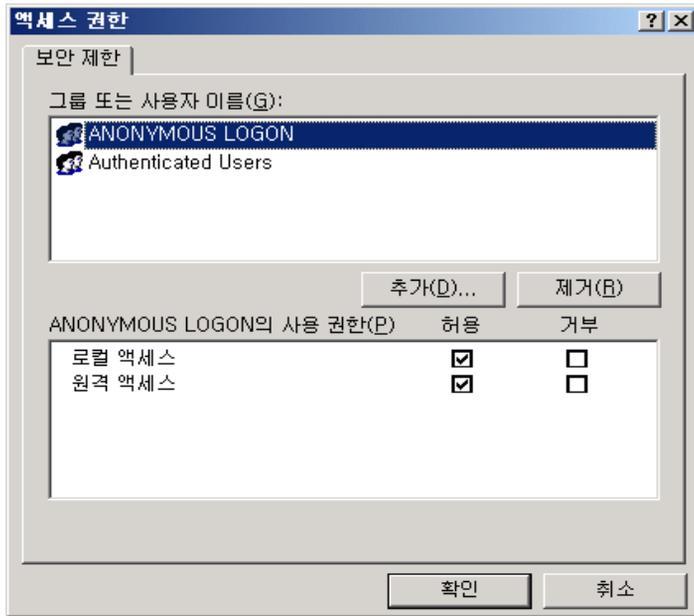
- 아래와 같은 “내 컴퓨터 등록 정보” 창이 표시됩니다.



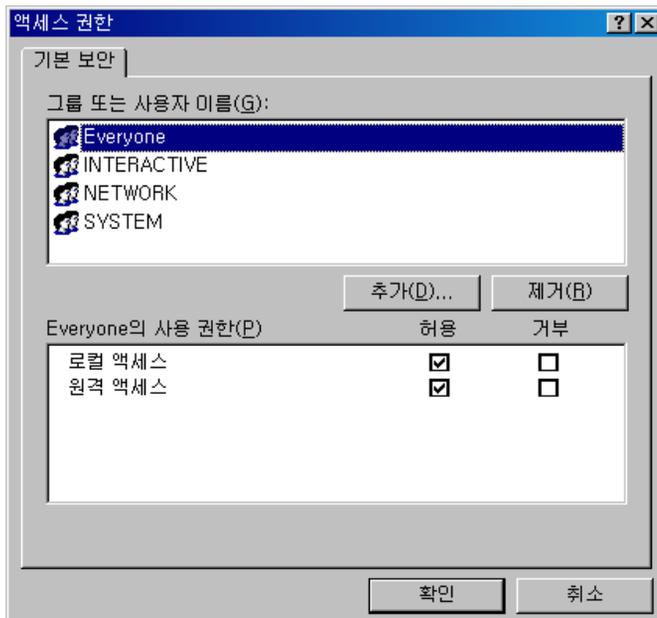
- “기본 속성” 탭을 선택합니다
- “이 컴퓨터에서 DCOM 사용” 체크 박스를 “사용함” 으로 체크합니다.
- “기본 인증 수준” 항목을 “연결” 로 선택합니다.
- “기본 가장 수준” 항목을 “ID” 로 선택 합니다.



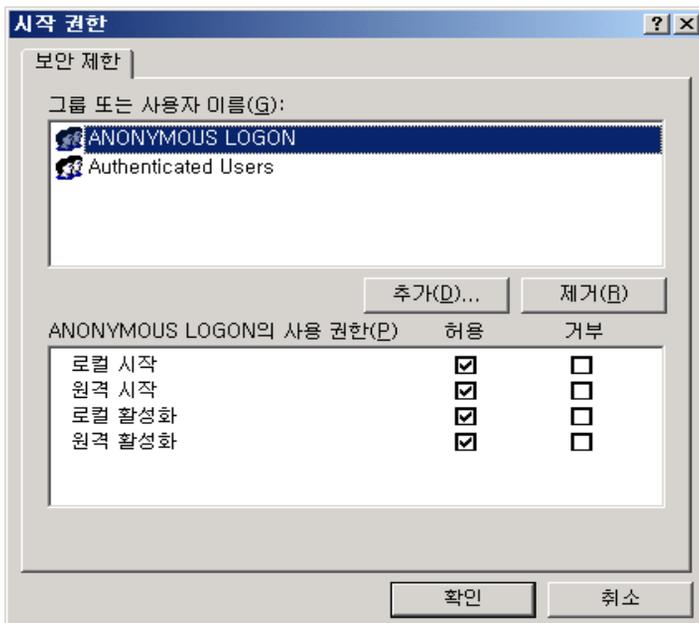
- “COM 보안” 탭을 선택 합니다.
- “액세스 권한” 을 설정하기 위해 “제한값 편집” 과 “기본값 편집” 버튼을 누르고 아래와 같이 설정합니다.
 - 제한값 편집 ; ANONYMOUS LOGON, Authenticated Users 사용자 각각에 대해 “로컬 액세스” 와 “원격 액세스” 항목을 모두 “허용함” 으로 체크를 합니다.



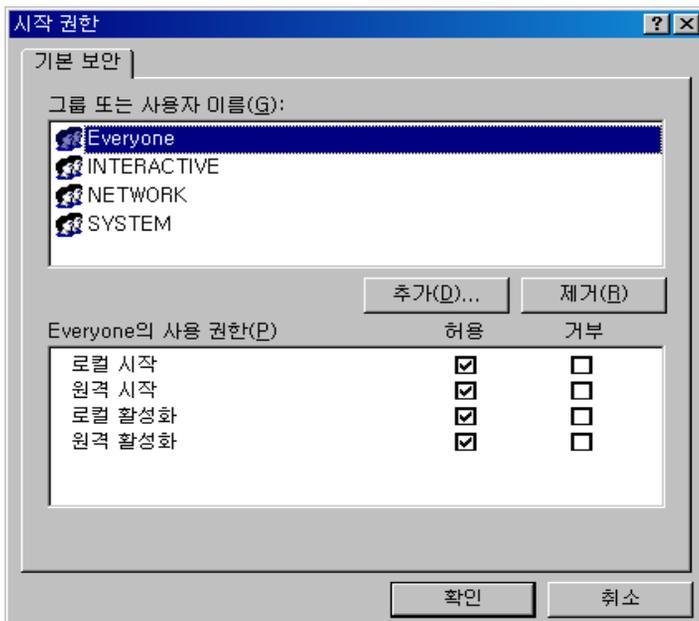
- 기본값 편집 : Everyone, INTERACTIVE, NETWORK, SYSTEM 사용자 각각에 대해 “로컬 액세스”와 “원격 액세스” 항목을 모두 “허용함”으로 체크를 합니다.



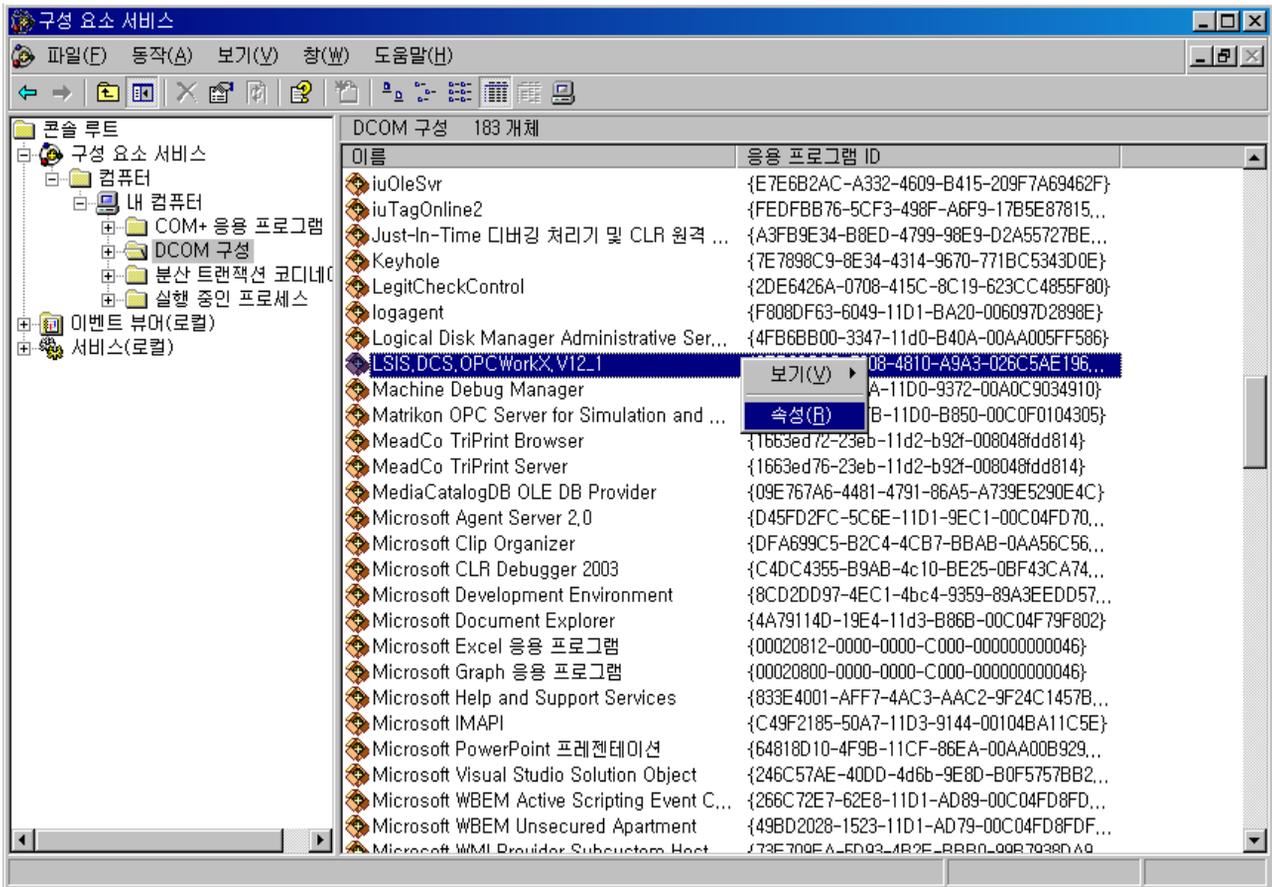
- “시작 및 활성화 권한”을 설정하기 위해 “기본값 편집” 버튼을 누르고 아래와 같이 설정합니다.
 - 제한값 편집 : ANONYMOUS LOGON, Authenticated Users 사용자 각각에 대해 “로컬 시작”, “원격 시작”, “로컬 활성화”, “원격 활성화” 항목을 모두 “허용함”으로 체크를 합니다.



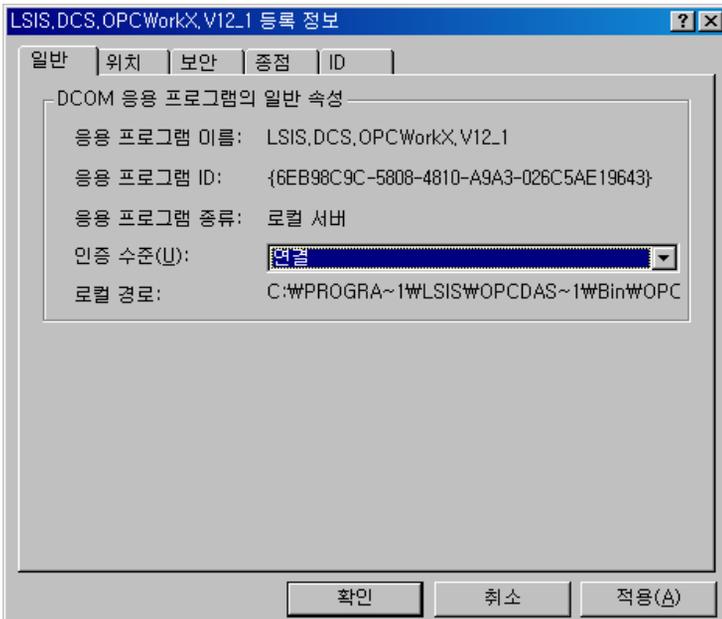
- 기본값 편집 : Everyone, INTERACTIVE, NETWORK, SYSTEM 사용자 각각에 대해 “로컬 시작” , “원격 시작” , “로컬 활성화” , “원격 활성화” 항목을 모두 “허용함” 으로 체크를 합니다.



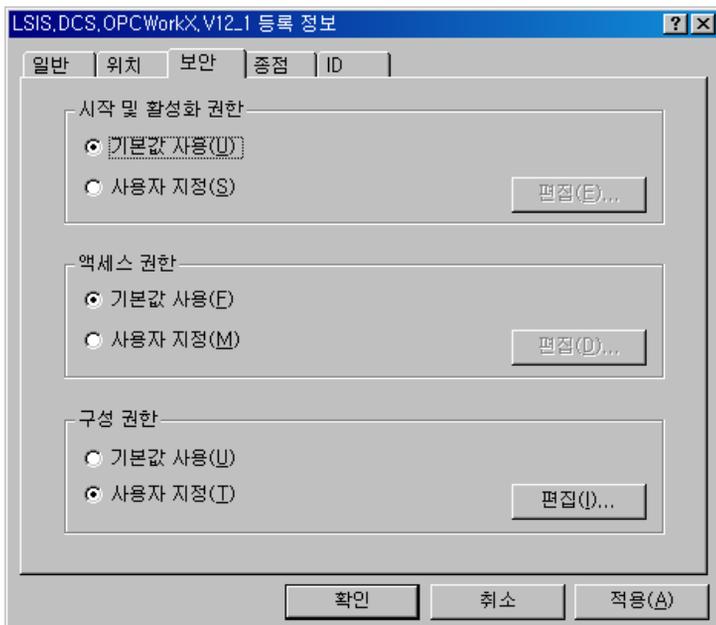
- 다음은 사용하고자 하는 OPC 서버에 대해 DCOM 설정을 합니다.
- 사용하고자 하는 OPC 서버는 모두 아래의 절차를 따라 각각 설정을 해야 합니다.



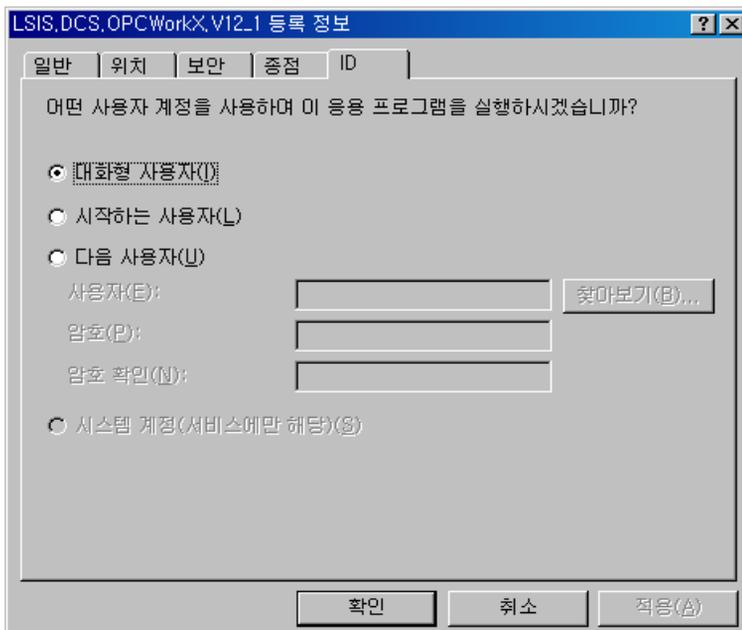
- 사용하기 원하는 OPC 서버를 찾아 “속성” 메뉴를 실행합니다.



- “일반” 탭에서 “인증 수준” 을 “연결” 로 선택합니다.

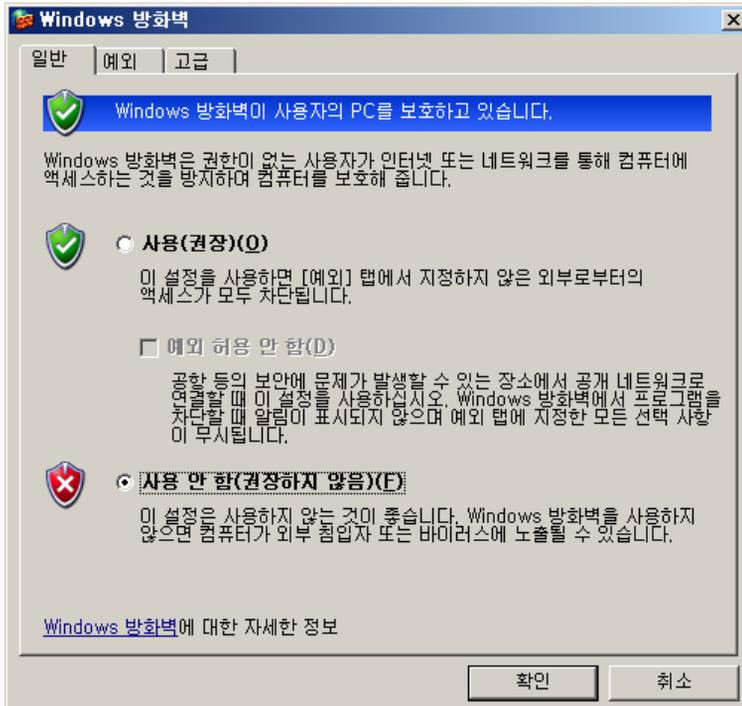


- “보안” 탭을 선택합니다.
- “시작 및 활성화 권한” 항목에 대해 “기본값 사용” 을 선택합니다.
- “액세스 권한” 항목에 대해 “기본값 사용” 을 선택합니다.

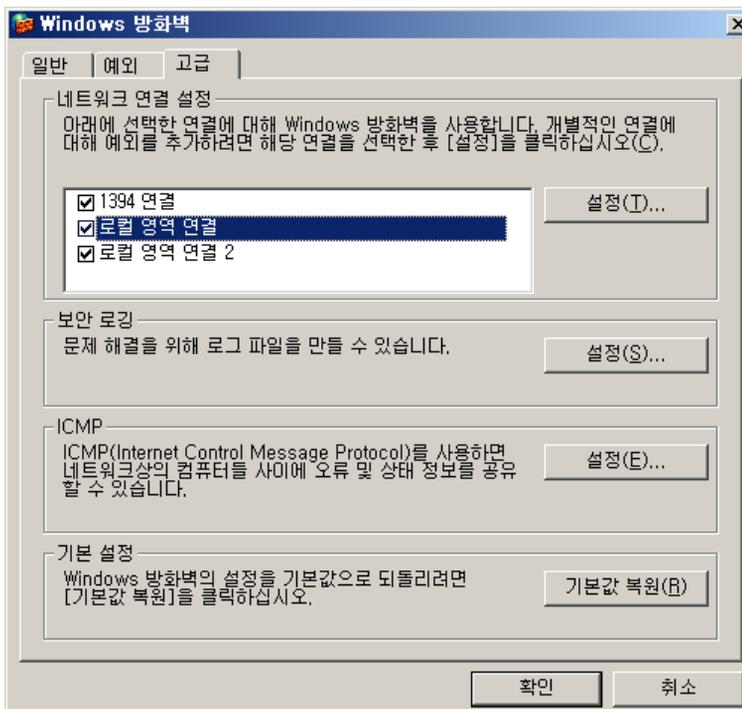


- “ID” 탭을 선택합니다.
- “대화형 사용자” 항목을 선택합니다.

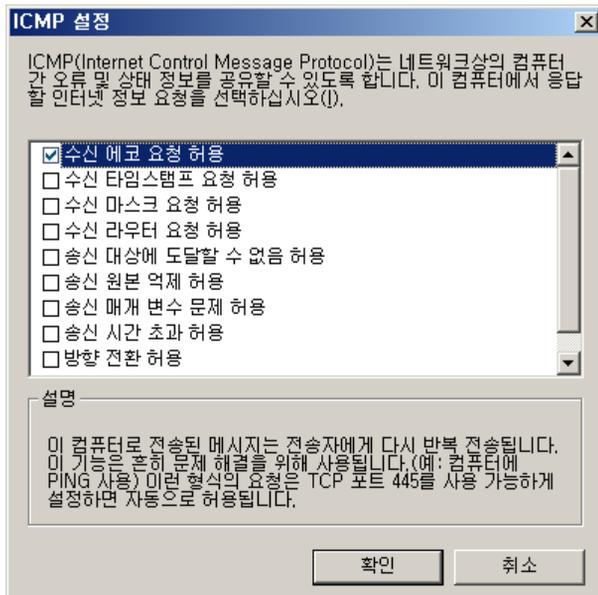
(C) 방화벽 설정



- Windows 방화벽 설정창을 표시합니다.
- “사용 안 함” 을 선택 합니다.
- “고급” 탭을 선택합니다.



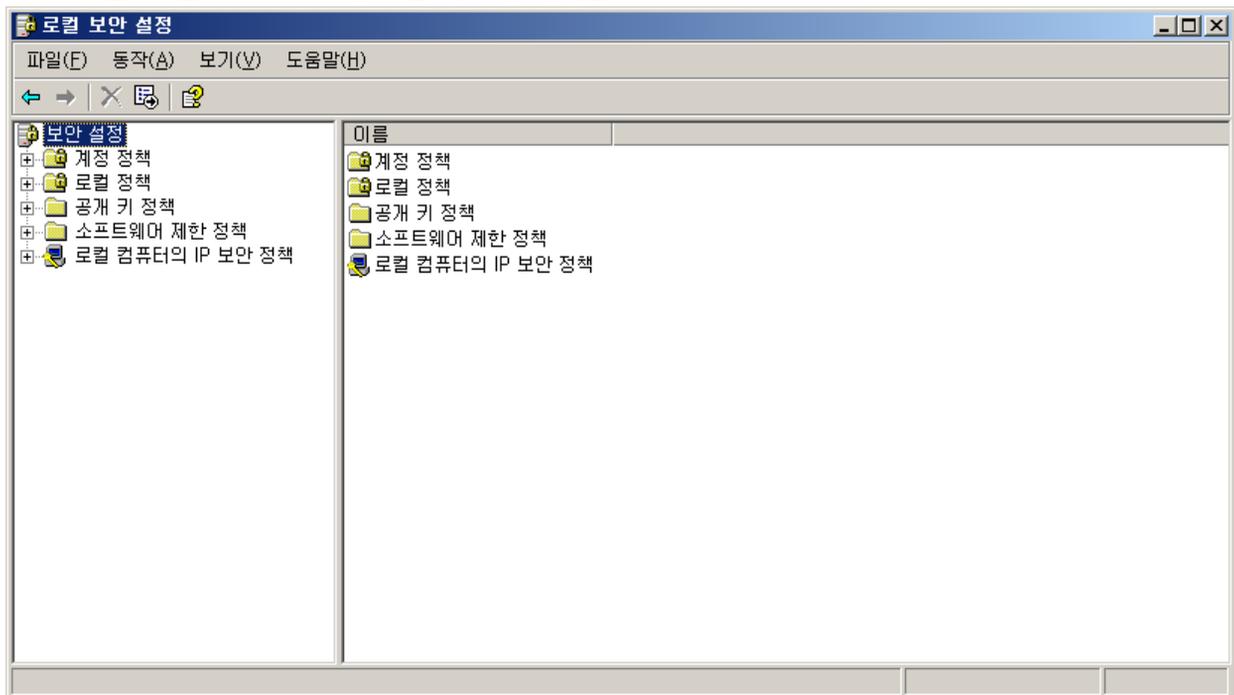
- ICMP 부분의 “설정” 버튼을 클릭 합니다.



- “수신 에코 요청 허용” 을 체크 합니다.

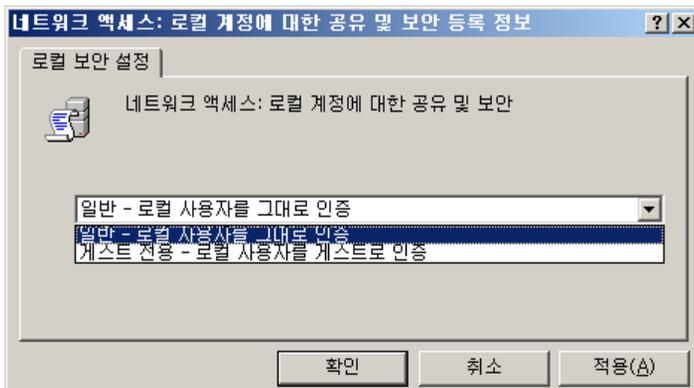
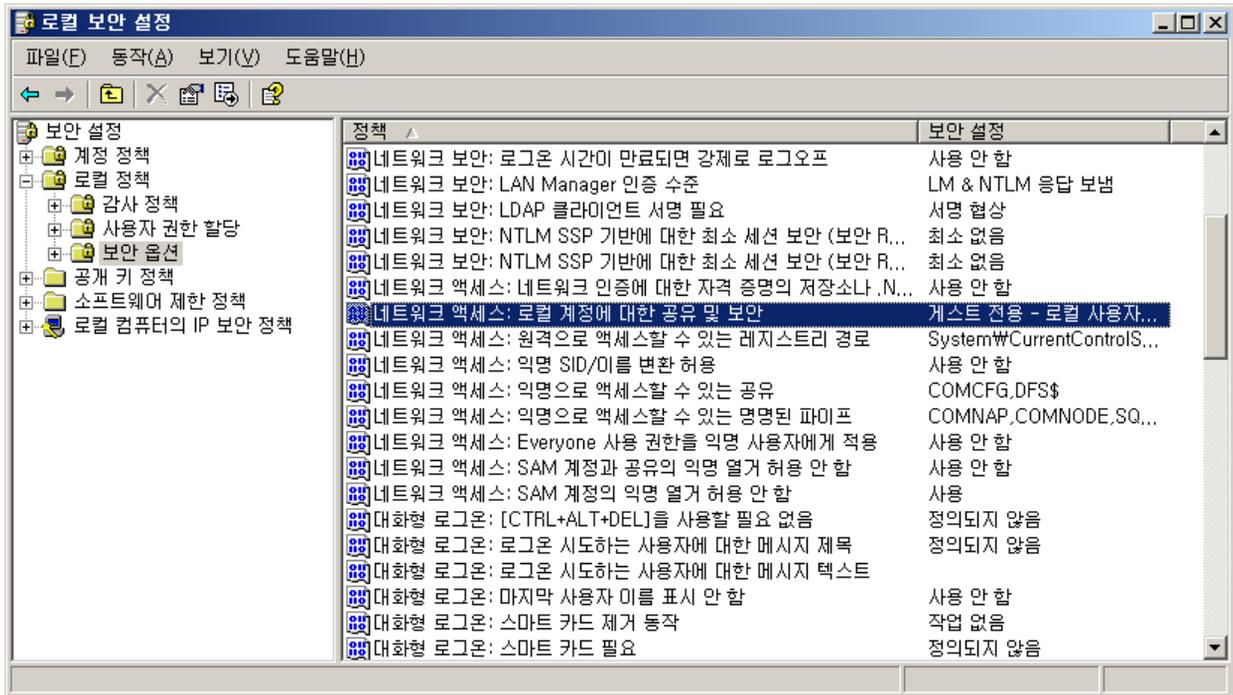
(d) 로컬 보안 설정

- 제어판 → 관리도구 → 로컬 보안 정책 을 실행합니다.



- 로컬 정책 → 보안 옵션 → “네트워크 액세스 : 로컬 계정에 대한 공유 및 보안” 항목을 실행합니다.

부록1 I/O 드라이버 설정방법



- “일반 - 로컬 사용자를 그대로 인증” 을 선택 합니다.

11. MeIsecQEnet 드라이버

(1) 통신채널 추가



I/O 드라이버 리스트에서 “MeIsecQEnet” 을 선택한 후 “추가” 버튼을 누릅니다.

- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- (d) 서버 IP Address #2 : 라인 이중화를 사용할 경우, 사용하게 될 두번째 IP Address를 입력합니다.
- (e) 최대 응답 시간 : 장비로부터 데이터를 한번 주고 받는데 걸리는 시간입니다.
- (f) 통신 Time Out : 장비에게 데이터를 요청한 후 타임아웃으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (g) 통신 주기 : 데이터를 요청하는 주기입니다.
- (h) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.

(2) Station 추가

통신채널 정보

통신채널 이름 : MelsecQEnet
 통신채널 설명 :

서버 IP Address #1 : 127 . 0 . 0 . 1
 서버 IP Address #2 : 0 . 0 . 0 . 0
 최대 응답시간 : 200 msec [100 ~ 2000]
 통신 Time Out : 200 msec [100 ~ 2000]
 통신 주기 : 400 msec [100 ~ 600000]
 재시도 횟수 : 3 회 [1 ~ 5]

Station 정보

Station 이름 :
 Station 설명 :

PLC CPU 종류 : Q_Q00J
 라인 이중화 장비 이중화

PLC IP Address #1-1 : 0 . 0 . 0 . 0 포트#1-1 : 1280
 PLC IP Address #1-2 : 0 . 0 . 0 . 0 포트#1-2 : 1280
 PLC IP Address #2-1 : 0 . 0 . 0 . 0 포트#2-1 : 1280
 PLC IP Address #2-2 : 0 . 0 . 0 . 0 포트#2-2 : 1280

통신 방식 : TCP
 프로토콜 형식 : Binary
 구조태그 유형 : 사용안함

네트워크 번호 : 0
 PLC 번호 : 255
 모듈 I/O 번호 : 1023
 모듈 국번호 : 0
 블럭 사이즈 : 100 [1 ~ 960 WORD]

신규 저장 닫기

- (a) Station 이름 : Station 이름을 입력합니다.
 (b) Station 설명 : Station 설명을 입력합니다.
 (c) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
 (d) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
 (e) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다.
 (f) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
 (g) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.
 (h) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
 (i) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.

- (j) 포트 #1-1 : IP Address에 해당하는 포트 번호를 입력합니다.
- (k) 포트 #1-2 : IP Address에 해당하는 포트 번호를 입력합니다.
- (l) 포트 #2-1 : IP Address에 해당하는 포트 번호를 입력합니다.
- (m) 포트 #2-2 : IP Address에 해당하는 포트 번호를 입력합니다.
- (n) 통신 방식 : TCP와 UDP 중의 한가지를 선택합니다.
- (o) 프로토콜 형식 : 아스키와 바이너리 중의 한가지를 선택합니다.
- (p) 네트워크 번호 : 네트워크 번호를 입력합니다.
- (q) PLC 번호 : PLC 번호를 입력합니다.
- (r) 모듈 I/O 번호 : 모듈 I/O 번호를 입력합니다.
- (s) 모듈 국번호 : 모듈 국번호를 입력합니다.
- (t) 블록 사이즈 : PLC 통신 블록의 크기를 입력합니다.

(3) 입출력 주소

(a) I/O 어드레스 맵

디바이스 종류	비트	워드	CPU 종류에 따른 어드레스 범위 (디플트 값)		표현		예 제(InfoU 어드레스)
			Q02(H), Q06H, Q12H, Q25H, Q12PH, Q25PH, Q2A, Q2A-S1, Q2AS, Q2AS-S1, Q2ASH, Q2ASH-1, Q3A, Q4A, Q4AR	Q00J, Q00, Q01	10 진수	16 진수	
SM	●	-	000000~002047	000000~001023	●	-	SM0 , SM10, SM197
SD	-	●	000000~002047	000000~001023	●	-	SD1, SD2047
X	●	-	000000~001FFF	000000~0007FF	-	●	X0~XF, X10~X1F, X1FFF
Y	●	-	000000~001FFF	000000~0007FF	-	●	Y0~YF, Y10~Y1F, YFFF
M	●	-	000000~008191	000000~008191	●	-	M0, M1F
L	●	-	000000~008191	000000~002047	●	-	L0~L11, L15, L100
F	●	-	000000~002047	000000~001023	●	-	F0~F17, F9, F2000
V	●	-	000000~002047	000000~001023	●	-	V1000, V2047
B	●	-	000000~001FFF	000000~0007FF	-	●	B1F, B1000, B1FFF
D	-	●	000000~012287	000000~011135	●	-	D0, D1000, D10000, D12287
W	-	●	000000~001FFF	000000~0007FF	-	●	W0, WF, W1F, W1FFF
TS	●	-	000000~002047	000000~000511	●	-	TS0, TS12, TS1000, TS2000
TC	●	-			●	-	TC1, TC17, TC200, TC2047
TN	-	●			●	-	TN0, TN2047
SS	●	-			●	-	SS0, SS2047
SC	●	-			●	-	SC0~SS2047
SN	-	●			●	-	SNO~SN2047
CS	●	-			●	-	CS0~CS1023
CC	●	-	000000~001023	000000~000511	●	-	CC0~CC1023
CN	-	●	●	-	CNO~CC1023		
SB	●	-	000000~0007FF	000000~0003FF	-	●	SB0~SB7FF
SW	-	●	000000~0007FF	000000~0003FF	-	●	SW0~SW7FF
S	●	-	000000~008191	000000~002047	●	-	S0~S8191 (Q00J, Q00, Q01 액세스불가)
DX	●	-	000000~001FFF	000000~0007FF	-	●	DX0~DX1FFF
DY	●	-	000000~001FFF	000000~0007FF	-	●	DY0~DY1FFF
Z	-	●	000000~000015	000000~000009	●	-	Z0~Z15
R	-	●	000000~032767	000000~032767	●	-	R0~R32767
ZR	-	●	000000~0FE7FF	000000~00FFFF	-	●	ZR0~ZRFE7FF

12. MeIsecQSerial 드라이버

(1) 통신채널 추가



I/O 드라이버 리스트에서 “MeIsecQSerial” 을 선택한 후 “추가” 버튼을 누릅니다.

MelsecQ Serial 설정

통신채널 정보

통신채널 이름

통신채널 설명

라인 이중화 장비 이중화

통신포트#1

통신포트#2

통신포트#3

통신포트#4

통신속도

패리티비트

데이터비트

스톱비트

명령어 최대 처리시간 msec [100 ~ 2000]

통신 Time Out msec [100 ~ 20000]

통신 주기 msec [100 ~ 1800000]

재시도 횟수 회 [1 ~ 5]

흐름제어 NONE RTS 제어

통신방식 R5232 R5422/485

Delay Time (Before) msec

Delay Time (After) msec

Station 정보

Station 이름

Station 설명

CPU Type

Station 번호

구조태그 유형

네트워크 번호 :

PLC 번호 :

모듈 I/O 번호 :

모듈 국번호 :

블럭 사이즈 :

- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (d) 통신포트#1 : 통신포트를 선택합니다.
- (e) 통신포트#2 : 통신포트를 선택합니다.
- (f) 통신포트#3 : 통신포트를 선택합니다.
- (g) 통신포트#4 : 통신포트를 선택합니다.
- (h) 통신속도 : 통신 속도를 선택합니다.
- (i) 패리티비트 : 패리티비트를 선택합니다.
- (j) 데이터비트 : 데이터비트를 선택합니다.

- (k) 스톱비트 : 스톱비트를 선택합니다.
- (l) 명령어 최대 처리시간 : 한번 데이터를 요청한뒤 응답을 받을때까지 걸리는 시간을 말합니다.
- (m) 통신 Time Out : 장비에게 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다.
통신 오류를 판별하는 근거가 됩니다.
- (n) 통신 주기 : 데이터를 요청하는 주기입니다.
- (o) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (p) 흐름제어 : NONE 과 RTS제어 중 하나를 선택합니다.
- (q) Delay Time (Before) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.
- (r) Delay Time (After) : 통신 Delay Time을 설정합니다. RTS 제어를 사용할 때에만 설정합니다.

(2) Station 추가

The screenshot shows the 'MelsecQ Serial 설정' dialog box. It is divided into two main sections:

- 통신채널 정보 (Communication Channel Information):**
 - 통신채널 이름: MelsecQSerial
 - 통신채널 설명: (empty)
 - 라인 미중화: (unchecked)
 - 장비 미중화: (unchecked)
 - 통신포트#1: COM1
 - 통신포트#2: (empty)
 - 통신포트#3: (empty)
 - 통신포트#4: (empty)
 - 통신속도: 9600
 - 패리티비트: No Parity
 - 데이터비트: 8
 - 스톱비트: 1
 - 흐름제어: RTS 제어 (selected), NONE
 - 통신방식: R5422/485 (selected), R5232
 - Delay Time (Before): 0 msec
 - Delay Time (After): 15 msec
 - 통신 Time Out: 500 msec [100 ~ 20000]
 - 재시도 횟수: 3 회 [1 ~ 5]
- Station 정보 (Station Information):**
 - Station 이름: (empty)
 - Station 설명: (empty)
 - CPU Type: Q_Q06H
 - Station 번호: 0
 - 구조태그 유형: 사용안함
 - 네트워크 번호: 0
 - PLC 번호: 255
 - 모듈 I/O 번호: 1023
 - 모듈 국번호: 0
 - 블럭 사이즈: 100 [1 ~ 480 WORD]

At the bottom of the dialog, there are three buttons: '신규' (New), '저장' (Save), and '닫기' (Close).

- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) CPU Type : CPU 타입을 선택합니다.

- (d) Station 번호 : Station 번호를 입력합니다.
- (e) 네트워크 번호 : 네트워크 번호를 입력합니다.
- (f) PLC 번호 : PLC 번호를 입력합니다.
- (g) 모듈 I/O 번호 : 모듈 I/O 번호를 입력합니다.
- (h) 모듈 국번호 : 모듈 국번호를 입력합니다.
- (i) 블록 사이즈 : PLC 통신 블록의 크기를 입력합니다.

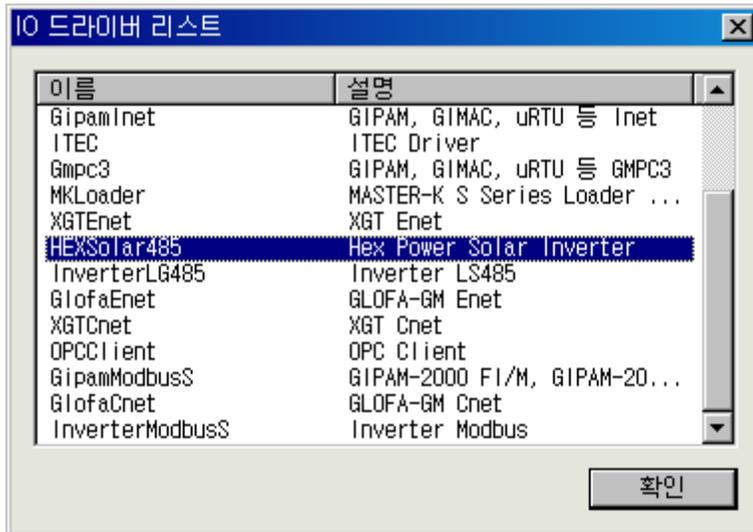
(3) 입출력 주소

(a) I/O Address Map

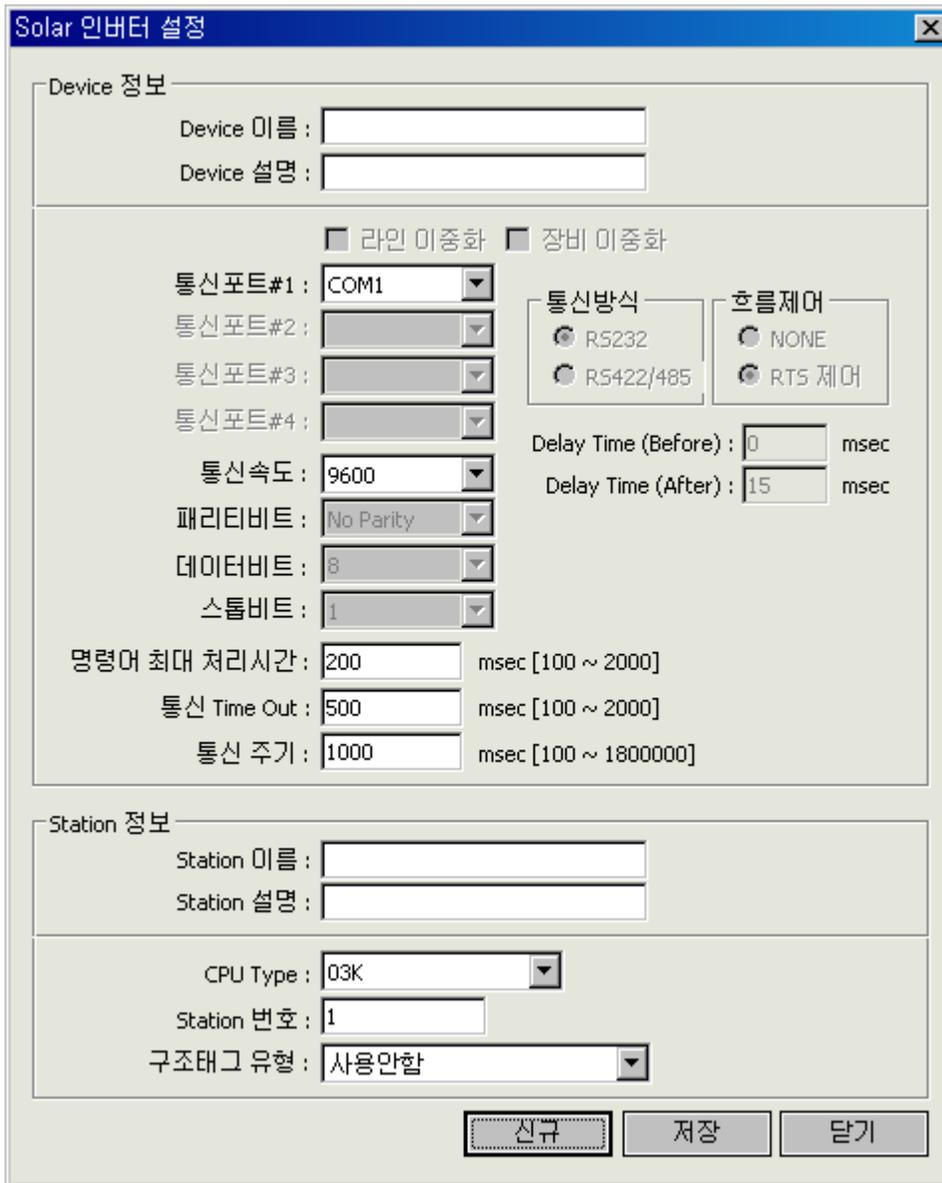
디바이스 종류	비트	워드	CPU 종류에 따른 어드레스 범위 (디폴트 값)		표현		예 제(InfoU 어드레스)
			Q02(H), Q06H, Q12H, Q25H, Q12PH, Q25PH, Q2A, Q2A-S1, Q2AS, Q2AS-S1, Q2ASH, Q2ASH-1, Q3A, Q4A, Q4AR	Q00J, Q00, Q01	10 진수	16 진수	
SM	●		000000~002047	000000~001023	●		SM0 , SM10, SM197
SD		●	000000~002047	000000~001023	●		SD1, SD2047
X	●		000000~001FFF	000000~0007FF		●	X0~XF, X10~X1F, X1FFF
Y	●		000000~001FFF	000000~0007FF		●	Y0~YF, Y10~Y1F, YFFF
M	●		000000~008191	000000~008191	●		M0, M1F
L	●		000000~008191	000000~002047	●		L0~L11, L15, L100
F	●		000000~002047	000000~001023	●		F0~F17, F9, F2000
V	●		000000~002047	000000~001023	●		V1000, V2047
B	●		000000~001FFF	000000~0007FF		●	B1F, B1000, B1FFF
D		●	000000~012287	000000~011135	●		D0, D1000, D10000, D12287
W		●	000000~001FFF	000000~0007FF		●	W0, WF, W1F, W1FFF
TS	●		000000~002047	000000~000511	●		TS0, TS12, TS1000, TS2000
TC	●				●		TC1, TC17, TC200, TC2047
TN		●			●		TN0, TN2047
SS	●				●		SS0, SS2047
SC	●				●		SC0~SS2047
SN		●			●		SN0~SN2047
CS	●		000000~001023	000000~000511	●		CS0~CS1023
CC	●				●		CC0~CC1023
CN		●			●		CN0~CC1023
SB	●		000000~0007FF	000000~0003FF		●	SB0~SB7FF
SW		●	000000~0007FF	000000~0003FF		●	SW0~SW7FF
S	●		000000~008191	000000~002047	●		S0~S8191 (Q00J, Q00, Q01 액세스불가)
DX	●		000000~001FFF	000000~0007FF		●	DX0~DX1FFF
DY	●		000000~001FFF	000000~0007FF		●	DY0~DY1FFF
Z		●	000000~000015	000000~000009	●		Z0~Z15
R		●	000000~032767	000000~032767	●		R0~R32767
ZR		●	000000~0FE7FF	000000~00FFFF		●	ZR0~ZRFE7FF

13.HexSolar 드라이버

(1) 통신채널 추가



- I/O 드라이버 리스트에서 “HEXSolar485” 을 선택한 후 “추가” 버튼을 누릅니다.



- (a) Device 이름 : Device 이름을 입력합니다.
- (b) Device 설명 : Device 설명을 입력합니다.
- (c) 통신포트#1 : 사용할 COM 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다. HexPower의 태양광 인버터는 9600으로 고정되어 있으므로 다른 값을 선택하면 안됩니다.
- (e) 명령어 최대 처리 시간 : 장비로부터 데이터를 한번 주고 받는데 걸리는 시간입니다.
- (f) 통신 Time Out : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (g) 통신 주기 : 데이터를 요청하는 주기입니다.

(2) Station 추가

Solar 인버터 설정

Device 정보

Device 이름 : INV
Device 설명 : 태양광 인버터

라인 이중화 장비 이중화

통신포트#1 : COM1
통신포트#2 :
통신포트#3 :
통신포트#4 :
통신속도 : 9600
패리티비트 : No Parity
데이터비트 : 8
스톱비트 : 1

통신방식
 RS232
 RS422/485

흐름제어
 NONE
 RTS 제어

Delay Time (Before) : 0 msec
Delay Time (After) : 15 msec

명령어 최대 처리시간 : 200 msec [100 ~ 2000]
통신 Time Out : 500 msec [100 ~ 2000]
통신 주기 : 1000 msec [100 ~ 1800000]

Station 정보

Station 이름 :
Station 설명 :
CPU Type : 03K
Station 번호 : 1
구조태그 유형 : 사용안함

신규 저장 닫기

- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) CPU Type : 인버터의 종류를 선택합니다.
- (d) Station 번호 : 인버터의 국번을 입력합니다.
- (e) 구조태그 유형 : 사용할 구조태그 타입을 선택합니다.

(3) 입출력 주소

(a) 3Kw

Solar Inverter Address Table (3kw)				
구분	데이터 종류	ADDRESS	태그종류	비고
Solar Fault Address (2Bytes)	태양전지 과전압 (Solar Cell OV fault)	DI000	Digital	0:normal, 1: fault
	태양전지 과전압 제한초과 (Software Fault) (Solar Cell OV limit fault)	DI001	Digital	0:normal, 1: fault
	태양전지 저전압 (Solar Cell UV fault)	DI002	Digital	0:normal, 1: fault
	태양전지 저전압 제한 초과 (Software Fault) (Solar Cell UV limit fault)	DI003	Digital	0:normal, 1: fault
Inverter Operation & Fault Address (2Bytes)	인버터 과전류 (Inverter over current fault)	DI004	Digital	0:normal, 1: fault
	인버터 과전류 시간초과 (Inverter over current overtime fault)	DI005	Digital	0:normal, 1: fault
	한전-인버터 위상 이상 (Inverter-Line async fault)	DI006	Digital	0:normal, 1: fault
	인버터 정지 (Inverter stop)	DI007	Digital	1:Stop
	인버터 과온 (Inverter over temperature fault)	DI008	Digital	0:normal, 1: fault
	인버터 동작 (Inverter run)	DI009	Digital	1:Run
	인버터 퓨즈 단선	DI015	Digital	0:normal, 1: fault
	인버터 MC 이상	DI016	Digital	0:normal, 1: fault
Line Fault Address (2Bytes)	한전 과전압 (Line over voltage fault)	DI010	Digital	0:normal, 1: fault
	한전 저전압 (Line under voltage fault)	DI011	Digital	0:normal, 1: fault
	한전 정전 (Line failure fault)	DI012	Digital	0:normal, 1: fault
	한전 과주파수 (Line over frequency fault)	DI013	Digital	0:normal, 1: fault
	한전 저주파수 (Line under frequency fault)	DI014	Digital	0:normal, 1: fault
Solar Cell voltage, current (4Bytes)	태양전지 전압 (Solar volt rms address)	AI000	Analog	Scale 1:1
	태양전지 전류 (Solar current rms address)	AI001	Analog	Scale 10:1
Inverter voltage, current, frequency (16Bytes)	인버터 전압 (Inverter voltage rms)	AI002	Analog	Scale 1:1
	인버터 전류 (Inverter current rms)	AI003	Analog	Scale 10:1
	인버터 주파수 (Inverter frequency)	AI004	Analog	Scale 10:1
Utility Line voltage, current, frequency (16Bytes)	한전 전압 (Line voltage rms)	AI005	Analog	Scale 1:1
	한전 전류 (Line current rms)	AI006	Analog	Scale 10:1
	한전 주파수 (Line frequency)	AI007	Analog	Scale 10:1
Solar & Inverter power (16Bytes)	태양 전지 현재 전력 (kW) (Solar kw address)	AI008	Analog	Scale 1000:1
	인버터 현재 전력 (kVA)	AI009	Analog	Scale 1000:1
	인버터 역률 (Inverter pf address)	AI010	Analog	Scale 10:1
	인버터 적산 전력량 (Inverter kWh)	AI011	Analog	Scale 1000:1
	인버터 최대출력 전력	AI012	Analog	Scale 1000:1
	인버터 하루 발전량	AI013	Analog	Scale 10:1

(a) 10, 20, 30, 40, 50kw

Solar Inverter Address Table (10, 20, 30, 40, 50kw)				
구분	데이터 종류	ADDRESS	태그종류	비고
Solar Inverter operation fault address (2Bytes)	태양전지 과전압 제한 초과 (software Fault) (Solar Cell OV limit fault)	DI000	Digital	0: fault, 1:normal
	태양전지 저전압 제한 초과 (software Fault) (Solar Cell UV limit fault)	DI001	Digital	0: fault, 1:normal
	한전 전압 R상 전압 이상 (Line R phase fault)	DI002	Digital	0: fault, 1:normal
	한전 전압 S상 전압 이상 (Line S phase fault)	DI003	Digital	0: fault, 1:normal
	한전 전압 T상 전압 이상 (Line T phase fault)	DI004	Digital	0: fault, 1:normal
	한전 전압 이상(정전) (Utility line failure)	DI005	Digital	0: fault, 1:normal
Inverter fault address (2Bytes)	태양전지 저전압 (hardware fault) (Solar Cell under voltage fault)	DI006	Digital	0: fault, 1:normal
	인버터 과전류 (Inverter over current fault)	DI007	Digital	0: fault, 1:normal
	태양전지 과전압 (hardware fault) (Solar Cell over voltage fault)	DI008	Digital	0: fault, 1:normal
Inverter state address (2Bytes)	인버터 MC on, off 상태 (Inverter contactor state)	DI009	Digital	0:on, 1:off
	인버터 입력단 fuse 상태 (Inverter fuse state)	DI010	Digital	0:normal, 1: fault
	인버터 과온 (Inverter over temperature)	DI011	Digital	0:normal, 1: fault
Inverter operation fault address (2Bytes)	인버터 MC 이상 (Inverter MC fault)	DI012	Digital	0: fault, 1:normal
	인버터 출력 전압 이상 (Inverter output voltage fault)	DI013	Digital	0: fault, 1:normal
	인버터 주파수 이상 (Inverter frequency fault)	DI014	Digital	0: fault, 1:normal
	한전-인버터 위상 이상 (Inverter-Line async fault)	DI015	Digital	0: fault, 1:normal
	한전 R,S,T 상 역상 (Line phase sequency fault)	DI016	Digital	0: fault, 1:normal
	한전 저전압 이상 (Line under voltage fault)	DI017	Digital	0: fault, 1:normal
	한전 과전압 이상 (Line over voltage fault)	DI018	Digital	0: fault, 1:normal
	한전 주파수 이상 (Line-frequency fault)	DI019	Digital	0: fault, 1:normal
	인버터 과전류 시간 초과 (Inverter over current overtime fault)	DI020	Digital	0: fault, 1:normal
	한전 과 주파수 이상 (Line over frequency fault)	DI021	Digital	0: fault, 1:normal
	Solar Cell voltage, current (4Bytes)	태양전지 전압 (Solar volt rms address)	AI000	Analog
태양전지 전류 (Solar current rms address)		AI001	Analog	Scale 1:1

부록1 I/O 드라이버 설정방법

구분	데이터 종류	ADDRESS	태그종류	비고
Inverter voltage, current, frequency (16Bytes)	인버터 U상 선간전압 (Inverter voltage rms u)	AI002	Analog	Scale 1:1
	인버터 V상 선간전압 (Inverter voltage rms v)	AI003	Analog	Scale 1:1
	인버터 W상 선간전압 (Inverter voltage rms w)	AI004	Analog	Scale 1:1
	인버터 U상 상전류 (Inverter current rms u)	AI005	Analog	Scale 1:1
	인버터 V상 상전류 (Inverter current rms v)	AI006	Analog	Scale 1:1
	인버터 W상 상전류 (Inverter current rms w)	AI007	Analog	Scale 1:1
	인버터 주파수 (Inverter frequency u,v,w)	AI008	Analog	Scale 10:1
Utility Line voltage, current, frequency (16Bytes)	한전 U상 선간전압 (Line voltage rms u)	AI009	Analog	Scale 1:1
	한전 V상 선간전압 (Line voltage rms v)	AI010	Analog	Scale 1:1
	한전 W상 선간전압 (Line voltage rms w)	AI011	Analog	Scale 1:1
	한전 U상 상전류 (Line current rms u)	AI012	Analog	Scale 1:1
	한전 V상 상전류 (Line current rms v)	AI013	Analog	Scale 1:1
	한전 W상 상전류 (Line current rms w)	AI014	Analog	Scale 1:1
	한전 주파수 (Line frequency u,v,w)	AI015	Analog	Scale 10:1
Solar & Inverter power kW, kWh (16Bytes)	태양 전지 현재 전력 (kW) (Solar kW address)	AI016	Analog	Scale 10:1
	인버터 적산 전력량 (Inverter kWh)	AI017	Analog	Scale 1:1
	인버터 현재 전력 (kVA) (Inverter kVA address)	AI018	Analog	Scale 10:1
	태양 전지 적산 전력량 (Solar kWh)	AI019	Analog	사용금지(프로토콜 변경으로 하루 발전량 값으로 대체 되었음)
	인버터 역률 (Inverter pf address)	AI020	Analog	Scale 10:1
	최대 출력 전력 kW	AI025	Analog	Scale 10:1
	하루 발전량 kWh	AI026	Analog	Scale 1:1
Solar Cell state (16Bytes)	경사면 일사량계 (T-radiation)	AI021	Analog	
	수평면 일사량계 (H-radiation)	AI022	Analog	
	외부 주위 온도 (Out temperture)	AI023	Analog	
	모듈 온도 (Module temperture)	AI024	Analog	

(a) DY_5010 및 DY_ADC

Solar Inverter Address Table (DY-5010 / DY-ADC)				
구분	데이터 종류	ADDRESS	태그종류	비고
아날로그입력#1		A1000	Analog	
아날로그입력#2		A1001	Analog	
아날로그입력#3		A1002	Analog	
.
아날로그입력#94		A1093	Analog	
아날로그입력#95		A1094	Analog	
아날로그입력#96		A1095	Analog	

어드레스 설정 : A1xxx

주의 : 여기서 xxx는 000-095 의 수이다, 반드시 3자리를 입력하여야 한다. 예) A1000 , A1001, A1095

예제1) DY-5010 사용, 21개의 channel을 사용할 때

A1000 : Channel 1 의 전압
 A1001 : Channel 1 의 전류
 A1002 : Channel 2 의 전압
 A1003 : Channel 2 의 전류
 A1004 : Channel 3 의 전압
 A1005 : Channel 3 의 전류

A1038 : Channel 20 의 전압
 A1039 : Channel 20 의 전류
 A1040 : Channel 21 의 전압
 A1041 : Channel 21 의 전류
 A1042 : 전체 출력 전압
 A1043 : 전체 출력 전류

예제2) DY-5010 사용, 32개의 channel을 사용할 때

A1000 : Channel 1 의 전압
 A1001 : Channel 1 의 전류
 A1002 : Channel 2 의 전압
 A1003 : Channel 2 의 전류
 A1004 : Channel 3 의 전압
 A1005 : Channel 3 의 전류

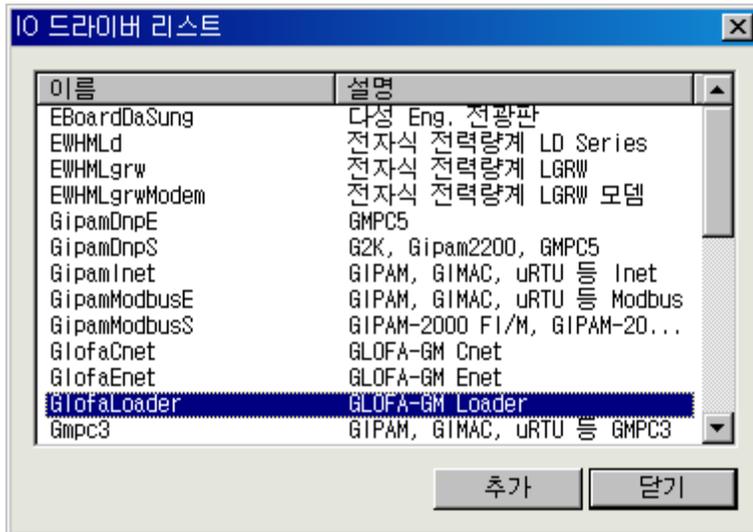
A1062 : Channel 31 의 전압
 A1063 : Channel 31 의 전류
 A1064 : Channel 32 의 전압
 A1065 : Channel 32 의 전류
 A1066 : 전체 출력 전압
 A1067 : 전체 출력 전류

예제3) DY-ADC 사용, 4개의 channel을 사용할 때

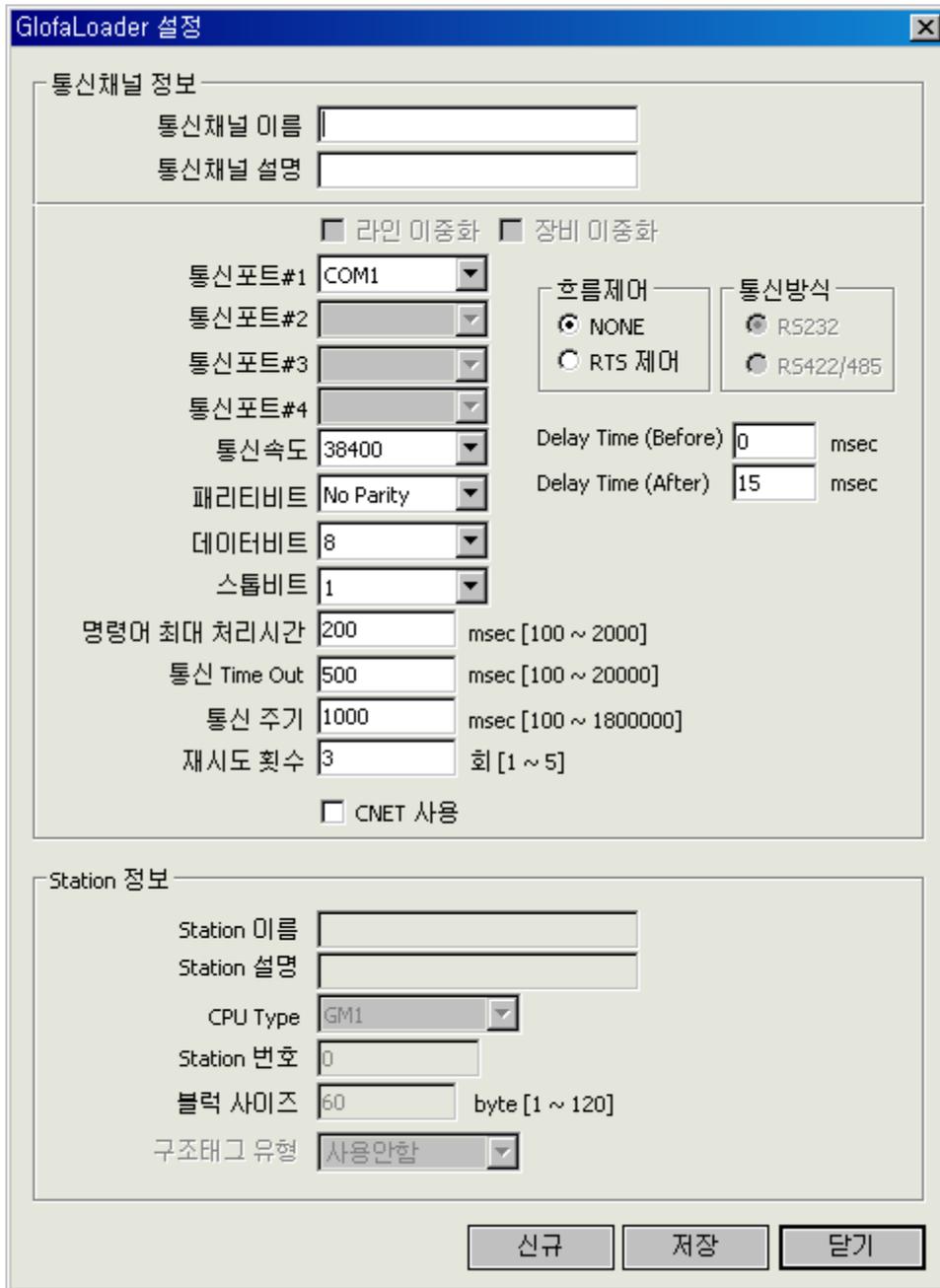
A1000 : Channel 1 의 값
 A1001 : Channel 2 의 값
 A1002 : Channel 3 의 값
 A1003 : Channel 4 의 값

14. GLOFA Loader 드라이버

(1) 통신채널 추가



- I/O 드라이버 리스트에서 “GLOFALoader” 을 선택한 후 “추가” 버튼을 누릅니다.



- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트#1 : 통신포트를 선택합니다.
- (d) 통신속도 : 통신 속도를 선택합니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (f) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 명령어 최대 처리시간 : 한번 데이터를 요청한뒤 응답을 받을때까지 걸리는 시간을 말합니다.
- (i) 통신 Time Out : 장비에게 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (j) 통신 주기 : 데이터를 요청하는 주기입니다.
- (k) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.

(2) Station 추가

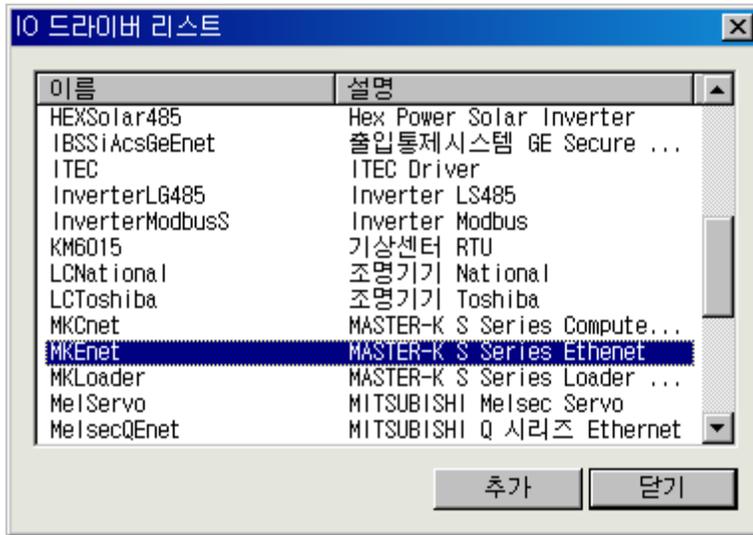
- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) CPU Type : CPU 타입을 선택합니다.
- (d) Station 번호 : Station 번호를 입력합니다.
- (e) 블록 사이즈 : PLC 통신 블록의 크기를 입력합니다.

(3) 입출력 주소

- I/O Address 형식
 - : %MXO, %MBO, %MNO, %MDO, %MLO
 - : %MBO.0, %MNO.0, %MDO.0, %MLO.0

15. MKEnet 드라이버

(1) 통신채널 추가



- I/O 드라이버 리스트에서 “MKEnet” 을 선택한 후 “추가” 버튼을 누릅니다.

The image shows the 'MKEnet' configuration dialog box. It is divided into two main sections: '통신채널 정보' (Communication Channel Information) and 'Station 정보' (Station Information).
 In the '통신채널 정보' section, there are input fields for '통신채널 이름' (Communication Channel Name) and '통신채널 설명' (Communication Channel Description). Below these are fields for '서버 IP Address #1' (127.0.0.1) and '서버 IP Address #2' (0.0.0.0). There are also fields for '최대 응답시간' (200 msec), '통신 Time Out' (500 msec), '통신 주기' (500 msec), and '재시도 횟수' (3 times).
 In the 'Station 정보' section, there are fields for 'Station 이름' (Station Name) and 'Station 설명' (Station Description). Below these is a dropdown for 'PLC CPU 종류' (K10005) and two checkboxes for '라인 이중화' (Line Redundancy) and '장비 이중화' (Equipment Redundancy). There are four fields for 'PLC IP Address #1-1', '#1-2', '#2-1', and '#2-2', all set to 0.0.0.0. There is a dropdown for '통신 방식' (TCP), a field for '포트 번호' (2004), and a field for '블럭 크기' (1400 byte).
 At the bottom, there are three buttons: '신규' (New), '저장' (Save), and '닫기' (Close).

- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- (d) 서버 IP Address #2 : 라인 이중화를 사용할 경우 사용하게 될 두번째 IP Address를 입력합니다.
- (e) 최대 응답 시간 : 장비로부터 데이터를 한번 주고 받는데 걸리는 시간입니다.
- (f) 통신 Time Out : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (g) 통신 주기 : 데이터를 요청하는 주기입니다.
- (h) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.

(2) Station 추가

The image shows the 'MKEnet' configuration dialog box. It is divided into two main sections: '통신채널 정보' (Communication Channel Information) and 'Station 정보' (Station Information).
 In the '통신채널 정보' section, there are fields for '통신채널 이름' (MKEnet), '통신채널 설명', '서버 IP Address #1' (127.0.0.1), '서버 IP Address #2' (0.0.0.0), '최대 응답시간' (200 msec), '통신 Time Out' (500 msec), '통신 주기' (500 msec), and '재시도 횟수' (3 회).
 In the 'Station 정보' section, there are fields for 'Station 이름', 'Station 설명', 'PLC CPU 종류' (K10005), '라인 이중화' and '장비 이중화' checkboxes, 'PLC IP Address #1-1' (0.0.0.0), 'PLC IP Address #1-2' (0.0.0.0), 'PLC IP Address #2-1' (0.0.0.0), 'PLC IP Address #2-2' (0.0.0.0), '통신 방식' (TCP), '포트 번호' (2004), and '블록 크기' (1400 byte).
 At the bottom, there are three buttons: '신규', '저장', and '닫기'.

- (a) Station 이름 : Station 이름을 입력합니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (d) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (e) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다.
- (f) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
- (g) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.
- (h) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
- (i) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.
- (j) 통신 방식 : TCP와 UDP 중의 한가지를 선택합니다.
- (k) 포트 번호 : 통신 방식 선택에 따라 자동으로 입력됩니다.
- (l) 블록 크기 : PLC 통신 블록의 크기를 입력합니다.

(3) 입출력 주소

(a) 지원하는 디바이스

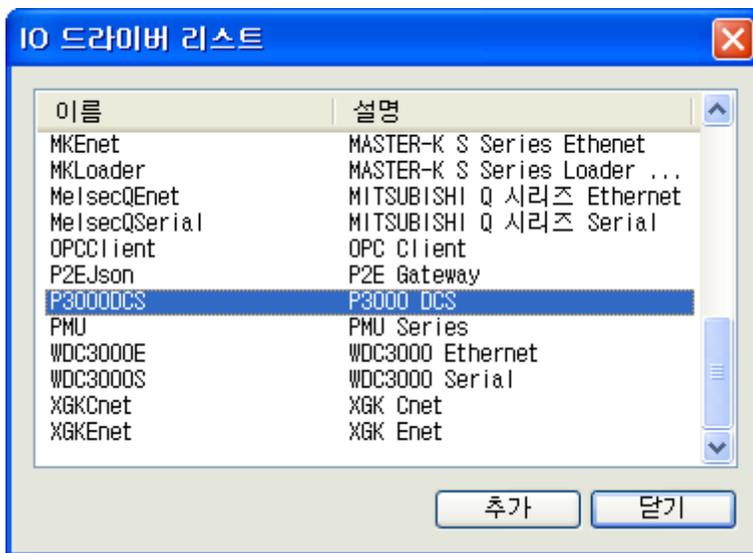
: P, M, K, F, L, D (T, C 디바이스는 지원하지 않습니다.)

(b) I/O Address 형식

: P000(Analog 태그: 디바이스 뒤의 숫자는 word 영역을 표시합니다.), P00F(Digital 태그 : 맨뒤의 한자리가 bit 영역을 나타냅니다.) 나머지 앞의 영역은 word 영역을 표시합니다.

16. P3000DCS 드라이버

(1) 통신채널 추가



- I/O 드라이버 리스트에서 “P3000 DCS” 를 선택한 후 “추가” 버튼을 클릭하면 아래 화면이 표시됩니다..

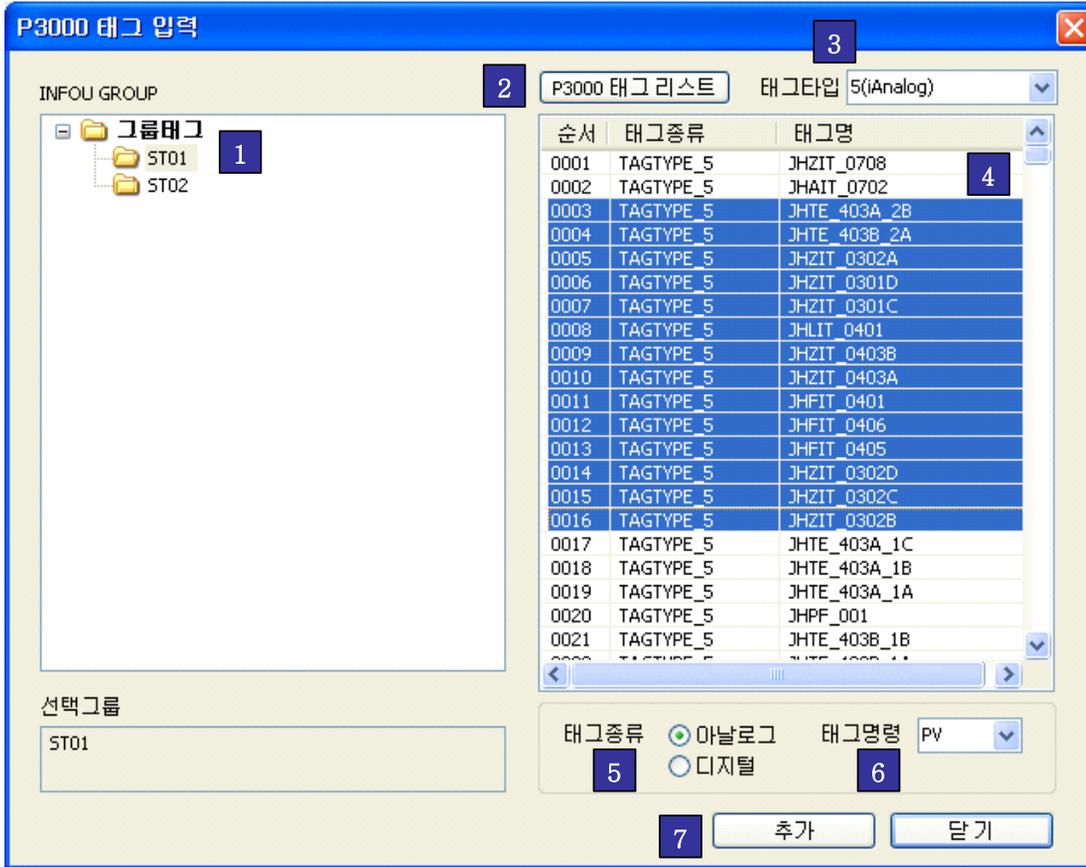
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다. 중복 될 수 없습니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 서버IP Address#1 설정 : InfoU 가 설치된 컴퓨터 IP 어드레스를 입력합니다.
- (d) 서버IP Address#2 설정 : 라인 이중화의 경우 보조 IP 어드레스로써 미입력 가능합니다.
- (e) 최대응답시간: Default 설정을 유지합니다.
- (f) 통신 TimeOut: Default 설정을 유지합니다.
- (g) 통신주기: 통신 부하율에 맞게 적절히 설정합니다.

(2) Station 추가

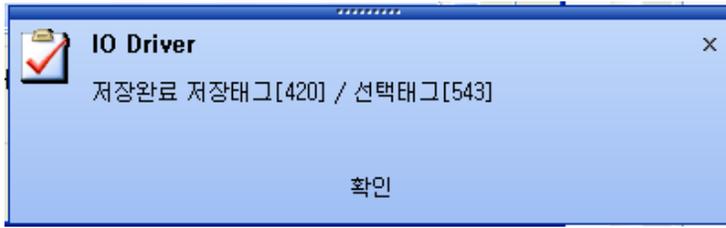
- (a) Station 이름 : Station 이름을 입력합니다. 중복 될 수 없습니다.
- (b) Station 설명 : Station 설명을 입력합니다.
- (c) P3000 Connect ID : 1~19 사이 선택할 수 있습니다.
- (d) DCS 종류 : P3000 UNIX, P3000 AT, P3000 NT중 설정가능하나 현재 이에 따른 별도 동작은 없고 다만 정보만 저장합니다. 향후 AT전용 API 추가 시 구분하여 동작 가능합니다.
- (e) DCS IP Address#1-1 : DCS가 설치된 IP 어드레스를 설정합니다.
- (f) DCS IP Address#1-2 : 라인이중화로 구성된 경우 DCS의 보조 IP 어드레스를 설정합니다.
- (g) DCS IP Address#2-1 : 이중화 DCS가 설치된 경우 Slave서버의 IP 어드레스를 설정합니다.
- (h) DCS IP Address#2-2 : 이중화 DCS가 라인이중화로 구성된 경우 Slave서버의 보조 IP 어드레스를 설정합니다.

(3) P3000 태그입력

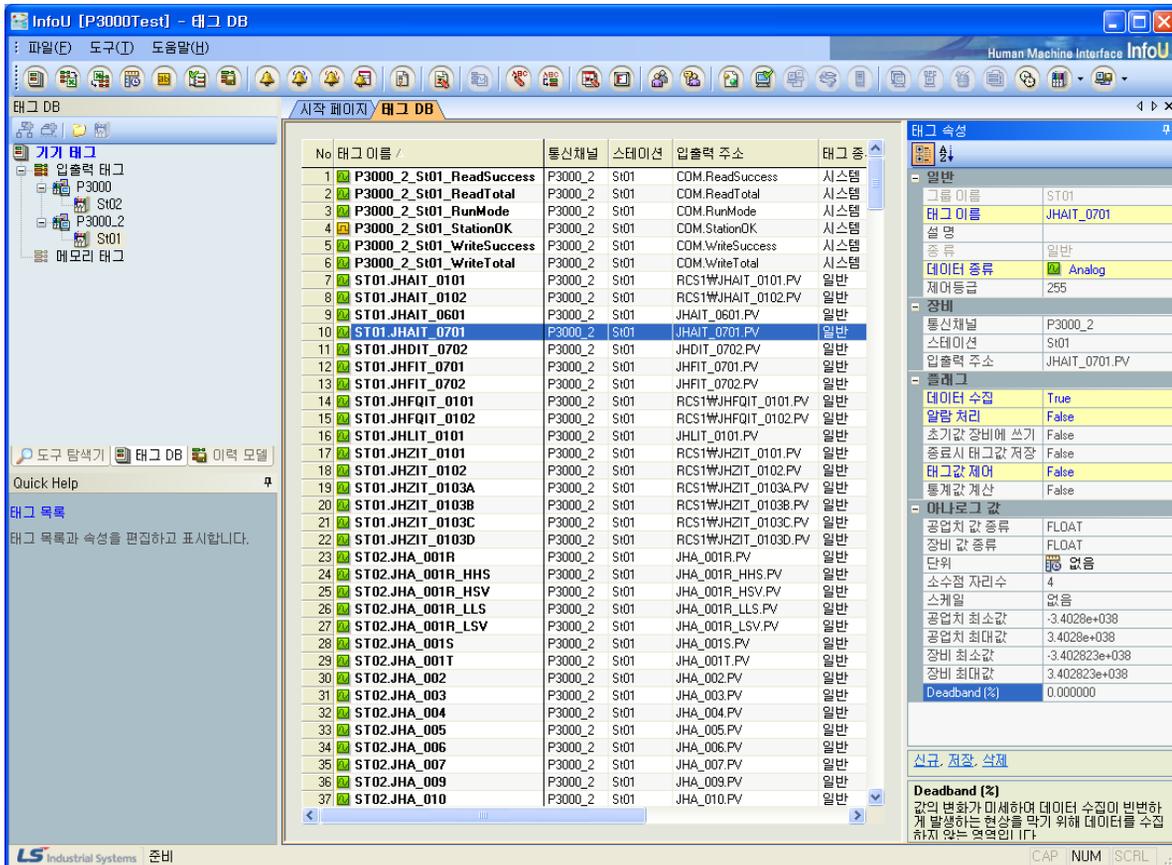
- 통신채널과 스테이션을 추가하고 상기 화면의 [P3000입력] 버튼을 클릭하여 태그 입력을 수행할 수 있습니다.



- 초기에 상기 화면이 표시될 때는 ③[태그타입], ⑤[태그종류], ⑥[태그명령], ⑦[태그추가] 입력 버튼들이 비활성화 되어있습니다.
- ②[P3000 태그리스트] 버튼을 클릭하여 DCS와 연결을 시도하여 성공하면 추가 시 필요한 선택버튼들이 활성화 됩니다.
- ③[태그타입] 콤보버튼을 클릭하여 태그 타입을 선택하면 해당 태그타입에 해당하는 P3000의 태그리스트 들이 ④ [DCS태그 리스트]에 표시되고 만약, 해당 태그타입에 대한 P3000 태그들이 존재하지 않으면 리스트는 빈 상태로 표시됩니다.
- 원하는 태그를 원하는 위치에 입력하기 위해 먼저 ①[INFOU GROUP리스트]에서 그룹 선택하여 아래 선택그룹 란에 InfoU 태그그룹이 표시되어야 합니다.(필수)
- 만약, InfoU에서 그룹을 생성하지 않았다면, GROUP은 리스트에 보이지 않으며 [그룹태그]라는 제목만 보이게 되며 [그룹태그]를 클릭하면 Root라고 표시되고 Root 그룹에 태그들이 추가됩니다.
- 만약, InfoU에서 그룹을 생성하지 않았다면 GROUP은 리스트에 보이지 않으며 [그룹태그]라는 제목만 보이게 되며 [그룹태그]를 클릭하면 Root라고 표시되고 Root 그룹에 태그들이 추가됩니다.
- ④[DCS태그 리스트]에서 입력하고자 하는 태그들을 선택합니다. (멀티선택 가능)
- ⑤[태그종류]에서 아날로그 디지털여부를 선택합니다.
- ⑥[태그명령]태그명령 선택합니다. 지원하는 항목은 [PV, SV, SUM, CSP0]입니다.
- ⑦[태그추가]를 클릭하여 태그 입력 작업을 실행합니다.
- 실행결과는 아래와 같이 표시되며 입력시도 선택한 태그 개수 대비 저장성공한 개수가 표시됩니다.



(l) 태그 속성은 Default로 입력되며 별도의 편집이 필요한 경우는 InfoU 태그편집기에서 수정하거나 csv import로 일괄편집도 가능합니다.



- (m) 상기화면은 아날로그 태그를 입력한 경우 InfoU 태그속성입니다.
- (n) P3000 입력다이얼로그화면에서 태그종류를 "아날로그", 태그명령을 "PV"로 입력하고, InfoU그룹을 "ST01" 를 선택하고 추가하였다면 주요속성들은 다음과 같습니다.
 - 1) 설명: 입력되지 않습니다.
 - 2) 데이터종류: 아날로그 "FLOAT", 디지털 "BOOL"
 - 3) 수집: True
 - 4) 태그값제어: FALSE, 만약 제어를 원한다면 True 변경가능합니다.
 - 5) 알람처리: False

(4) 주의사항

- (a) P3000과 통신 방법
 - 1) P3000은 InfoU에서 통신채널 1개당 1개의 DCS만 통신이 가능합니다. (P3000 API 속성)
 - 2) 여러 개의 P3000을 붙이려면 통신채널을 추가 한다. (즉, 통신채널당 1개의 P3000)

(b) 태그입력 실패의 원인

- 1) 동일한 태그이름 이미 존재할 경우 P3000태그는 실제로 AAA.PV, AAA.SV 등이 별도로 존재할 수 있지만(리스트에는 모두 AAA로 표시됩니다.) InfoU의 태그이름은 동일하게 AAA로 처리되고 IP 어드레스에서 구분하여야 합니다. 이 경우 먼저 PV값을 일괄로 입력하고 나머지 태그명령에 대해서는 별도의 태그이름을 부여하여 신규로 입력하여야 합니다.
- 2) P3000의 태그 이름에 InfoU 태그에서 수용할 수 없는 금칙문자가 포함된 경우 [#\${}-{}] 기호 등의 특수 기호, 한자 Root에 입력하는 경우 태그이름이 첫 문자에 숫자가 입력될 수 없습니다.

(c) P3000 드라이버에서 태그입력이 불가능 한 경우

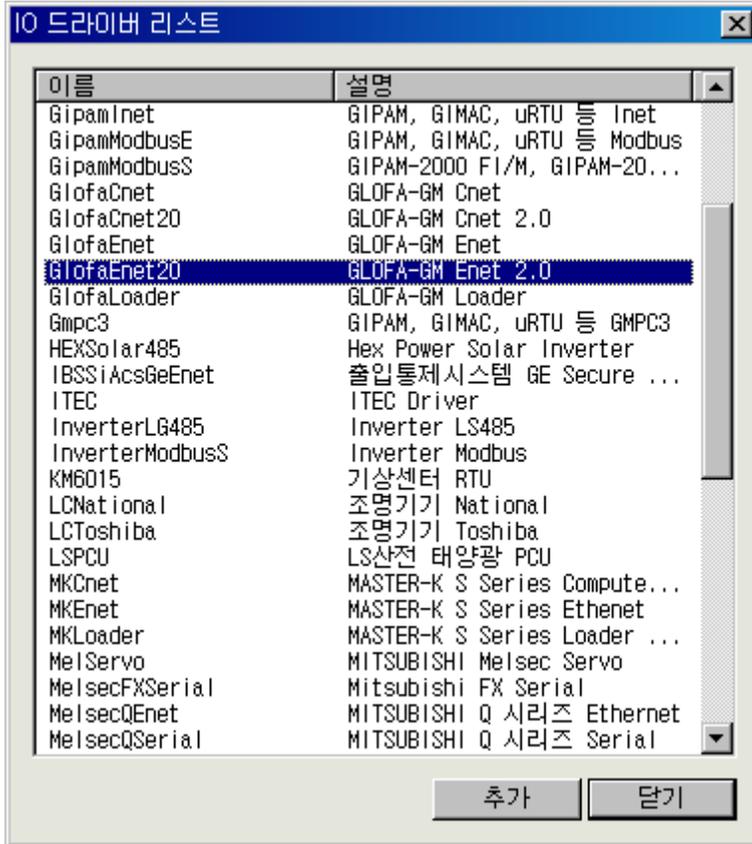
- 1) P3000 통신드라이버는 등록태그의 Address와 태그종류(아날로그, 디지털)를 구별하여 Index 테이블을 생성하여 통신하는데 Index 테이블 아날로그 및 디지털이 구별되며 최대 8개까지 생성할 수 있습니다. 그리고 한 개의 Index 테이블로 1600개의 P3000태그를 가져올 수 있습니다.
- 2) 즉 한 타입에 1600개 이상의 P3000태그가 입력되어 있다면 API에서 가져오지 못한 태그는 입력 할 수 없습니다.

(d) 기타 사용상의 제약 사항

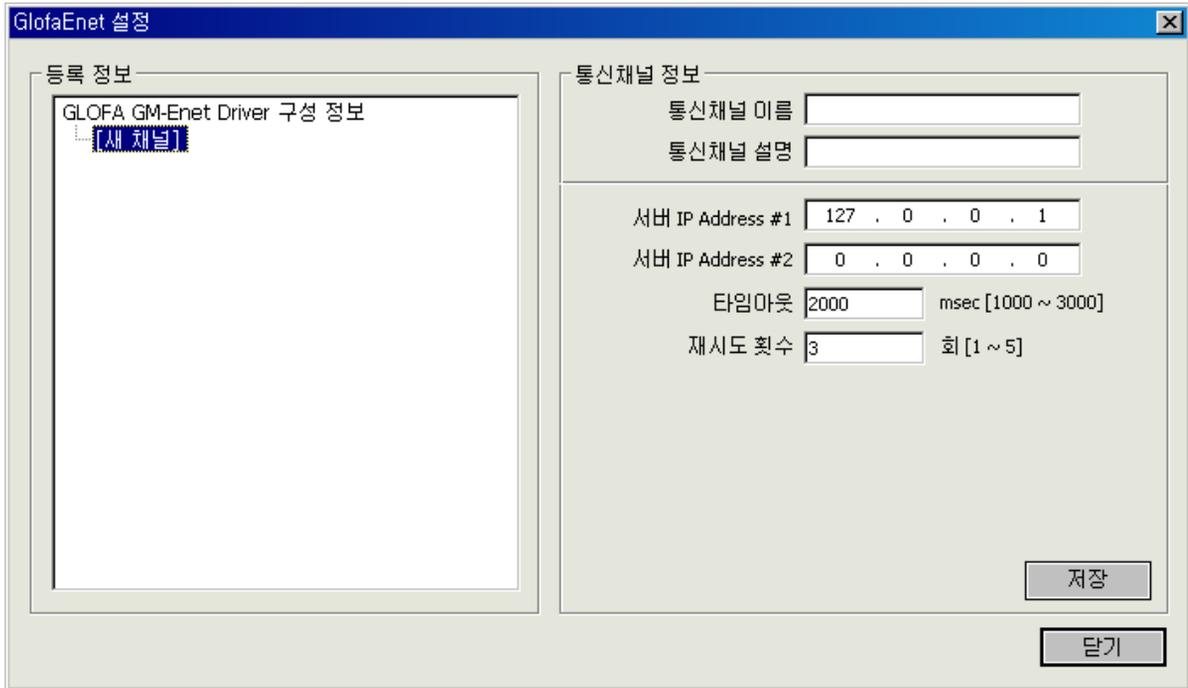
- 1) P3000AT 그룹이름 지원 불가: 그룹이 달라도 동일한 태그이름으로 표시되며 등록은 되지만 실제 통신은 되지 않습니다. 이 경우 수동으로 IO 어드레스 입력란에 그룹을 입력해 주어야 합니다.
예) 입출력 주소 : JHAIT_0601.PV → RCS01WJHAIT_0601.PV
- 2) P3000 태그 이름 및 그룹을 포함하여 20자가 넘는 경우 통신에러는 없지만 입력값은 '0' 을 취하게 됩니다. 즉, InfoU 태그의 IO 어드레스 입력란에 20자를 초과하면 통신상의 문제는 없지만 P3000에서는 없는 태그로 인식하여 '0' 값을 전송하므로 InfoU에서 보여지는 값은 항상 '0' 이 됩니다.

17. GlofaEnet20 드라이버

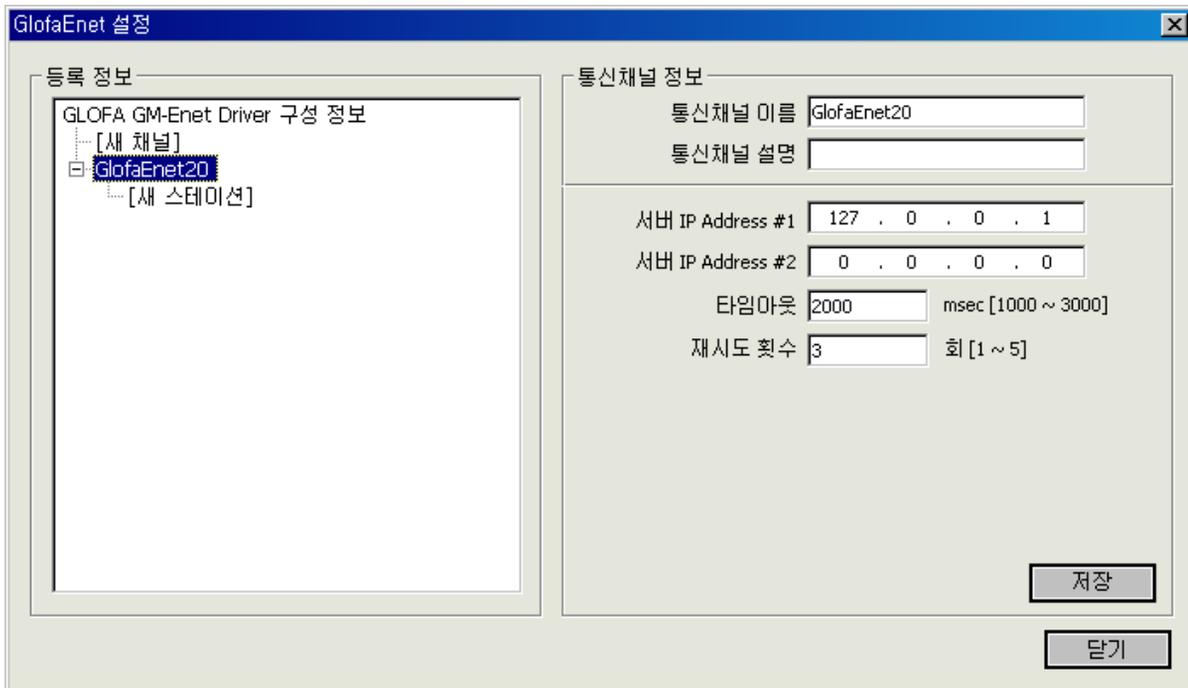
(1) 통신채널 추가



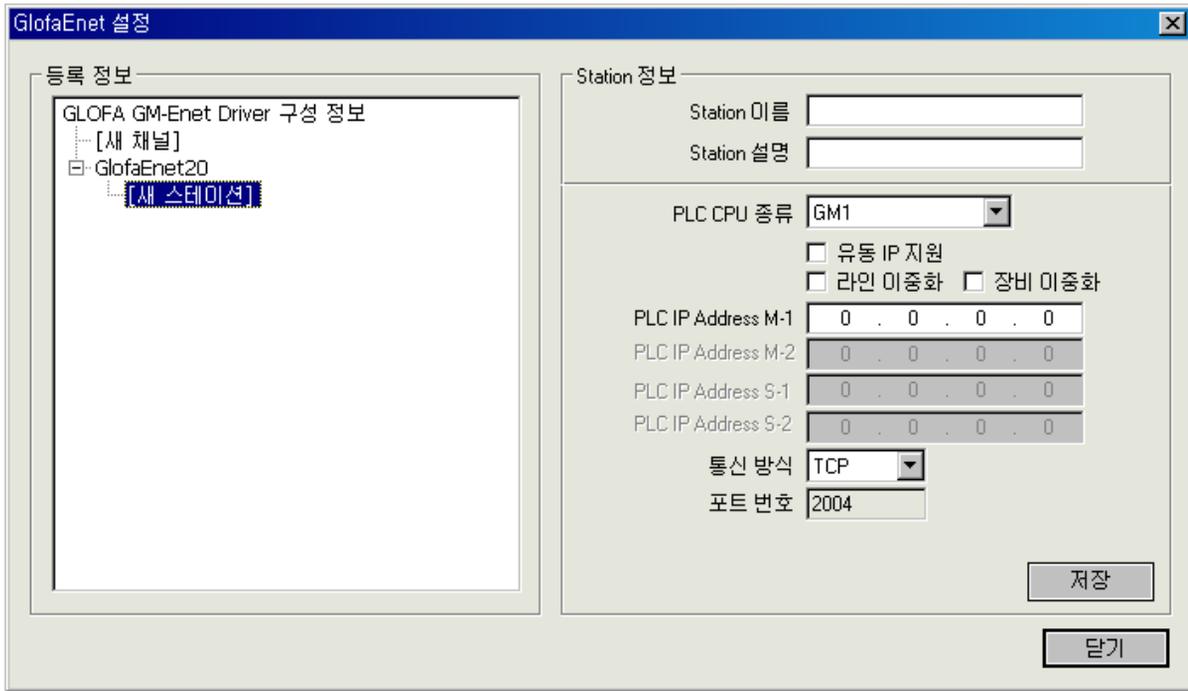
- IO 드라이버 리스트에서 “GlofaEnet20” 을 선택한 후 “추가” 버튼을 누릅니다.



- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- (d) 서버 IP Address #2 : 라인 이중화를 사용할 경우 사용하게 될 두번째 IP Address를 입력합니다.
- (e) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (f) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(2) Station 추가



- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다. 컴퓨터에 랜카드 2개를 설치하고, PLC에 Enet 통신 모듈을 2개 설치하여, 그림과 같이 네트워크를 분리하여 통신하고자 할 때 사용합니다. 네트워크 라인에 이상이 있을때를 대비하기 위한 이중화 옵션 입니다.

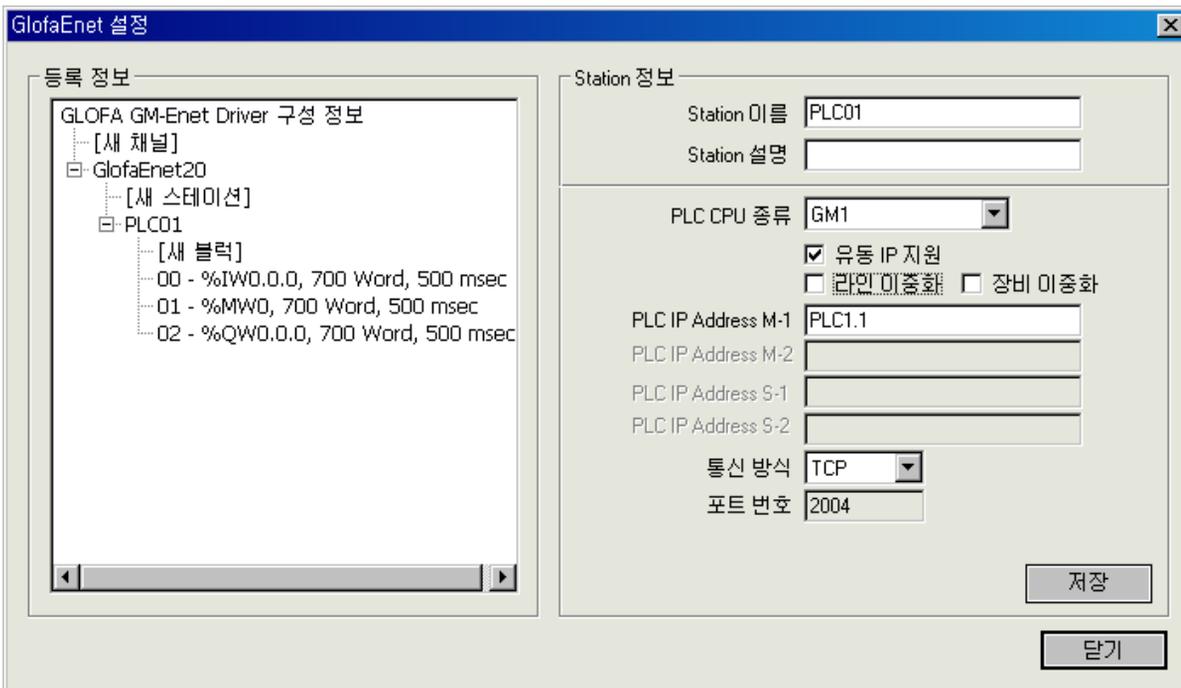
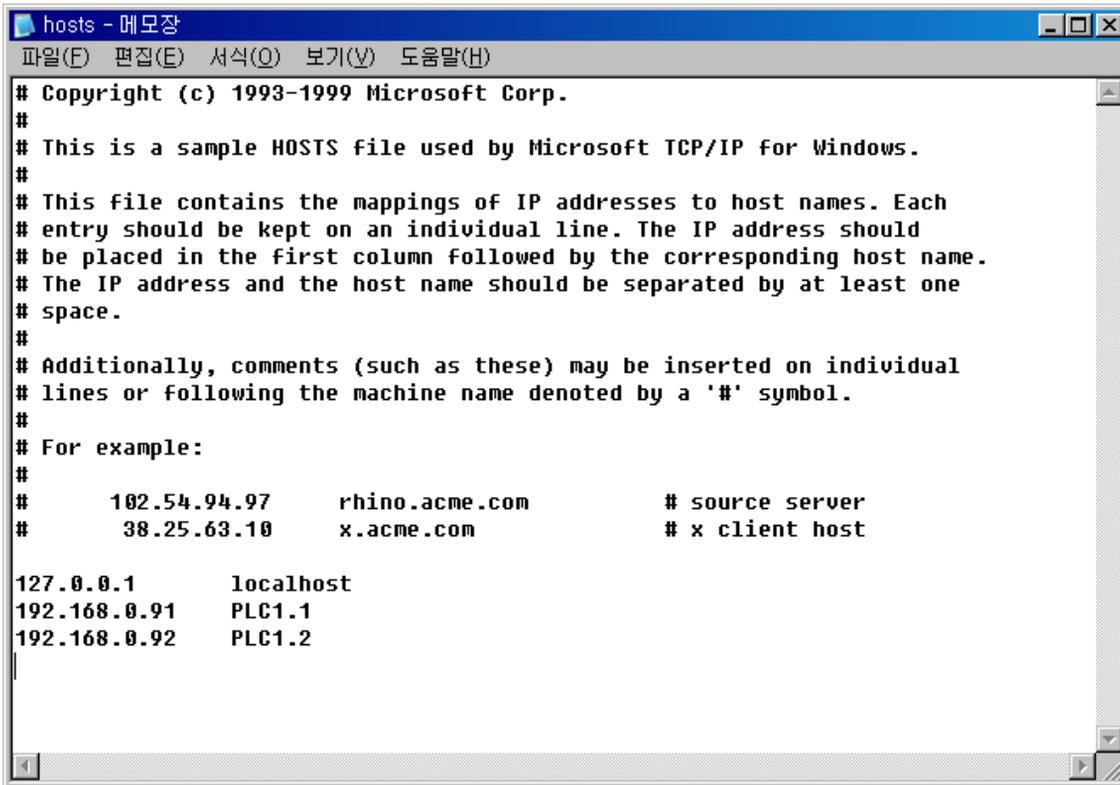


- (f) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다. 컴퓨터에 랜카드 1개를 설치하고, PLC에 Enet 통신 모듈

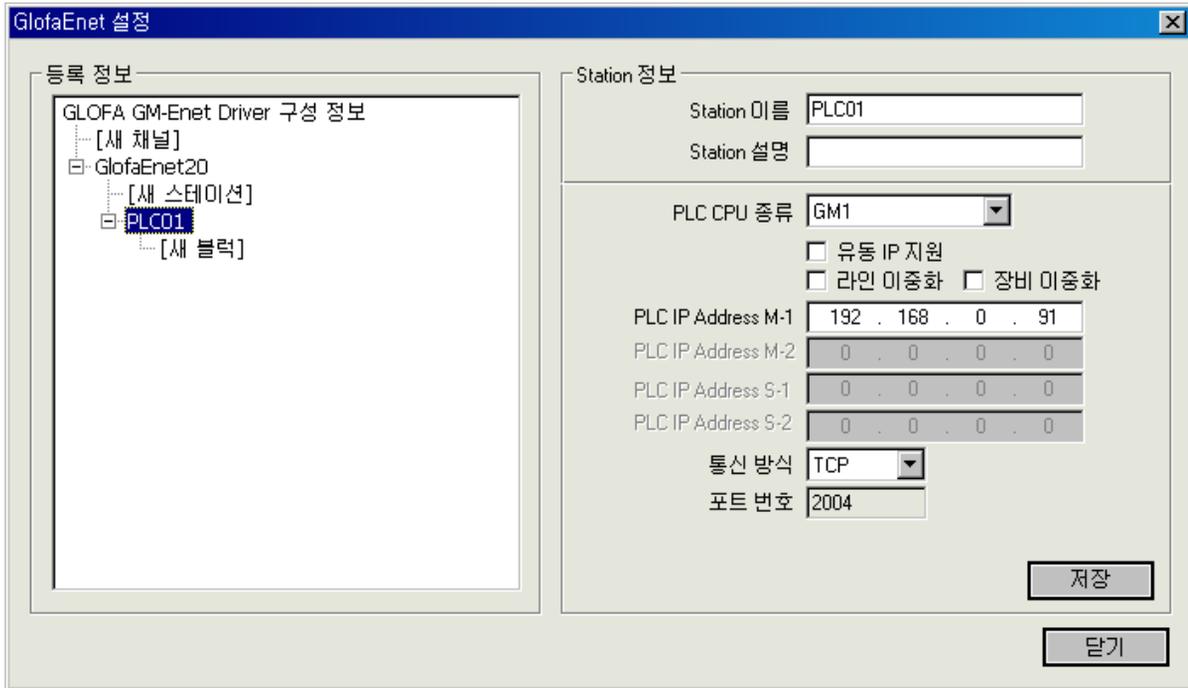
을 2개 설치하여, 그림과 같이 통신 모듈이 분리되어 있는 경우 사용합니다. PLC 통신 모듈에 이상이 있을때를 대비하기 위한 이중화 옵션 입니다.



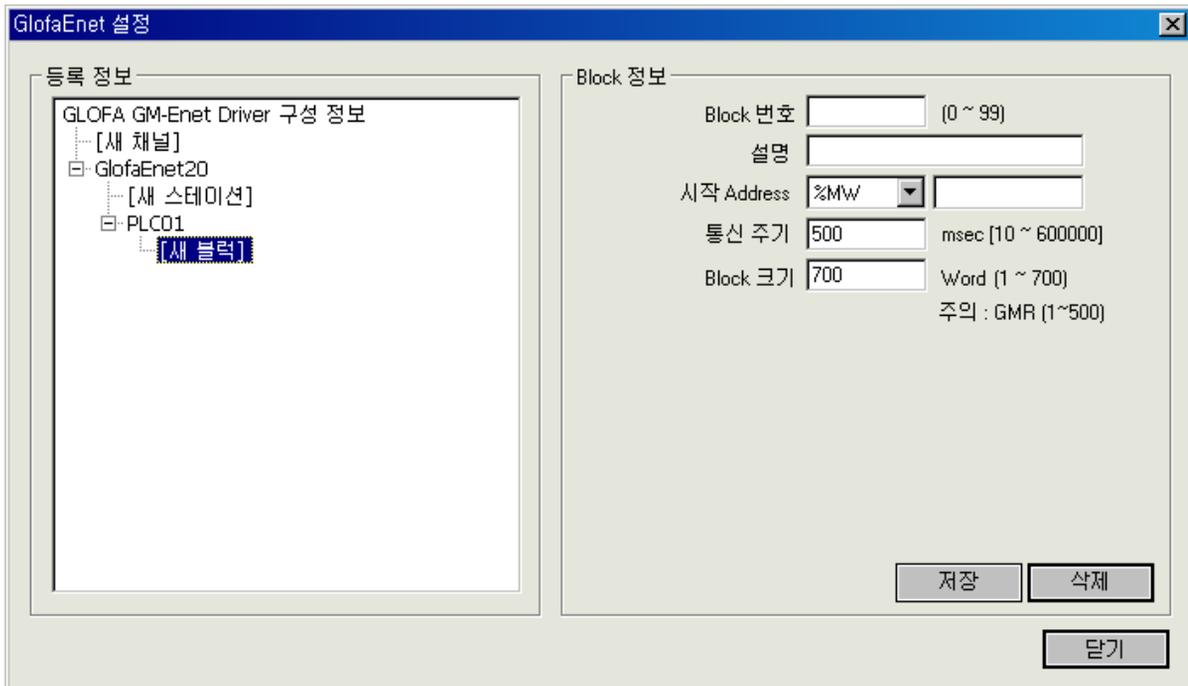
- (g) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
- (h) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.
- (i) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
- (j) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.
- (k) 통신 방식 : TCP와 UDP 중의 한가지를 선택합니다.
- (l) 포트 번호 : 통신 방식 선택에 따라 자동으로 입력됩니다.
- (m) 유동 IP 지원 : 유동 IP 를 사용하고자 할 때, 체크합니다. 이 옵션은, hosts 파일 또는 DNS 서버로 부터 IP 어드레스를 얻어와서 통신할때 사용합니다. 따라서 hosts 파일 또는 DNS 서버에 호스트 이름 또는 URL이 등록되어 있어야 합니다. hosts 파일을 이용하는 방식은 다음과 같습니다. hosts 파일은 C:\WINDOWS\system32\drivers\etc 위치에 있습니다. hosts 파일이 아래와 같이 저장되어 있을 때, 유동 IP 는 다음과 같이 설정합니다. (DNS 서버를 이용할 때도, InfoU 설정방식은 동일 합니다.)



(n) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



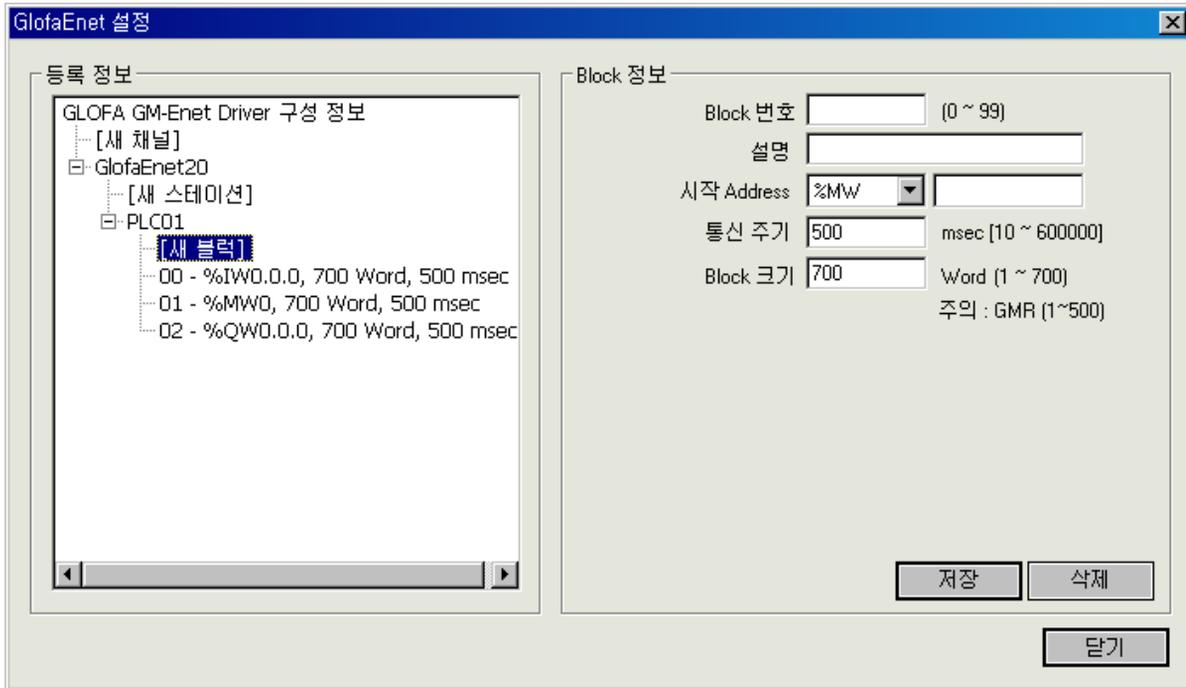
(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 3가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.
 - 올바른 예제: %MWO, %MM20, %IWO.0.0, %QW1.0.0
 - 잘못된 예제: %MWO.0.0, %IWO, %QWO

부록1 I/O 드라이버 설정방법

- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 Block의 Block 크기를 Word(2 byte) 단위로 입력합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

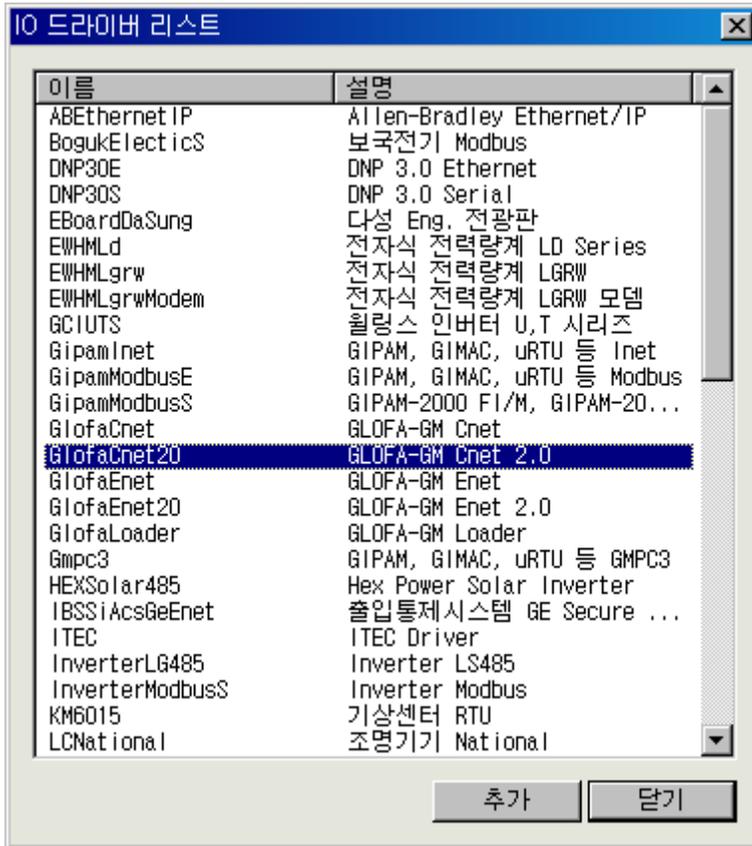
- 아날로그 : %MXO, %MBO, %MWO, %MDO, %MLO, %IWO.0.0, %QWO.0.0
- 디지털 : %MBO.0, %MWO.0, %MDO.0, %MLO.0, %IXO.0.0, %QXO.0.0

(b) 사용가능한 디바이스

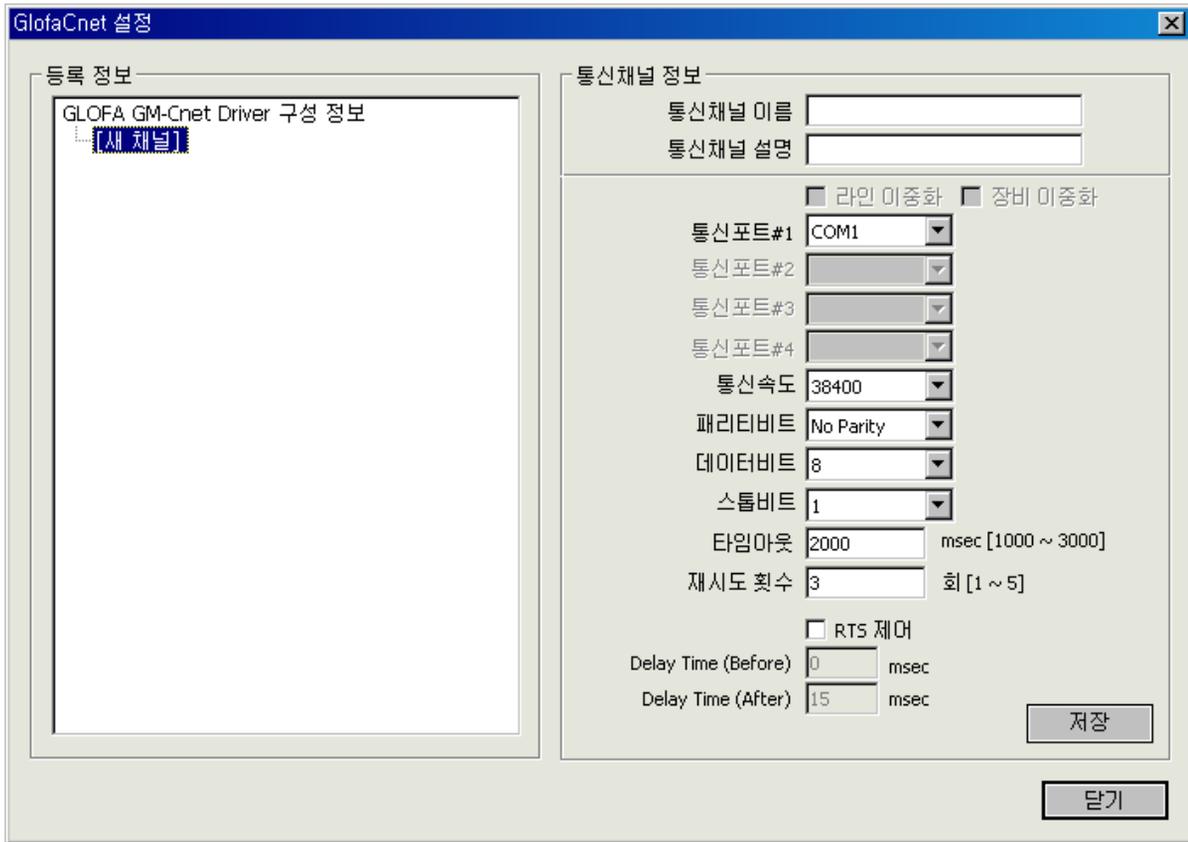
- I, M, Q

18. GlofaCnet20 드라이버

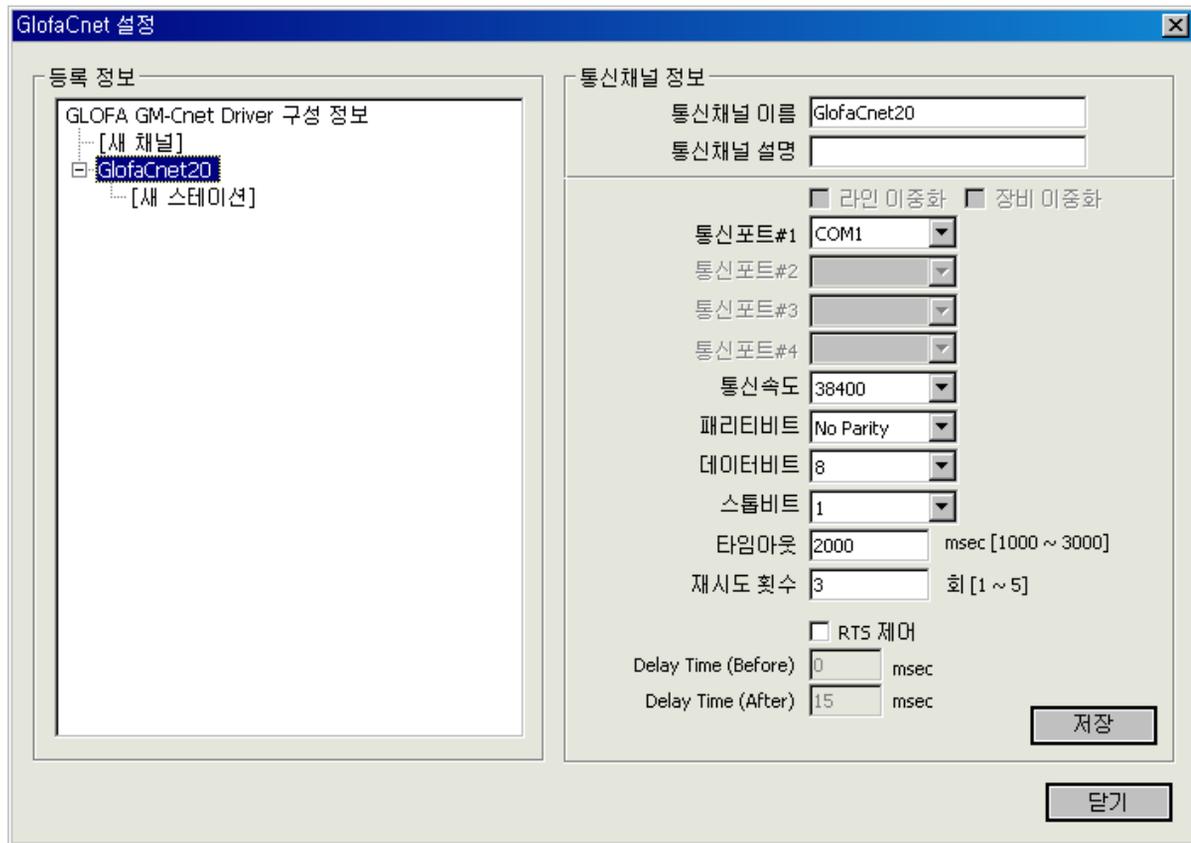
(1) 통신채널 추가



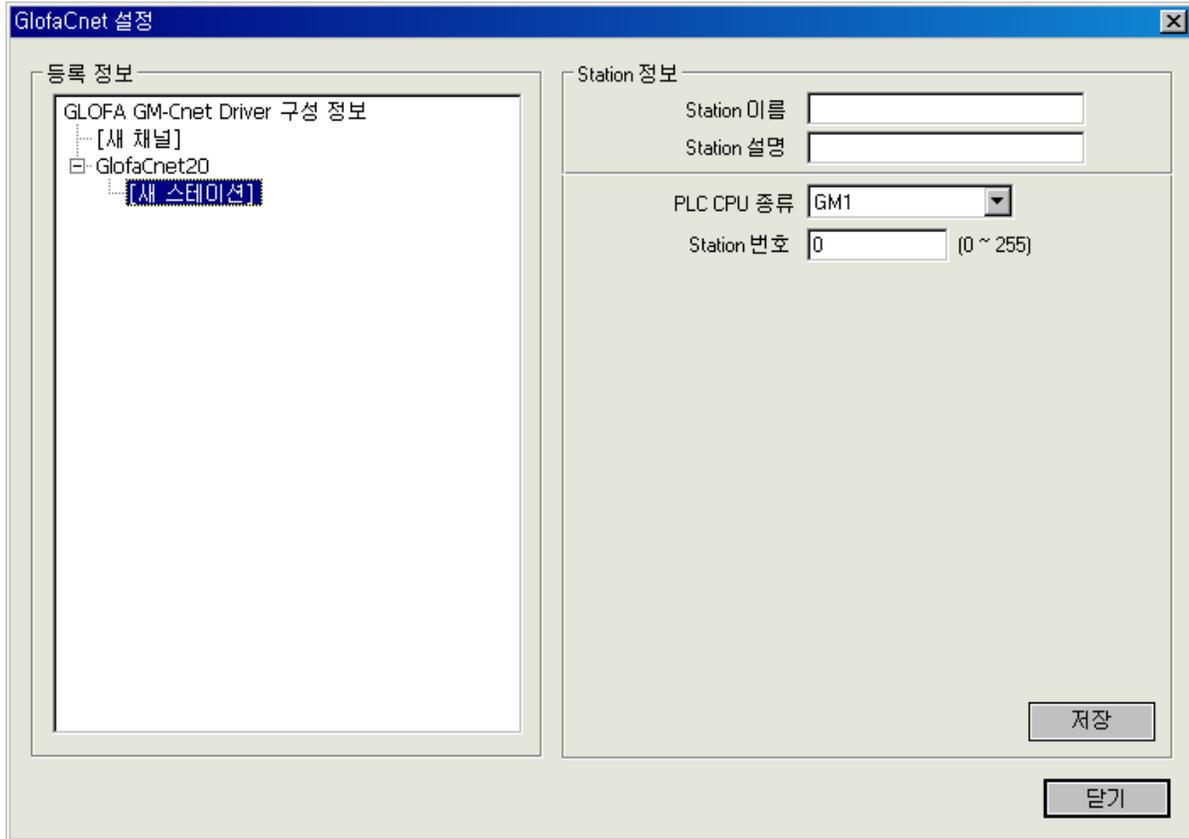
- IO 드라이버 리스트에서 “GlofaCnet20” 을 선택한 후 “추가” 버튼을 누릅니다.



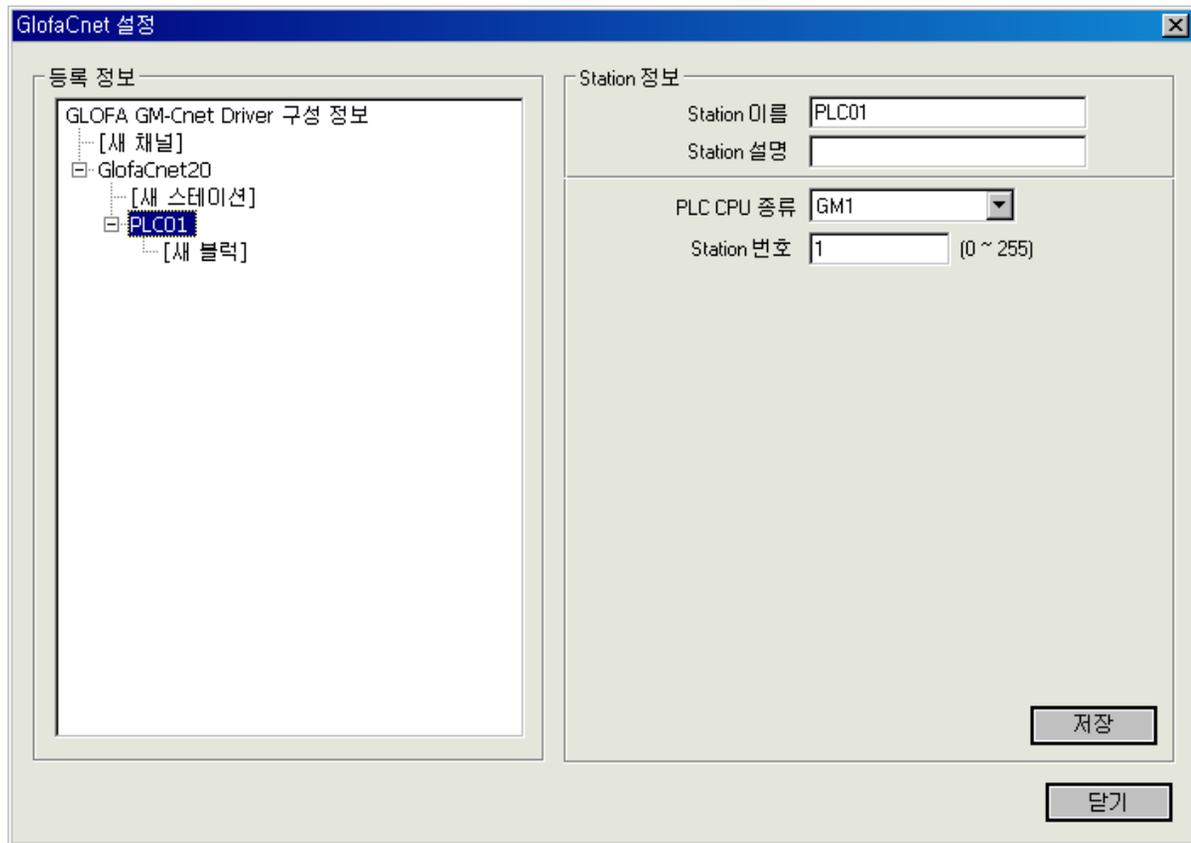
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (f) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) RTS 제어 : 485, 422 통신을 사용할 경우, 사용함으로 체크를 합니다.
- (k) Delay Time (Before) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 하기 직전의 Delay Time 입니다.
- (l) Delay Time (After) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 한 직후의 Delay Time 입니다.
- (m) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



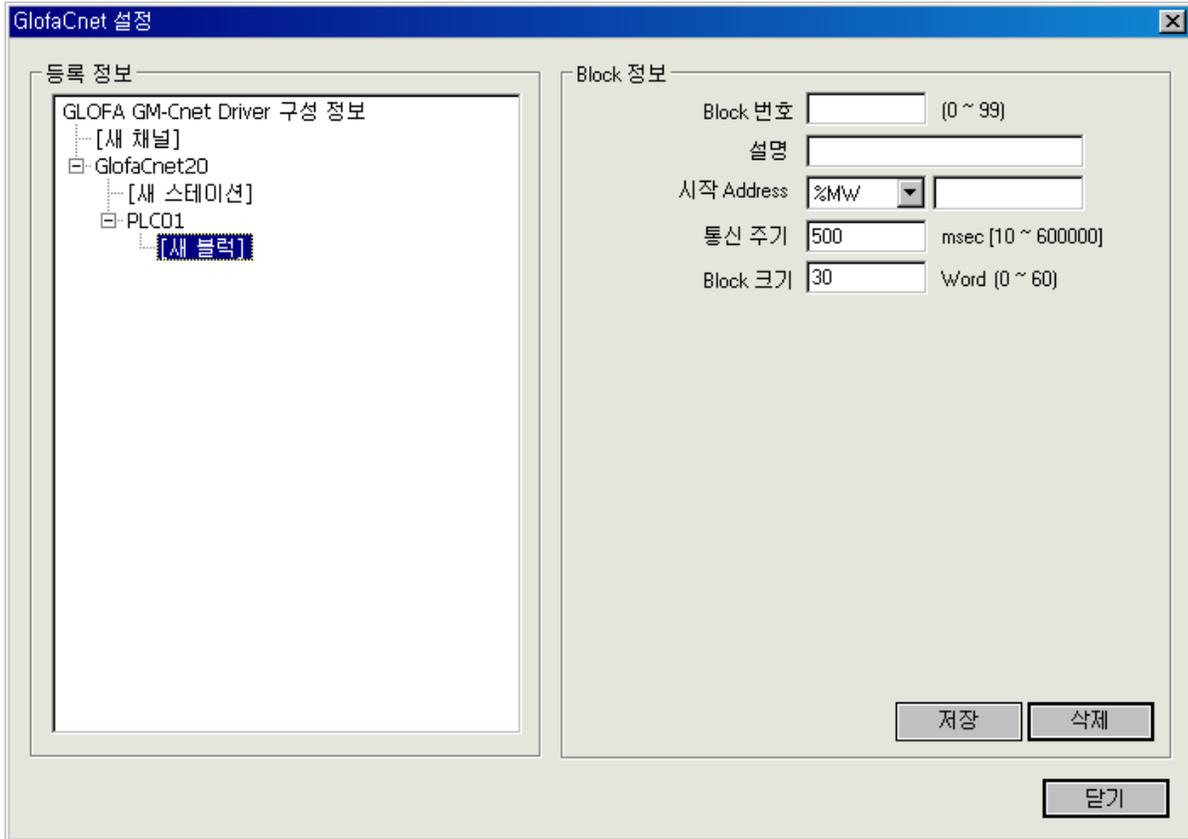
(2) Station 추가



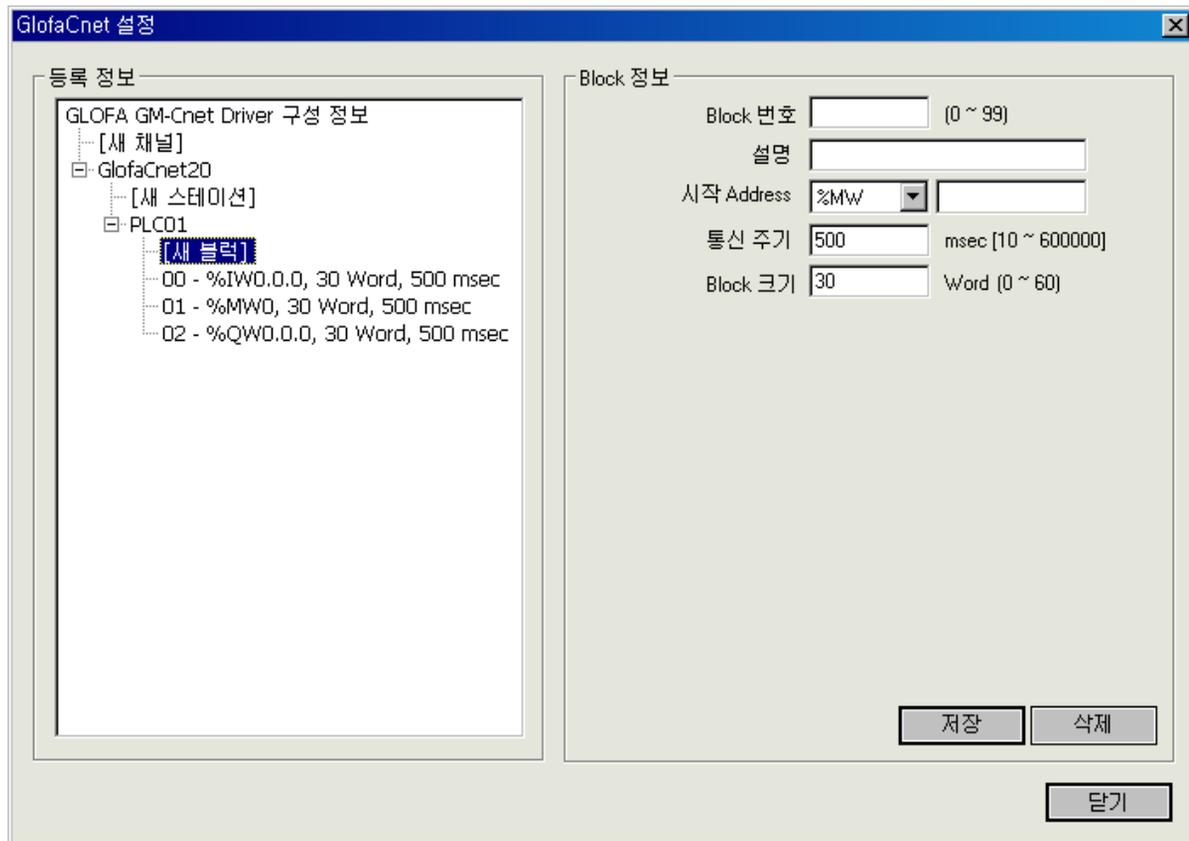
- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) Station 번호 : PLC Cnet 모듈의 국번을 입력합니다.
- (f) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 3가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.
 - 올바른 예제: %MW0, %MW20, %IWO.0.0, %QW1.0.0
 - 잘못된 예제: %MW0.0.0, %IWO, %QWO
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 Block의 Block 크기를 Word(2 byte) 단위로 입력합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

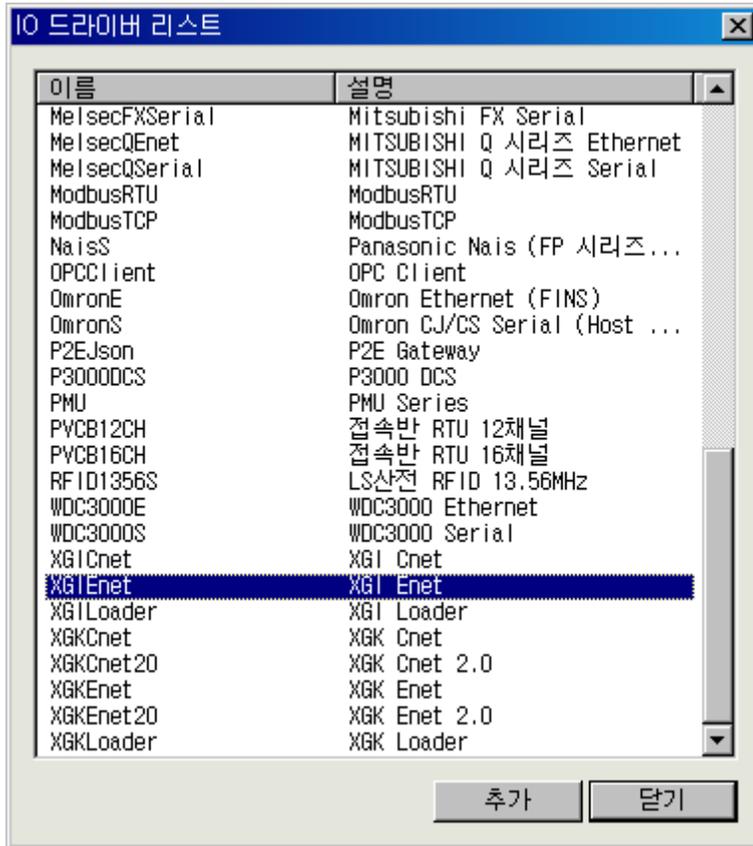
- . 아날로그 : %MX0, %MBO, %MNO, %MDO, %MLO, %IWO.0.0, %QWO.0.0
- . 디지털 : %MBO.0, %MNO.0, %MDO.0, %MLO.0, %IX0.0.0, %QX0.0.0

(b) 사용가능한 디바이스

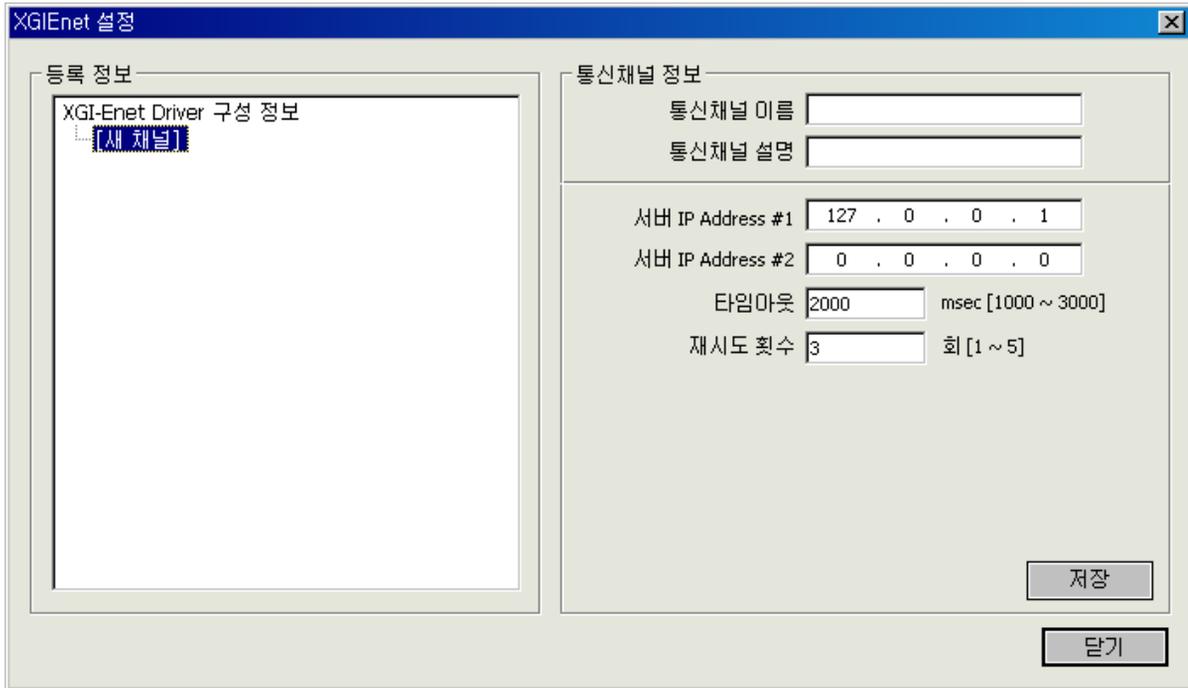
- . I, M, Q

19. XGEnet 드라이버

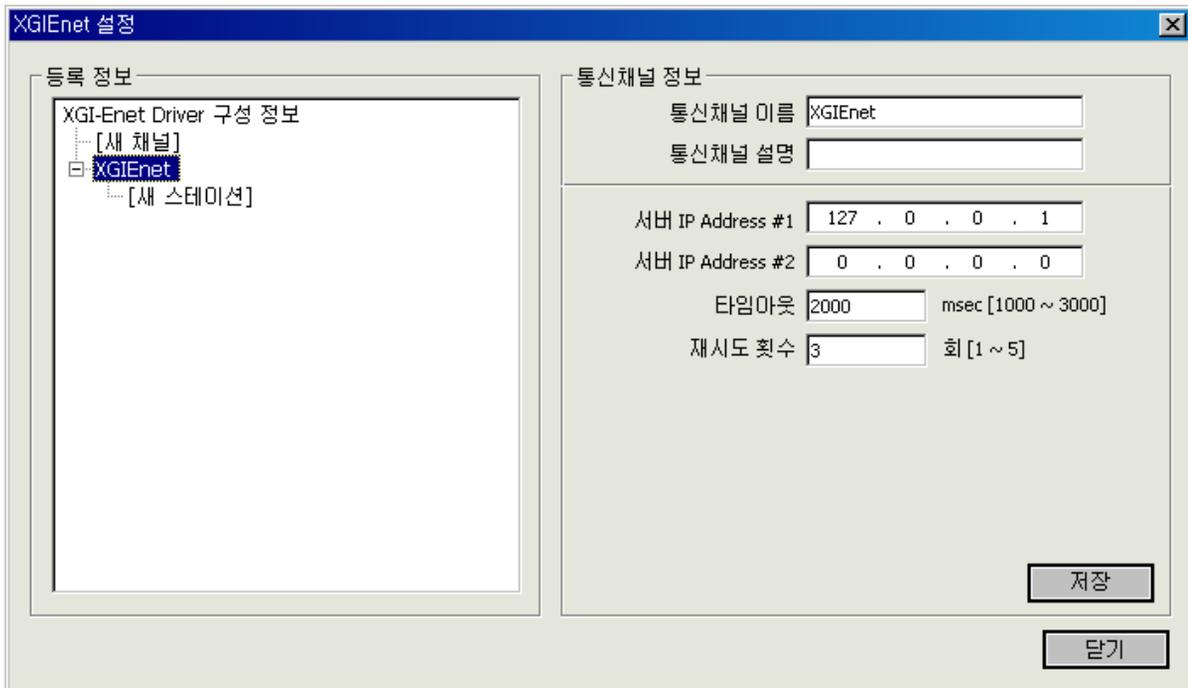
(1) 통신채널 추가



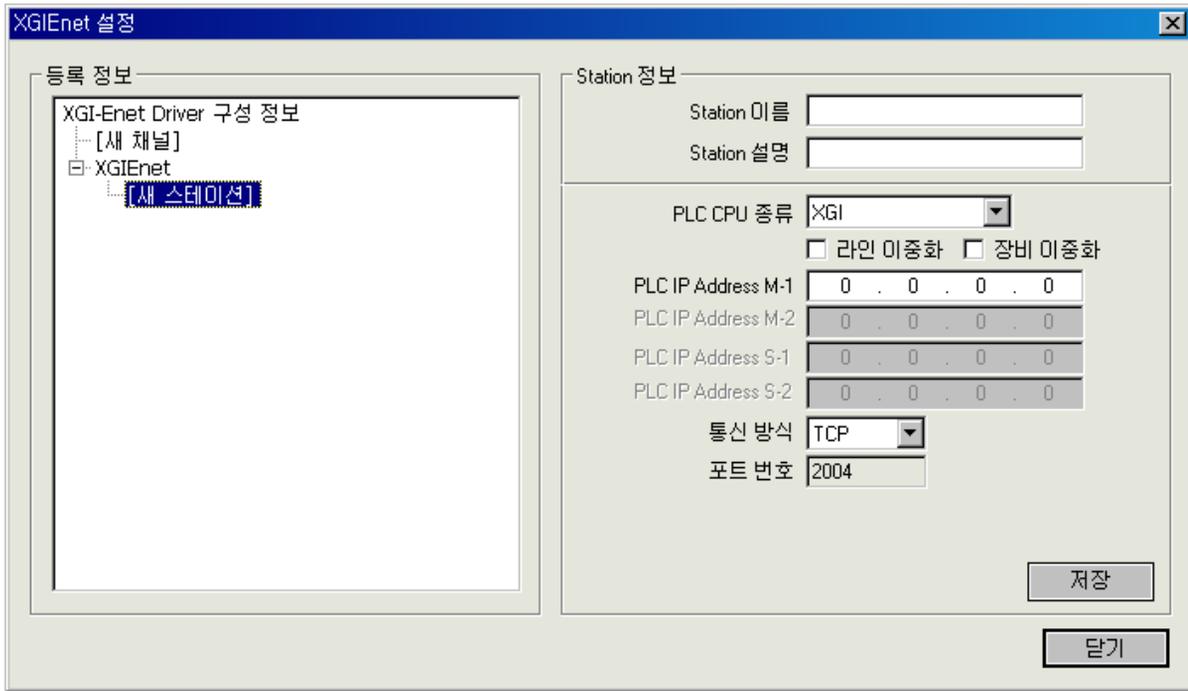
- IO 드라이버 리스트에서 “XGEnet” 을 선택한 후 “추가” 버튼을 누릅니다.



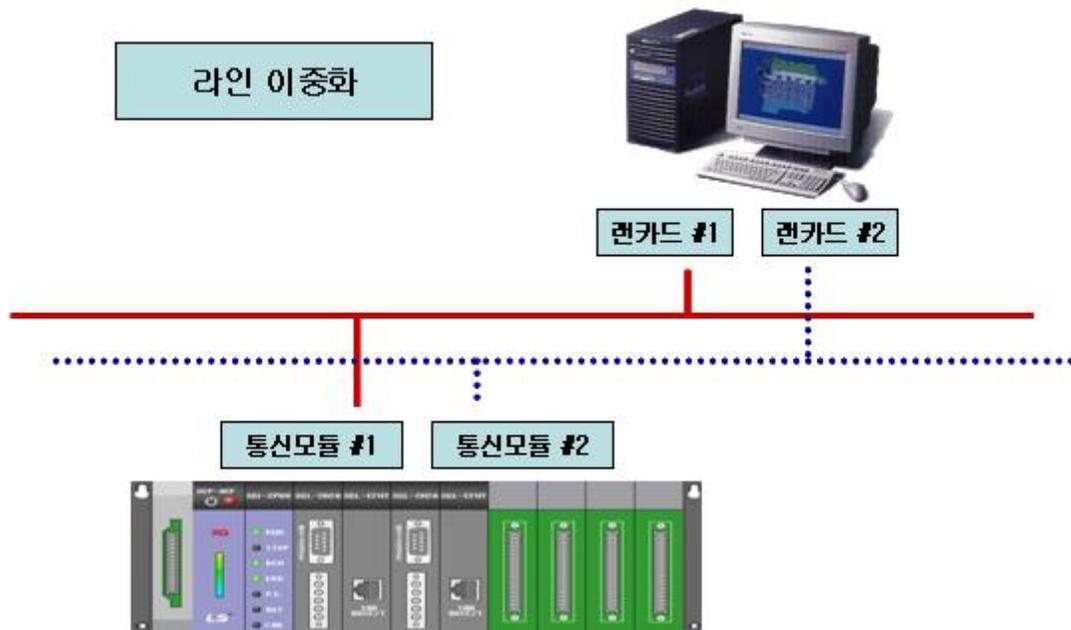
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- (d) 서버 IP Address #2 : 라인 이중화를 사용할 경우 사용하게 될 두번째 IP Address를 입력합니다.
- (e) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (f) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(2) Station 추가



- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다. 컴퓨터에 랜카드 2개를 설치하고, PLC에 Enet 통신 모듈을 2개 설치하여, 그림과 같이 네트워크를 분리하여 통신하고자 할 때 사용합니다. 네트워크 라인에 이상이 있을때를 대비하기 위한 이중화 옵션 입니다.

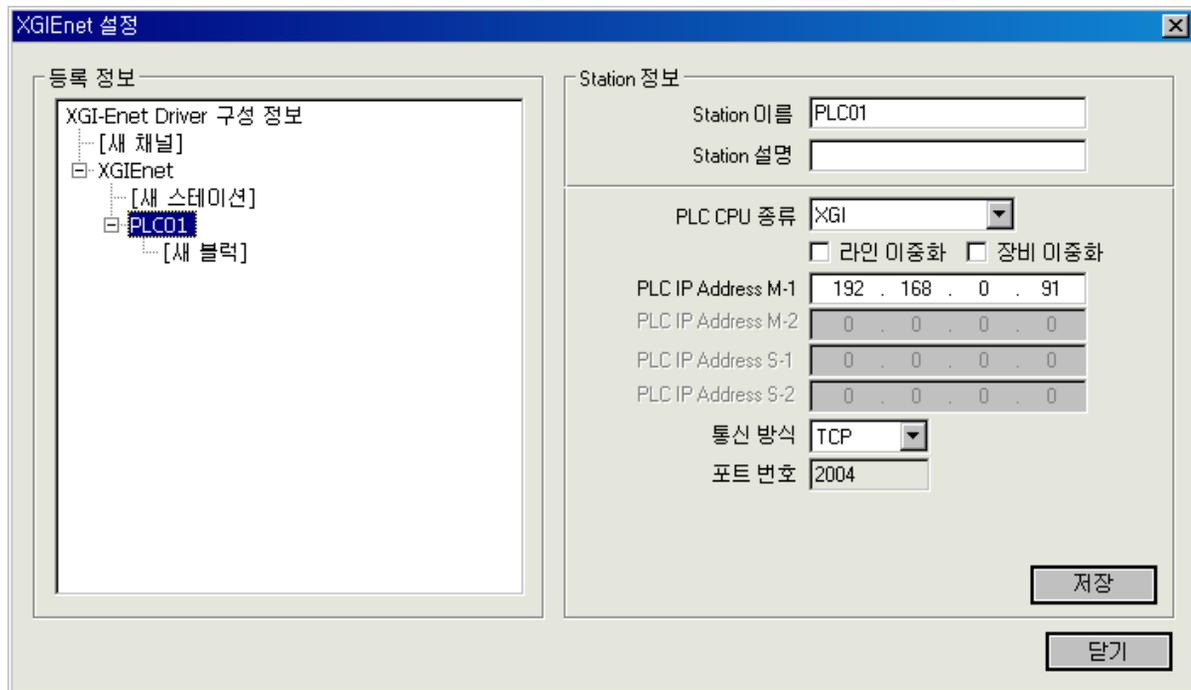


- (f) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다. 컴퓨터에 랜카드 1개를 설치하고, PLC에 Enet 통신 모듈

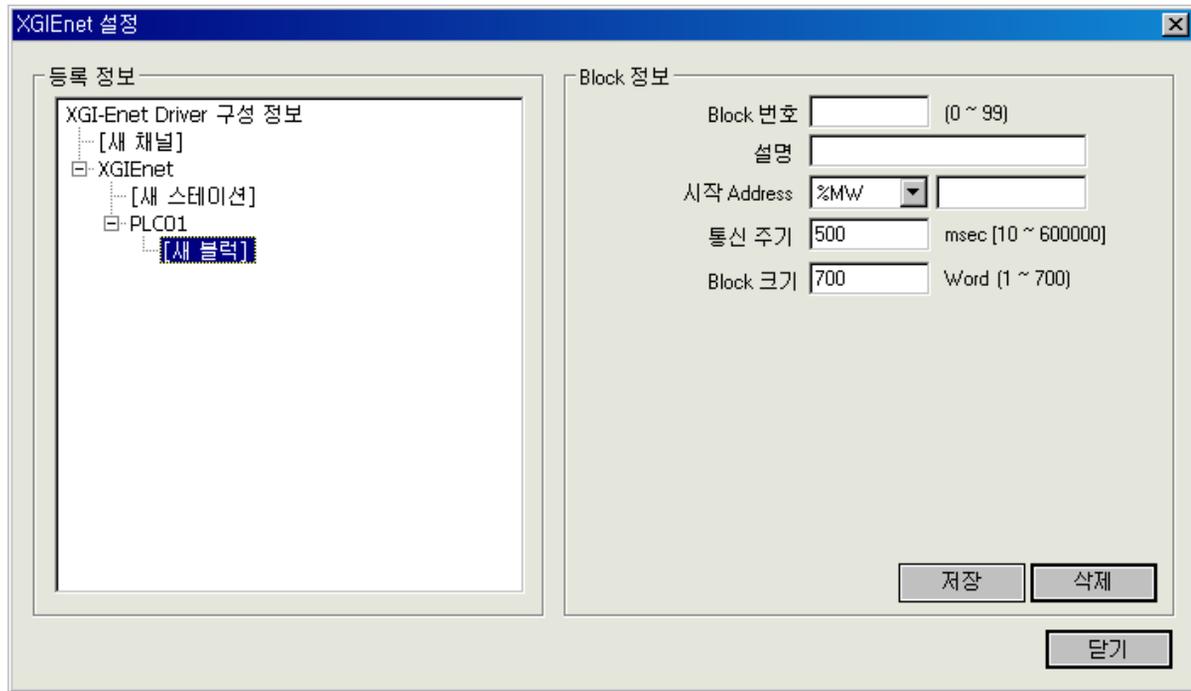
을 2개 설치하여, 그림과 같이 통신 모듈이 분리되어 있는 경우 사용합니다. PLC 통신 모듈에 이상이 있을때를 대비하기 위한 이중화 옵션 입니다.



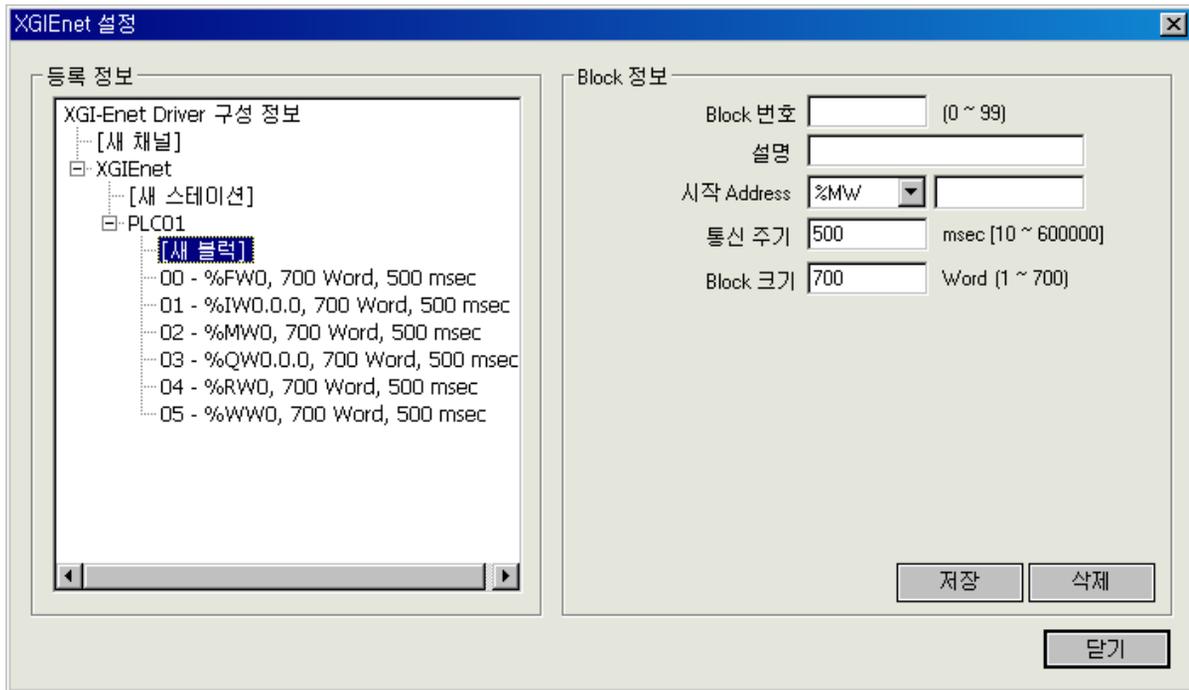
- (g) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
- (h) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.
- (i) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
- (j) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.
- (k) 통신 방식 : TCP와 UDP 중의 한가지를 선택합니다.
- (l) 포트 번호 : 통신 방식 선택에 따라 자동으로 입력됩니다.
- (m) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 6가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.
 - 올바른 예제: %MW0, %MW20, %IWO.0.0, %QW1.0.0
 - 잘못된 예제: %MW0.0.0, %IWO, %QWO
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 Block의 Block 크기를 Word(2 byte) 단위로 입력합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

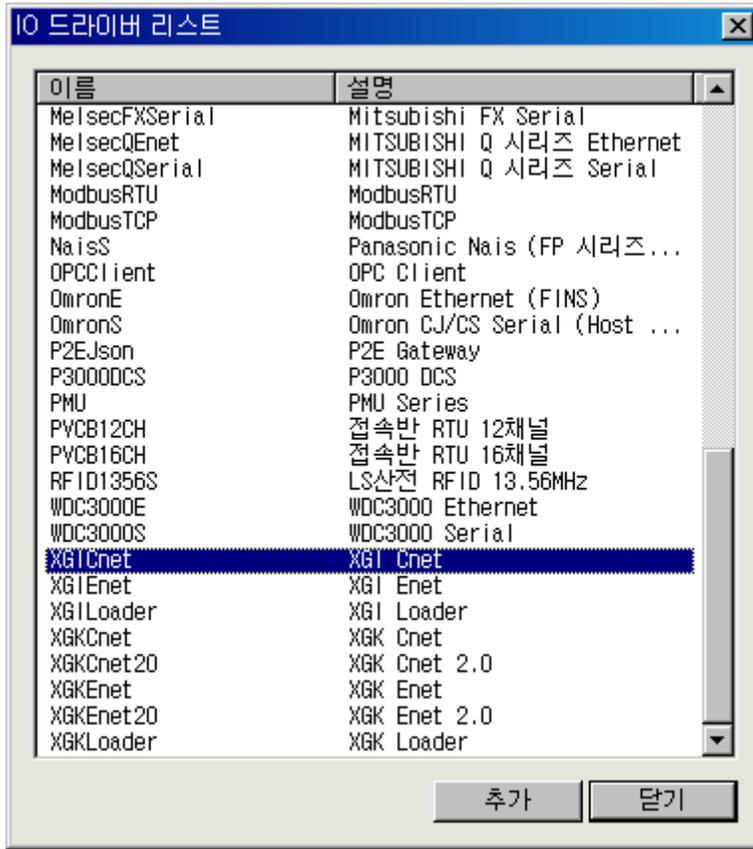
- . 아날로그 : %MX0, %MBO, %MNO, %MDO, %MLO, %IWO.0.0, %QWO.0.0
- . 디지털 : %MBO.0, %MNO.0, %MDO.0, %MLO.0, %IX0.0.0, %QX0.0.0

(b) 사용가능한 디바이스

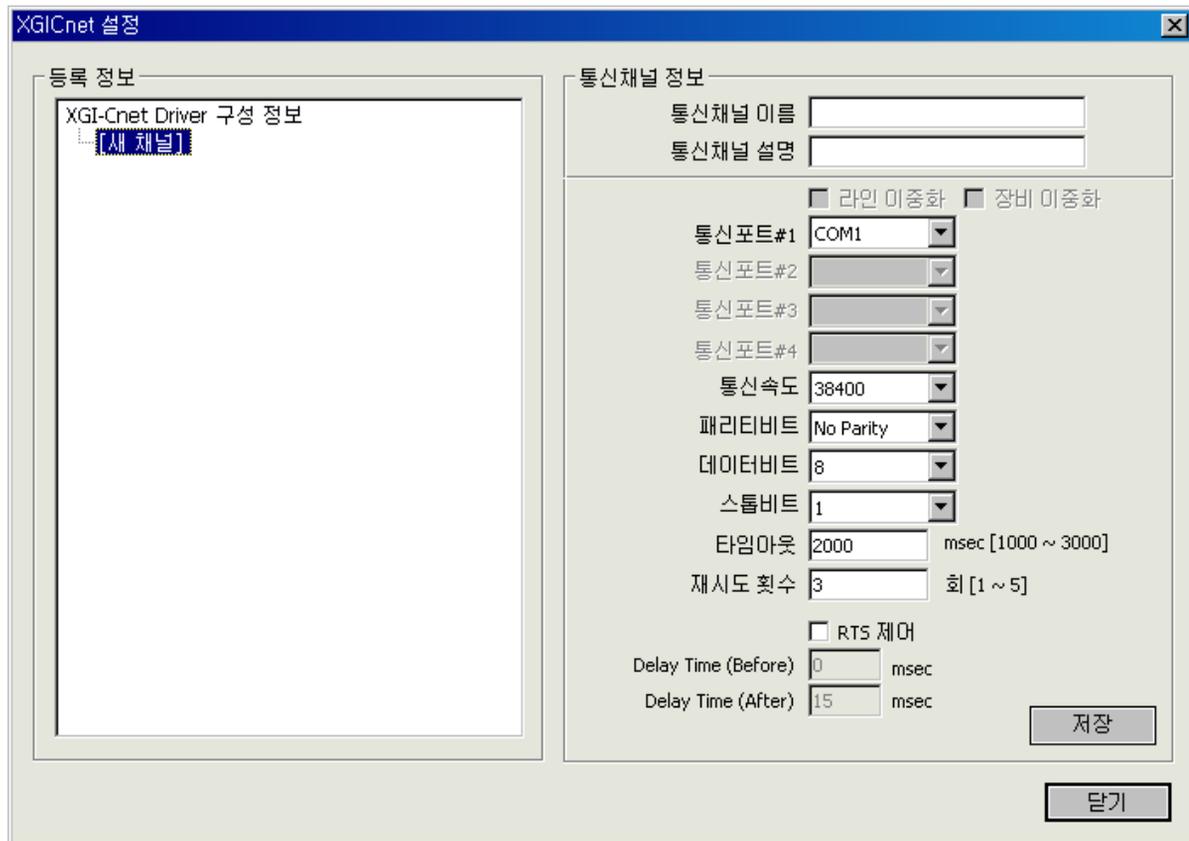
- . F, I, M, Q, R, W

20. XGICnet 드라이버

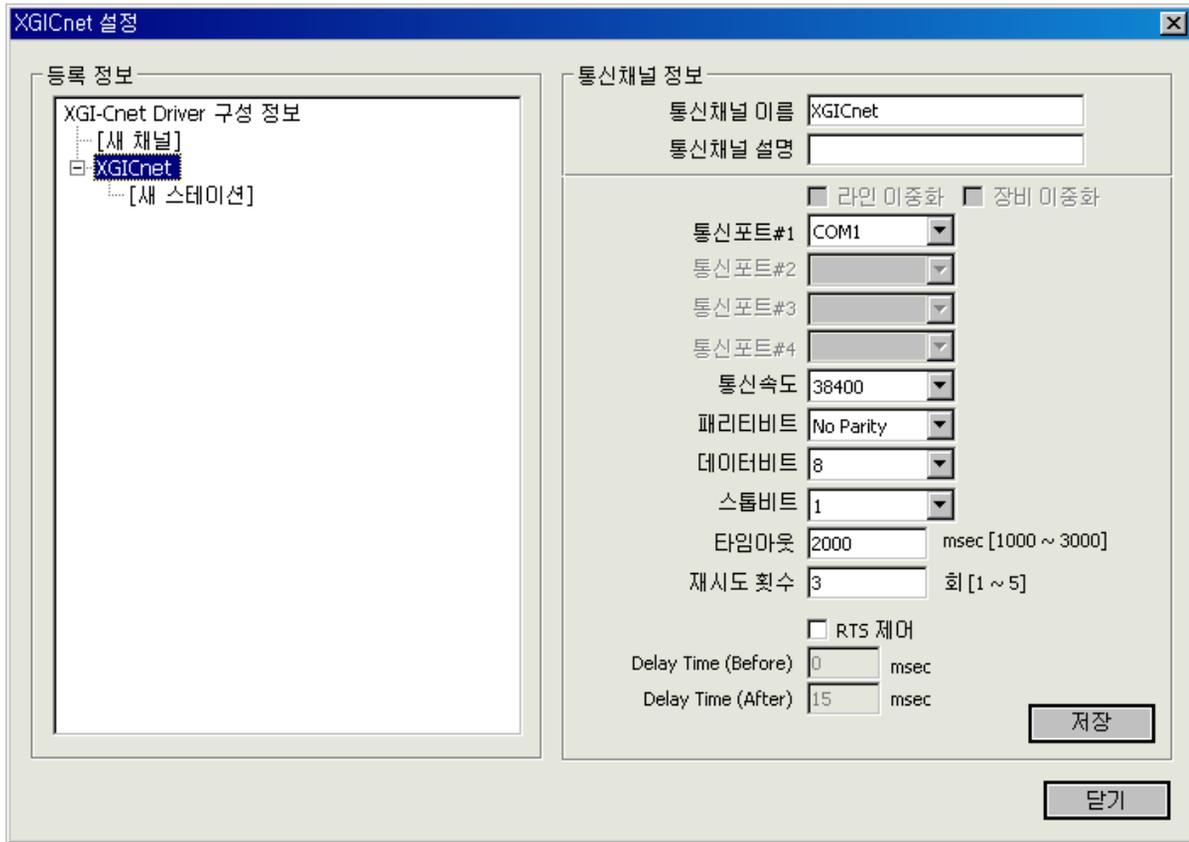
(1) 통신채널 추가



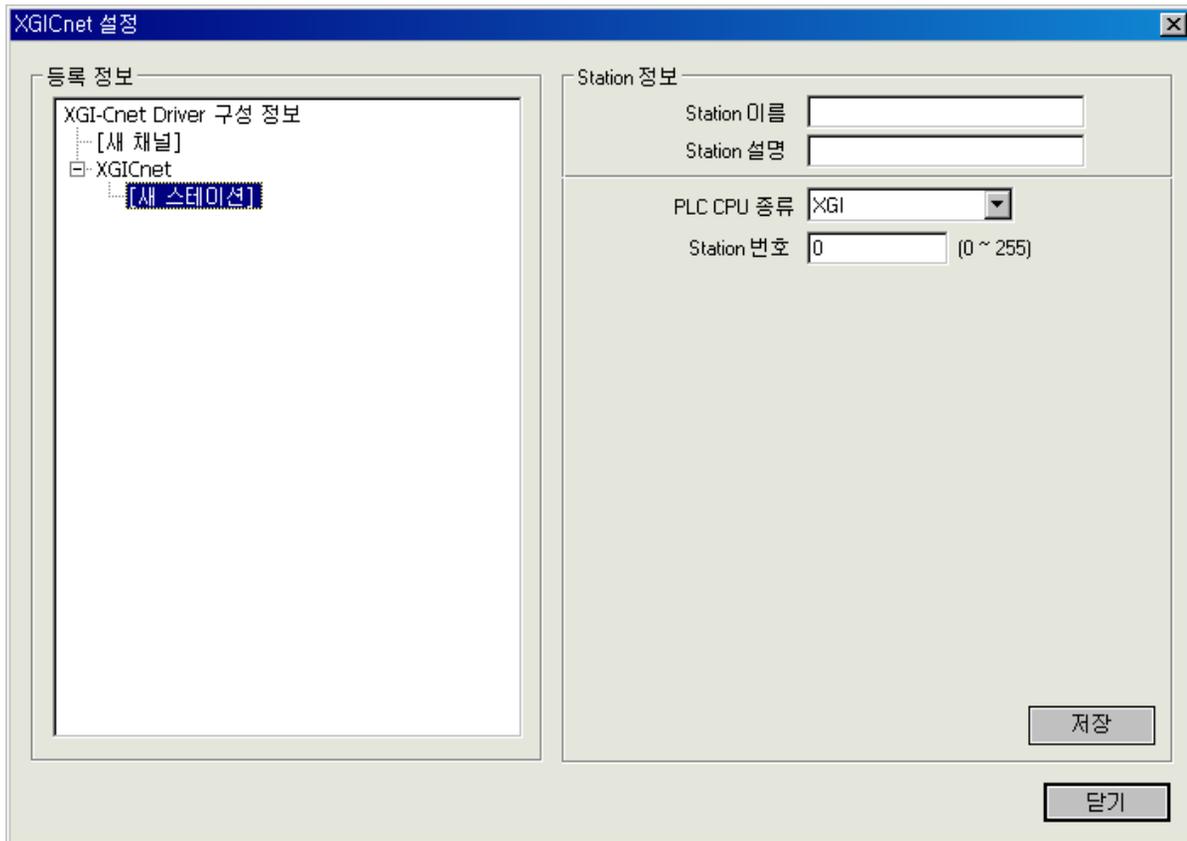
- IO 드라이버 리스트에서 “XGICnet” 을 선택한 후 “추가” 버튼을 누릅니다.



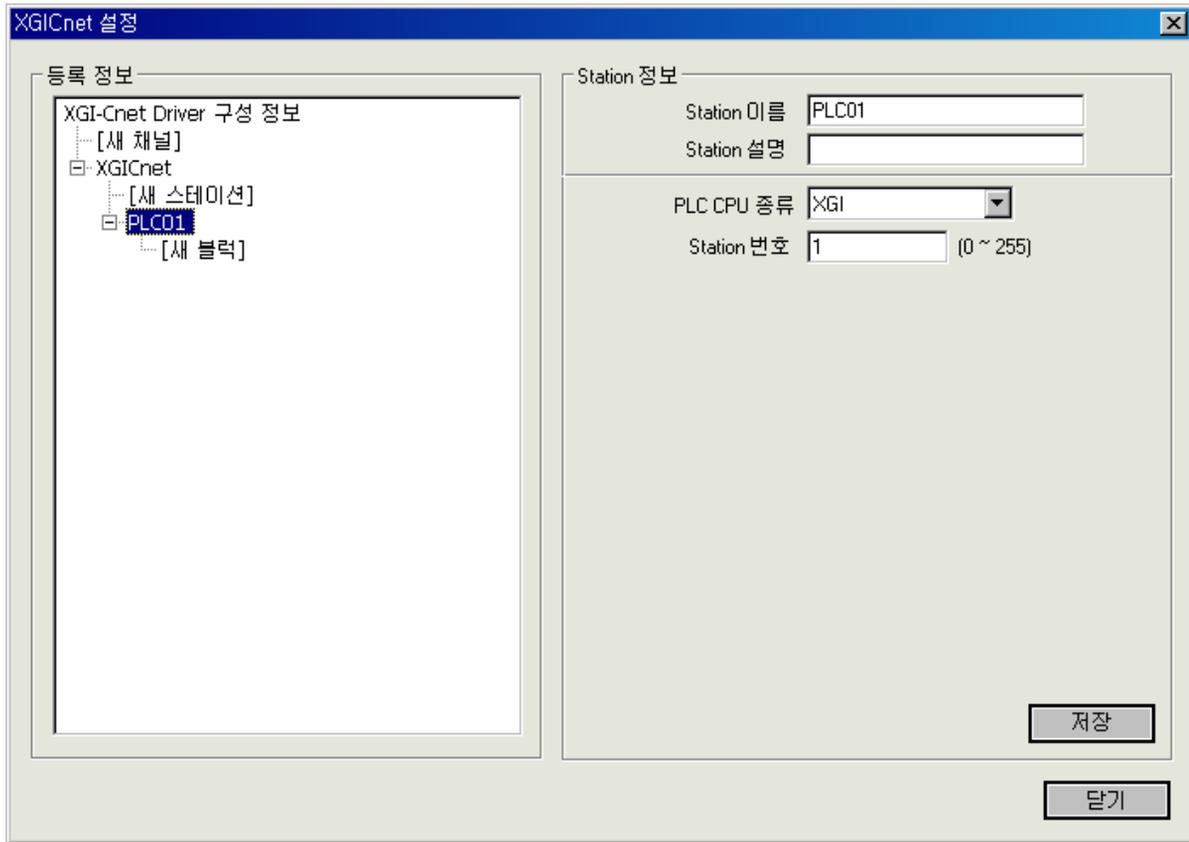
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (f) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) RTS 제어 : 485, 422 통신을 사용할 경우, 사용함으로 체크를 합니다.
- (k) Delay Time (Before) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 하기 직전의 Delay Time 입니다.
- (l) Delay Time (After) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 한 직후의 Delay Time 입니다.
- (m) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



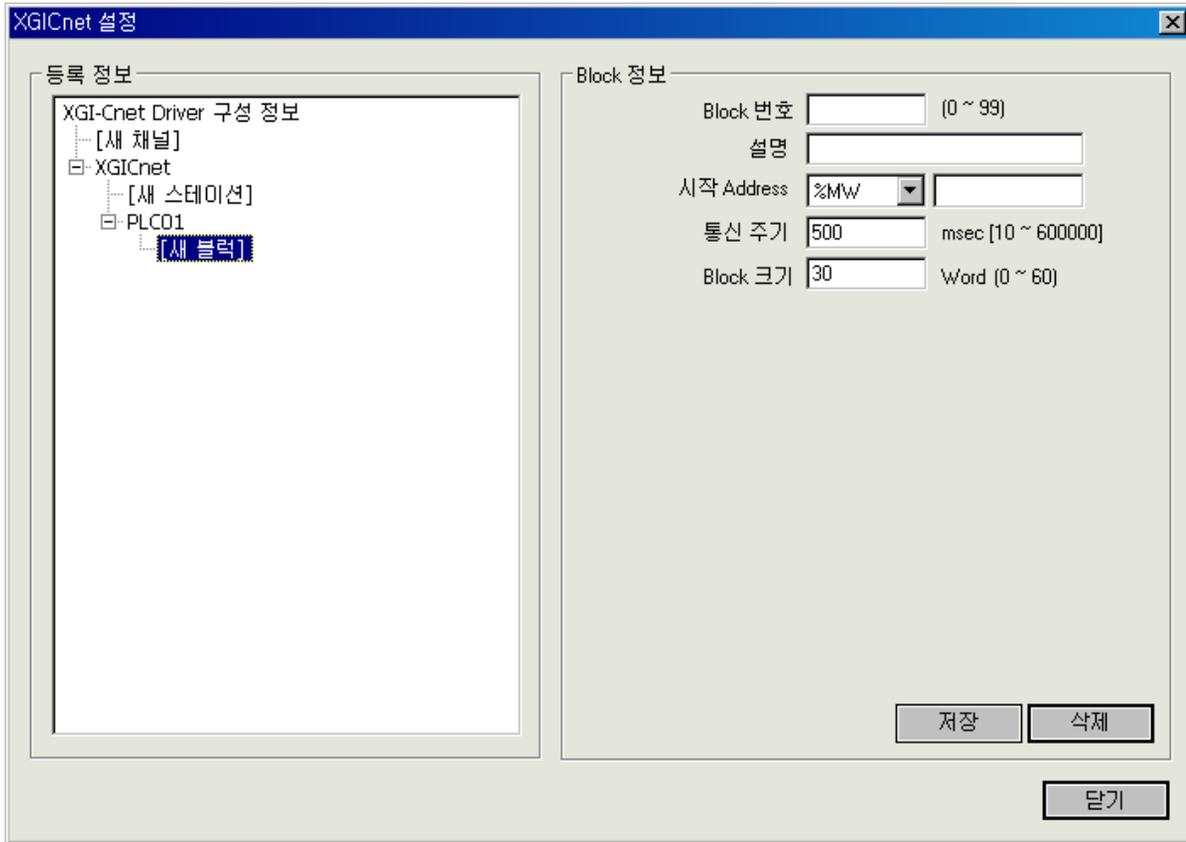
(2) Station 추가



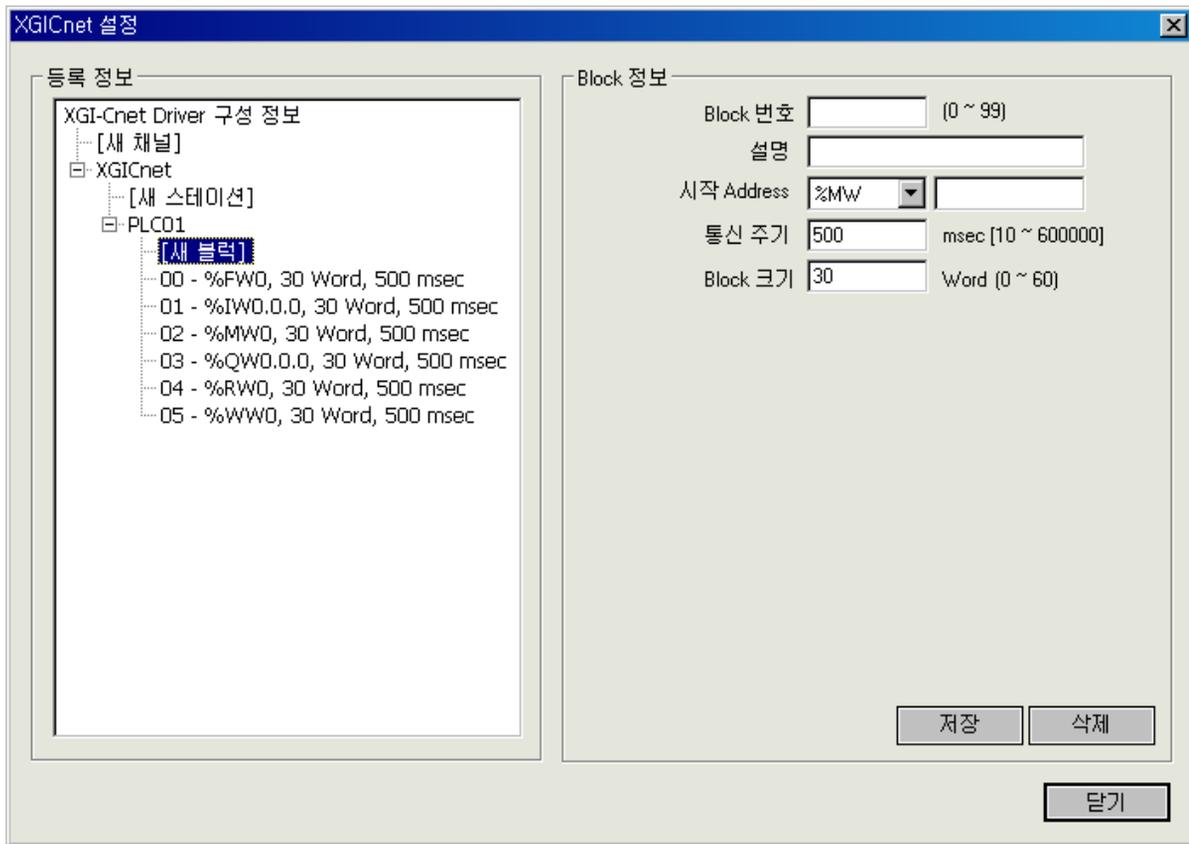
- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) Station 번호 : PLC Cnet 모듈의 국번을 입력합니다.
- (f) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 6가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.
 - 올바른 예제: %MW0, %MW20, %IWO.0.0, %QW1.0.0
 - 잘못된 예제: %MW0.0.0, %IWO, %QWO
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 Block의 Block 크기를 Word(2 byte) 단위로 입력합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

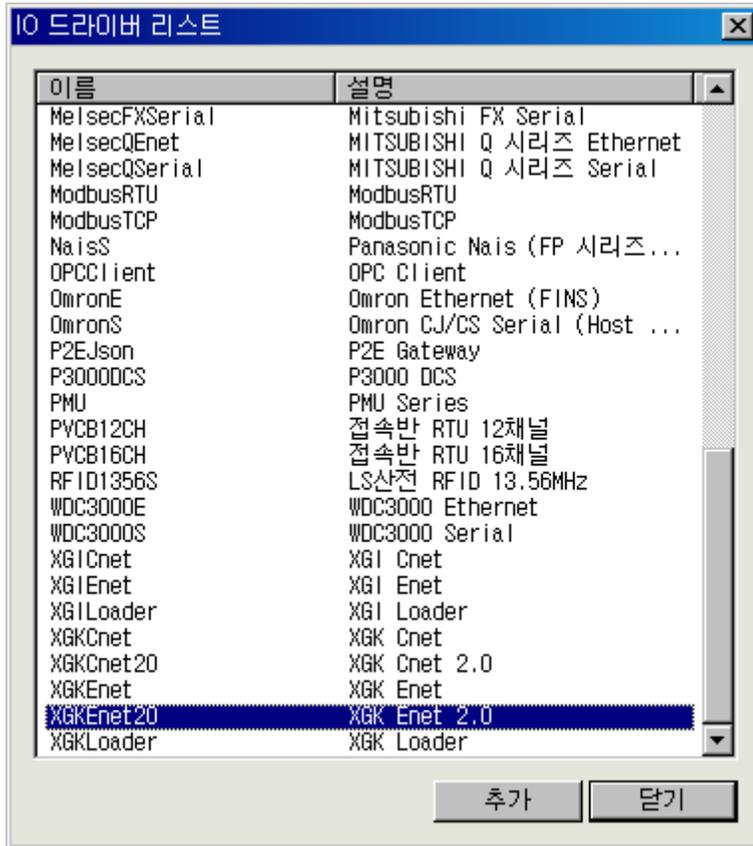
- . 아날로그 : %MX0, %MBO, %MNO, %MDO, %MLO, %IWO.0.0, %QWO.0.0
- . 디지털 : %MBO.0, %MNO.0, %MDO.0, %MLO.0, %IX0.0.0, %QX0.0.0

(b) 사용가능한 디바이스

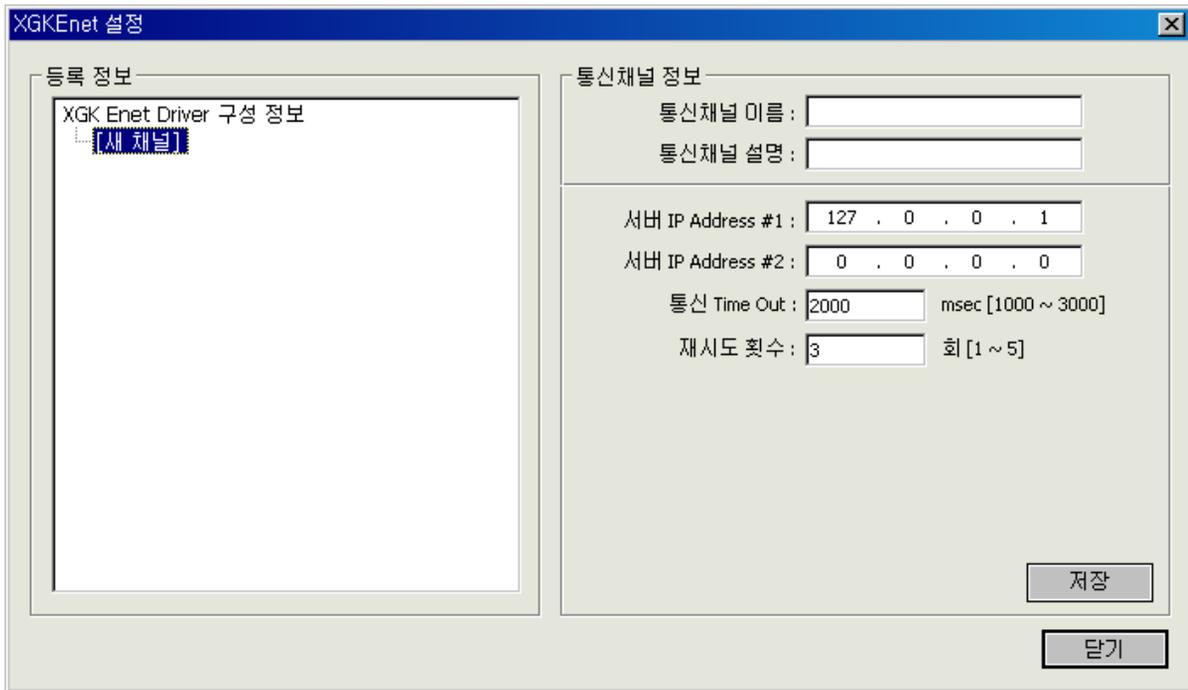
- . F, I, M, Q, R, W

21. XGKEnet20 드라이버

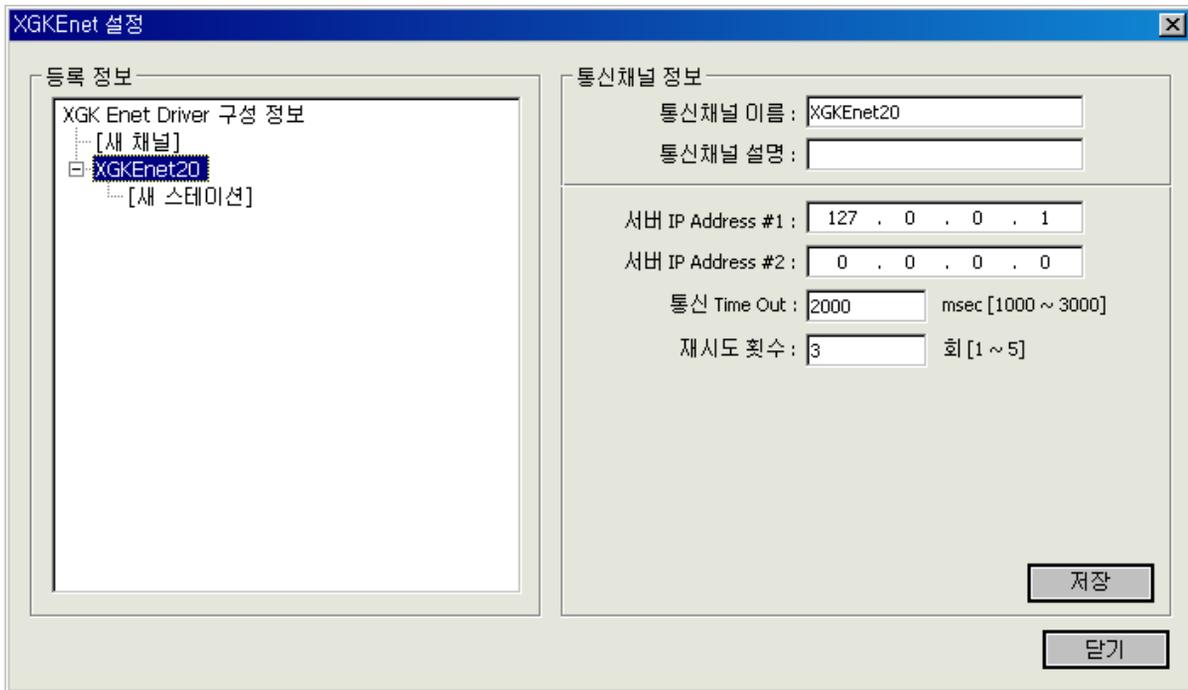
(1) 통신채널 추가



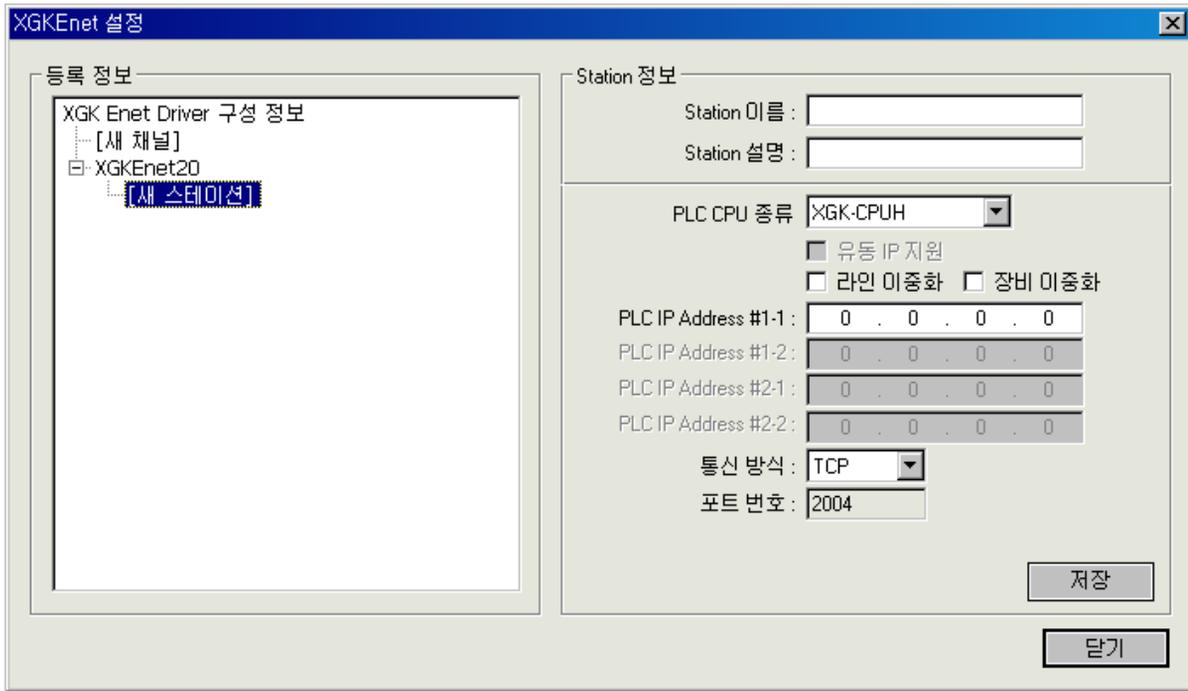
- IO 드라이버 리스트에서 “XGKEnet20” 을 선택한 후 “추가” 버튼을 누릅니다.



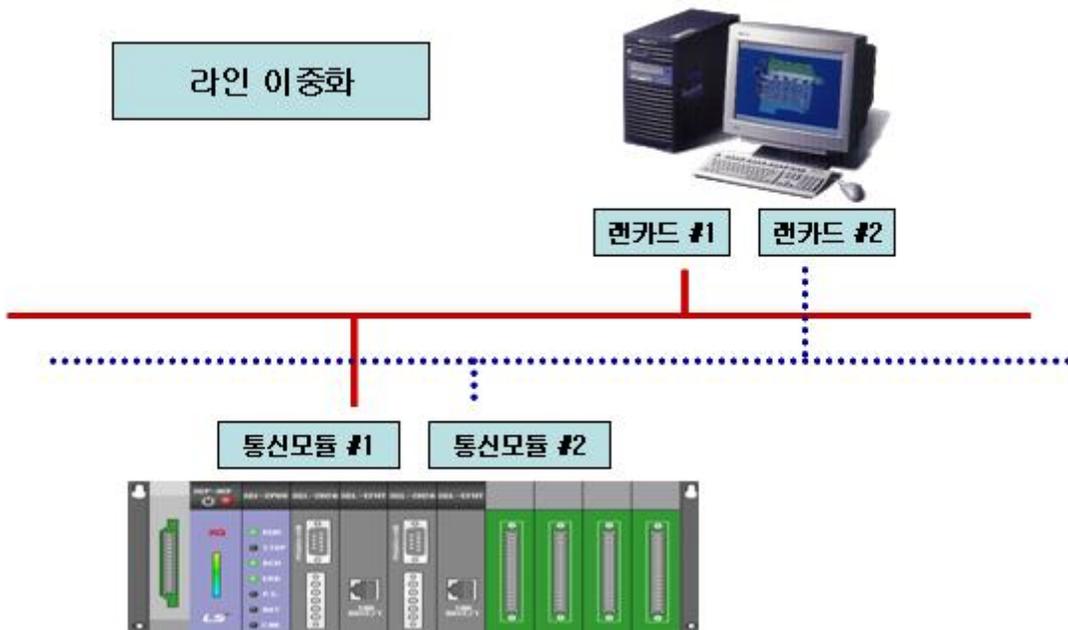
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- (d) 서버 IP Address #2 : 라인 이중화를 사용할 경우 사용하게 될 두번째 IP Address를 입력합니다.
- (e) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (f) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(2) Station 추가



- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다. 컴퓨터에 랜카드 2개를 설치하고, PLC에 Enet 통신 모듈을 2개 설치하여, 그림과 같이 네트워크를 분리하여 통신하고자 할 때 사용합니다. 네트워크 라인에 이상이 있을때를 대비하기 위한 이중화 옵션 입니다.

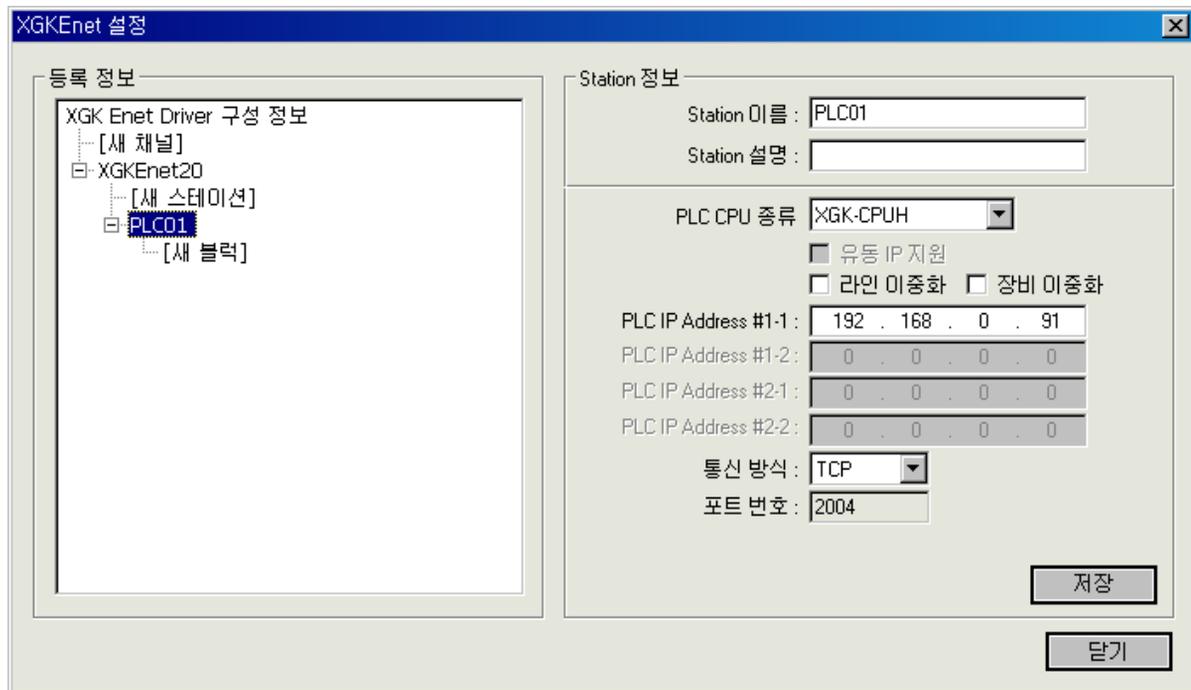


- (f) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다. 컴퓨터에 랜카드 1개를 설치하고, PLC에 Enet 통신 모듈

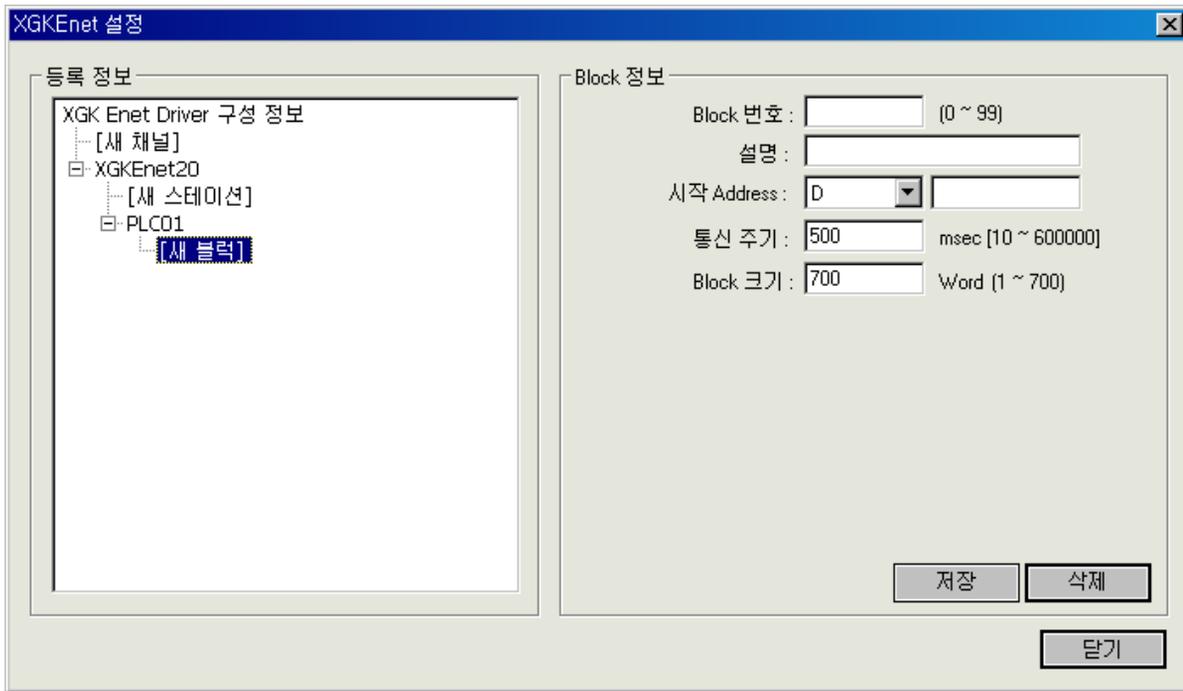
을 2개 설치하여, 그림과 같이 통신 모듈이 분리되어 있는 경우 사용합니다. PLC 통신 모듈에 이상이 있을때를 대비하기 위한 이중화 옵션 입니다.



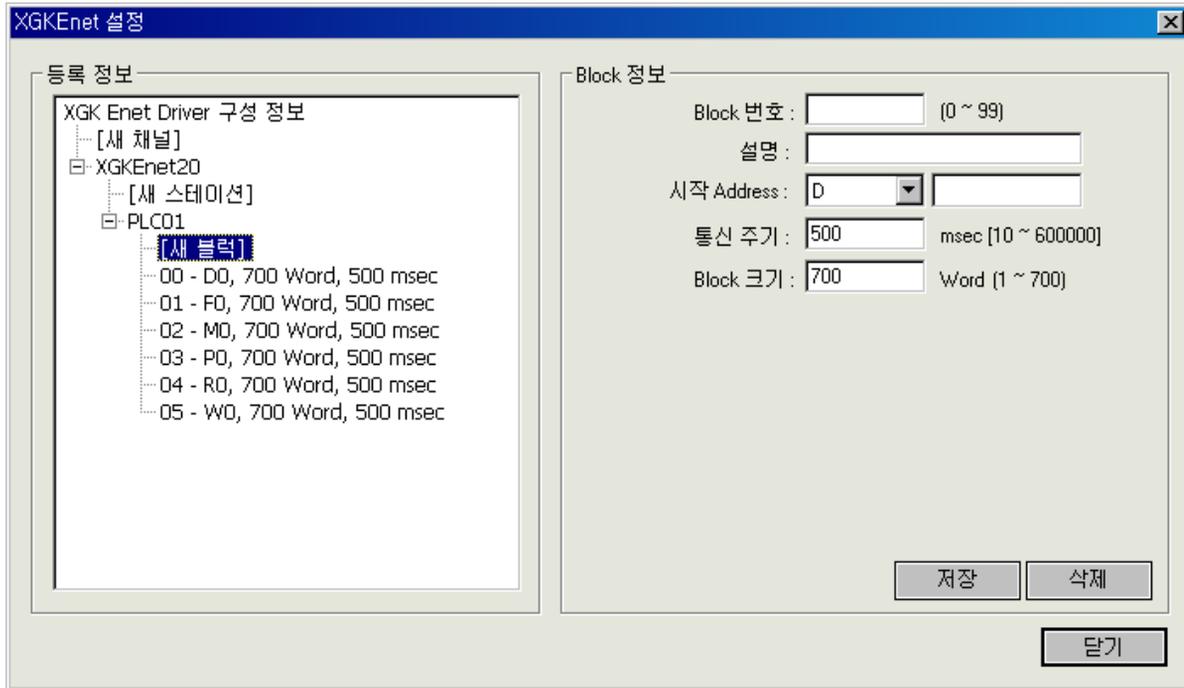
- (g) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
- (h) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.
- (i) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
- (j) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.
- (k) 통신 방식 : TCP와 UDP 중의 한가지를 선택합니다.
- (l) 포트 번호 : 통신 방식 선택에 따라 자동으로 입력됩니다.
- (m) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호 입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 6가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.
 - . 올바른 예제: D0, F20, M10, P30, R20, W20
 - . 잘못된 예제: M0.0.0, F11A
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 Block의 Block 크기를 Word(2 byte) 단위로 입력합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

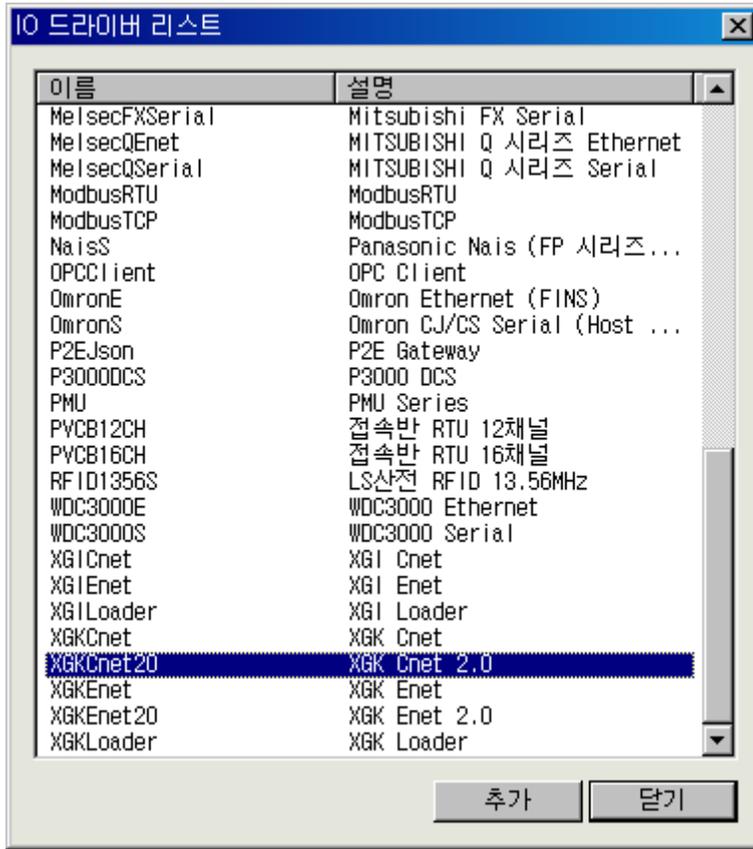
- . 아날로그 : D20, F10, M30, P0, R20
- . 디지털 : D20A, F105, M30F, P01, R20F (마지막 문자가 해당 Word 값의 비트 위치를 나타냅니다. 0 ~ F 까지 사용 가능)

(b) 사용가능한 디바이스

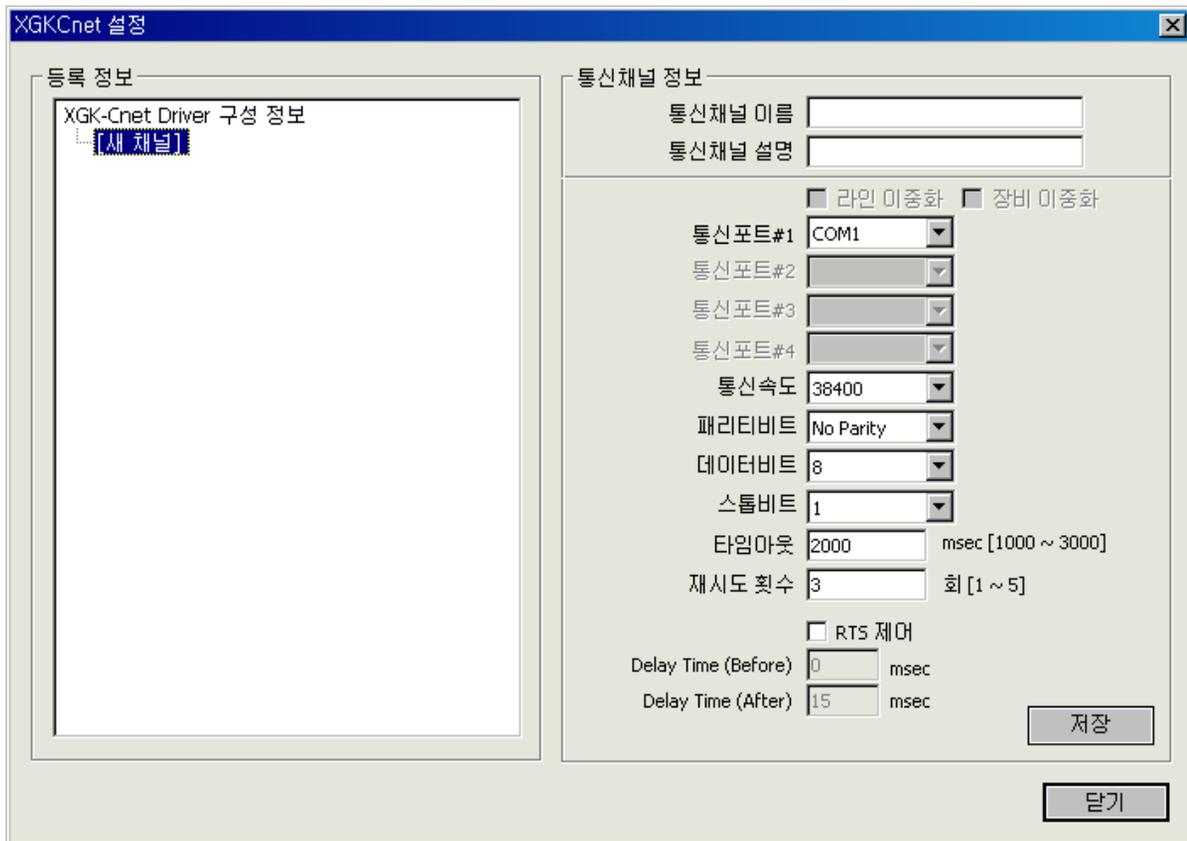
- . D, F, M, P, R, W

22. XGKnet20 드라이버

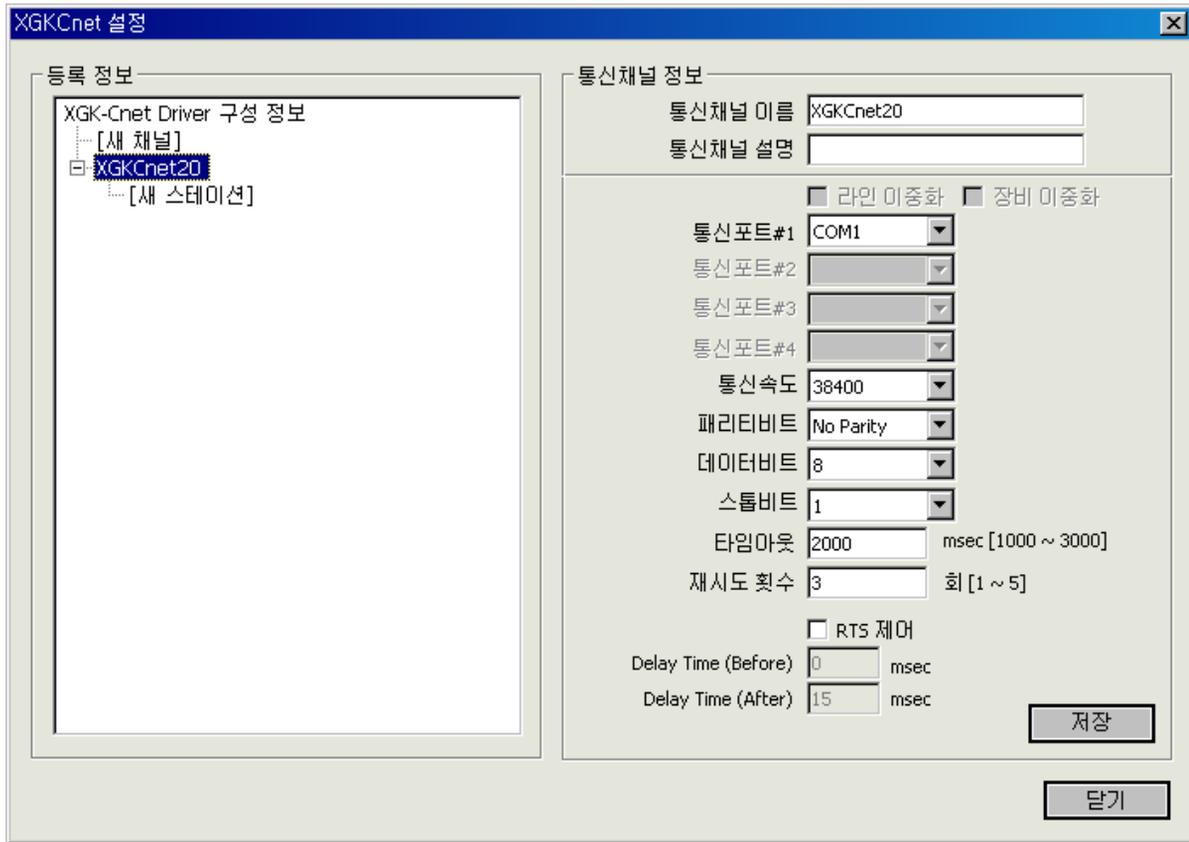
(1) 통신채널 추가



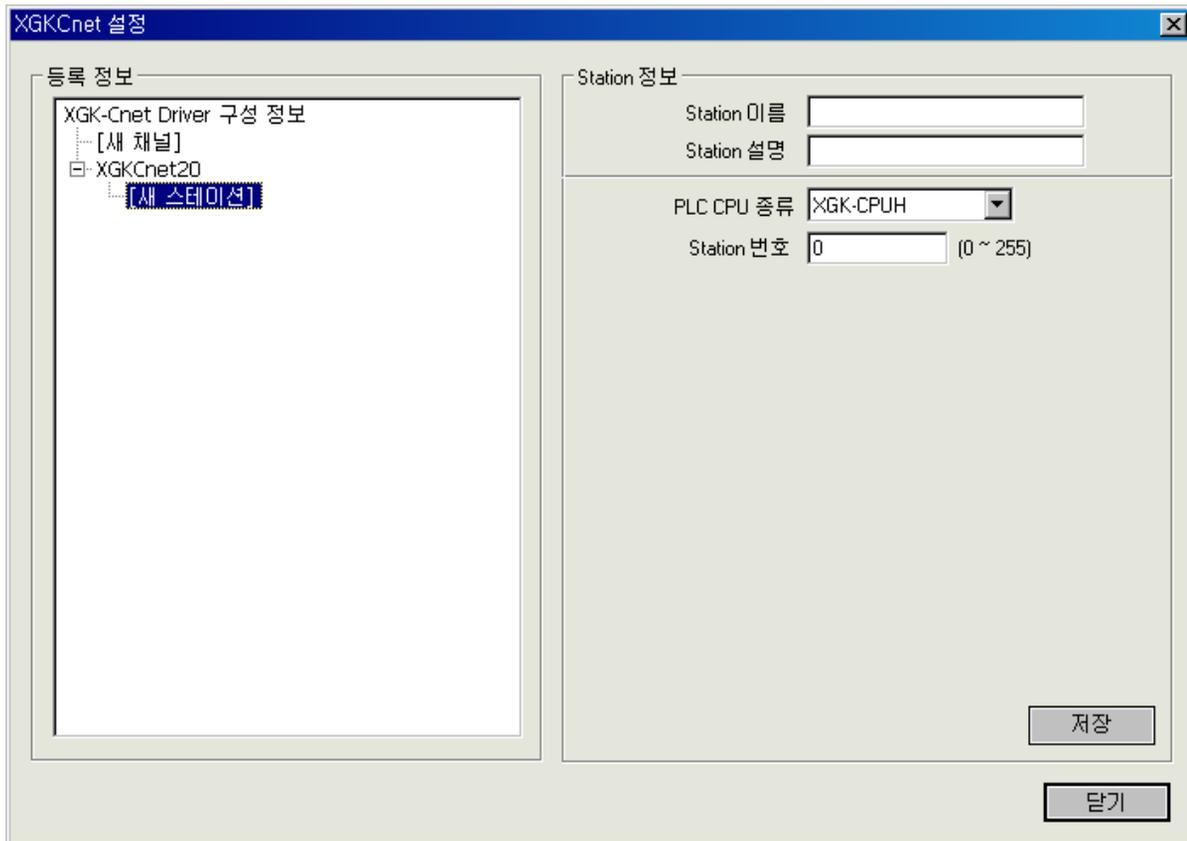
- IO 드라이버 리스트에서 “XGKnet20” 을 선택한 후 “추가” 버튼을 누릅니다.



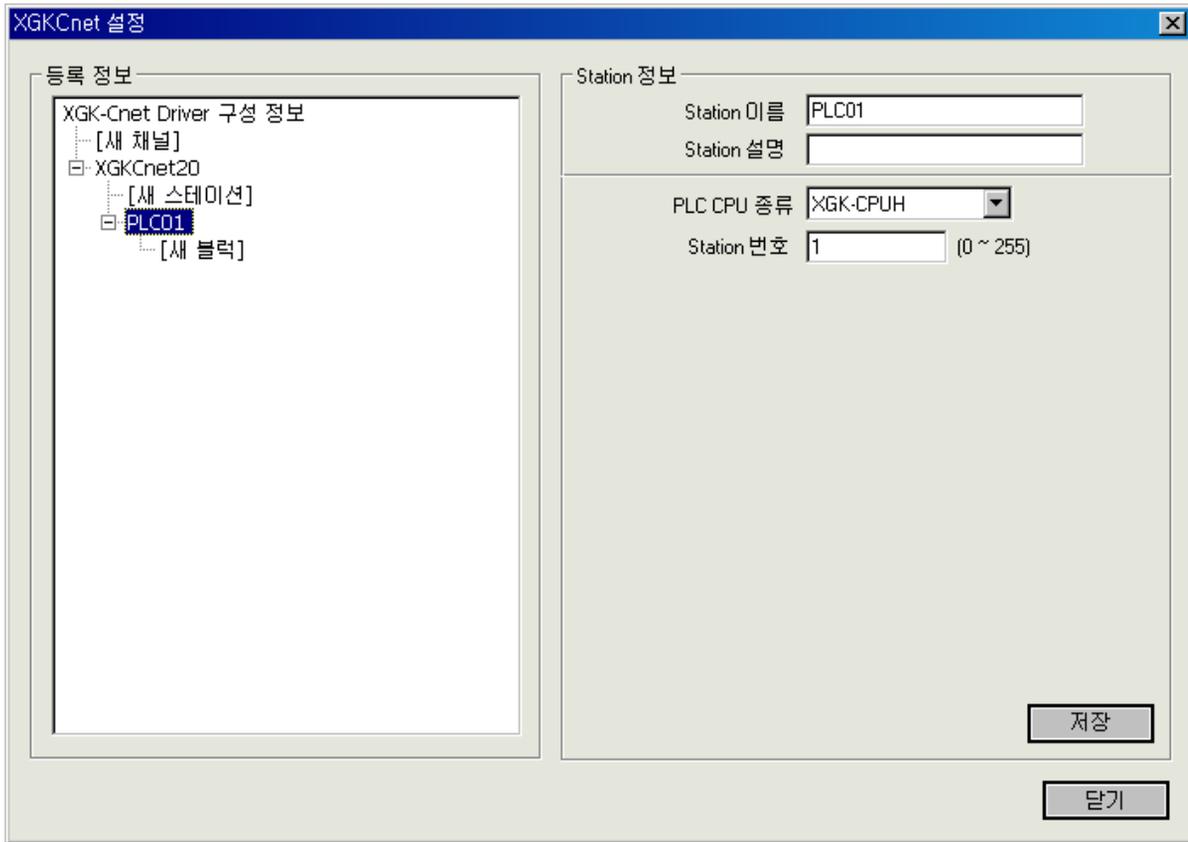
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (f) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) RTS 제어 : 485, 422 통신을 사용할 경우, 사용함으로 체크를 합니다.
- (k) Delay Time (Before) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 하기 직전의 Delay Time 입니다.
- (l) Delay Time (After) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 한 직후의 Delay Time 입니다.
- (m) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



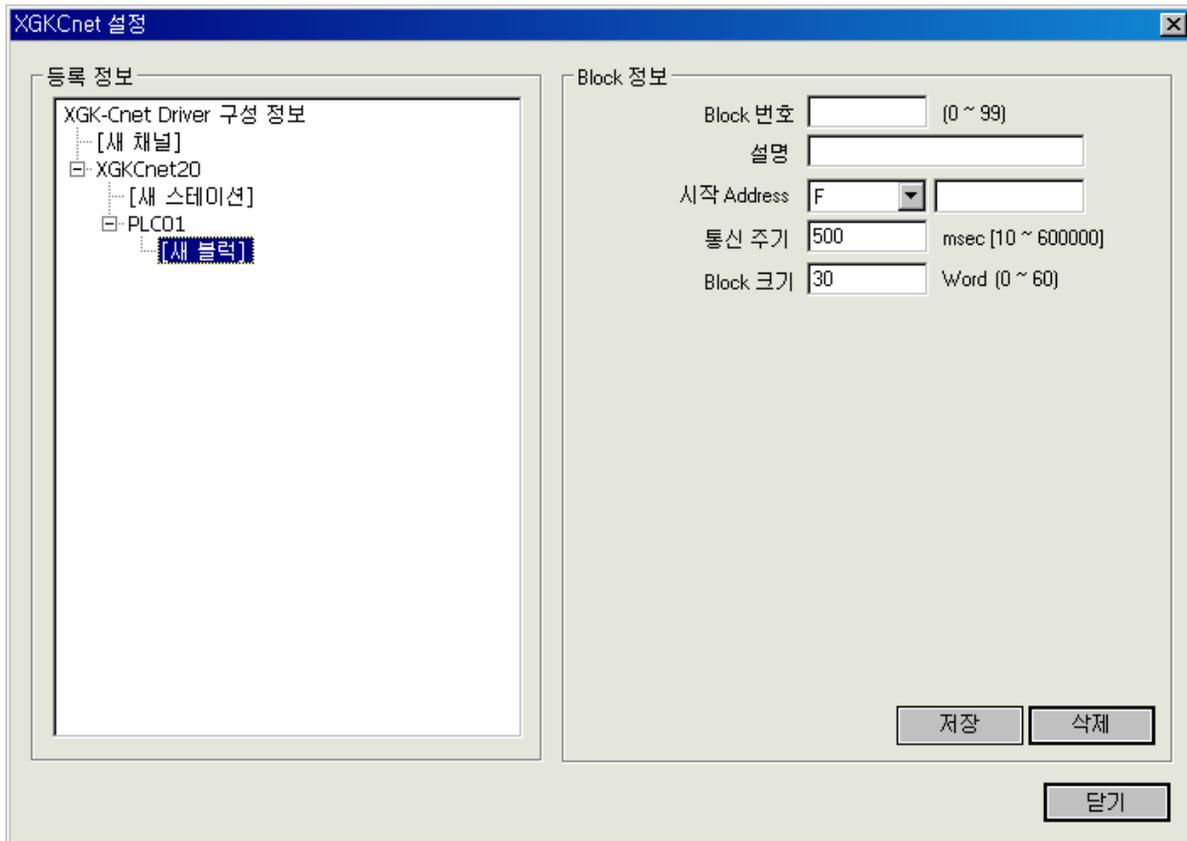
(2) Station 추가



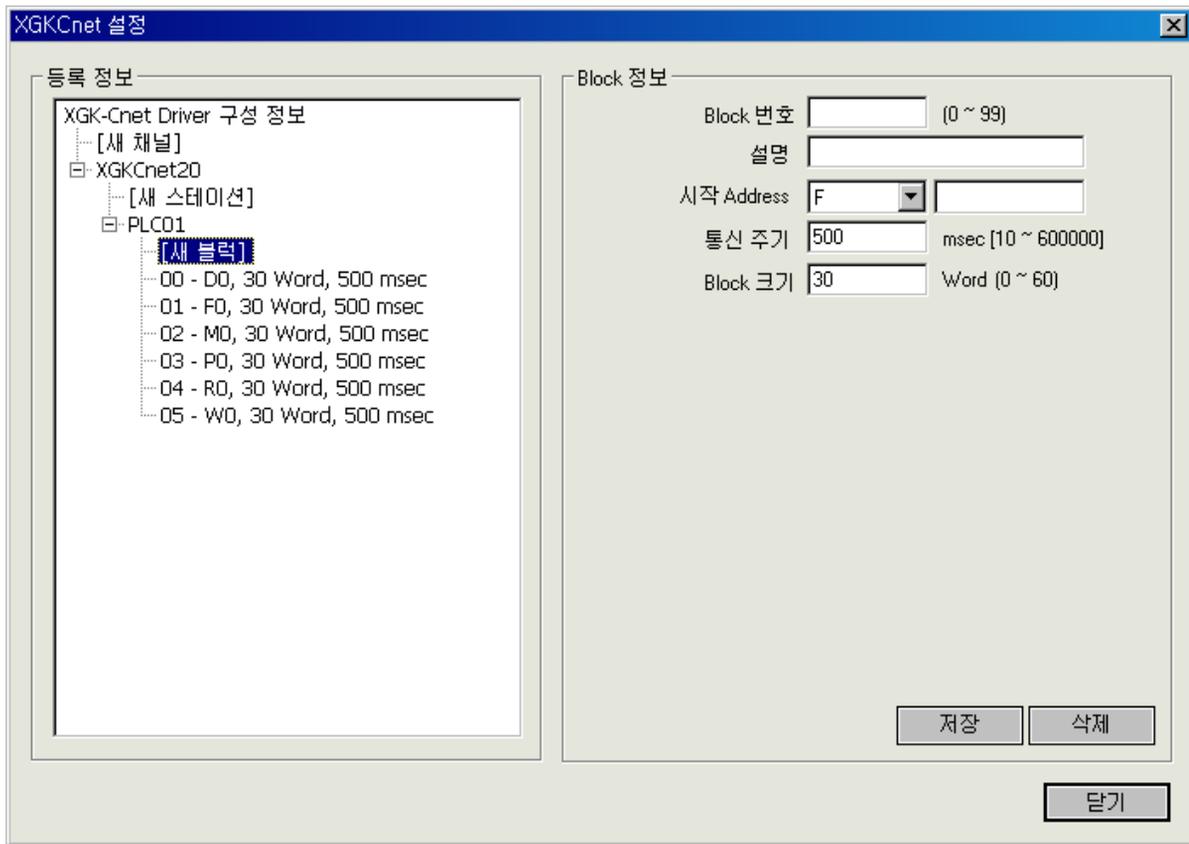
- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) Station 번호 : PLC Cnet 모듈의 국번을 입력합니다.
- (f) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 6가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.
 - 올바른 예제: D0, F20, M10, P30, R20, W20
 - 잘못된 예제: M0.0.0, F11A
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 Block의 Block 크기를 Word(2 byte) 단위로 입력합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

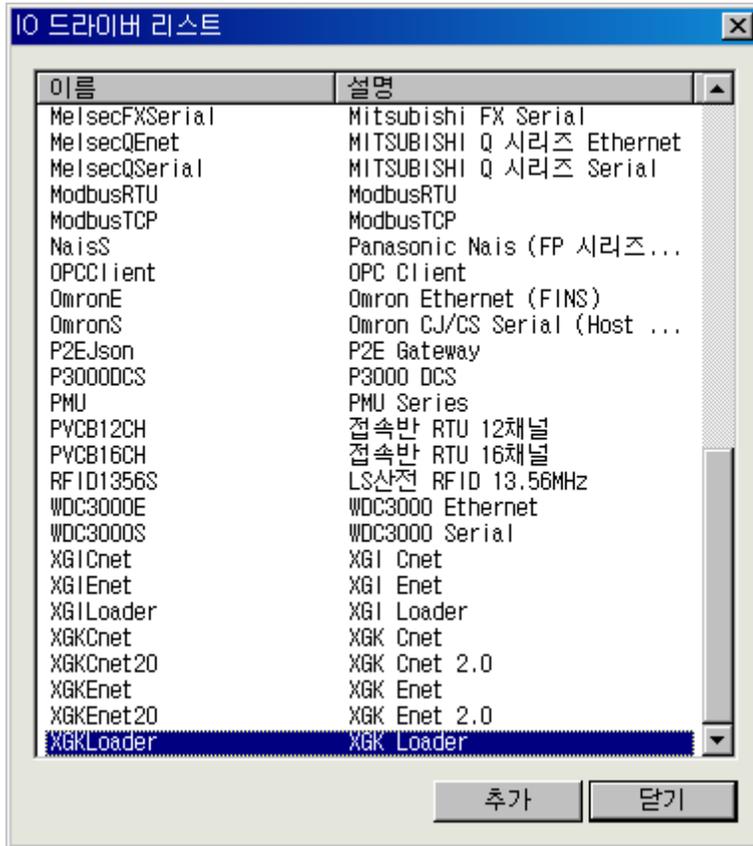
- . 아날로그 : D20, F10, M30, P0, R20
- . 디지털 : D20A, F105, M30F, P01, R20F (마지막 문자가 해당 Word 값의 비트 위치를 나타냅니다. 0 ~ F 까지 사용 가능)

(b) 사용가능한 디바이스

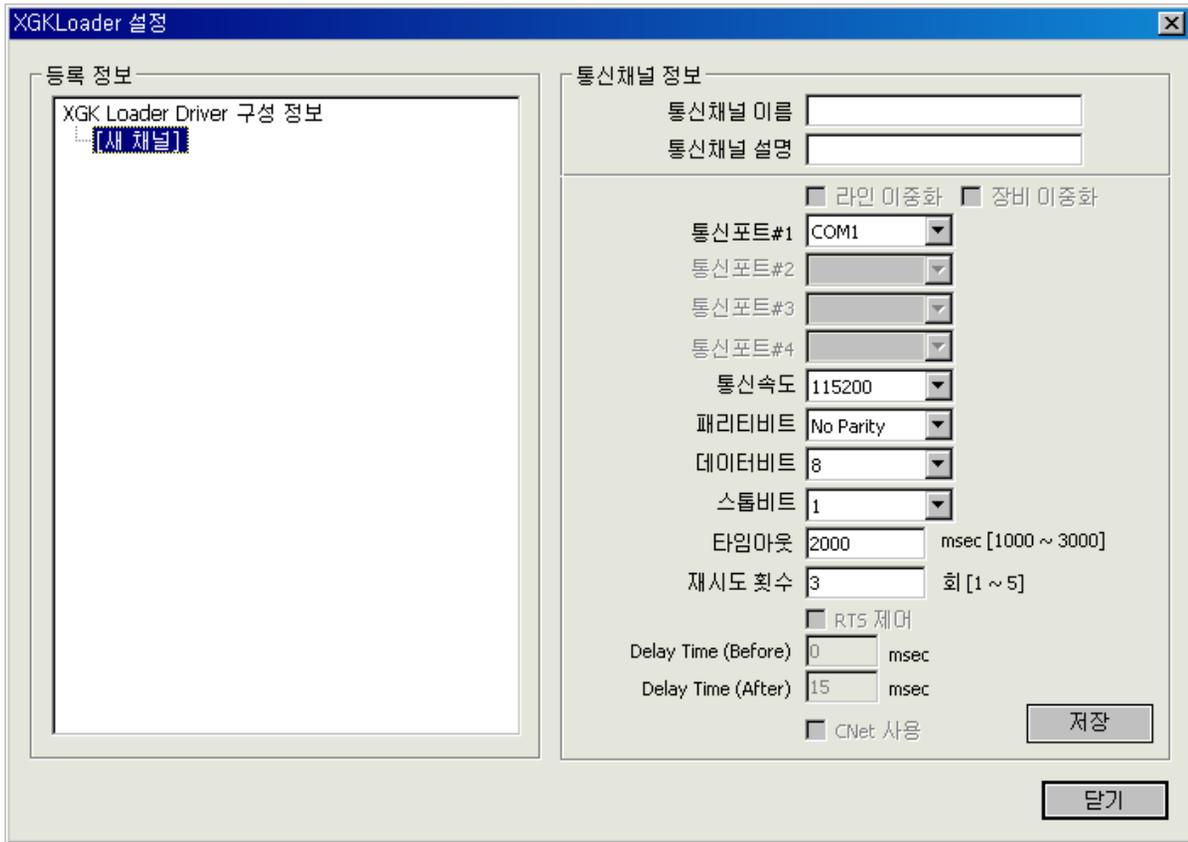
- . D, F, M, P, R, W

23. XGKLoader 드라이버

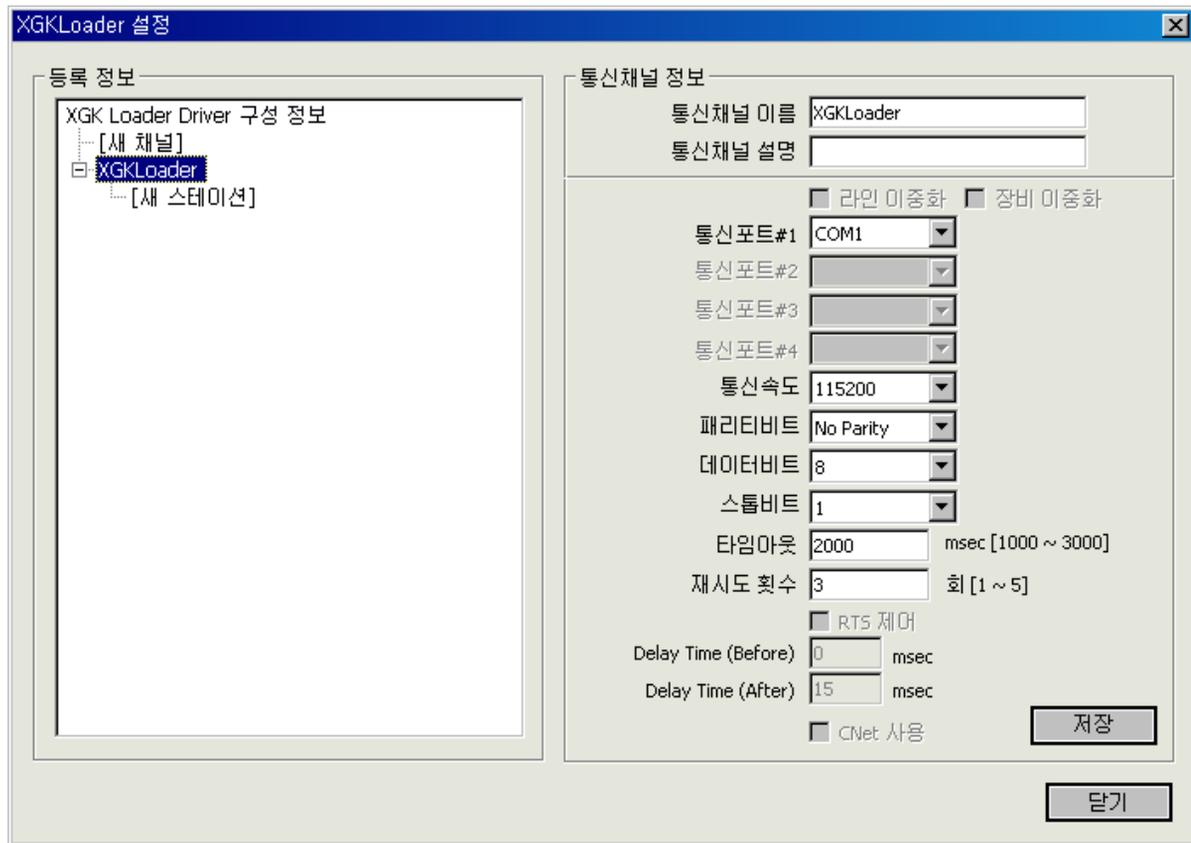
(1) 통신채널 추가



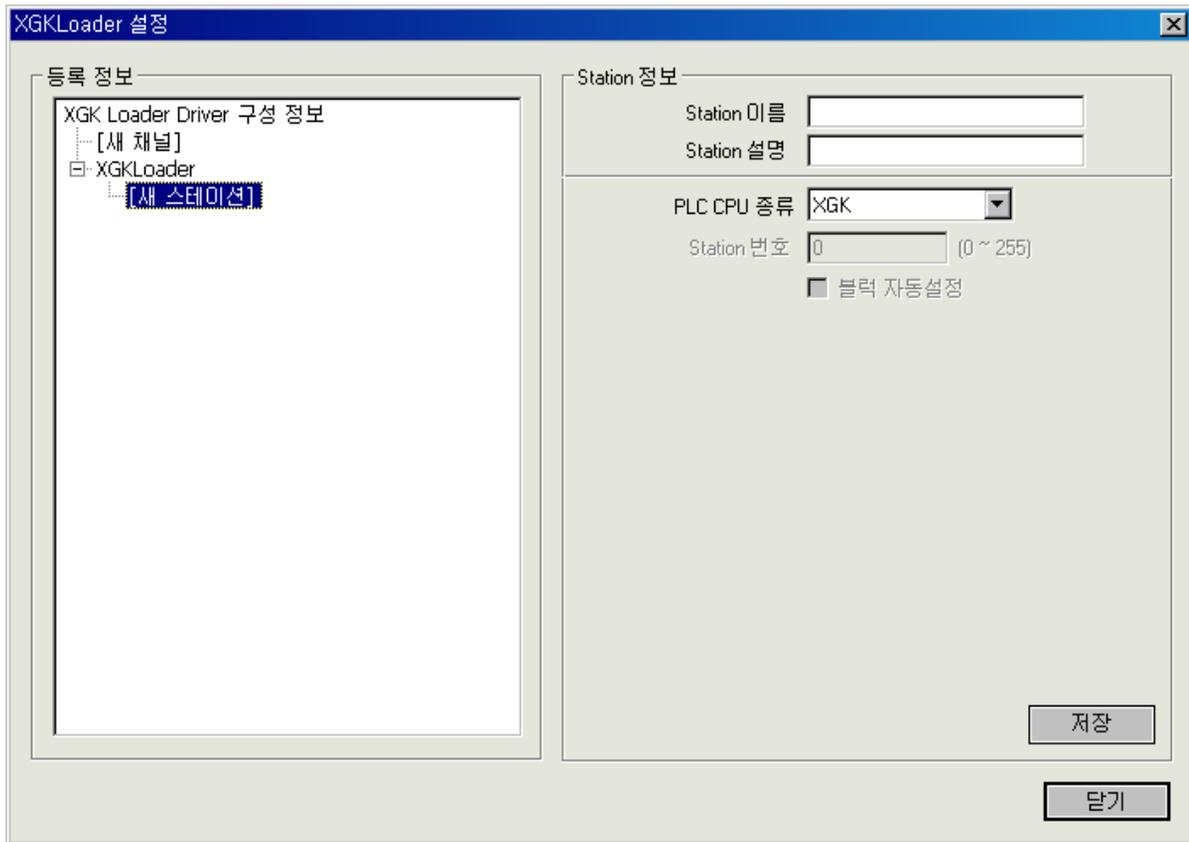
- IO 드라이버 리스트에서 “XGKLoader” 을 선택한 후 “추가” 버튼을 누릅니다.



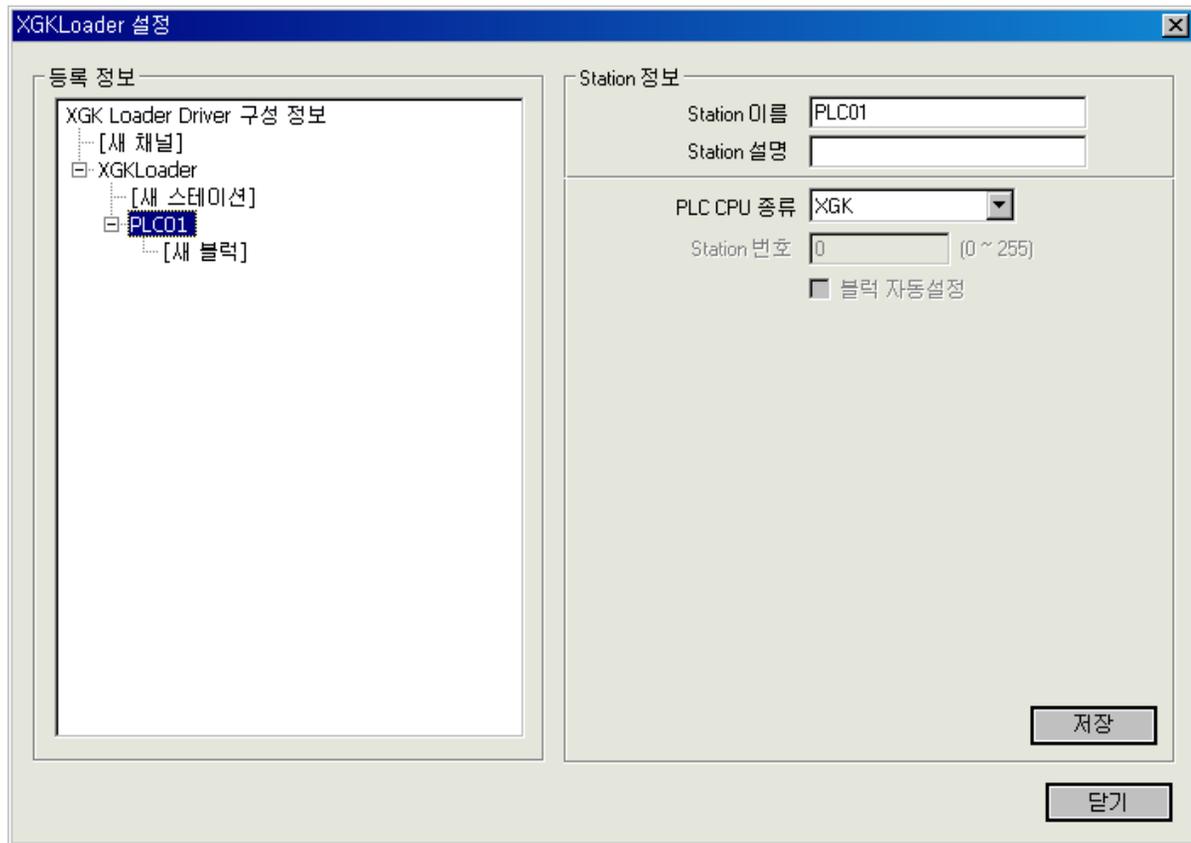
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다. PLC Loader 포트의 통신 속도는 115200 bps 입니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (f) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



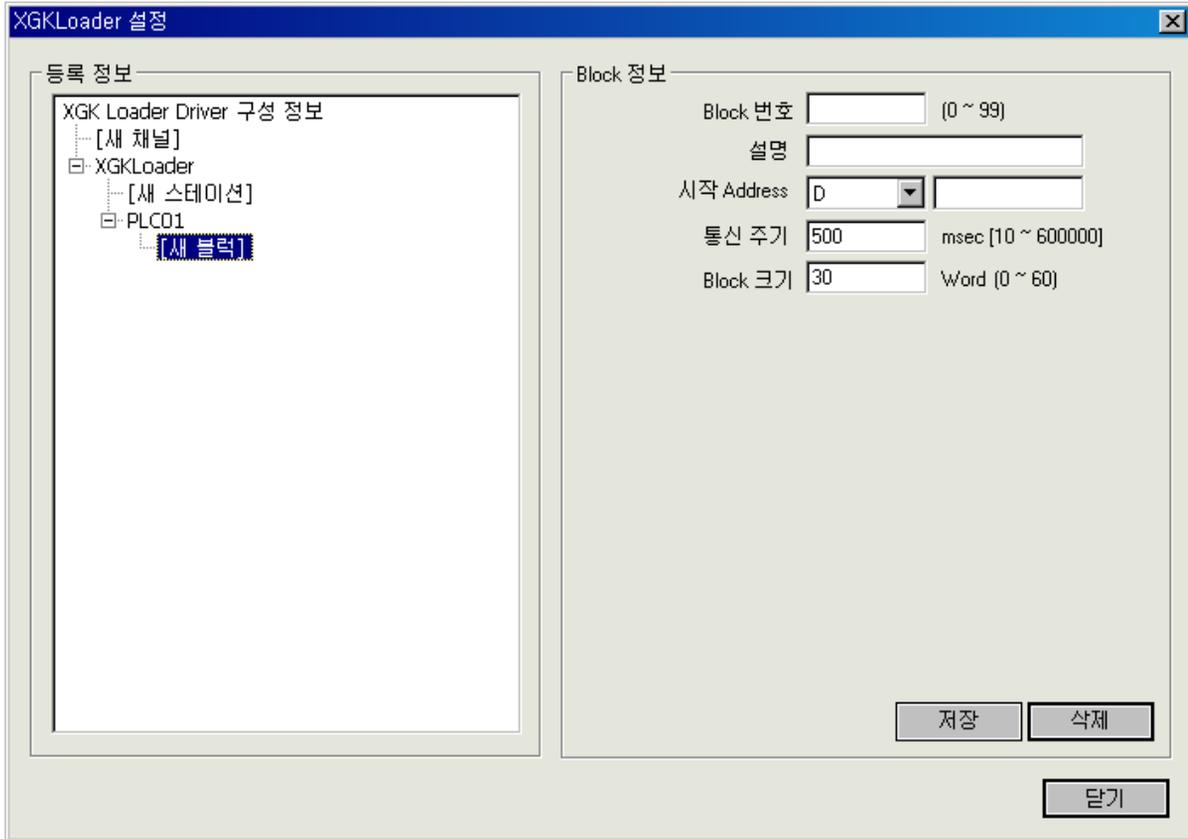
(2) Station 추가



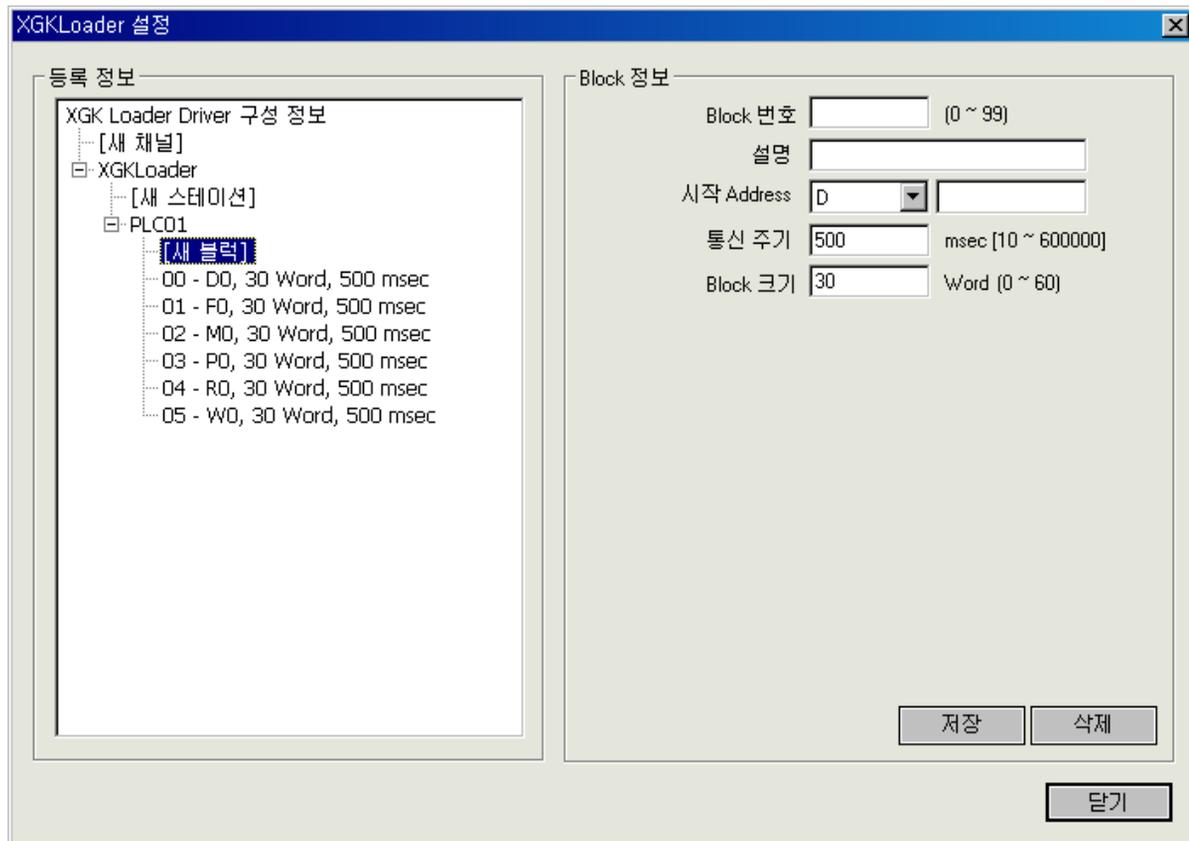
- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호 입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 6가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.
 - 올바른 예제: D0, F20, M10, P30, R20, W20
 - 잘못된 예제: M0.0.0, F11A
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 Block의 Block 크기를 Word(2 byte) 단위로 입력합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

- . 아날로그 : D20, F10, M30, P0, R20

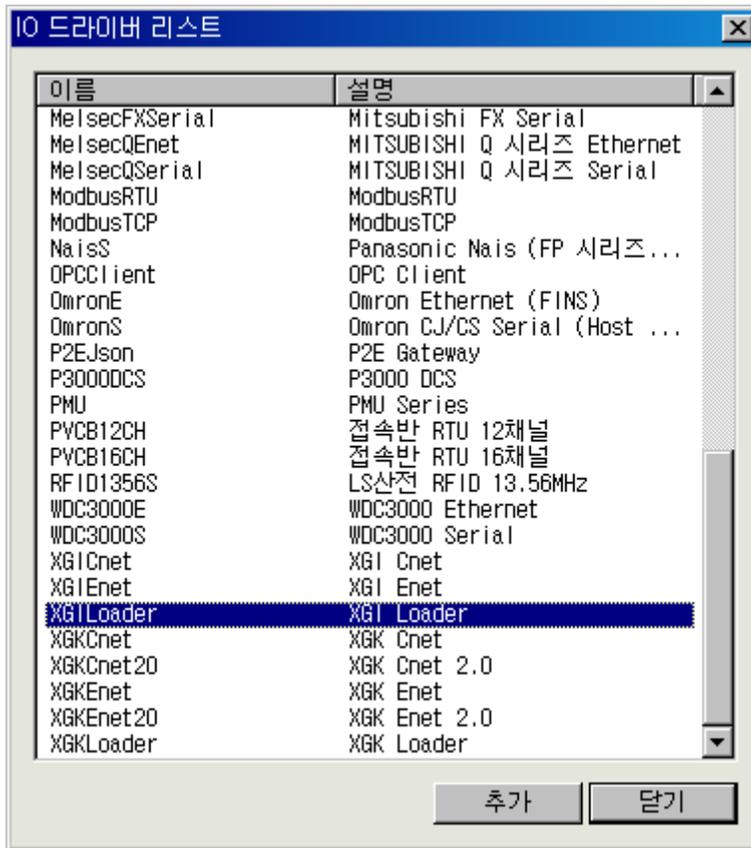
- . 디지털 : D20A, F105, M30F, P01, R20F (마지막 문자가 해당 Word 값의 비트 위치를 나타냅니다. 0 ~ F 까지 사용 가능)

(b) 사용가능한 디바이스

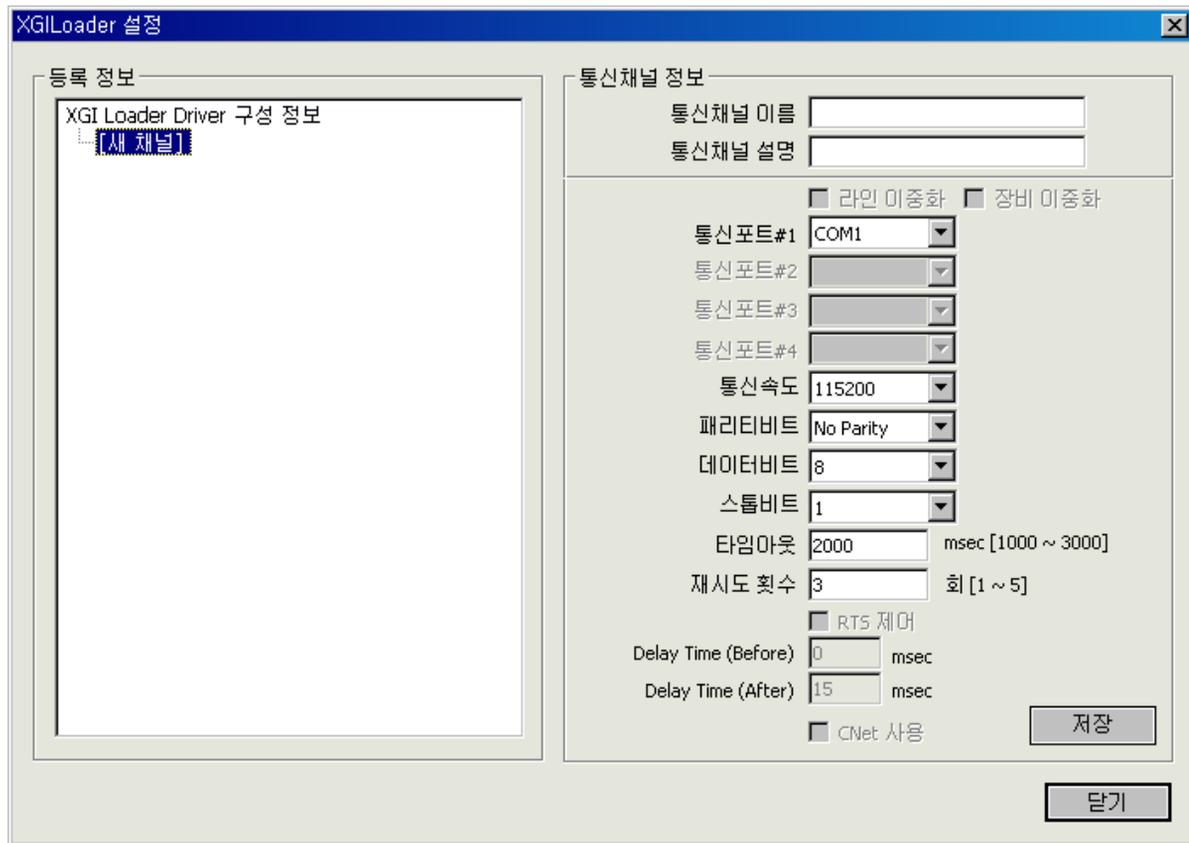
- . D, F, M, P, R, W

24. XGILoader 드라이버

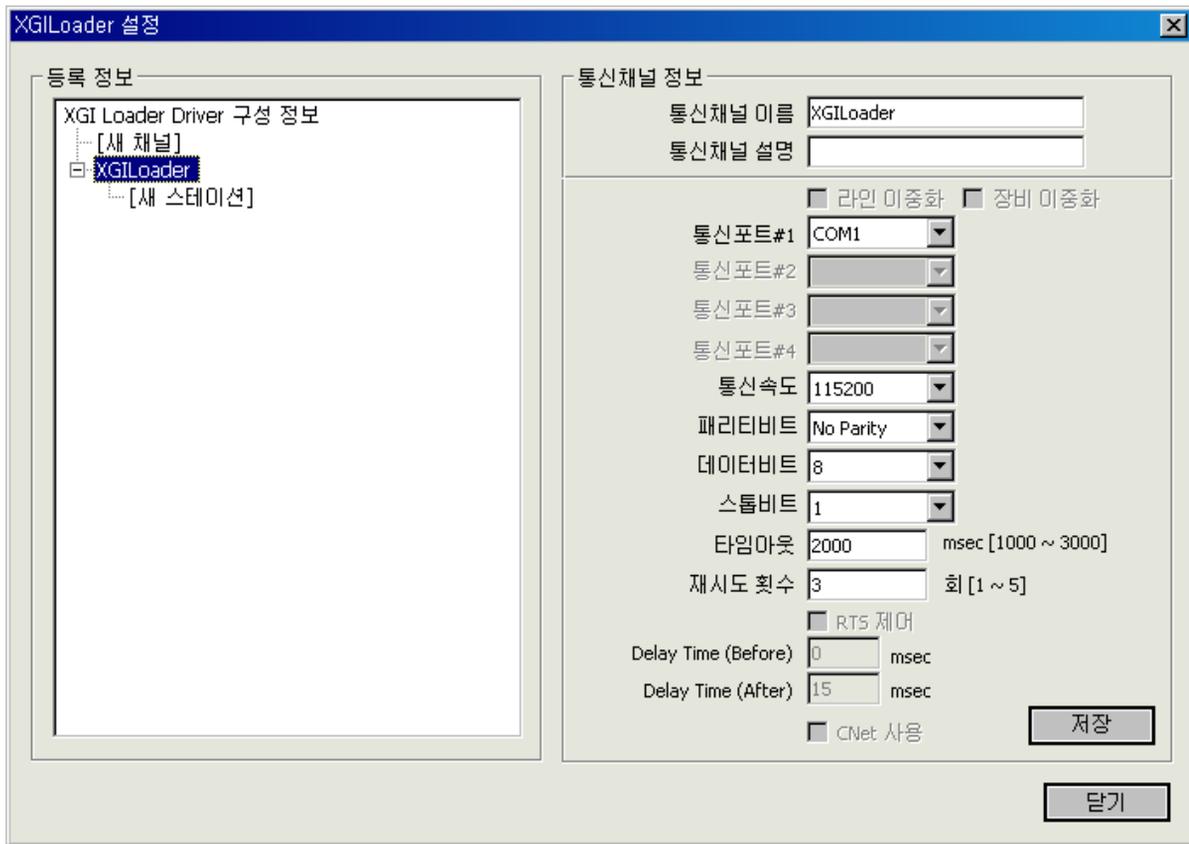
(1) 통신채널 추가



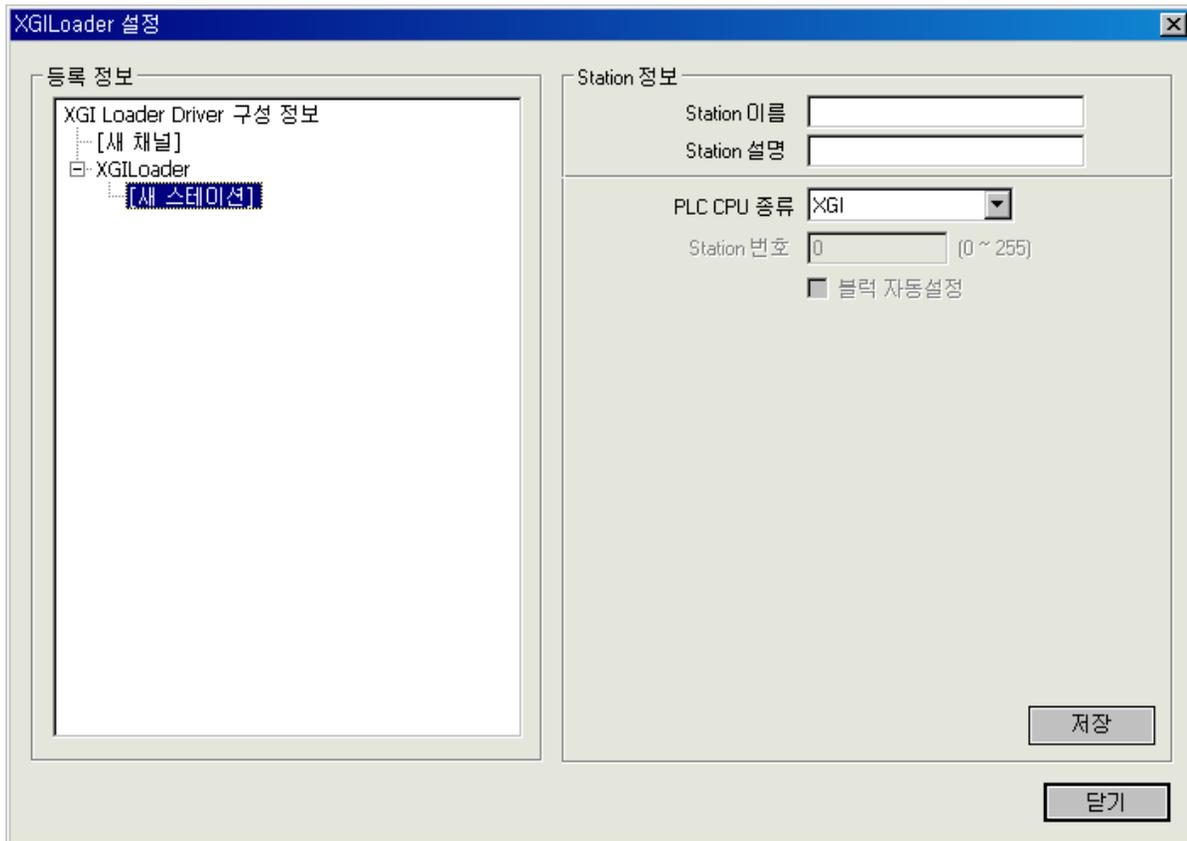
- IO 드라이버 리스트에서 “XGILoader” 을 선택한 후 “추가” 버튼을 누릅니다.



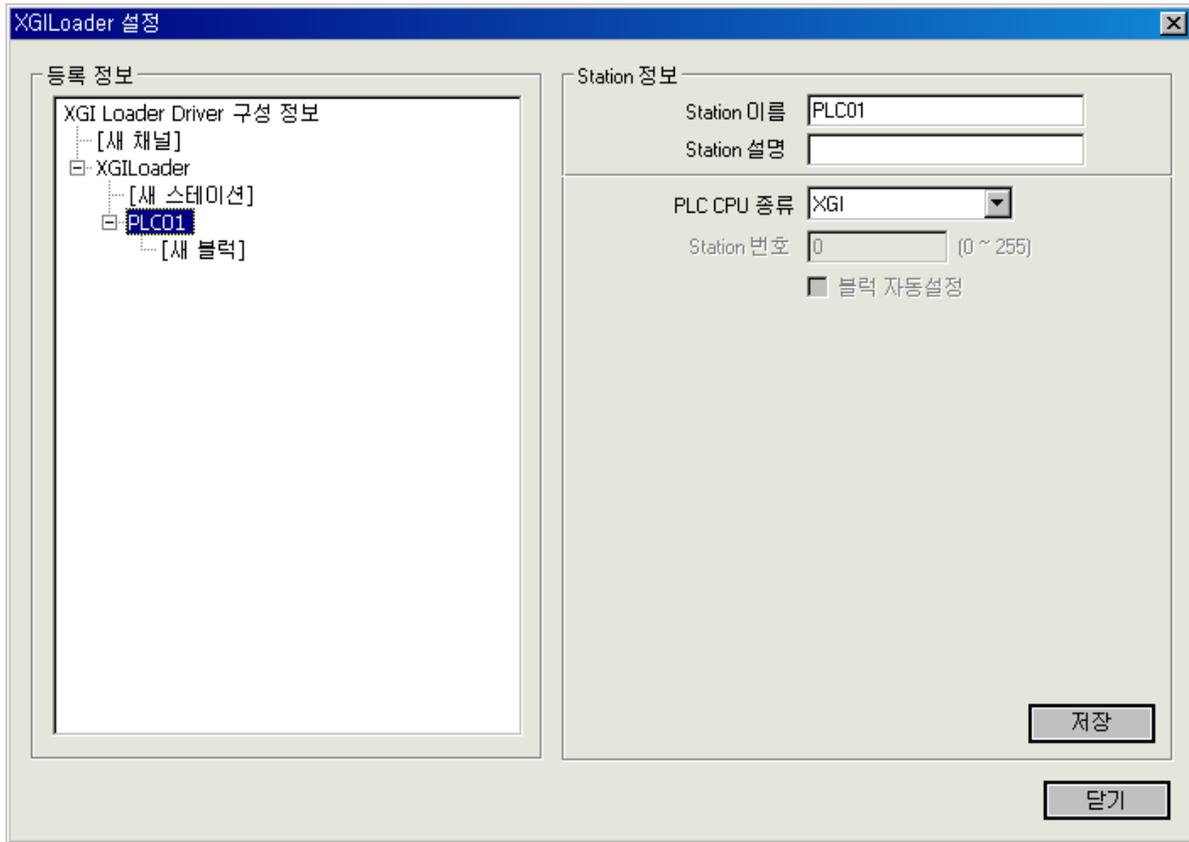
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다. PLC Loader 포트의 통신 속도는 115200 bps 입니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (f) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



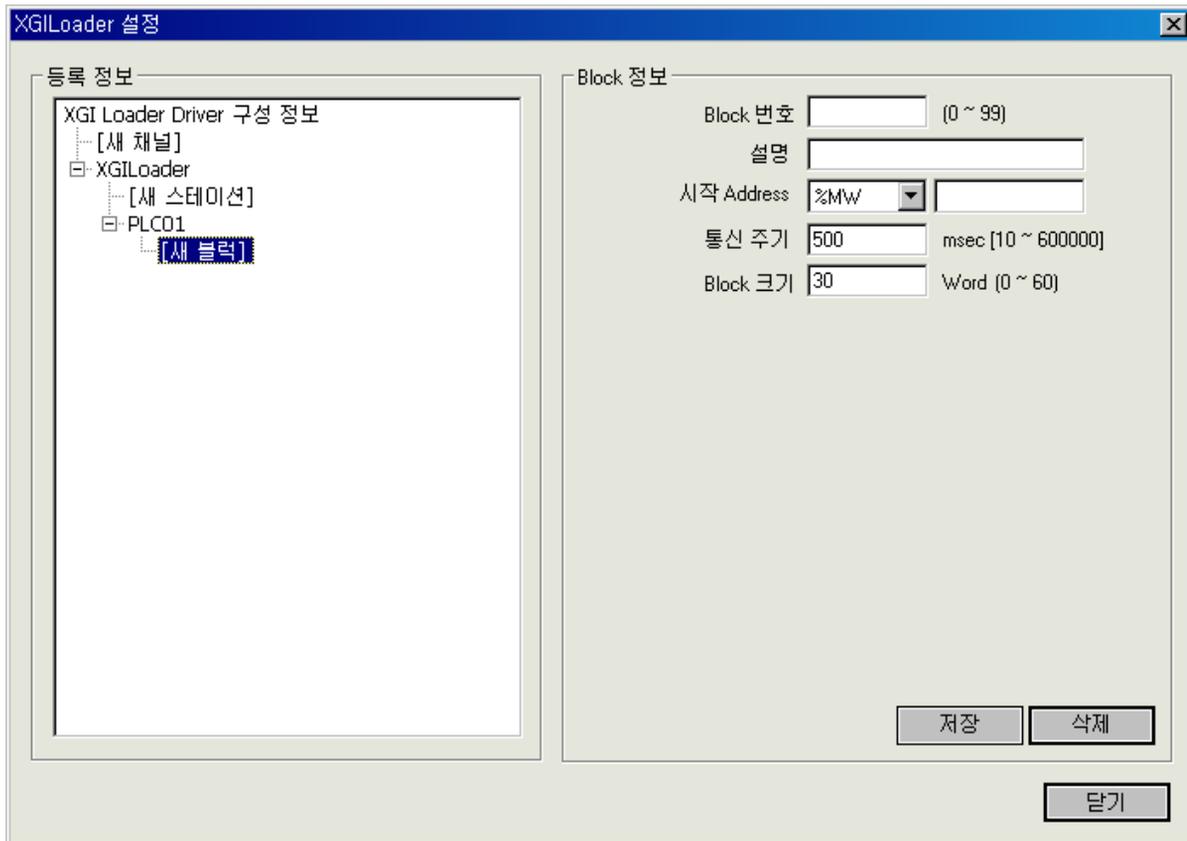
(2) Station 추가



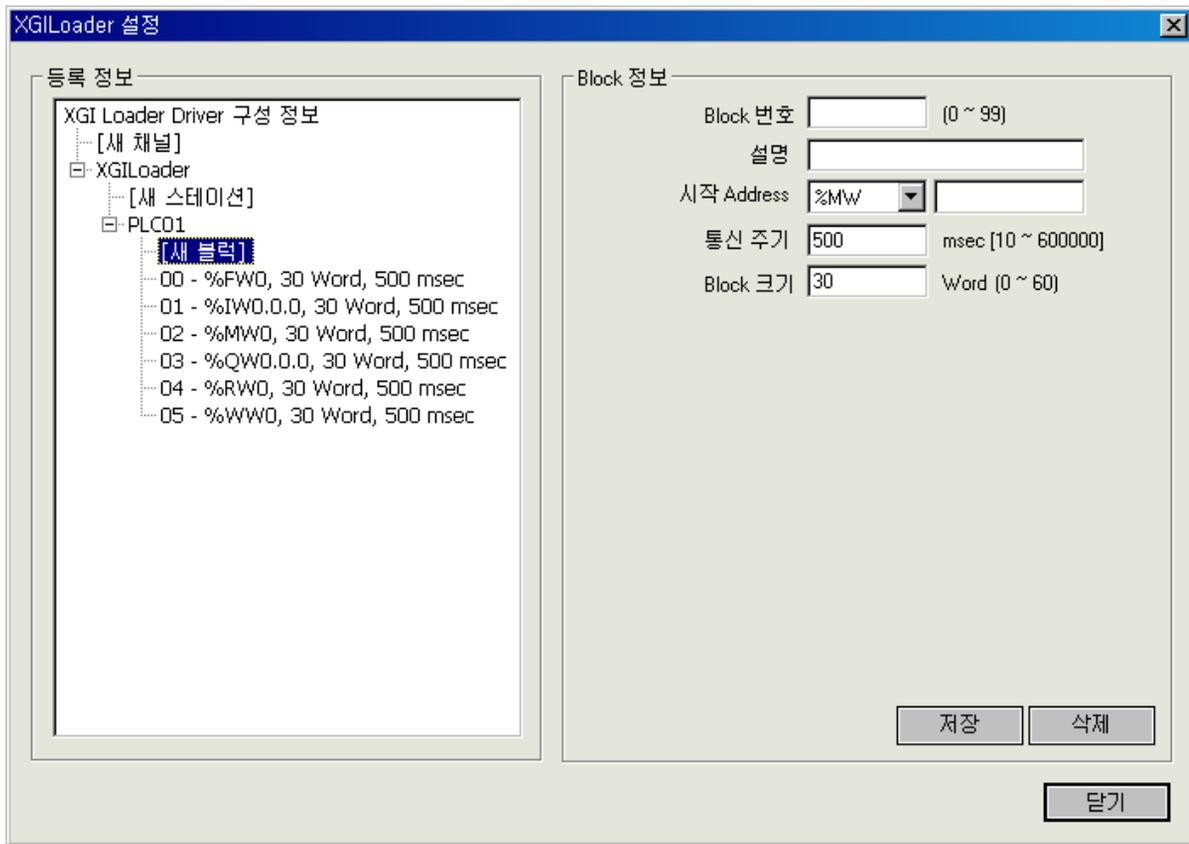
- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 6가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.
 - 올바른 예제: %MW0, %MW20, %IWO.0.0, %QW1.0.0
 - 잘못된 예제: %MW0.0.0, %IWO, %QWO
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 Block의 Block 크기를 Word(2 byte) 단위로 입력합니다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

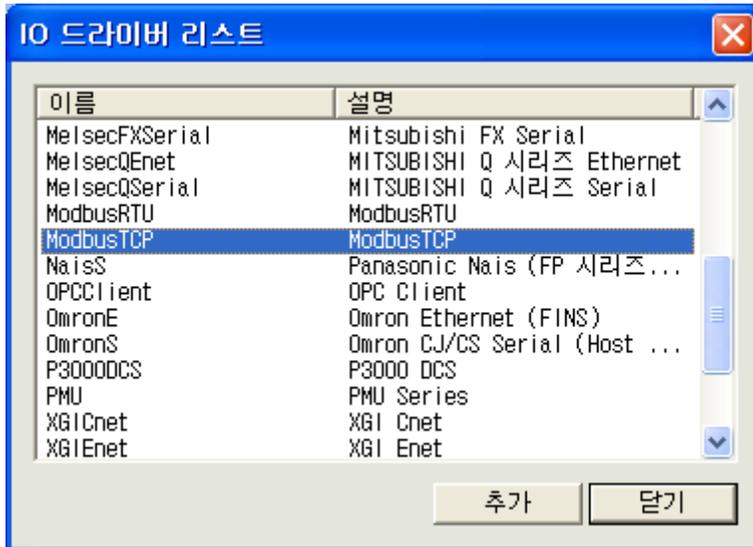
- . 아날로그 : %MX0, %MBO, %MNO, %MDO, %MLO, %IWO.0.0, %QWO.0.0
- . 디지털 : %MBO.0, %MNO.0, %MDO.0, %MLO.0, %IX0.0.0, %QX0.0.0

(b) 사용가능한 디바이스

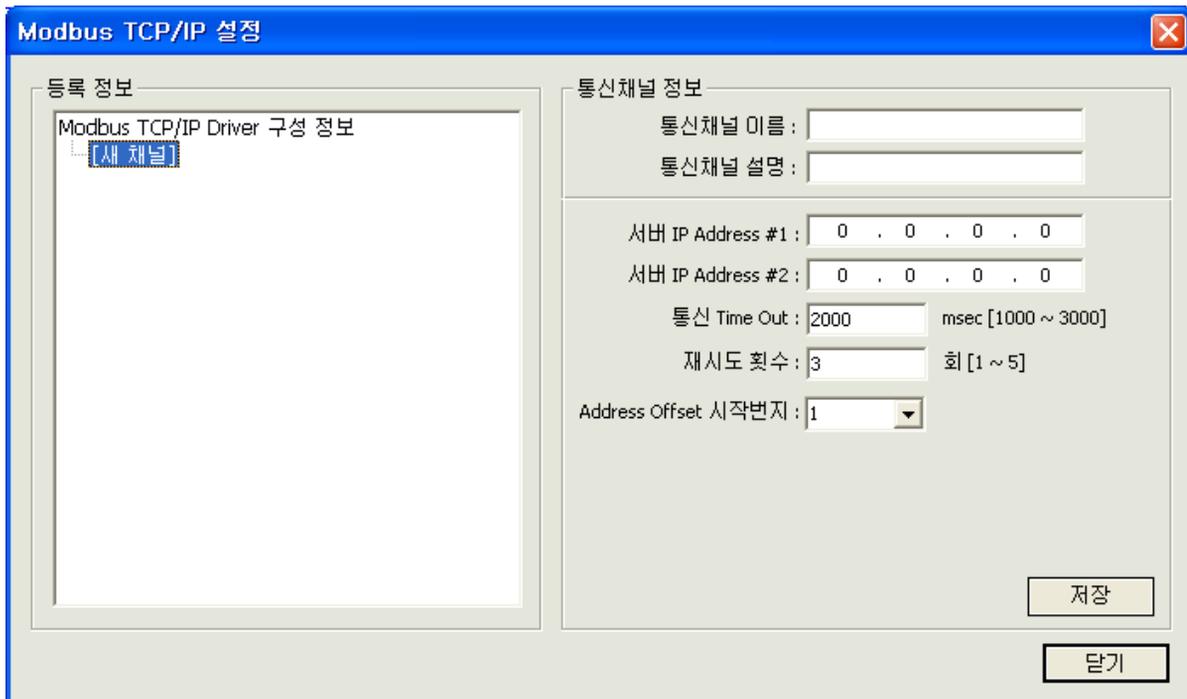
- . F, I, M, Q, R, W

25. ModbusTCP 드라이버

(1) 통신채널 추가



- IO 드라이버 리스트에서 “ModbusTCP” 을 선택한 후 “추가” 버튼을 누릅니다.

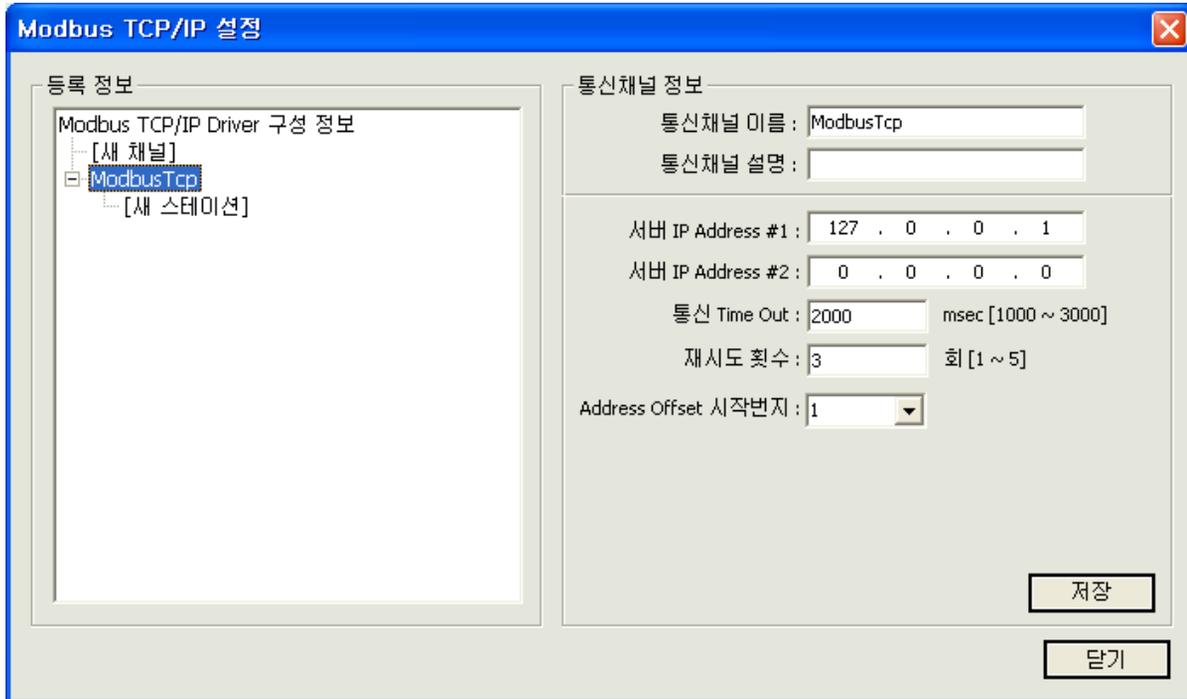


- 통신채널 이름 : 통신채널 이름을 입력합니다.
- 통신채널 설명 : 통신채널 설명을 입력합니다.
- 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- 서버 IP Address #2 : 라인 이중화를 사용할 경우 사용하게 될 두번째 IP Address를 입력합니다.
- 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는

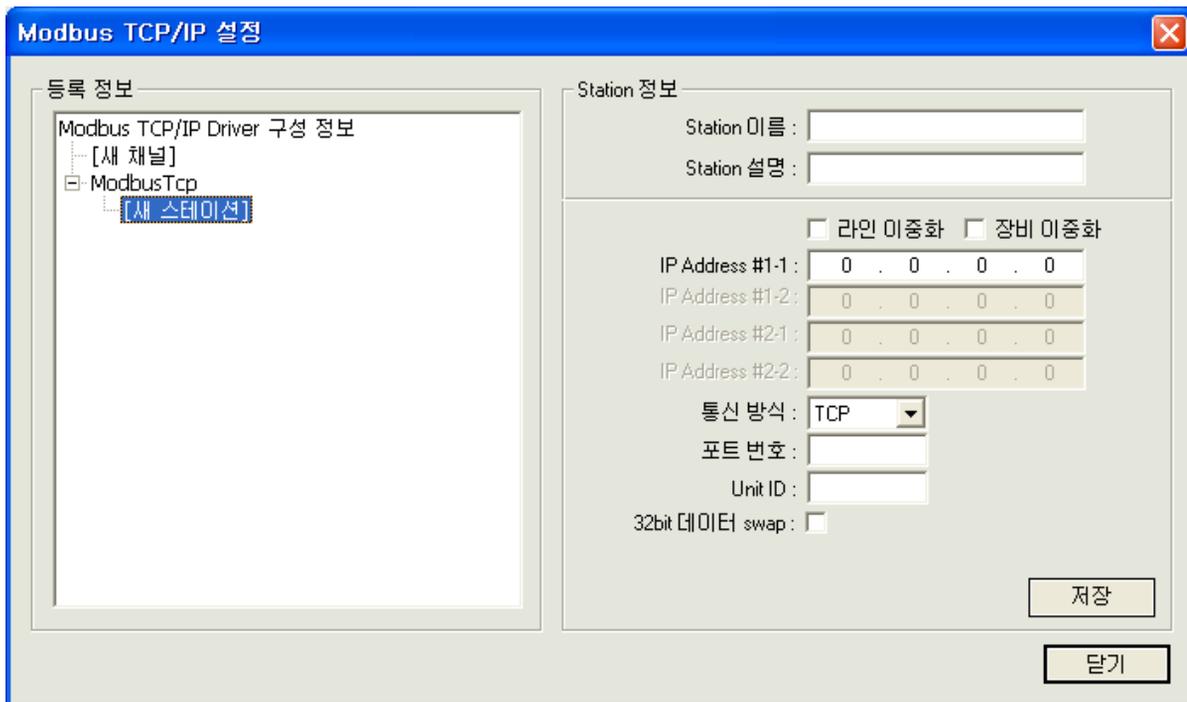
부록1 I/O 드라이버 설정방법

근거가 됩니다.

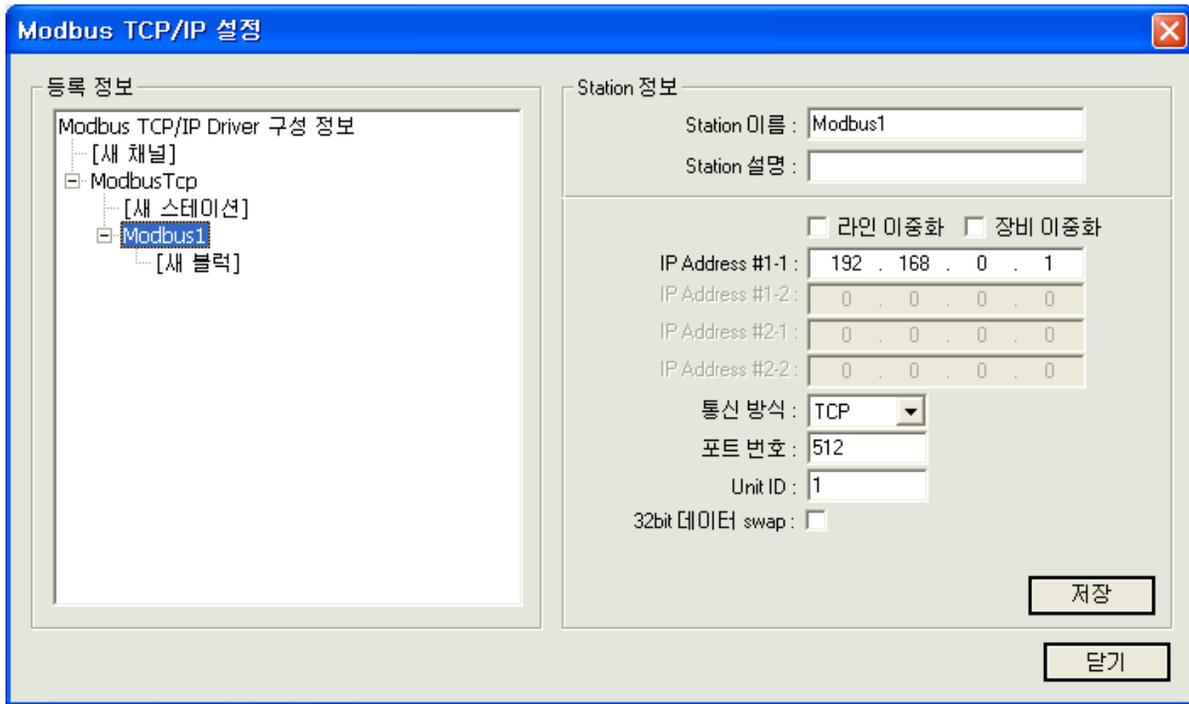
- (f) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (g) Address Offset 시작번지 : 0 또는 1을 가진다. 시작번지를 0번부터 또는 시작번지를 1부터 시작하는가의 여부를 설정한다.
- (h) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



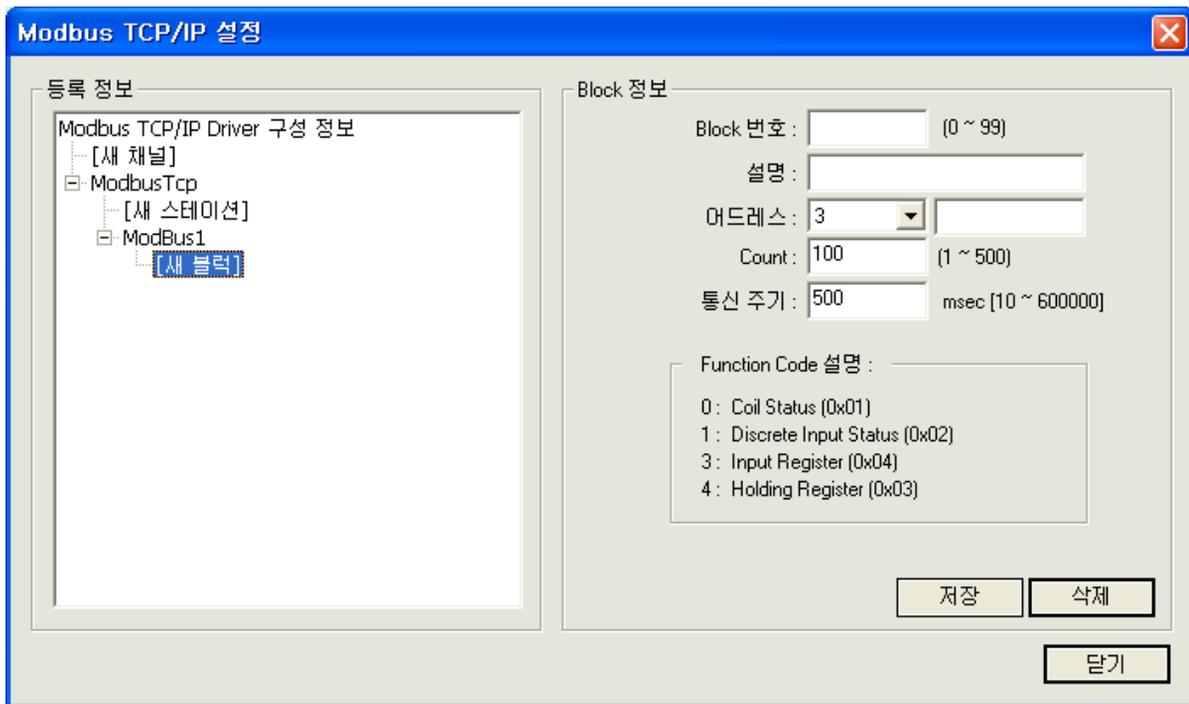
(2) Station 추가



- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (f) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다.
- (g) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
- (h) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.
- (i) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
- (j) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.
- (k) 통신 방식 : TCP와 UDP 중의 한가지를 선택합니다.
- (l) 포트 번호 : 통신 방식 선택에 따라 자동으로 입력됩니다.
- (m) UnitID : 통신하고자 하는 Modbus TCP 장비의 ID를 입력한다.
- (n) 32Bit데이터 Swap: 32Bit인 경우 [상위워드 + 하위워드] 또는 [하위워드 + 상위워드]의 조합으로 32Bit를 나타낼 수 있는데 이를 위하여 상위워드와 하위워드 Swap을 사용한다.
- (o) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.

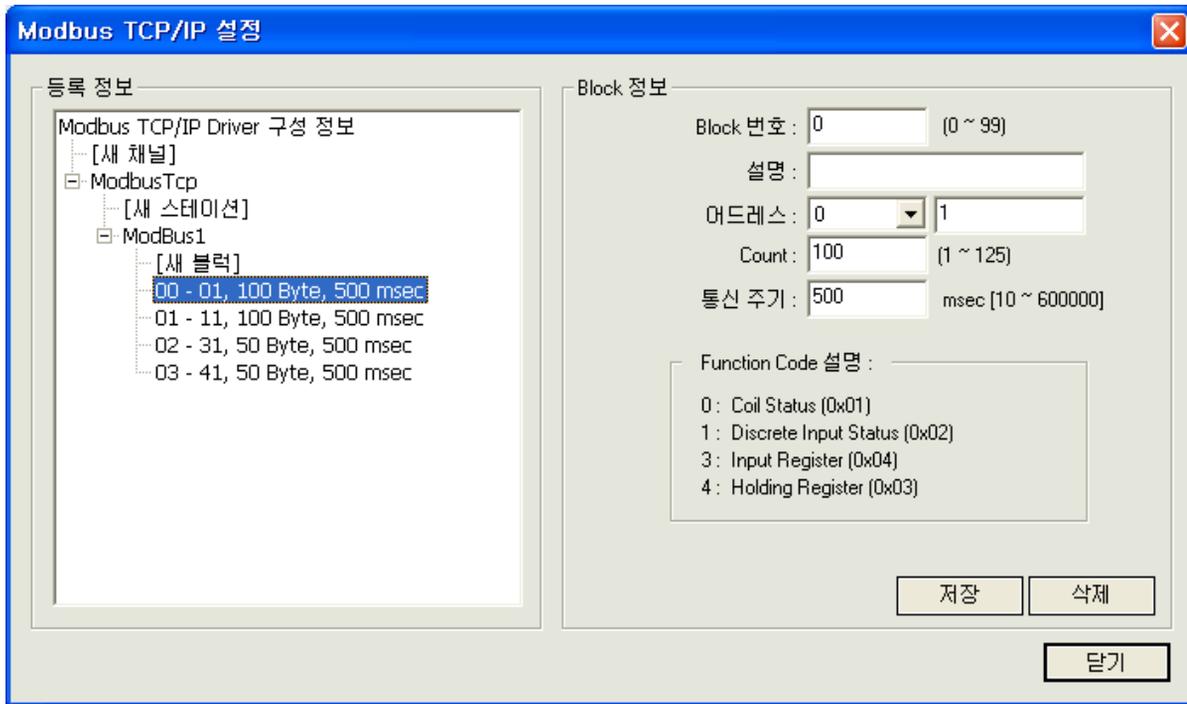


(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호 입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 4가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.

- . Address 구별자 0(Coil Status), 1(Discrete Input Status), 3(Input Register), 4(Holding Register)
- . Address는 0-65535 이며, 만약, 채널정보에서 Address Offset 시작번지를 1로 설정하였다면, 1-65536
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 구별자별 읽고자하는 갯수
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

- . 아날로그 : 구별자(3 또는 4) + 0~65535 (만약, 채널정보에서 Address Offset 시작주소를 1로 설정하였다면, 1~65536)

예1) 만약, 3번(Input Register : 읽기만 가능)의 0번지(0~65535중 0)를 읽고 싶다면
 Address Offset 시작주소를 0으로 설정하였다면, 300000 또는 30으로 설정
 Address Offset 시작주소를 1으로 설정하였다면, 300001 또는 31으로 설정

예2) 만약, 4번(Holding Register)의 17번지(0~65535중 17)를 읽고 싶다면
 Address Offset 시작주소를 0으로 설정하였다면, 400017 또는 417으로 설정
 Address Offset 시작주소를 1으로 설정하였다면, 400018 또는 418으로 설정

- . 디지털 : 구별자(0 또는 1 또는 3 또는 4) + 0~65535 (만약, 채널정보에서 Address Offset 시작주소를 1로 설정하였다면, 1~65536)

예1) 만약, 0번(Coil Status : 읽기/쓰기 가능)의 0번지(0~65535중 0)를 읽고 싶다면
 Address Offset 시작주소를 0으로 설정하였다면, 000000 또는 00으로 설정
 Address Offset 시작주소를 1으로 설정하였다면, 000001 또는 01으로 설정

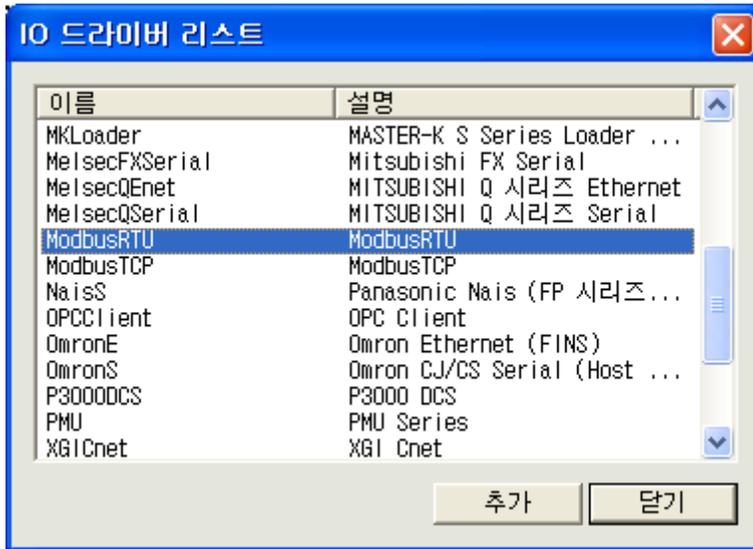
예2) 만약, 1번(Discrete Input Status : 읽기만 가능)의 17번지(0~65535중 17)를 읽고 싶다면
Address Offset 시작주소를 0으로 설정하였다면, 100017 또는 117으로 설정
Address Offset 시작주소를 1으로 설정하였다면, 100018 또는 118으로 설정

예3) 만약, 3번(Input Register : 읽기만 가능)의 17번지(0~65535중 17)의 10번째 Bit를 읽고 싶다면
Address Offset 시작주소를 0으로 설정하였다면, 300017.A 또는 317.A으로 설정
Address Offset 시작주소를 1으로 설정하였다면, 300018.A 또는 318.A으로 설정

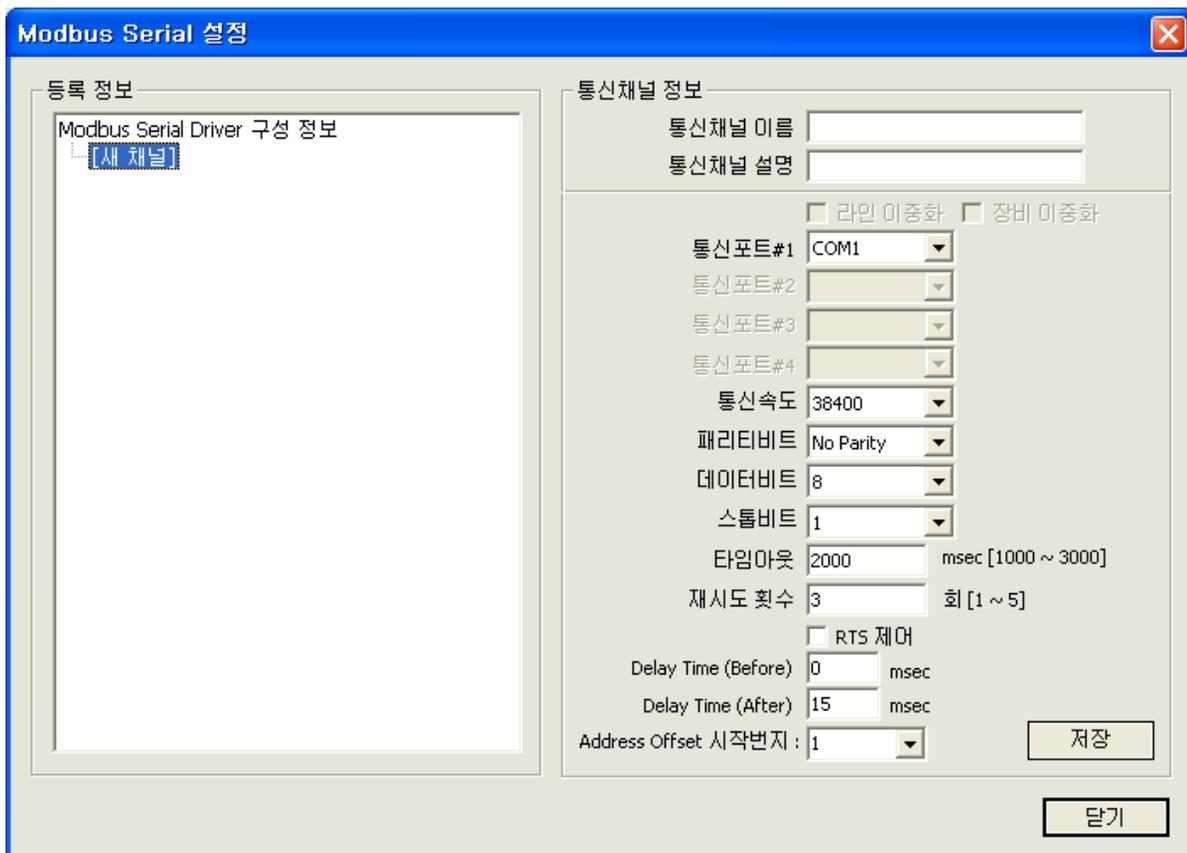
예4) 만약, 4번(Holding Register : 읽기만 가능)의 0번지(0~65535중 17)의 15번째 Bit를 읽고 싶다면
Address Offset 시작주소를 0으로 설정하였다면, 300000.F 또는 30.F으로 설정
Address Offset 시작주소를 1으로 설정하였다면, 300000.F 또는 31.F으로 설정

26. ModbusRTU 드라이버

(1) 통신채널 추가



- IO 드라이버 리스트에서 “ModbusRTU” 을 선택한 후 “추가” 버튼을 누릅니다.

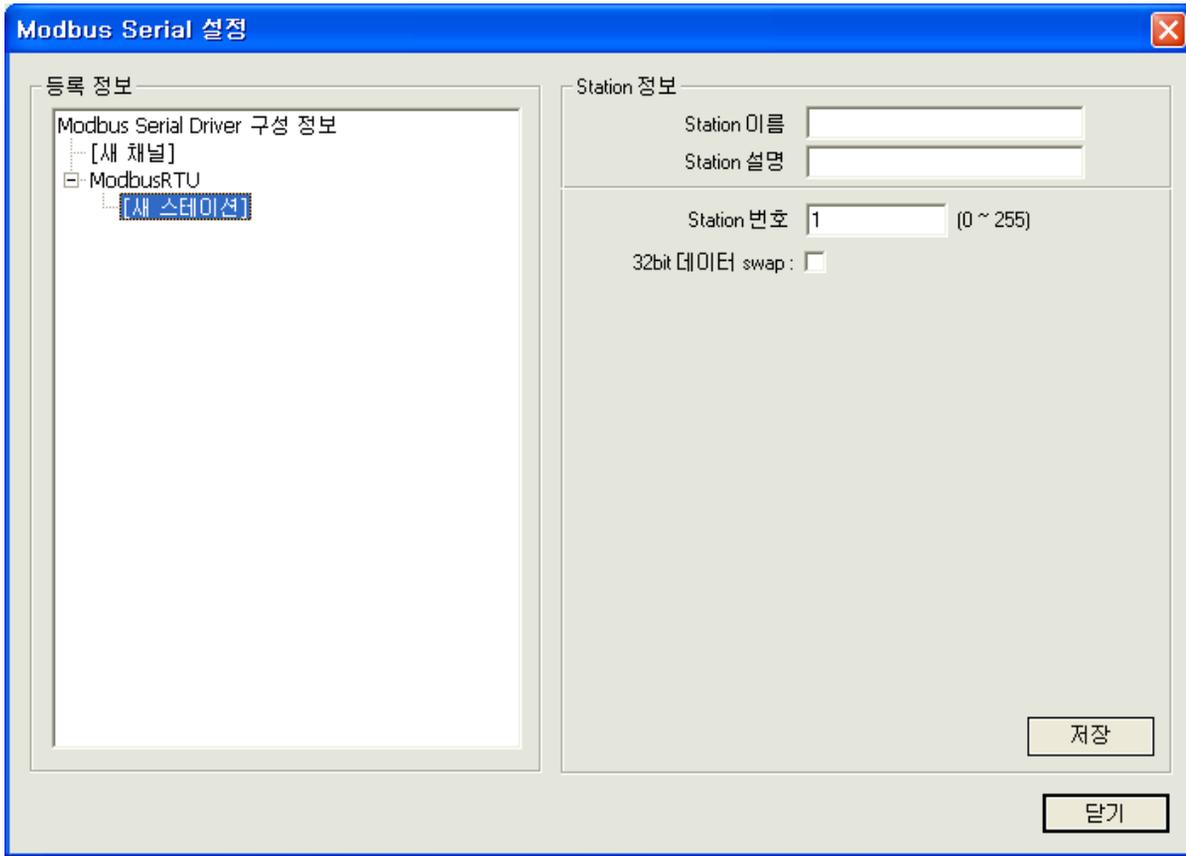


- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.

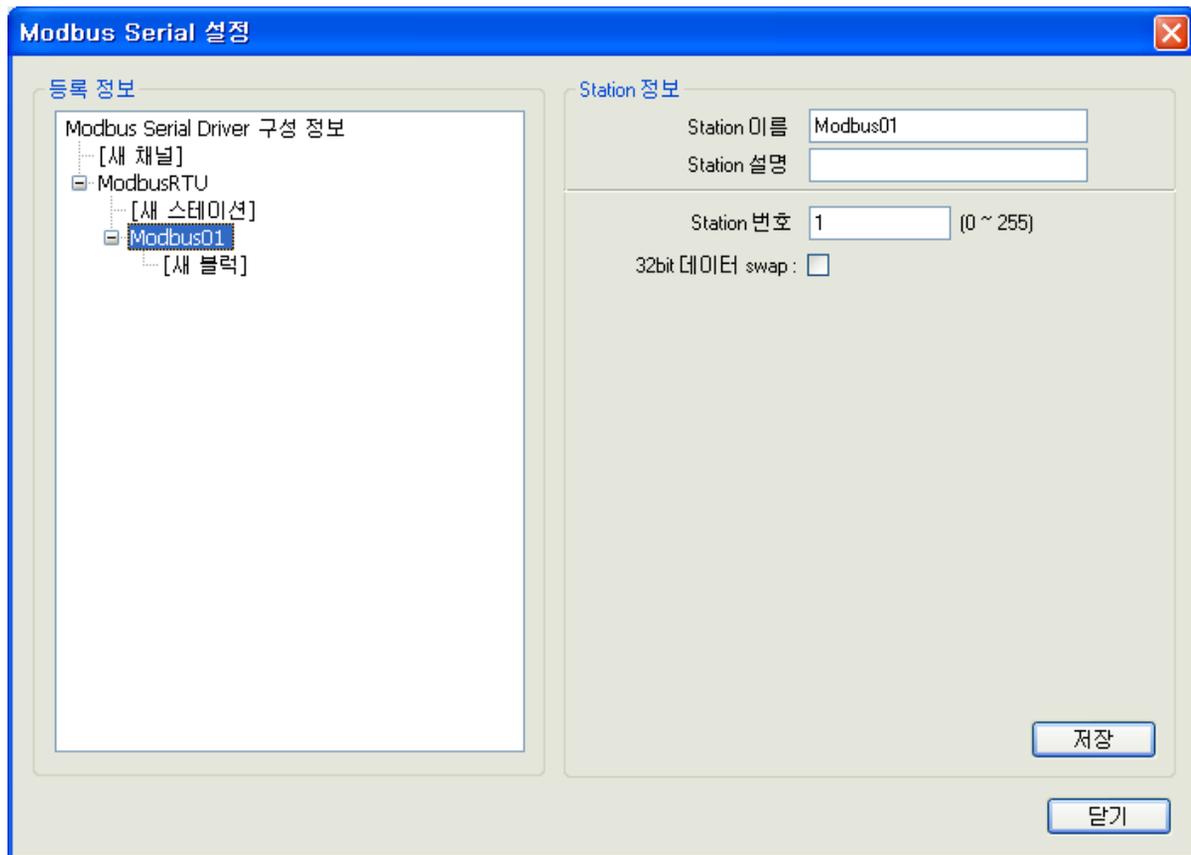
부록1 I/O 드라이버 설정방법

- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (f) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) RTS 제어 : 485, 422 통신을 사용할 경우, 사용함으로 체크를 합니다.
- (k) Delay Time (Before) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 하기 직전의 Delay Time 입니다.
- (l) Delay Time (After) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 한 직후의 Delay Time 입니다.
- (m) Address Offset 시작번지 : 0 또는 1을 가진다. 시작번지를 0번부터 또는 시작번지를 1부터 시작하는가의 여부를 설정한다.
- (n) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.

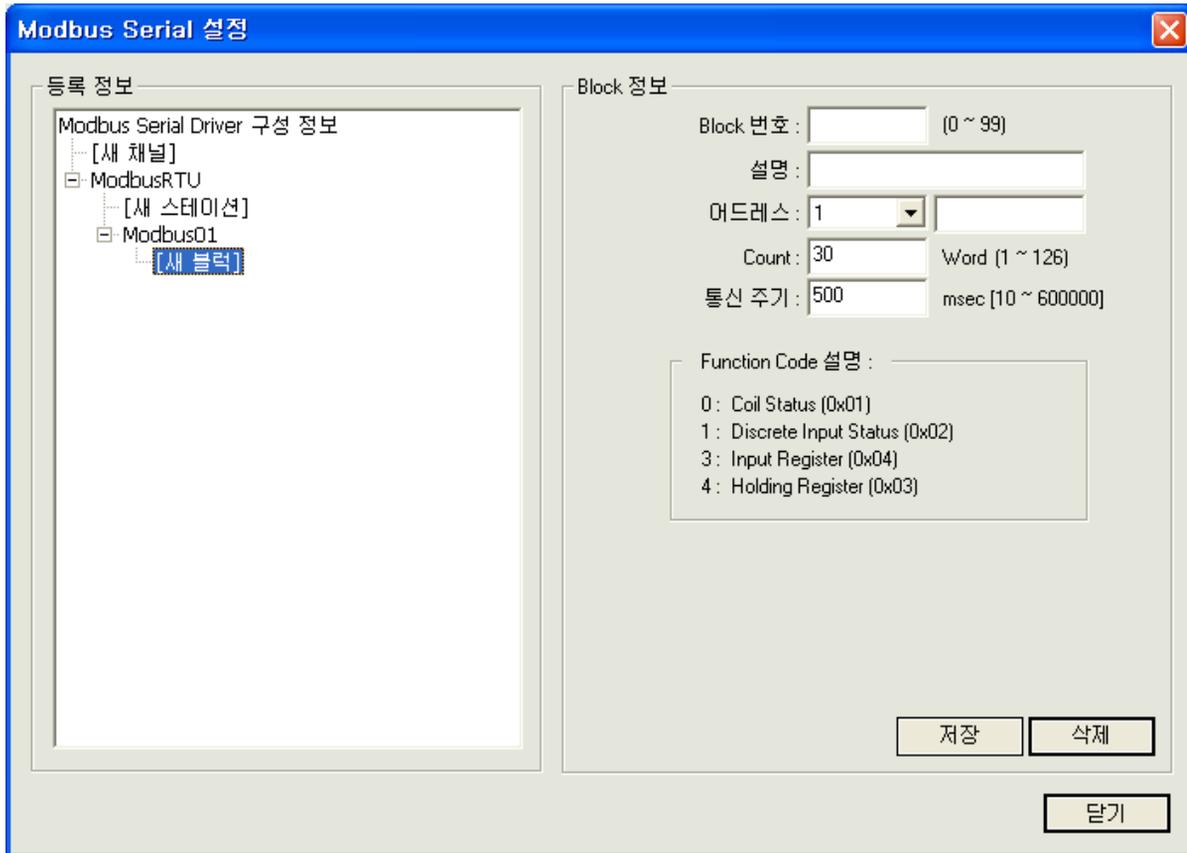
(2) Station 추가



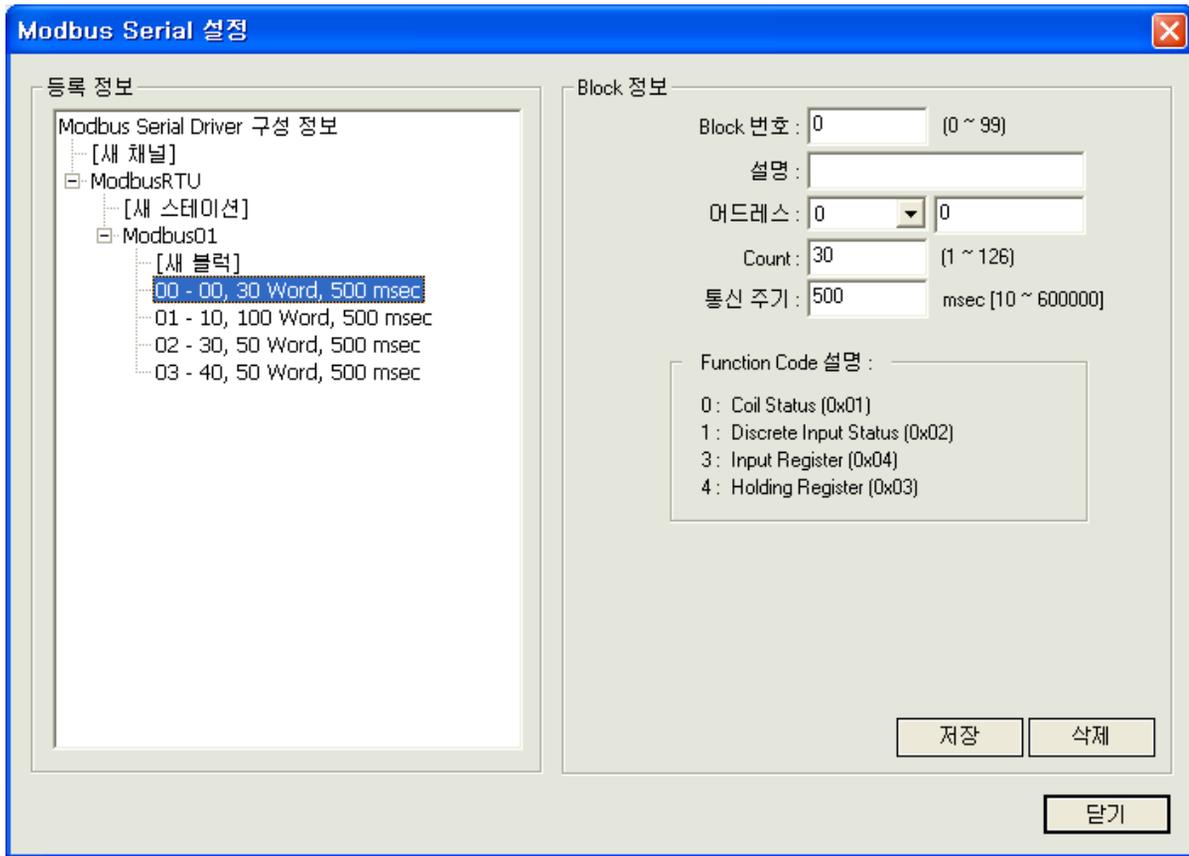
- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) Station 번호 : PLC Cnet 모듈의 국번을 입력합니다.
- (f) 32Bit데이터 Swap: 32Bit인 경우 [상위워드 + 하위워드] 또는 [하위워드 + 상위워드]의 조합으로 32Bit를 나타낼 수 있는데 이를 위하여 상위워드와 하위워드 Swap을 사용한다.
- (g) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다. 4가지 종류가 있으며, 각각 아래와 같은 방법으로 Address를 지정합니다.
 - . Address 구별자 0(Coil Status), 1(Discrete Input Status), 3(Input Register), 4(Holding Register)
 - . Address는 0~65535 이며, 만약, 채널정보에서 Address Offset 시작번지를 1로 설정하였다면, 1~65536
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 구별자별 읽고자하는 갯수
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

- 아날로그 : 구별자(3 또는 4) + 0~65535 (만약, 채널정보에서 Address Offset 시작주소를 1로 설정하였다면, 1~65536)

예1) 만약, 3번(Input Register : 읽기만 가능)의 0번지(0~65535중 0)를 읽고 싶다면
 Address Offset 시작주소를 0으로 설정하였다면, 30000 또는 30으로 설정
 Address Offset 시작주소를 1으로 설정하였다면, 30001 또는 31으로 설정

예2) 만약, 4번(Holding Register)의 17번지(0~65535중 17)를 읽고 싶다면
 Address Offset 시작주소를 0으로 설정하였다면, 40017 또는 417으로 설정
 Address Offset 시작주소를 1으로 설정하였다면, 40018 또는 418으로 설정

- 디지털 : 구별자(0 또는 1 또는 3 또는 4) + 0~65535 (만약, 채널정보에서 Address Offset 시작주소를 1로 설정하였다면, 1~65536)

예1) 만약, 0번(Coil Status : 읽기/쓰기 가능)의 0번지(0~65535중 0)를 읽고 싶다면
 Address Offset 시작주소를 0으로 설정하였다면, 00000 또는 00으로 설정
 Address Offset 시작주소를 1으로 설정하였다면, 00001 또는 01으로 설정

예2) 만약, 1번(Discrete Input Status : 읽기만 가능)의 17번지(0~65535중 17)를 읽고 싶다면
 Address Offset 시작주소를 0으로 설정하였다면, 10017 또는 117으로 설정

Address Offset 시작주소를 1으로 설정하였다면, 100018 또는 118으로 설정

예3) 만약, 3번(Input Register : 읽기만 가능)의 17번지(0~65535중 17)의 10번째 Bit를 읽고 싶다면

Address Offset 시작주소를 0으로 설정하였다면, 300017.A 또는 317.A으로 설정

Address Offset 시작주소를 1으로 설정하였다면, 300018.A 또는 318.A으로 설정

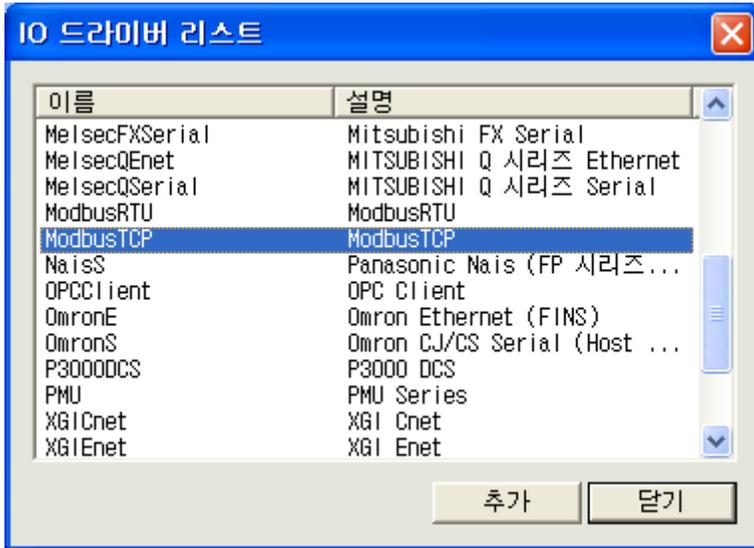
예4) 만약, 4번(Holding Register : 읽기만 가능)의 0번지(0~65535중 17)의 15번째 Bit를 읽고 싶다면

Address Offset 시작주소를 0으로 설정하였다면, 300000.F 또는 30.F으로 설정

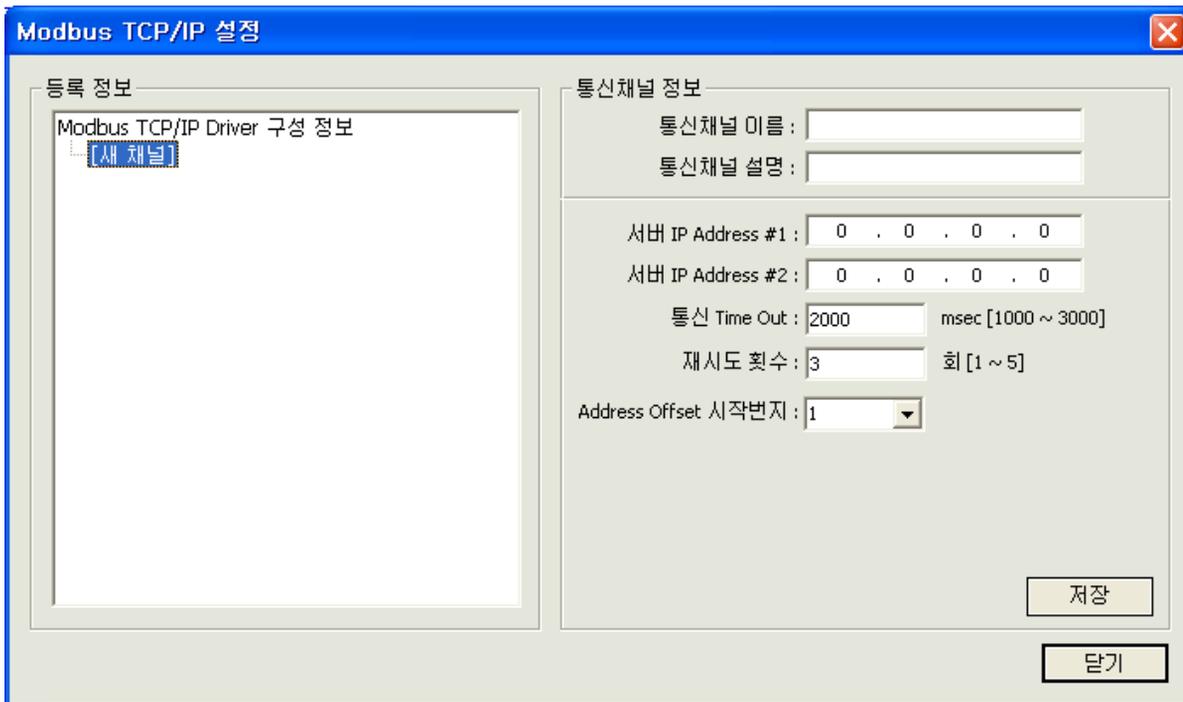
Address Offset 시작주소를 1으로 설정하였다면, 300000.F 또는 31.F으로 설정

27. OmronE 드라이버

(1) 통신채널 추가



- IO 드라이버 리스트에서 “OmronE” 을 선택한 후 “추가” 버튼을 누릅니다.

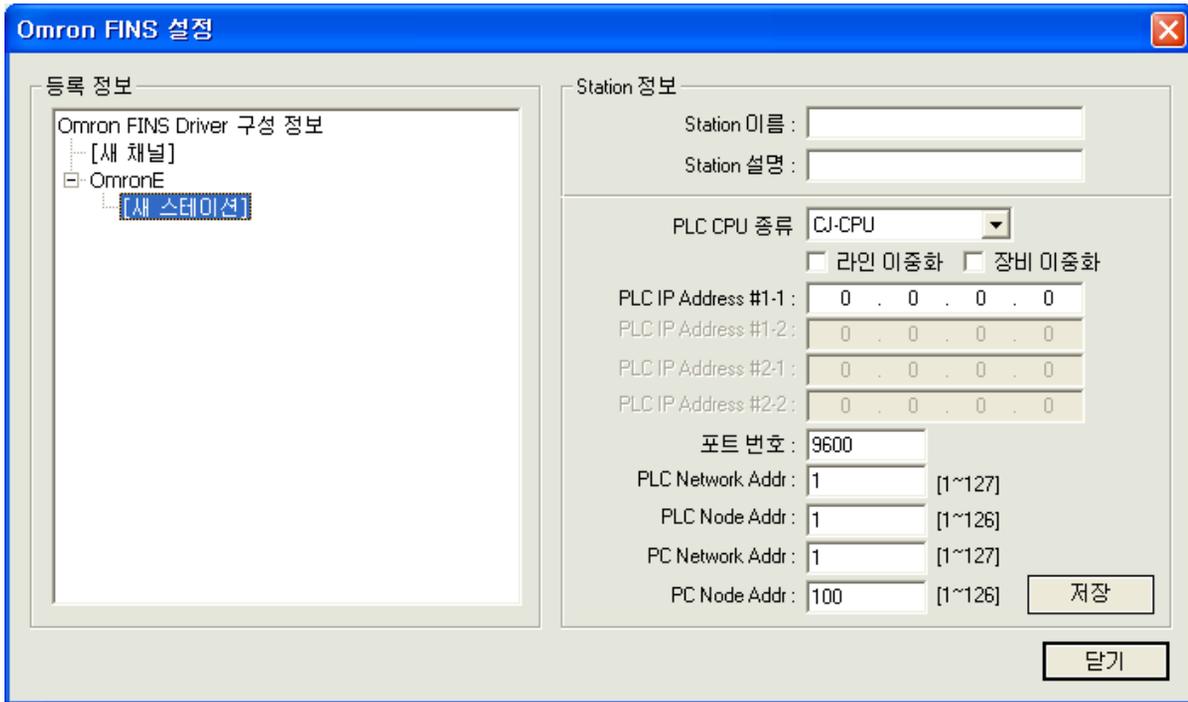


- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- (d) 서버 IP Address #2 : 라인 이중화를 사용할 경우 사용하게 될 두번째 IP Address를 입력합니다.

- (e) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (f) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (g) Address Offset 시작번지 : 0 또는 1을 가진다. 시작번지를 0번부터 또는 시작번지를 1부터 시작하는가의 여부를 설정한다.
- (h) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



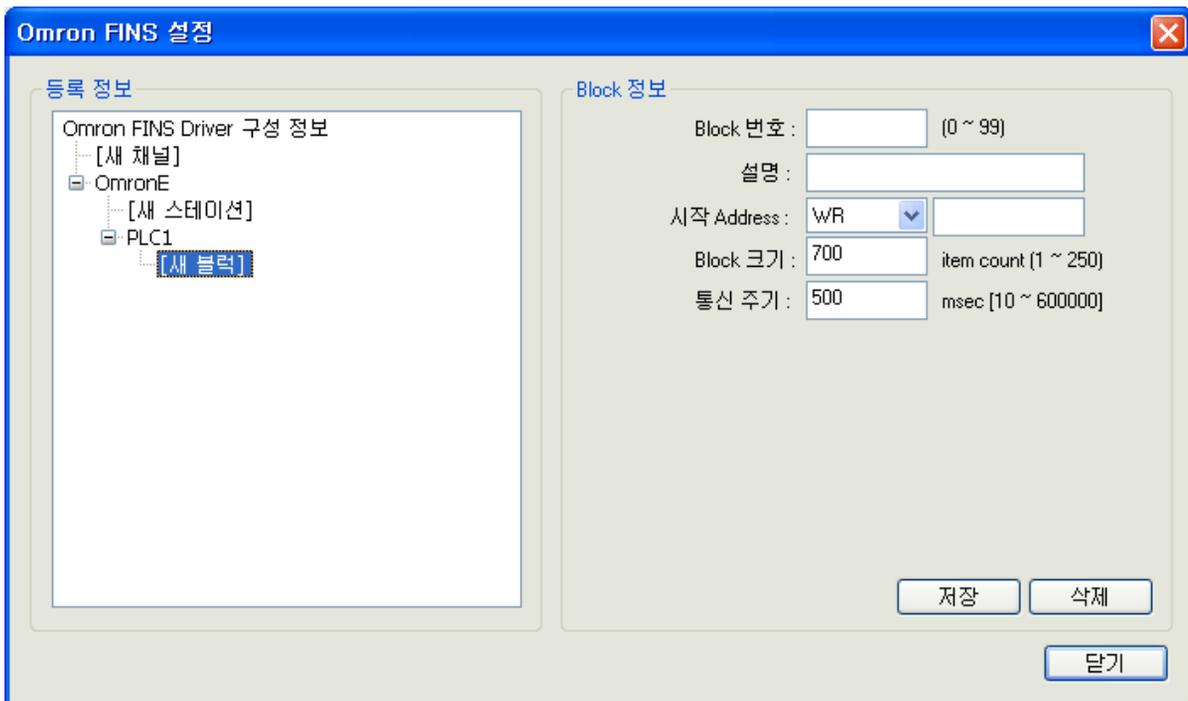
(2) Station 추가



- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (f) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다.
- (g) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
- (h) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.
- (i) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
- (j) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.
- (k) 통신 방식 : UDP 를 사용하므로 화면상에는 표시하지 않았다.
- (l) 포트 번호 : FINS는 공장출하시 Default로 9600을 사용하며, 다른 포트로 설정하고 싶다면, PLC설정값을 바꾸어 사용하여도 무방하다.
- (m) PLC Network Addr : 통신하고자 하는 Omron PLC 장비의 Network Address 를 입력한다. Default ‘1’
- (n) PLC Node Addr : 통신하고자 하는 Omron PLC 장비의 Node Address를 입력한다. Default ‘1’
- (o) PC Network Addr : 통신하고 있는 PC(InfoU PC)의 Network Address를 입력한다. Default ‘1’
- (p) PC Node Addr : 통신하고 있는 PC(InfoU PC)의 Node Address를 입력한다. Default ‘100’
- (q) UnitID : 0-F중 설정 한다. (국번)
- (r) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다.
 - 시작Address : 메모리타입(C10,WR,HR,AR,TF,CF,EMO)+시작번지

- Block 크기 : 읽어올 갯수

예제1) 시작Address : C1010, Block 크기 : 100 이라 하면, C10 메모리(0부터 시작)의 10번째 WORD부터 100개의 WORD를 읽어온다.

예제2) 시작Address : TF20, Block크기 : 200이라 하면, TF 메모리(0부터 시작)의 20번째 상태 Bit부터 20개의 상태 Bit를 읽어온다.

※ C10, WR, HR, AR, TIM, CNT, DM, EM, EMO, EM1, EM2, EM3, EM4, EM5, EM6, EM7, EM8, EM9, EMA, EMB, EMC, EM_CURR, DR 은 WORD단위의 디바이스 이며, IR은 DWORD단위의 디바이스 이다.

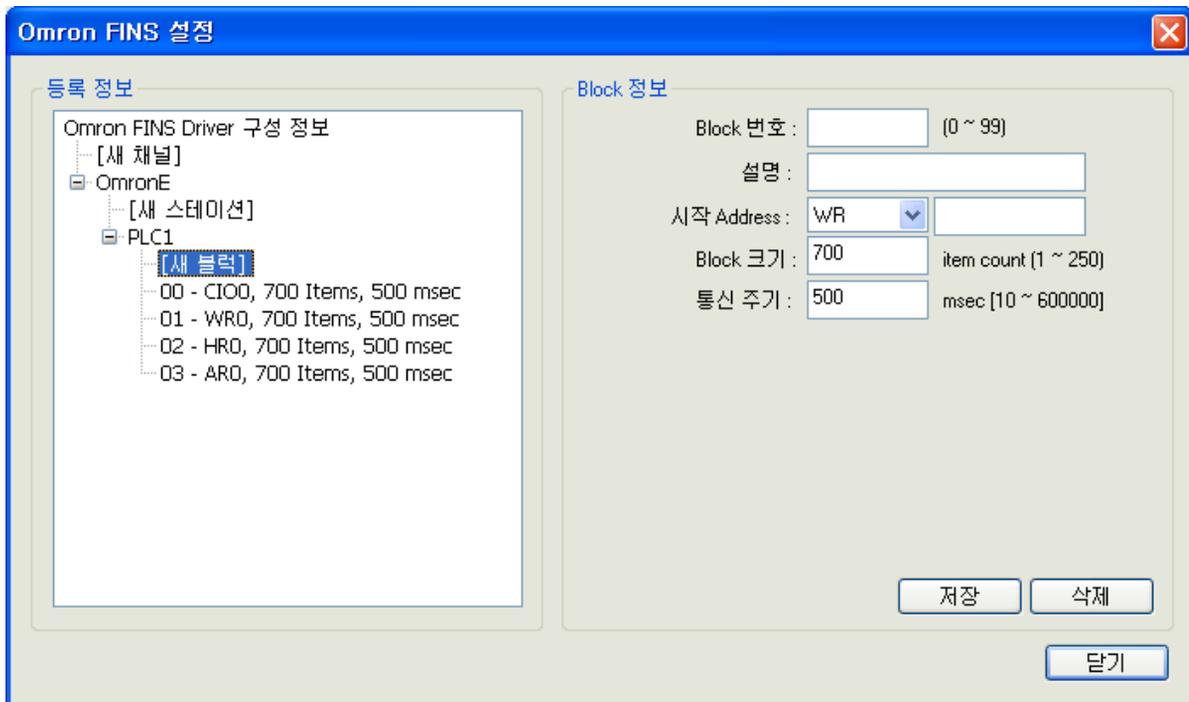
TKF, TKS, TF, CF 는 Bit단위의 디바이스 이다.

(e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.

(f) Block 크기 : 해당 구별자별 읽고자하는 갯수

(g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.

(h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

- 아날로그 입력Address 사용예제

C10_100 : C10영역의 100번째 Word값을 가져온다

EM7_20000 : EM7영역의 20000번째 Word값을 가져온다.

- 디지털 입력Address 사용예제

C10_90A : C10영역의 90번째 WORD의 10(A)번째 Bit 값을 가져온다.

EM7_07 : EM7영역의 0번째 WORD의 7번째 Bit 값을 가져온다.

TF_123 : Timer Flag영역의 123번째 타이머 Completion Flag 값을 가져온다.

- Omron PLC 메모리 디바이스 맵

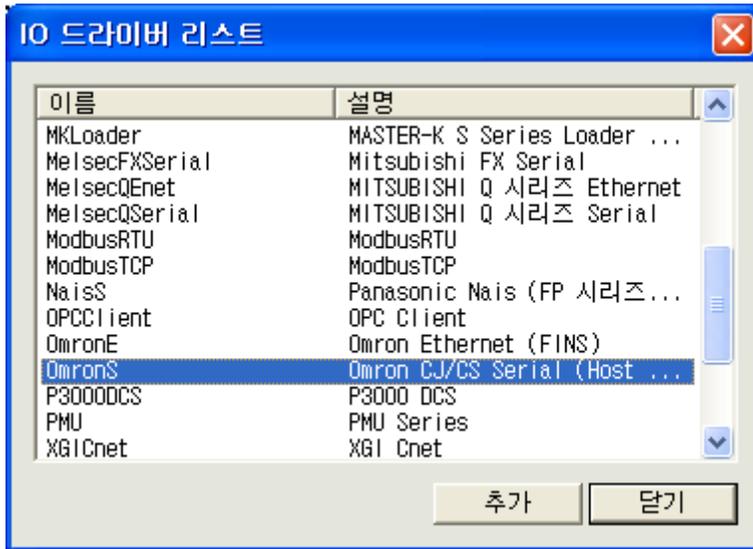
메모리코드	특성	데이터 형식	CS/CJ CPU 메모리번지	CV CPU 메모리번지	비고	어드레스 적용예제
CIO	CIO Area	WORD	0~6143	0~2555		AI : CIO_xxxx (xxxx 는 0~6143 의 십진수) CIO_100 DI : CIO_xxxx0~F(xxxx 는 0~6143 의 십진수) CIO_100A
WR	Work Area		0~511			AI : WR_xxxx (xxxx 는 0~511 의 십진수) WR_100 DI : WR_xxxx0~F(xxxx 는 0~511 의 십진수) WR_0F
HR	Holding Bit Area					AI : HR_xxxx (xxxx 는 0~511 의 십진수) HR_234 DI : HR_xxxx0~F(xxxx 는 0~511 의 십진수) HR_19
AR	Auxiliary Bit Area		0~959	0~959	0~447(읽기전용), 448~959(읽기/쓰기)	AI : AR_xxxx (xxxx 는 0~511 의 십진수) AR_800 DI : AR_xxxx0~F(xxxx 는 0~511 의 십진수) AR_9001
TF	Timer Area	Completion Flag(BIT)	0~4095	0~2047 또는 0~1023	읽기전용	DI : TF_xxxx(xxxx 는 0~4095 의 십진수), TF_2047
CF	Counter Area					DI : CF_xxxx(xxxx 는 0~4095 의 십진수), CF_2048
TIM	Timer Area PV	WORD	0~4095	0~2047 또는 0~1023		AI : TIM_xxxx (xxxx 는 0~4095 의 십진수) TIM_1000
CNT	Counter Area PV				AI : CNT_xxxx (xxxx 는 0~4095 의 십진수) CNT_2000	
DM	DM Area	WORD	0~32767	0~32767		AI : DM_xxxxx (xxxxx 는 0~32767 의 십진수) DM_10000 DI : DM_xxxxx0~F(xxxxxx 는 0~32767 의 십진수) DM_900E
EM0~EM7	EM Area					AI : EM0_xxxxx (xxxxx 는 0~32767 의 십진수) EM0_9000 DI : EM0_xxxxx0~F(xxxxxx 는 0~32767 의 십진수) EM0_7000A
EM7~EMC					AI : EMA_xxxxx (xxxxx 는 0~32767 의 십진수) EMA_9000 DI : EMA_xxxxx0~F(xxxxxx 는 0~32767 의 십진수) EMA_7000A	

부록1 I/O 드라이버 설정방법

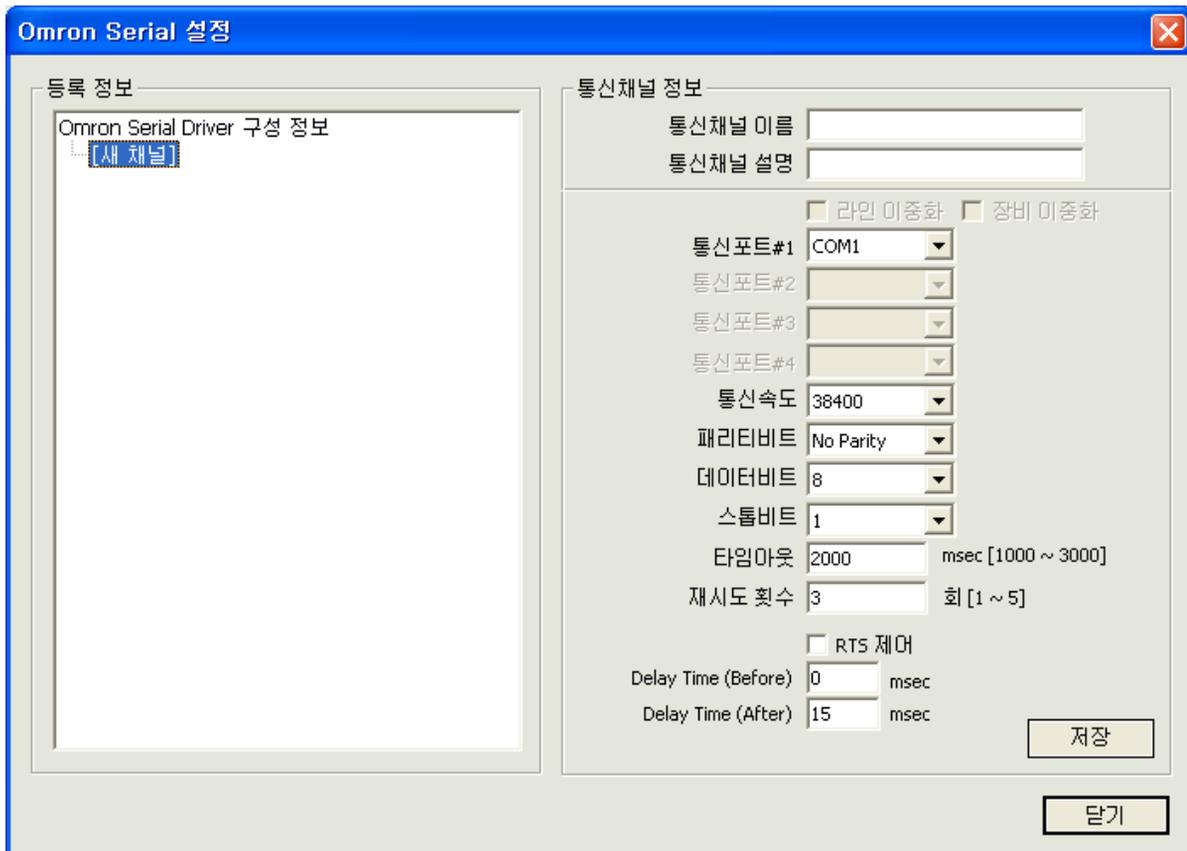
메모리코드	특성	데이터 형식	CS/CJ CPU 메모리번지	CV CPU 메모리번지	비고	어드레스 적용예제
EM						AI : EM_xxxxx (xxxxx 는 0~32767의 십진수) EM_9000 DI : EM_xxxxx0~F(xxxxx 는 0~32767의 십진수) EM_7000A
EM_CURR			0번지 1개 (현재 Bank No)			AI : EM_0
TKF	Task Flag BIT	BIT	0~31		읽기전용	DI : TKF_xxxx(xxxx 는 0~31의 십진수), TKF_21
TKS	Task Flag Status			—		DI : TKS_xxxx(xxxx 는 0~31의 십진수), TKS_1
IR	Index Register	DWORD	0~15	—		AI : IR_xxxx (xxxx 는 0~15의 십진수) IR_5
DR	Data Register	WORD		0~15		AI : DR_xxxxx (xxxxx 는 0~15의 십진수) DR_9 DI : DR_xxxxx0~F(xxxxx 는 0~15의 십진수) DR_1A

28. OmronS 드라이버

(1) 통신채널 추가



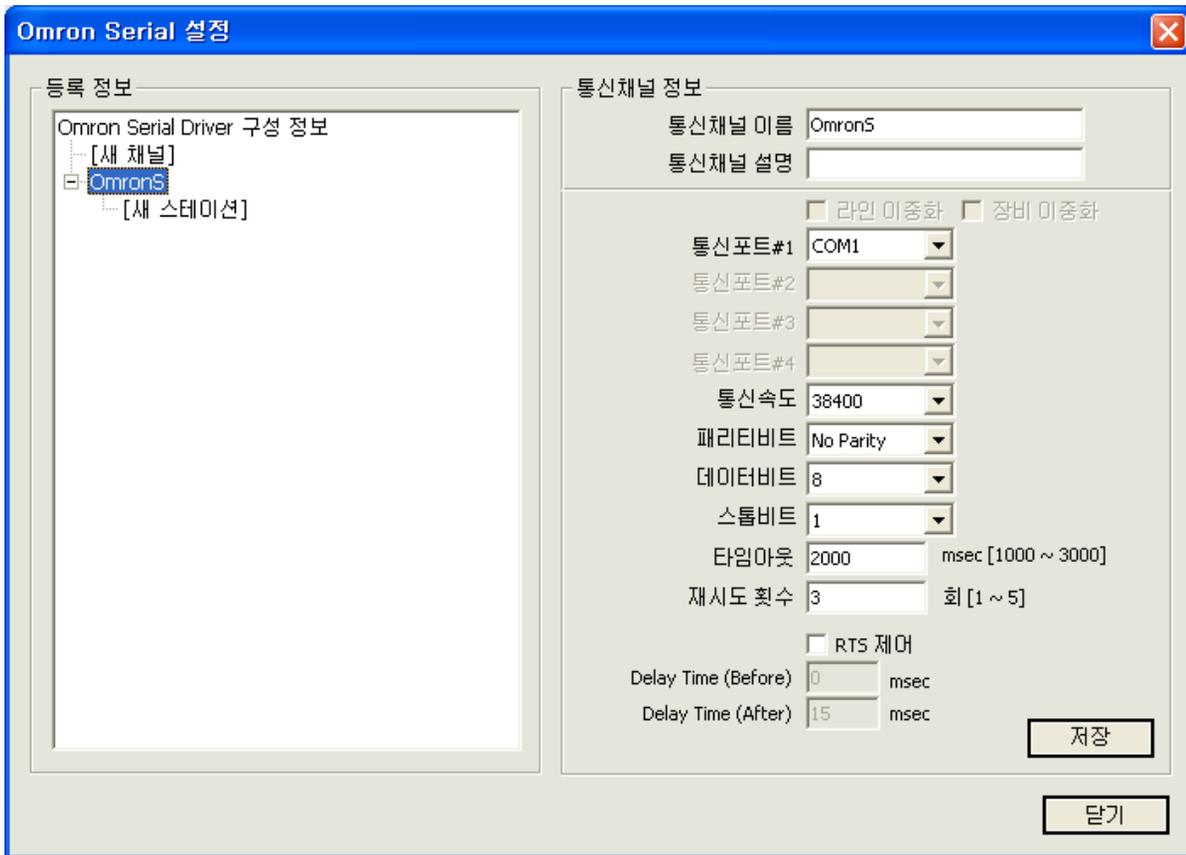
- IO 드라이버 리스트에서 “OmronS” 을 선택한 후 “추가” 버튼을 누릅니다.



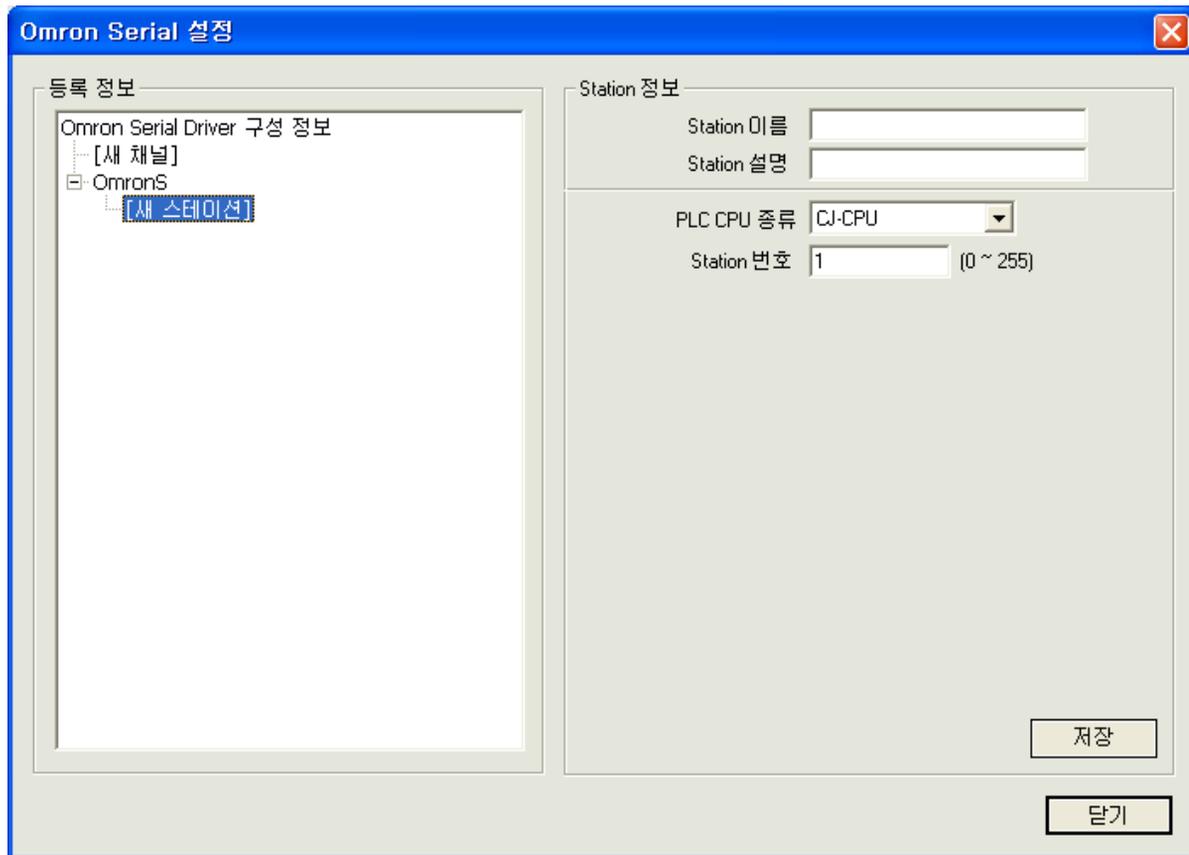
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.

부록1 I/O 드라이버 설정방법

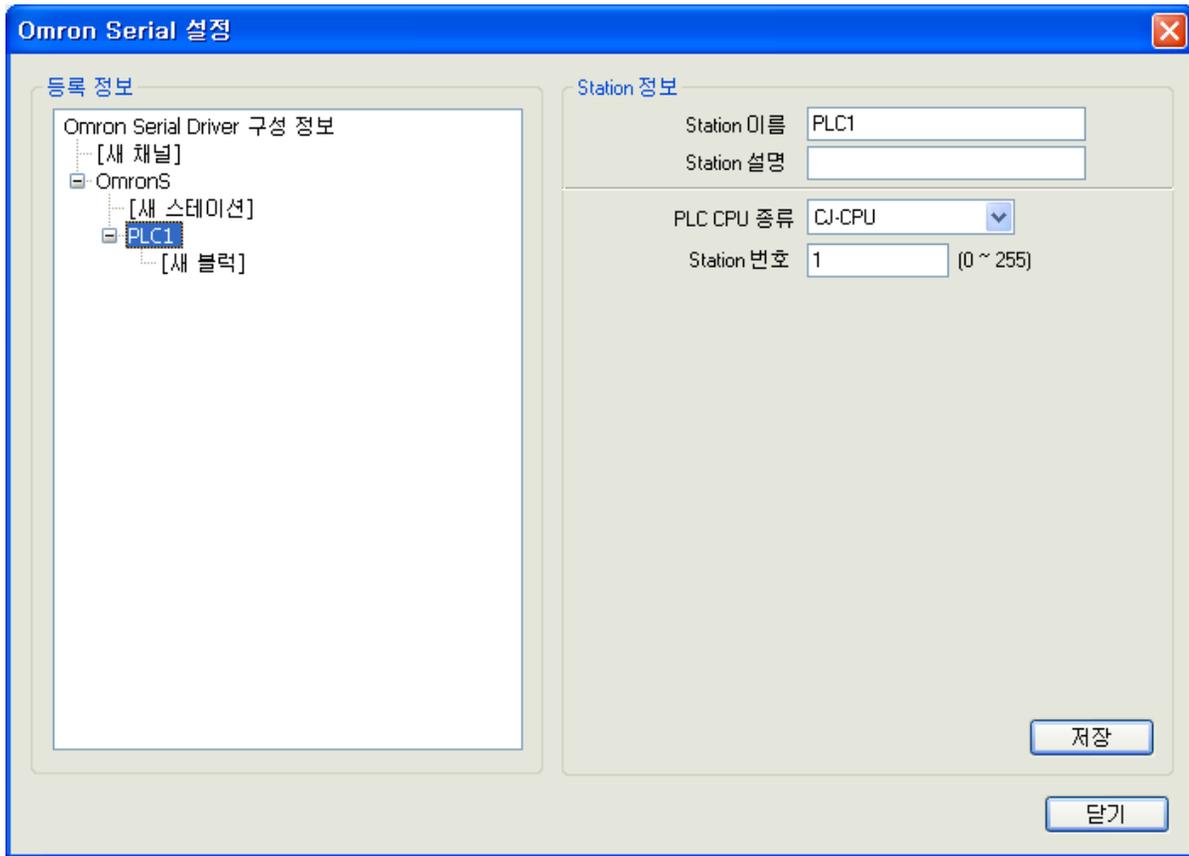
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (f) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) RTS 제어 : 485, 422 통신을 사용할 경우, 사용함으로 체크를 합니다.
- (k) Delay Time (Before) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 하기 직전의 Delay Time 입니다.
- (l) Delay Time (After) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 한 직후의 Delay Time 입니다.
- (m) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



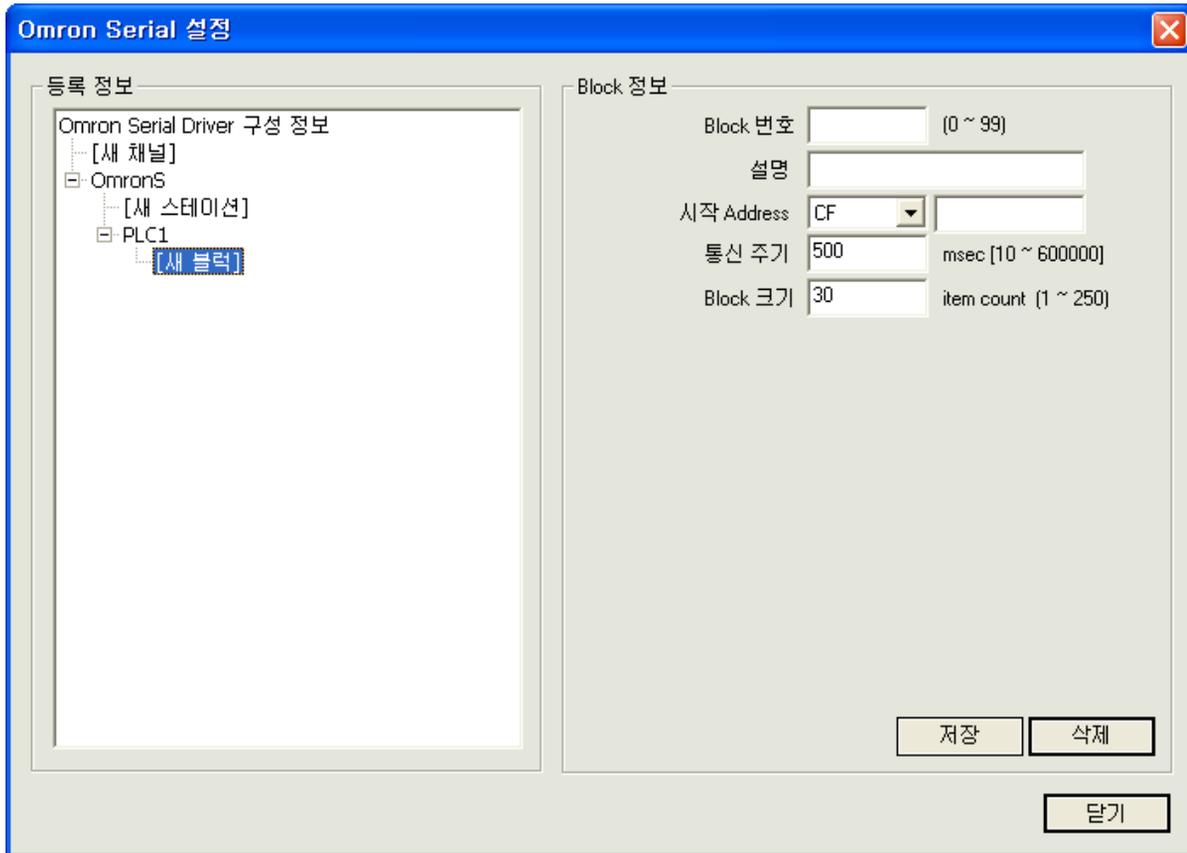
(2) Station 추가



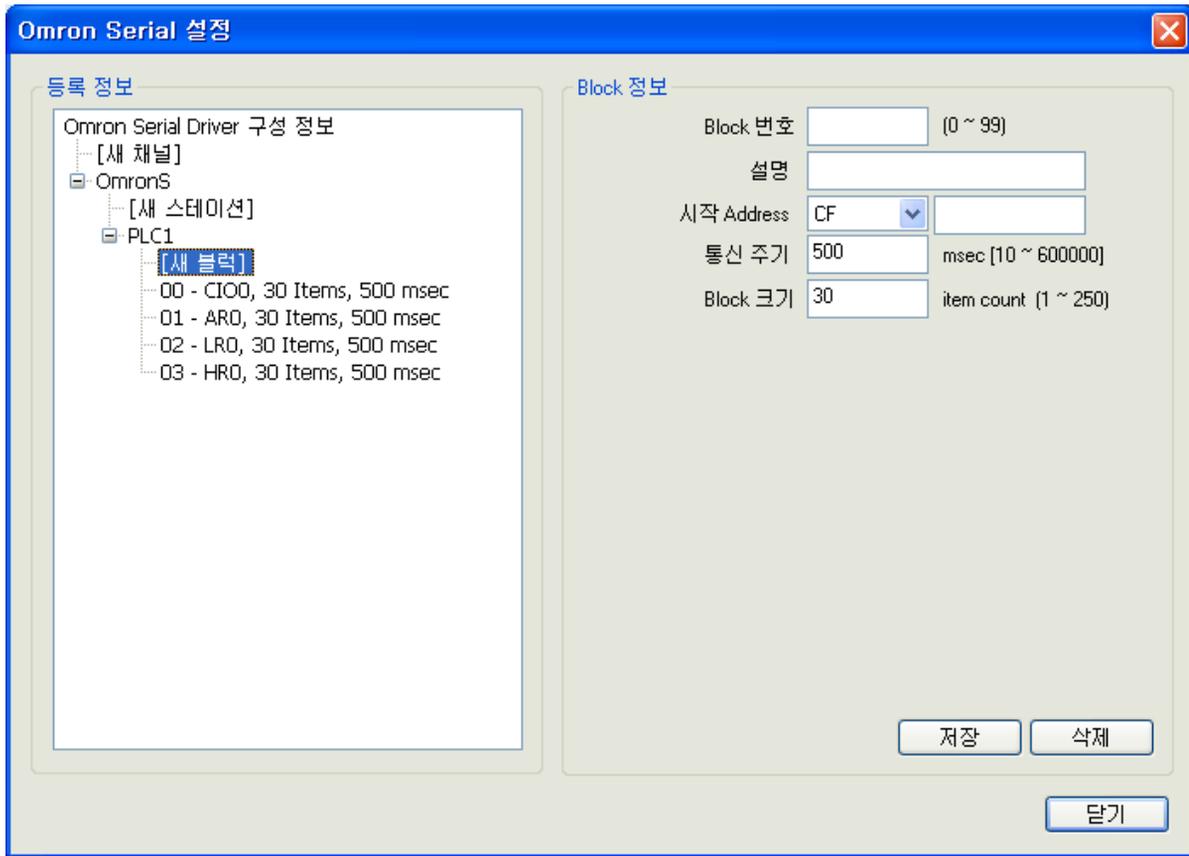
- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) Station 번호 : PLC Cnet 모듈의 국번을 입력합니다.
- (f) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호 입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다.
- 시작Address : 메모리타입(C10,WR,HR,AR,TF,CF,EMO)+시작번지
 - Block 크기 : 읽어올 갯수
- 예제1) 시작Address : C1010, Block 크기 : 100 이라 하면, C10 메모리(0부터 시작)의 10번째 WORD부터 100개의 WORD를 읽어온다.
- 예제2) 시작Address : TF20, Block크기 : 20이라 하면, TF 메모리(0부터 시작)의 20번째 상태 Bit부터 20개의 상태 Bit를 읽어온다.
- 주의) C10,LR,HR,AR,TIM,CNT,DM,EM,EMO,EM1,EM2,EM3,EM4,EM5,EM6,EM7,EM8,EM9,EMA,EMB,EMC 은 WORD단위의 디바이스 다.
- 주의) TF,CF 는 Bit단위의 디바이스 이다.
- 주의) WORD단위의 디바이스는 30개이상, Bit단위의 디바이스는 120개 이상이면, Multi Frame으로 읽게 된다.
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 구별자별 읽고자하는 갯수
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

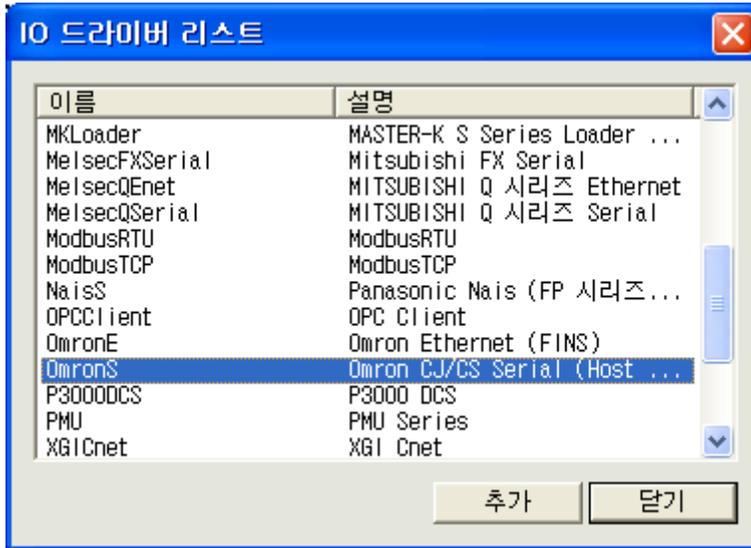
- . 아날로그 입력Address 사용예제
 - C10_100 : C10영역의 100번째 Word값을 가져온다
 - EM7_20000 : EM7영역의 20000번째 Word값을 가져온다.
- . 디지털 입력Address 사용예제
 - C10_90A : C10영역의 90번째 WORD의 10(A)번째 Bit 값을 가져온다.
 - EM7_07 : EM7영역의 0번째 WORD의 7번째 Bit 값을 가져온다.
 - TF_123 : Timer Flag영역의 123번째 타이머 Completion Flag 값을 가져온다.

- Omron PLC 메모리 디바이스 맵

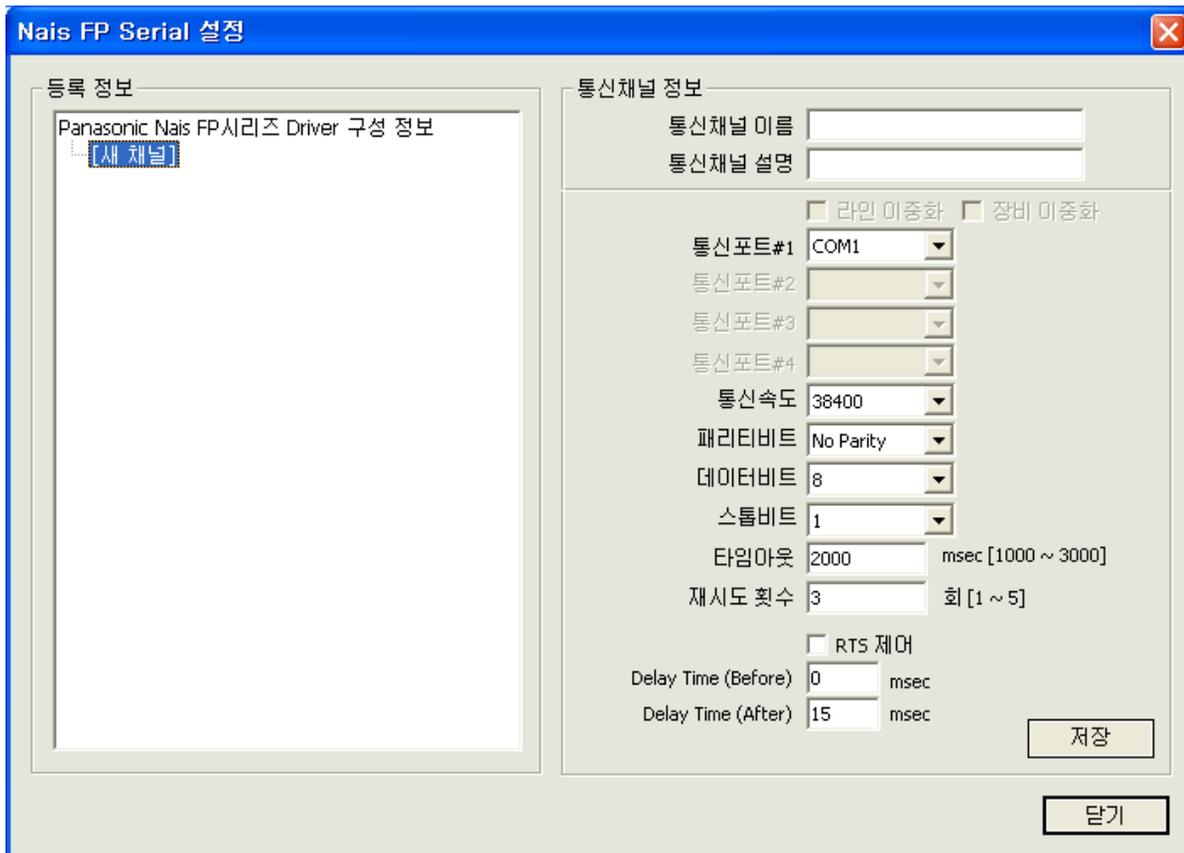
메모리 코드	특성	데이터 형식	CS/CJ CPU 메모리번지	CV CPU 메모리번지	비고	어드레스 적용예제
CIO	CIO Area	WORD	0~6143	0~2555		AI : CIO_xxxx (xxxx 는 0~6143 의 십진수) CIO_100 DI : CIO_xxxx0~F(xxxx 는 0~6143 의 십진수) CIO_100A
LR	CIO1000~CIO1199		0~199	—		AI : WR_xxxx (xxxx 는 0~511 의 십진수) WR_100 DI : WR_xxxx0~F(xxxx 는 0~511 의 십진수) WR_OF
HR	Holding Bit Area		0~511			AI : HR_xxxx (xxxx 는 0~511 의 십진수) HR_234 DI : HR_xxxx0~F(xxxx 는 0~511 의 십진수) HR_19
AR	Auxiliary Bit Area		0~959	0~959	0~447(읽기전용), 448~959(읽기/쓰기)	AI : AR_xxxx (xxxx 는 0~511 의 십진수) AR_800 DI : AR_xxxx0~F(xxxx 는 0~511 의 십진수) AR_9001
TF	Timer Area	Completion Flag(BIT)	0~2047	0~2047 또는 0~1023	읽기전용	DI : TF_xxxx(xxxx 는 0~4095 의 십진수), TF_2047
CF	Counter Area					DI : CF_xxxx(xxxx 는 0~4095 의 십진수), CF_2048
TIM	Timer Area PV	WORD	0~2047	0~2047 또는 0~1023		AI : TIM_xxxx (xxxx 는 0~4095 의 십진수) TIM_1000
CNT	Counter Area PV				AI : CNT_xxxx (xxxx 는 0~4095 의 십진수) CNT_2000	
DM	DM Area		0~9999	—		AI : DM_xxxxx (xxxxx 는 0~32767 의 십진수) DM_10000 DI : DM_xxxxx0~F(xxxxxx 는 0~32767 의 십진수) DM_900E
EM0~EM7	EM Area					
EM7~EMC						AI : EMA_xxxxx (xxxxx 는 0~32767 의 십진수) EMA_9000 DI : EMA_xxxxx0~F(xxxxxx 는 0~32767 의 십진수) EMA_7000A
EM			0~9999		AI : EM_xxxxx (xxxxx 는 0~32767 의 십진수) EM_9000 DI : EM_xxxxx0~F(xxxxxx 는 0~32767 의 십진수) EM_7000A	

29. NaisS 드라이버

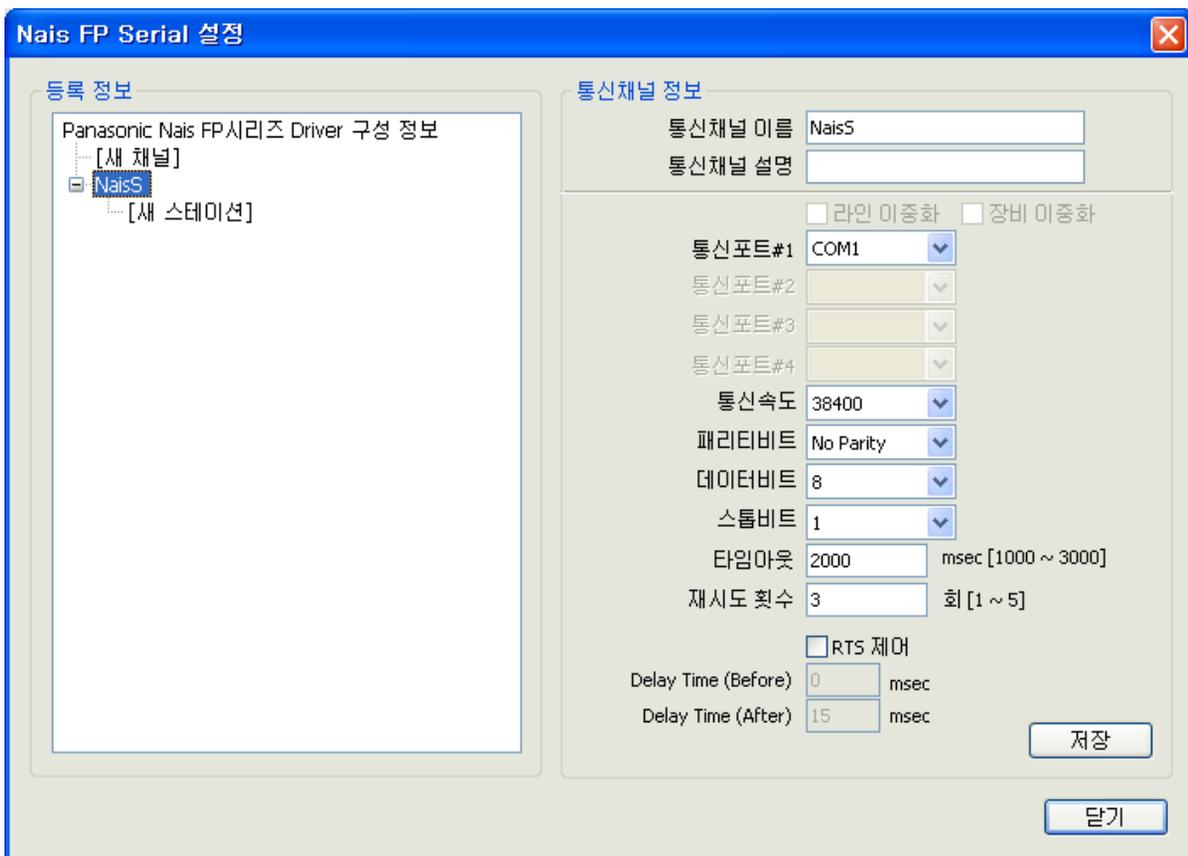
(1) 통신채널 추가



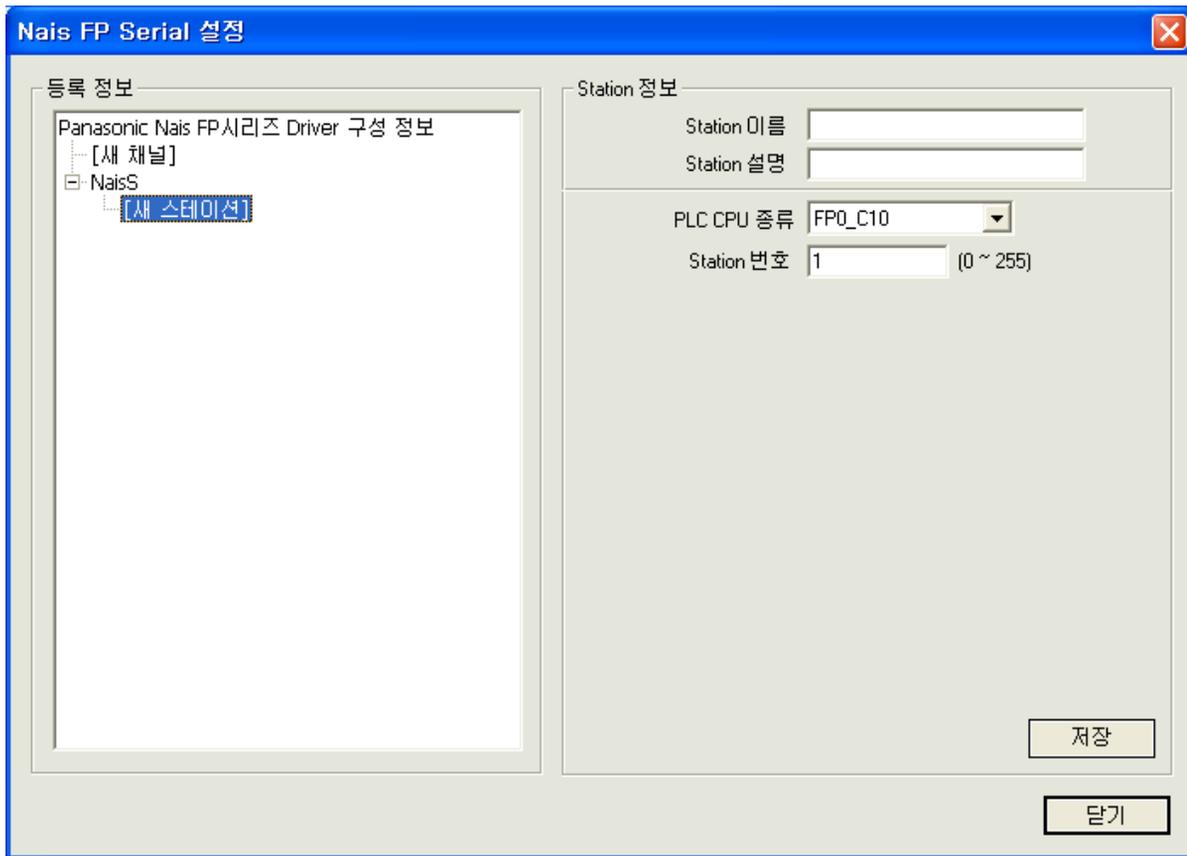
- IO 드라이버 리스트에서 “OmronS” 을 선택한 후 “추가” 버튼을 누릅니다.



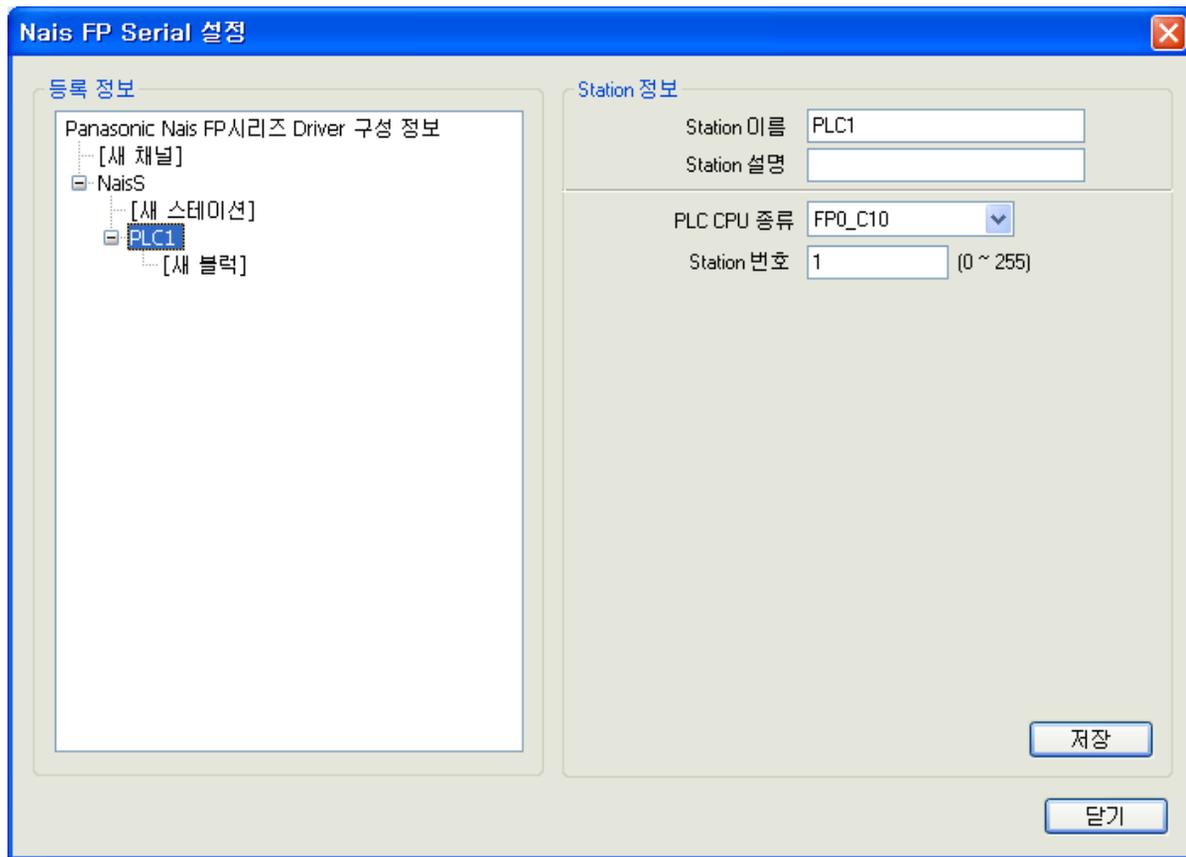
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (r) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) RTS 제어 : 485, 422 통신을 사용할 경우, 사용함으로 체크를 합니다.
- (k) Delay Time (Before) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 하기 직전의 Delay Time 입니다.
- (l) Delay Time (After) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 한 직후의 Delay Time 입니다.
- (m) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



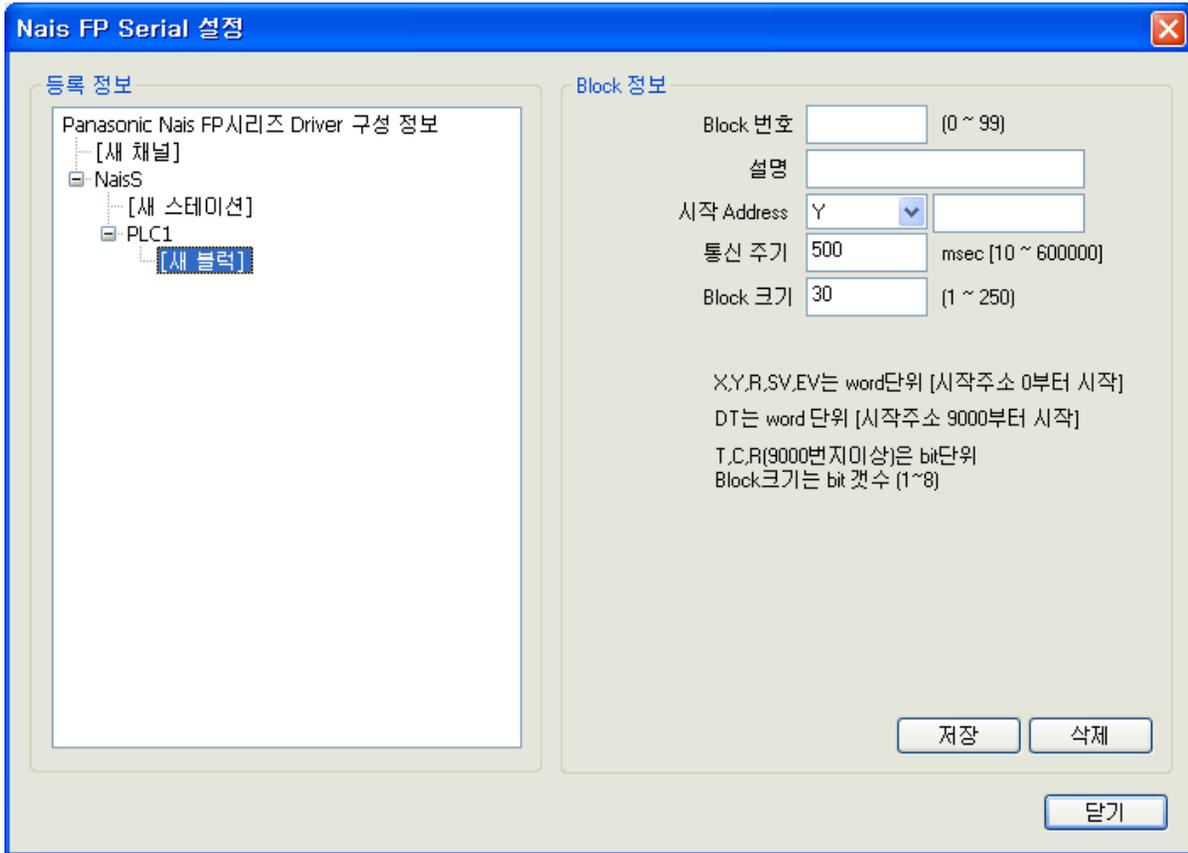
(2) Station 추가



- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) Station 번호 : PLC Cnet 모듈의 국번을 입력합니다.
- (f) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.

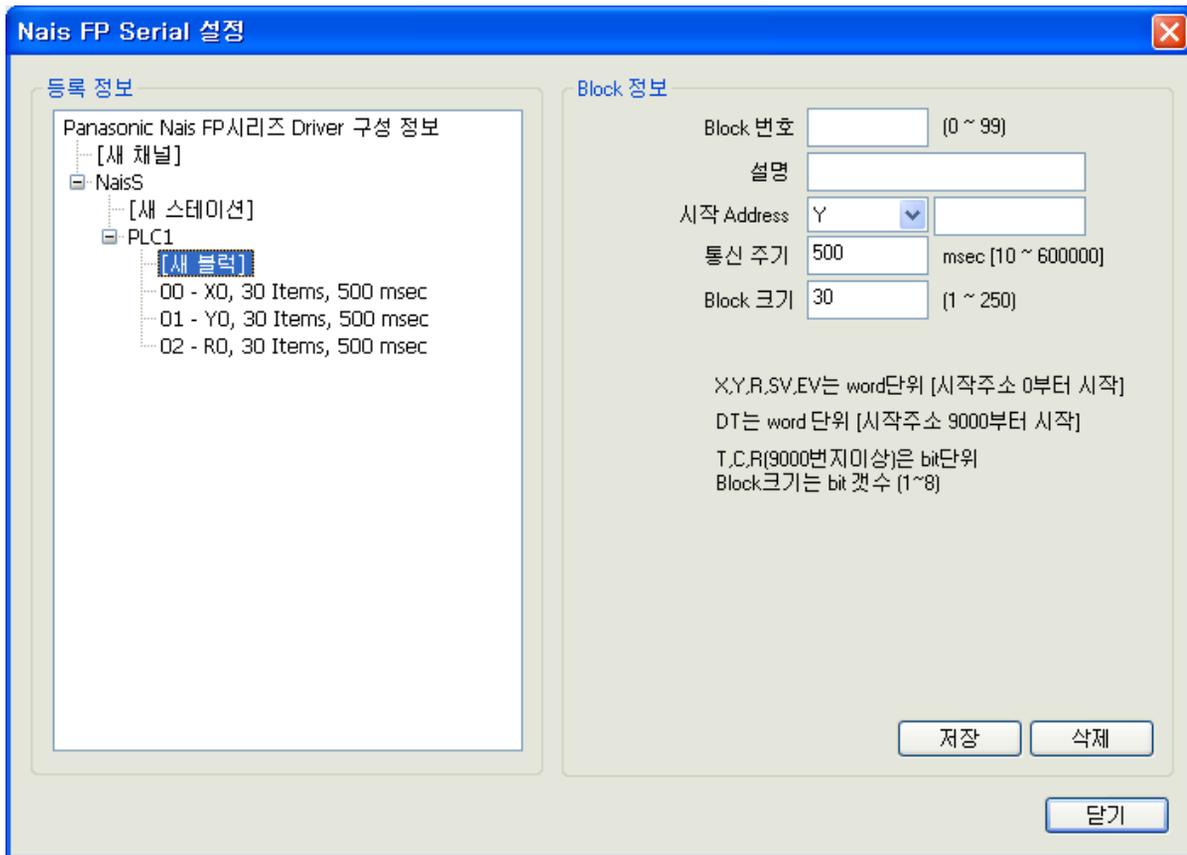


(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호 입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다.
 - X,Y,R,DT,SV,EV, IX, IY 는 WORD단위 읽기를 사용한다.
 - 예1) X,Y,R은 Bit단위 메모리이지만 WORD단위 읽기를 사용한다. X입력은 208점으로 X0~X12F까지 사용한다.
시작Address : X1, Block 크기 : 5한다면, X10~X5F의 데이터를 가져온다.
 - 예2) EV,SV는 WORD단위 메모리이다. 따라서 WORD단위 읽기를 사용한다.
시작Address : EV0, Block 크기 : 20으로 한다면, EV0~EV19의 데이터를 가져온다.
 - 예3) IX, IY는 WORD단위 메모리이지만 크기는 1WORD씩 즉, IX0, IY0만 존재한다.
시작Address : IX0, Block 크기 : 1으로 설정한다. IY도 마찬가지로 이다.
 - 예4) DT는 WORD단위 메모리이지만 범위는 DT9000~DT9111만 존재한다.
시작Address : DT9000, Block 크기 : 10으로 설정한다면, DT9000~DT9009의 데이터를 가져온다.
 - R9000이상 ,T,C는 Bit단위 읽기를 사용한다. 시작Address는 Bit단위이며, Block크기는 1~8개만 가진다.
 - 예1) 시작Address : T0, Block 크기 : 8한다면, T0~T7의 데이터를 가져온다.
 - 예2) 시작Address : R9000, Block 크기 : 8한다면, R9000~R9007의 데이터를 가져온다.
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 구별자별 읽고자하는 갯수
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.

(h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

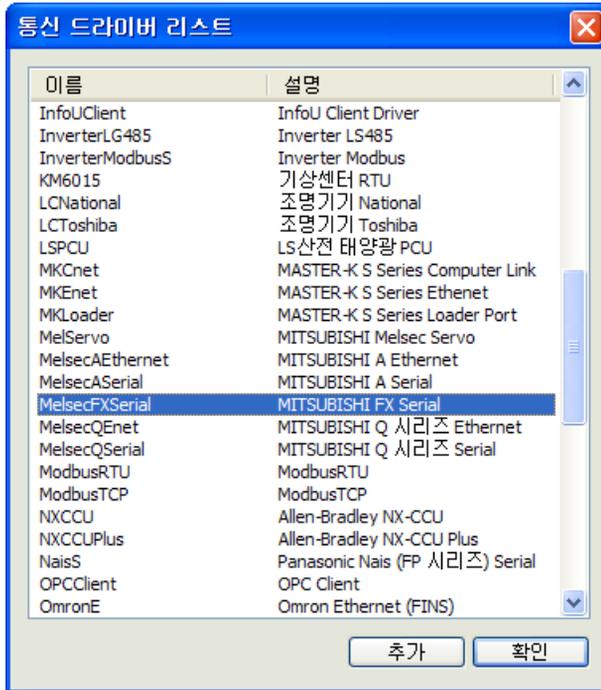
(a) 형식

- 아날로그 입력Address 사용예제
 - DT100 : DT영역의 100번째 Word값을 가져온다.
 - SV10 : 타이머 영역의 T10의 설정치값을 가져온다
 - SV111 : 카운터 영역의 C111의 설정치값을 가져온다
- 디지털 입력Address 사용예제
 - X10F : X영역의 $10 \times 16 + F(15) = 160 + 15 = 175$ 번째 접점의 값을 가져온다.
 - R501 : 내부릴레이 R영역의 $50 \times 16 + 1(1) = 800 + 1 = 801$ 번째 접점의 값을 가져온다.
 - R9001 : 특수내부릴레이 R9001의 접점의 값을 가져온다.
 - DT1280F : 데이터레지스터 1280워드의 F(15)번째의 Bit값을 가져온다.
- Nais PLC 메모리 디바이스 맵

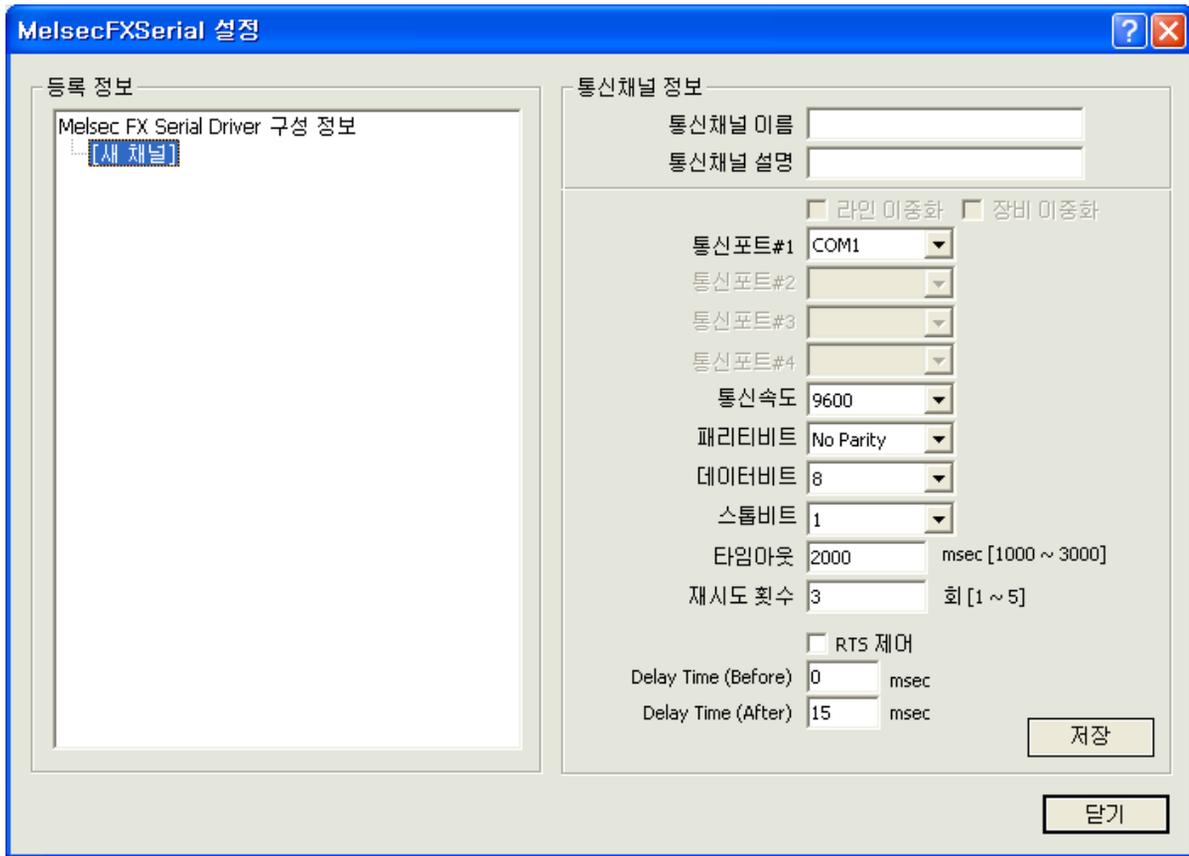
메모리 코드	특성	데이터 형식	C10/C14/C16 CPU 메모리번지	C32/SL1 CPU 메모리번지	어드레스 적용예제
X	외부입력	BIT	208 점(X0~X12F)		DI : Xxxxx0~F(xxxx 는 0~12의 십진수) X12A
Y	외부출력		208 점(Y0~Y12F)		DI : Yxxxx0~F(xxxx 는 0~12의 십진수) Y0F
R	내부릴레이		1008 점(R0~X62F) 특수내부릴레이 64 점 (R9000~R9063)		DI : Rxxxx0~F(xxxx 는 0~62의 십진수) R19F (특수) DI : Rxxxx (xxxx 는 9000~9063의 십진수) R9010
T	타이머		T0~T99		DI : Txxxx (xxxx 는 0~99의 십진수) T90
C	카운터		C100~C143		DI : Cxxxx(xxxx 는 100~143의 십진수), C108
DT	데이터레지스터		WORD	1660 워드 (DT0~DT1659)	6144 워드 (DT0~DT6143)
SV	타이머/카운터 설정치 영역	144 워드(SV0~SV143) 타이머 : 0~99, 카운터 : 100~143		AI : SVxxxx (xxxx 는 0~143) SV1 or SV100	
EV	타이머/카운터 경과치 영역	144 워드(EV0~EV143) 타이머 : 0~99, 카운터 : 100~143		AI : EVxxxx (xxxx 는 0~143) EV40 or EV129	
IX	인덱스레지스터	1 워드 (IX0)		AI : IX0 (단지 1개워드) DI : IX_0(0~F) IX0F	
IY	인덱스레지스터	1 워드 (IY0)		AI : IY0 (단지 1개워드) DI : IY_0(0~F) IY01	

30. MelsecFXSerial 드라이버

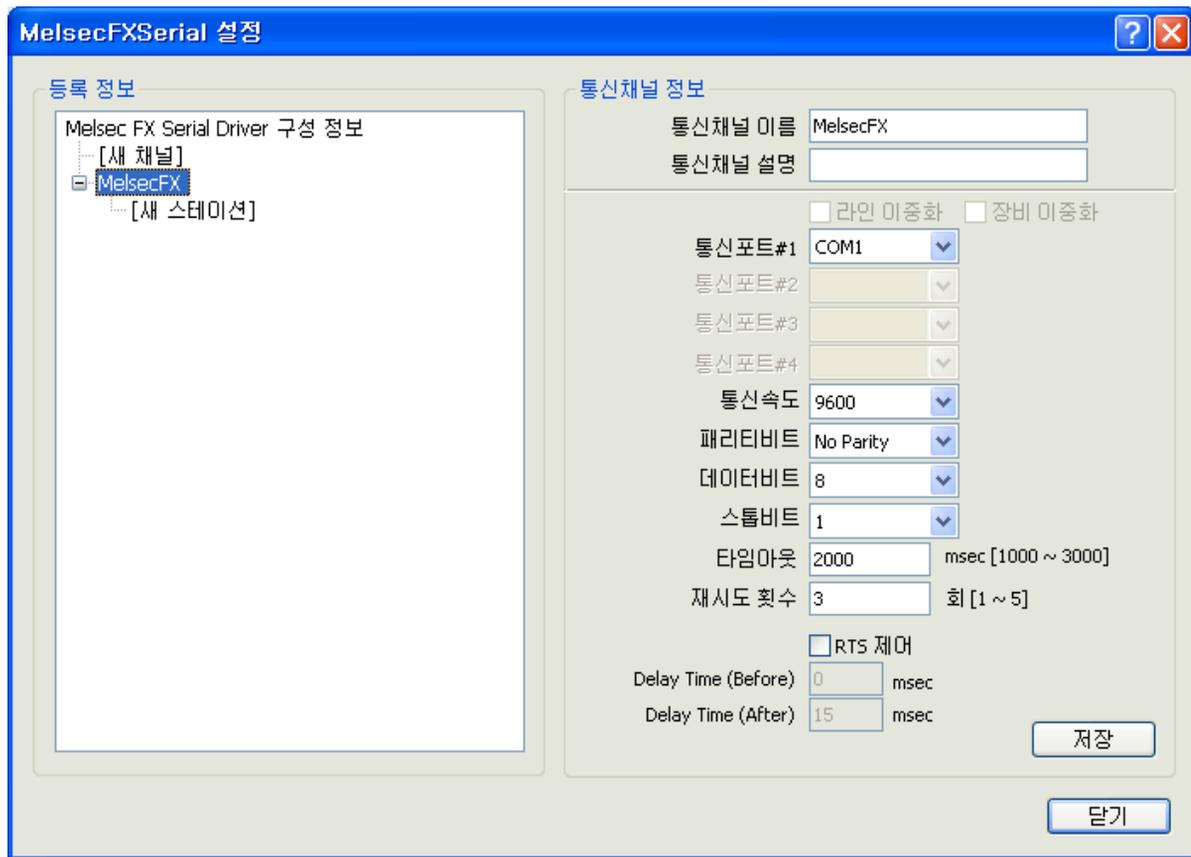
(1) 통신채널 추가



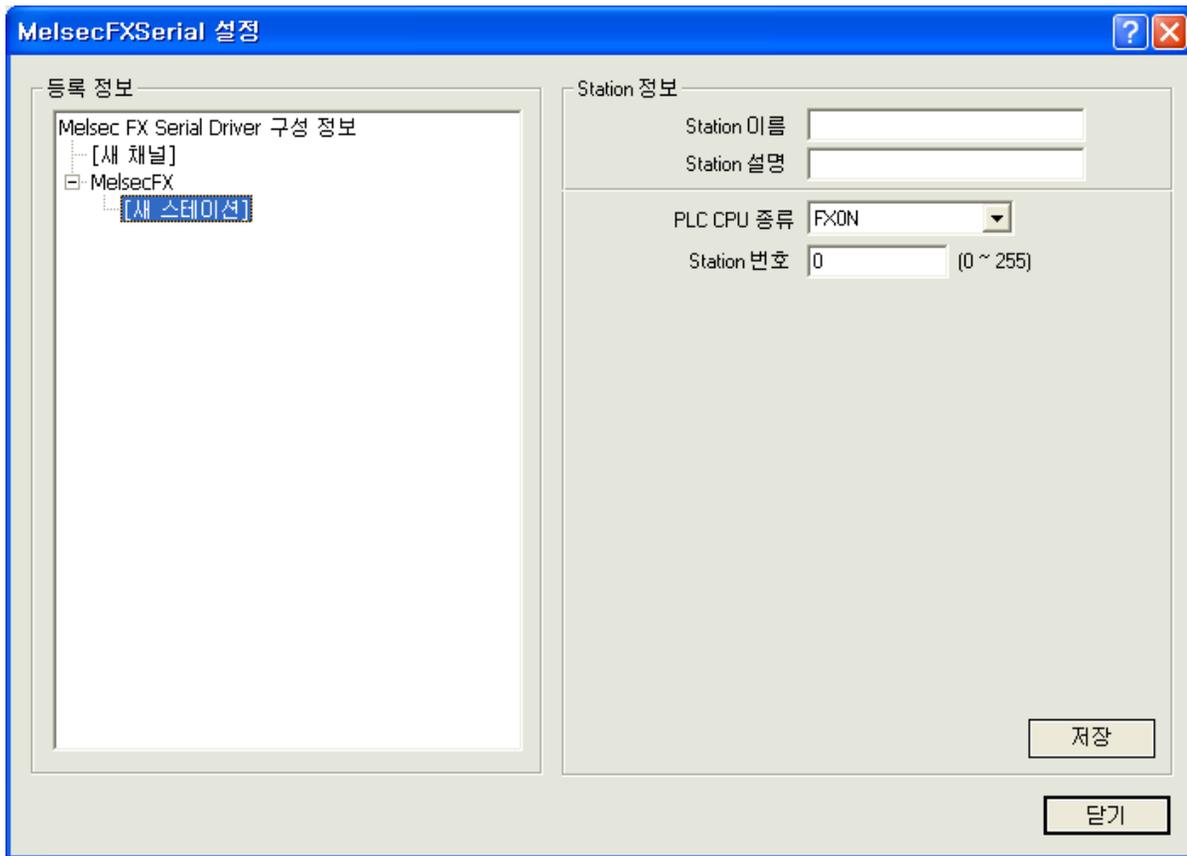
- I/O 드라이버 리스트에서 “MelsecFXSerial” 을 선택한 후 “추가” 버튼을 누릅니다.



- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (r) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) RTS 제어 : 485, 422 통신을 사용할 경우, 사용함으로 체크를 합니다.
- (k) Delay Time (Before) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 하기 직전의 Delay Time 입니다.
- (l) Delay Time (After) : 485, 422 통신에서만 사용하는 정보입니다. PLC에게 데이터 요청을 한 직후의 Delay Time 입니다.
- (m) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.

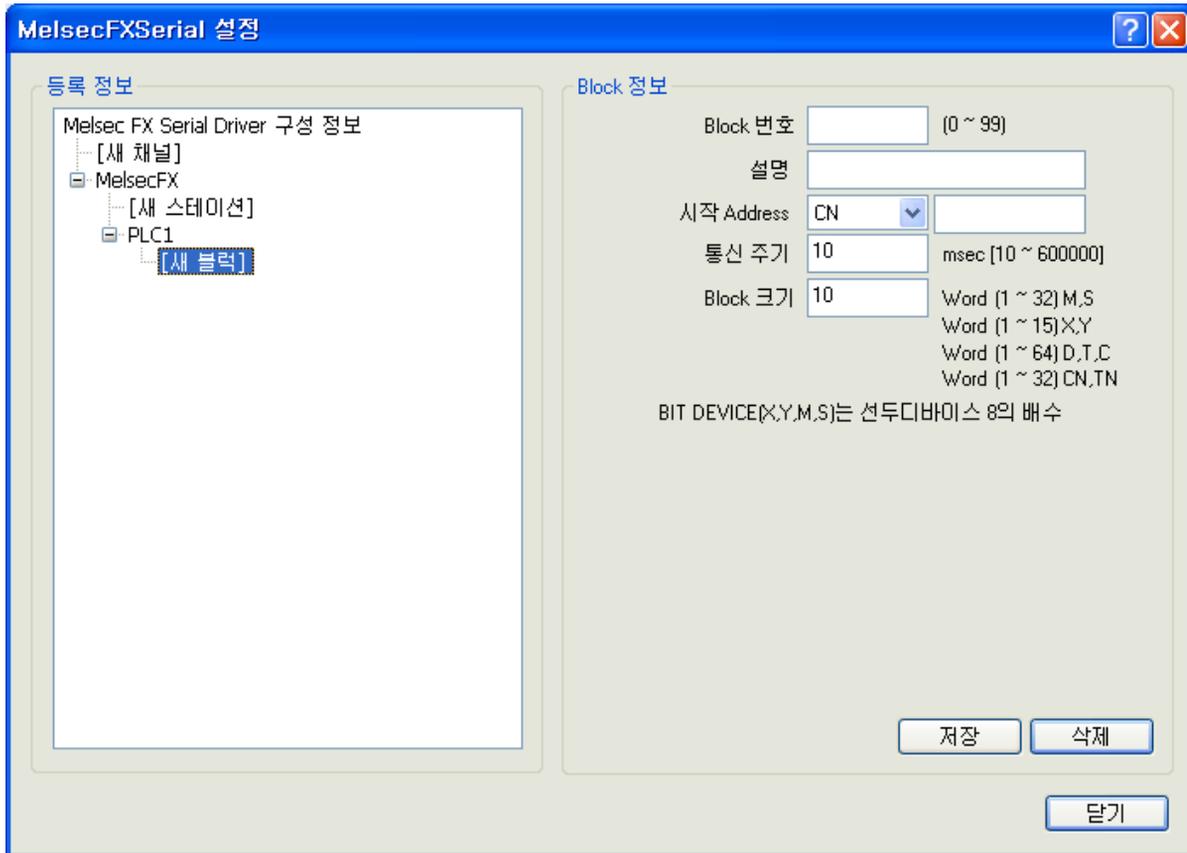


(2) Station 추가

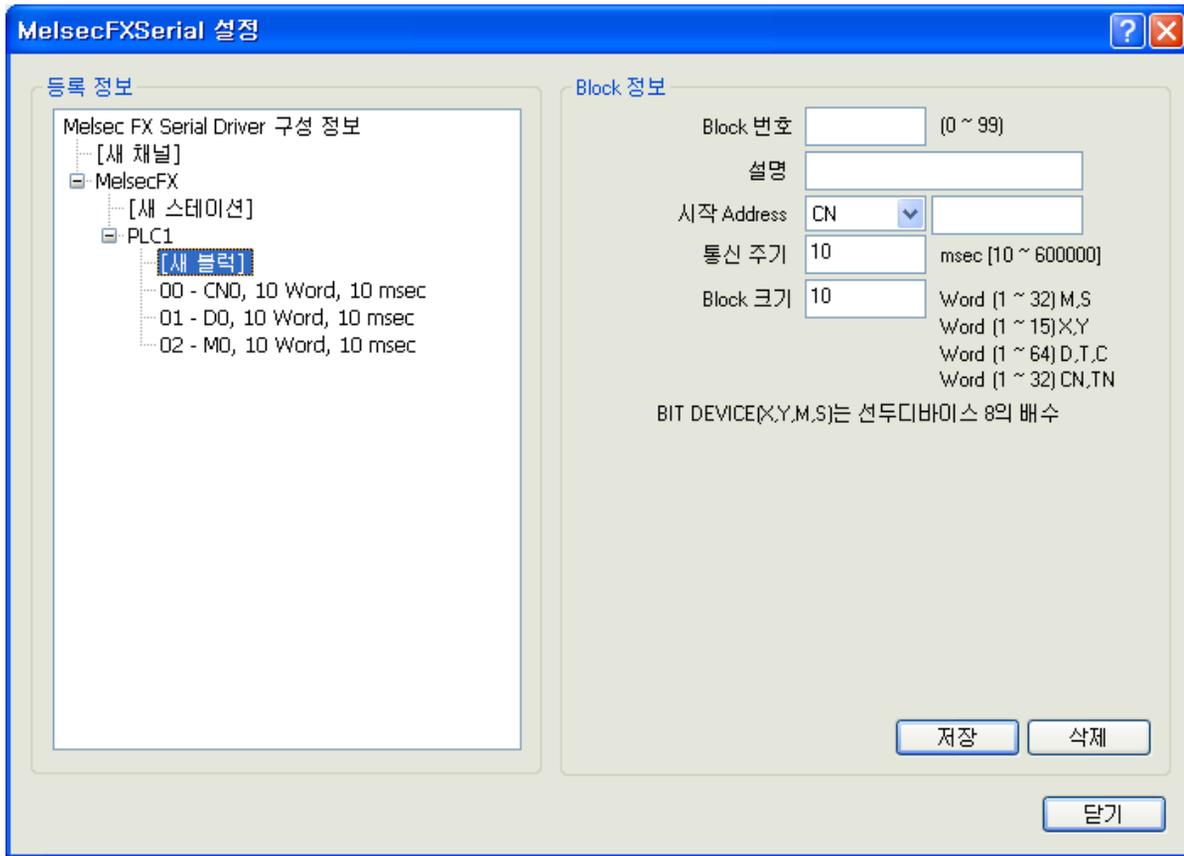


- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) PLC CPU 종류 : PLC CPU 종류를 선택합니다.
- (e) Station 번호 : PLC Cnet 모듈의 국번을 입력합니다.
- (f) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.

(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : Block의 시작 Address를 입력합니다.
- X,Y,M,S는 Bit단위의 디바이스 메모리이며, 읽기는 WORD단위로 불러온다.
 예제1) X,Y는 8진수단위의 Bit메모리 이다. Address는 X7 다음은 X10으로 표현한다. 시작Address는 8진수로 8의 배수이어야 한다. 즉, X0(0번째 비트), X10(8번째 비트), X20(16번째 비트), X30(24번째 비트)....
 그리고, Block크기는 WORD단위 이다. 만약16점을 읽어오려면, 1을 넣어주면 된다.
 시작 Address : X0, Block크기 : 2로 하였다면, X0~X7, X10~X17, X20~X27, X30~X37의 32Bit를 읽어온다.
 예제2) M,S는 10진수단위의 Bit메모리 이다. Address는 M9 다음은 M10으로 표현한다. 시작Address는 10진수로 8의 배수이어야 한다. 즉, M0(0번째 비트), M8(8번째 비트), M16(16번째 비트), M24(24번째 비트)....
 그리고, Block크기는 WORD단위 이다. 만약16점을 읽어오려면, 1을 넣어주면 된다.
 시작 Address : M0, Block크기 : 2로 하였다면, M0~M31의 32Bit를 읽어온다.
 - D,T,C,CN,TN은 WORD단위의 디바이스 메모리이다.
 시작Address : D10, Block크기 : 10으로 하였다면, D10~D19까지의 10개 WORD값을 읽어온다.
- (e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.
- (f) Block 크기 : 해당 구별자별 읽고자하는 갯수
- (g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



(4) 입출력 주소

(a) 형식

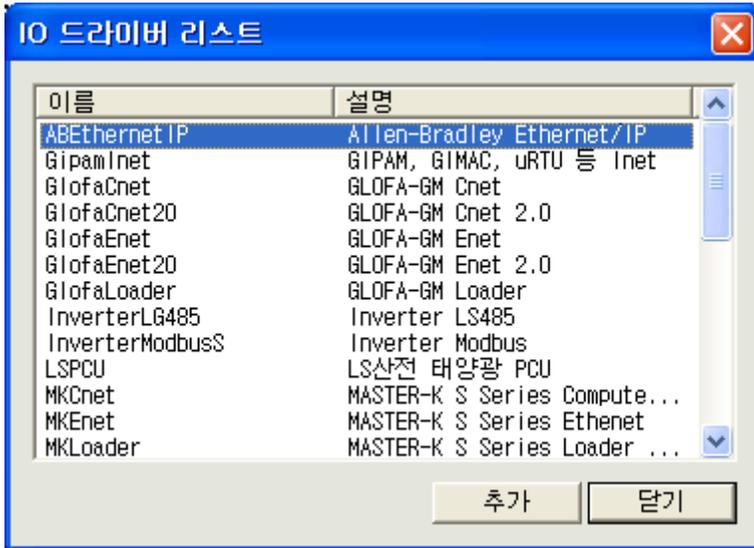
- . 아날로그 입력Address 사용예제
 - D100 : D영역의 100번째(0번째부터 시작) Word값을 가져온다.
 - TN6 : 타이머 영역의 6번째(0번째부터 시작) 현재값을 가져온다
 - CN10 : 카운터 영역의 10번째(0번째부터 시작) 현재값을 가져온다
- . 디지털 입력Address 사용예제
 - X7 : 입력릴레이의 7번째(0번째부터 시작) Bit값을 가져온다.
 - Y10 : 출력릴레이 10번째(0번째부터 시작)Bit값을 가져온다.
 - M100 : 보조릴레이 100번째(0번째부터 시작)Bit값을 가져온다.

- MelsecFX Serial PLC 메모리 디바이스 맵

디바이스 종류		디바이스 코드	디바이스 번호 범위 (디폴트 할당 시)					표현		비고	예제(InfoU 어드레스)
비트	워드		FX1S	FX0N	FX2, FX2C	FX1NC, FX1N	FX2N, FX2NC	8진수	10진수		
●		X	X0000~X0017	X0000~X0177	X0000~X0337	X0000~X0177	X0000~X0267	●		입력릴레이	X0,X07,X007,X0007
●		Y	Y0000~Y0015	Y0000~Y0177	Y0000~Y0337	Y0000~Y0177	Y0000~Y0267	●		출력릴레이	Y0,Y10,Y110,Y0110
●		M	M0000~M0511		M0000~M1535		M0000~M3071		●	보조릴레이	M0,M04,M111,M1234
●		S	S0000~S127		S0000~S0999				●	스테이트	S1,S50,S200,S0005
●		M	M8000~M8254		M8000~M8255				●	특수보조릴레이	-
	●	TN	TN000~TN063		TN000~TN255				●	타이머 현재값	TN0,TN01,TN001
	●	CN	CN000~CN031, CN235~CN254		CN000~CN255				●	카운터 현재값	CN6,CN10,CN100
	●	D	D0000~D0255		D0000~D0999	D0000~D7999			●	데이터 레지스터	D1,D10,D100,D1000
	●	D	-	D1000~D2499	D1000~D2999	-			●	파일 레지스터	-
	●	D	-		D6000~D7999	-			●	RAM 파일 레지스터	-
	●	D	D8000~D8255		D8000~D8255				●	특수 데이터 레지스터	-

31. ABEthernet IP 드라이버

(1) 통신채널 추가

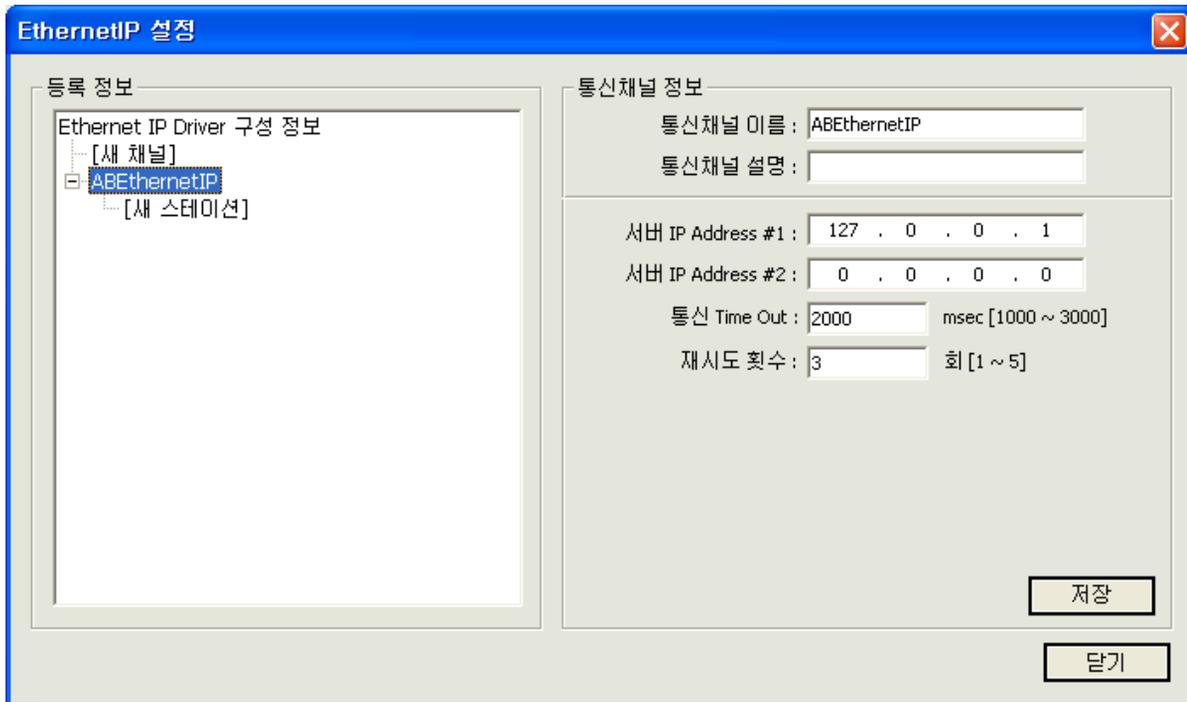


- IO 드라이버 리스트에서 “OmronE” 을 선택한 후 “추가” 버튼을 누릅니다.

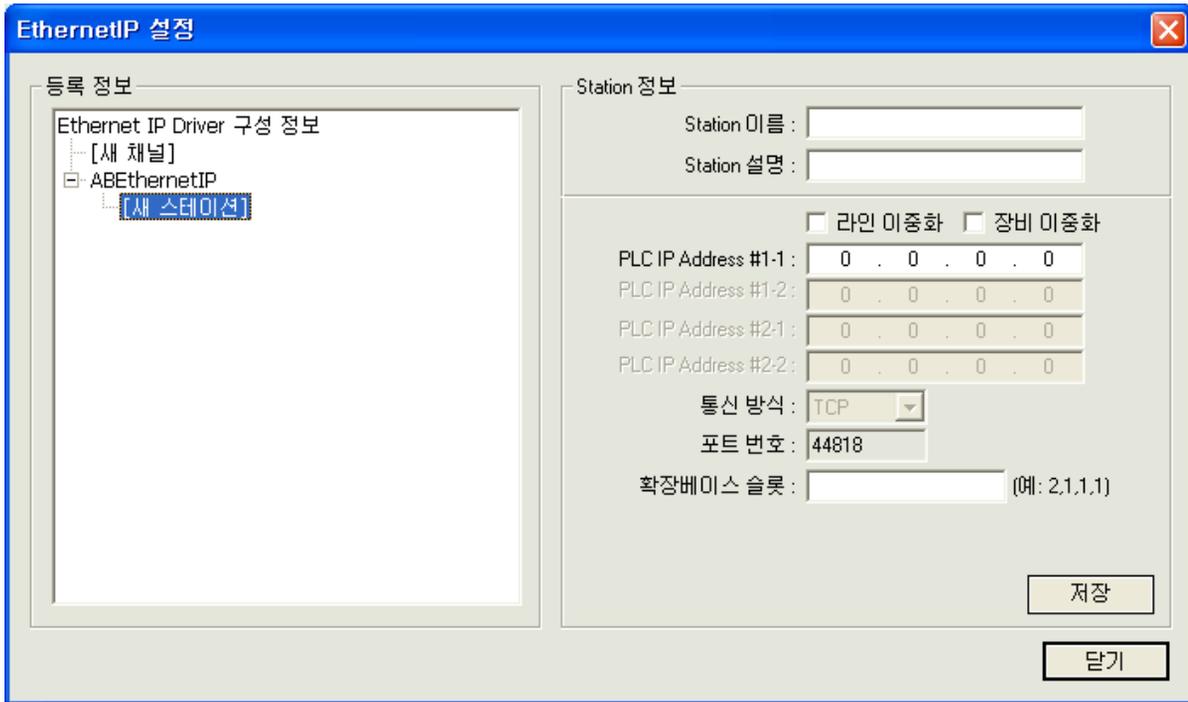


- 통신채널 이름 : 통신채널 이름을 입력합니다.
- 통신채널 설명 : 통신채널 설명을 입력합니다.
- 서버 IP Address #1 : PC의 IP Address를 입력합니다.
- 서버 IP Address #2 : 라인 이중화를 사용할 경우 사용하게 될 두번째 IP Address를 입력합니다.

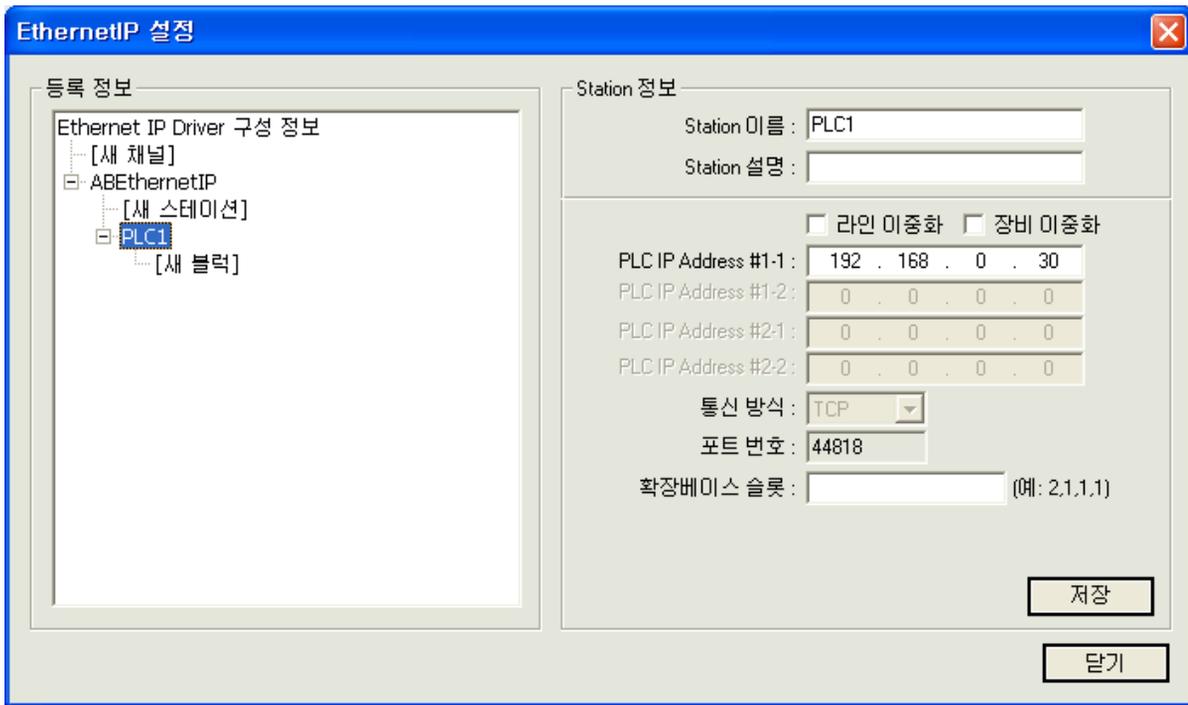
- (e) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (f) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (g) Address Offset 시작번지 : 0 또는 1을 가진다. 시작번지를 0번부터 또는 시작번지를 1부터 시작하는가의 여부를 설정한다.
- (h) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타 납니다.



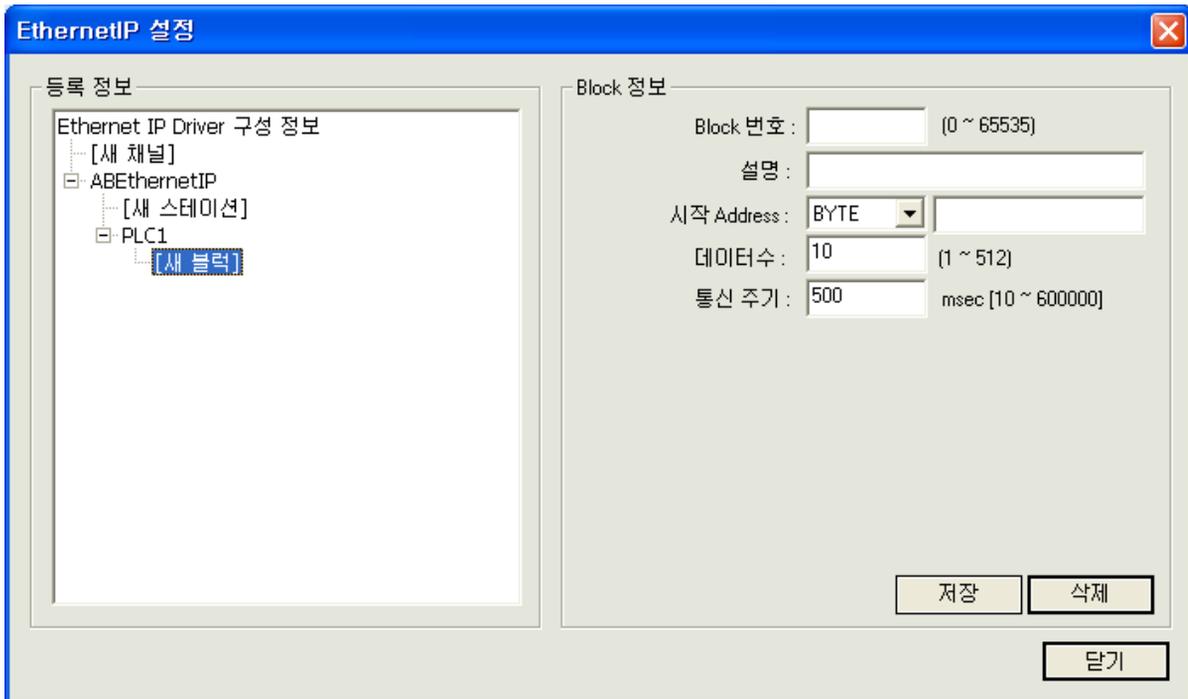
(2) Station 추가



- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) 라인 이중화 : 라인 이중화를 사용할 경우 체크 합니다.
- (f) 장비 이중화 : 장비 이중화를 사용할 경우 체크 합니다.
- (g) PLC IP Address #1-1 : PLC 의 IP Address를 입력합니다.
- (h) PLC IP Address #1-2 : PLC 의 IP Address를 입력합니다. 라인 이중화를 사용할 때 입력합니다.
- (i) PLC IP Address #2-1 : PLC 의 IP Address를 입력합니다. 장비 이중화를 사용할 때 입력합니다.
- (j) PLC IP Address #2-2 : PLC 의 IP Address를 입력합니다. 라인 이중화와 장비 이중화를 함께 사용할 때 입력합니다.
- (k) 통신 방식 : TCP사용으로 고정이다.
- (l) 포트 번호 : Ethernet IP는 44818포트로 고정되어 있다.
- (m) 확장베이스슬롯 : EtherNet/IP 통신 모듈이 장착되어 있는 슬롯이 확장 베이스에 장착되어 있을 경우에 확장 베이스 슬롯 번호를 입력한다



(3) Block 추가



- (a) “등록 정보” 트리에서 [새 블록]을 선택합니다.
- (b) Block 번호 : Block의 고유 번호입니다. 각각의 Block은 서로 다른 Block 번호를 지정해 주어야 합니다.
- (c) 설명 : Block 설명을 입력합니다.
- (d) 시작 Address : EtherNet/IP의 통신블록의 시작Address는 PLC에서 등록된 태그 이름을 어드레스로 사용한다.
 - . 만약, PLC에서 INT_A1_TAG 이름으로 Data타입을 INT형으로 배열0~99까지 100개를 할당했다면,

부록1 I/O 드라이버 설정방법

시작어드레스는 WORD타입, INT_A1_TAG[0]으로 데이터수는 100으로 설정해준다.

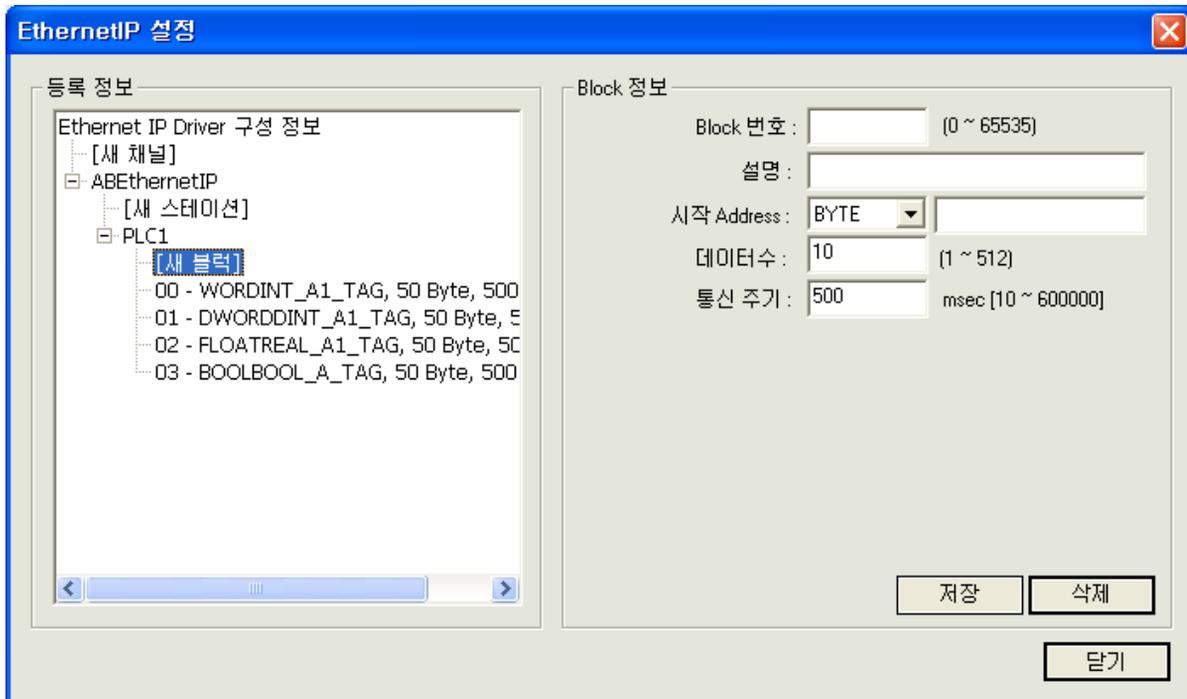
- 만약, PLC에서 BOOL_A_TAG 이름으로 Data타입을 BOOL형으로 배열0~99까지 100개를 할당했다면, 시작어드레스는 BOOL타입, BOOL_A_TAG[0]으로 데이터수는 100으로 설정해준다.
BOOL형의 시작주소는 항상 32의 배수이어야 한다.

(e) 통신 주기 : 해당 Block의 데이터 수집 주기를 msec 단위로 입력합니다.

(f) Block 크기 : 해당 구별자별 읽고자하는 갯수

(g) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.

(h) 삭제 : 삭제 버튼을 누르면, 현재 선택된 Block이 삭제됩니다.



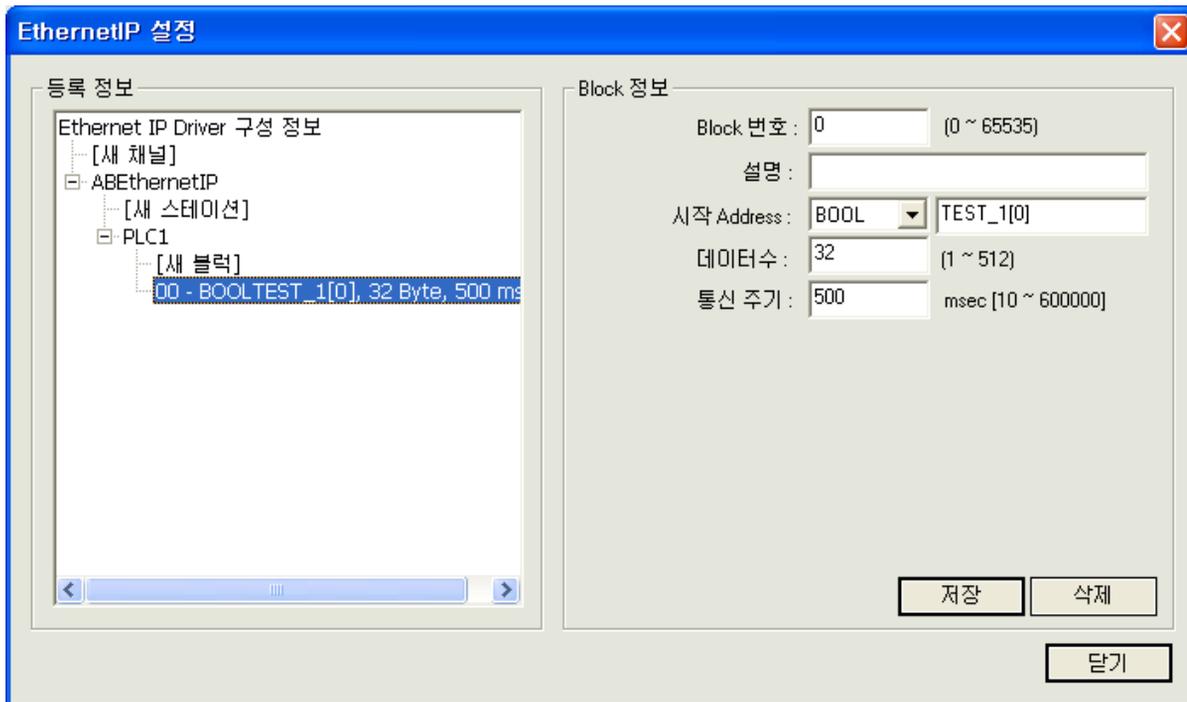
(4) 입출력 주소

(a) 형식

- 태그이름
- 태그이름[Array 번호]
- 태그이름[Array 번호].0 ~ 태그이름[Array 번호].15
- 태그이름[Array 번호].0 ~ 태그이름[Array 번호].31
- Remote1:1:I.Data.0 ~ Remote1:1:I.Data.31

- BOOL Array Type의 통신 블록 설정 및 입출력 주소 입력

RS Logix 5000 에서 태그 설정 BOOL형의 Array TEST_1[0]부터 32개를 설정하였다면, InfoU에서는 다음과 같이 설정 한다. PLC의 BOOL Array 타입TEST_1[32]의 값을 읽어 올 경우에 통신블록에 아래와 같이 입력합니다.



※ [주의] 시작 어드레스는 반드시 32의 배수가 되어야 합니다
 예) TEST_1[0], TEST_1[32], TEST_1[512]

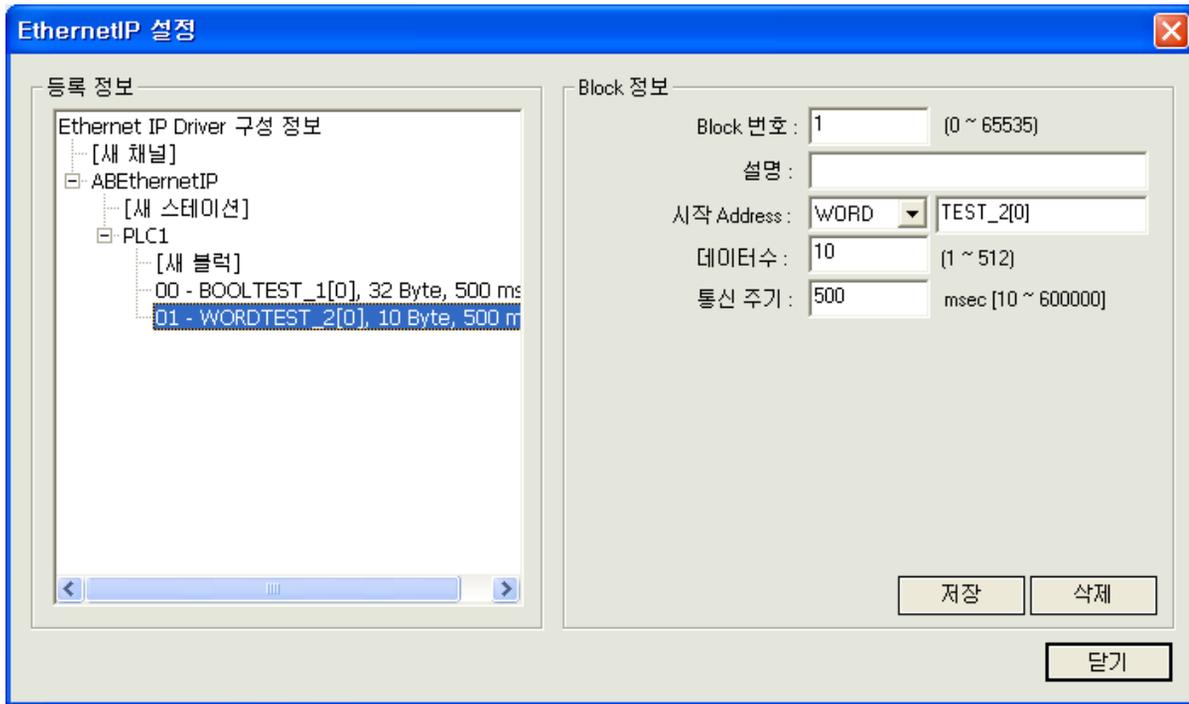
InfoU에서 데이터베이스 입력 시 아래와 같이 태그이름[Array 번호]를 입력합니다.

태그종류 : 디지털태그 BOOL형

Address : TEST_1[0], TEST_1[1], TEST_1[2], ... TEST_1[31]

- INT Array Type의 통신 블록 설정 및 입출력 주소 입력

RS Logix 5000 에서 태그 설정 INT형의 Array TEST_2[0]부터 10개를 설정하였다면, InfoU에서는 다음과 같이 설정 한다. PLC의 INT Array 타입TEST_2[10]의 값을 읽어 올 경우에 통신블록에 아래와 같이 입력합니다.



InfoU에서 데이터베이스 입력 시 아래와 같이 태그이름[Array 번호]를 입력합니다.

태그 종류 : 아날로그

데이터 타입: INT16, UINT16

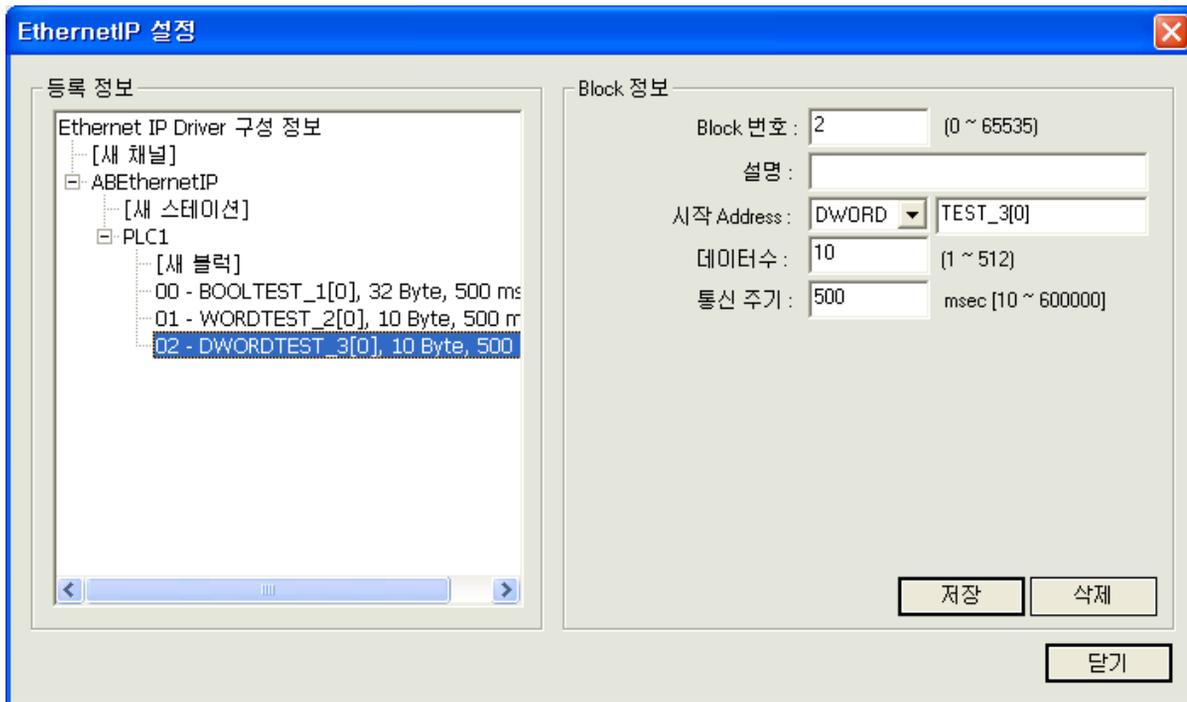
Address : TEST_2[0], TEST_2[1], ... TEST_2[9]

태그 종류 : 디지털

Address : TEST_2[0].0, TEST_2[0].1, ... TEST_2[0].15

- DINT Array Type의 통신 블록 설정 및 입출력 주소 입력

RS Logix 5000 에서 태그 설정 DINT형의 Array TEST_3[0]부터 10개를 설정하였다면, InfoU에서는 다음과 같이 설정 한다. PLC의 DINT Array 타입TEST_3[10]의 값을 읽어 올 경우에 통신블록에 아래와 같이 입력합니다.



InfoU에서 데이터베이스 입력 시 아래와 같이 태그이름[Array 번호]를 입력합니다.

태그 종류 : 아날로그

데이터 타입: INT32, UINT32

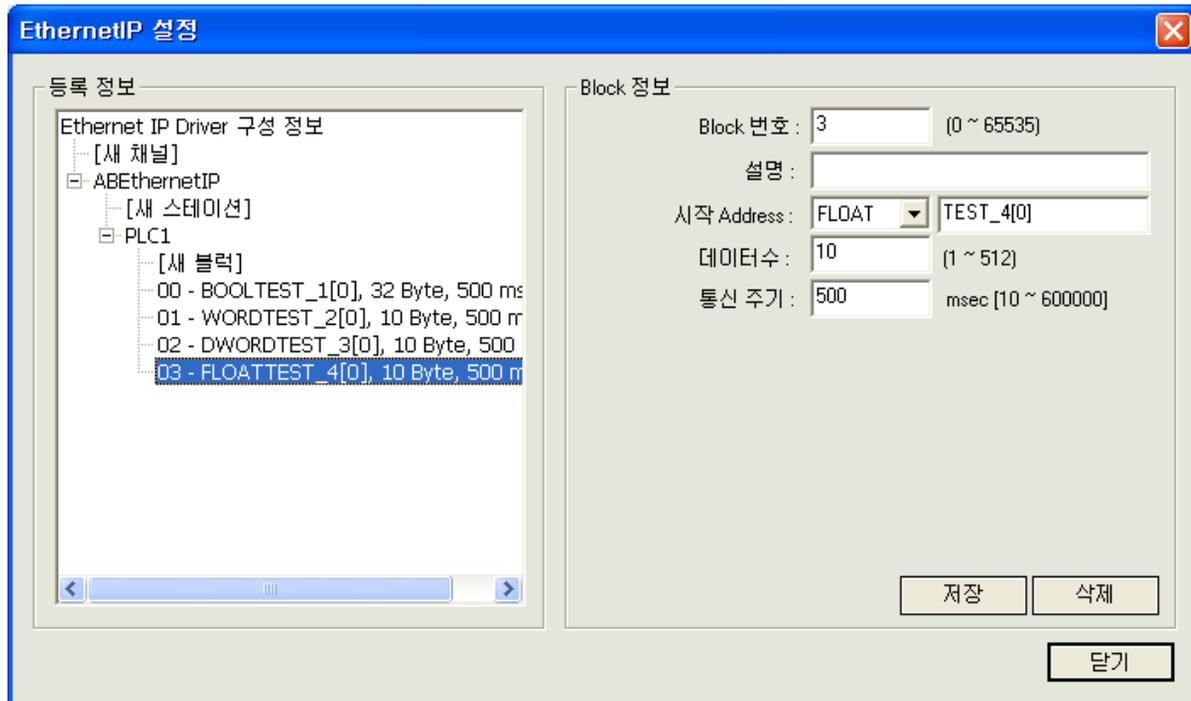
Address : TEST_3[0], TEST_3[1], ... TEST_3[9]

태그 종류 : 디지털

Address : TEST_3[0].0, TEST_3[0].1, ... TEST_3[0].31

- . REAL Array Type의 통신 블록 설정 및 입출력 주소 입력

RS Logix 5000 에서 태그 설정 REAL형의 Array TEST_4[0]부터 10개를 설정하였다면, InfoU에서는 다음과 같이 설정 한다. PLC의 REAL Array 타입TEST_4[10]의 값을 읽어 올 경우에 통신블록에 아래와 같이 입력합니다.



InfoU에서 데이터베이스 입력 시 아래와 같이 태그이름[Array 번호]를 입력합니다.

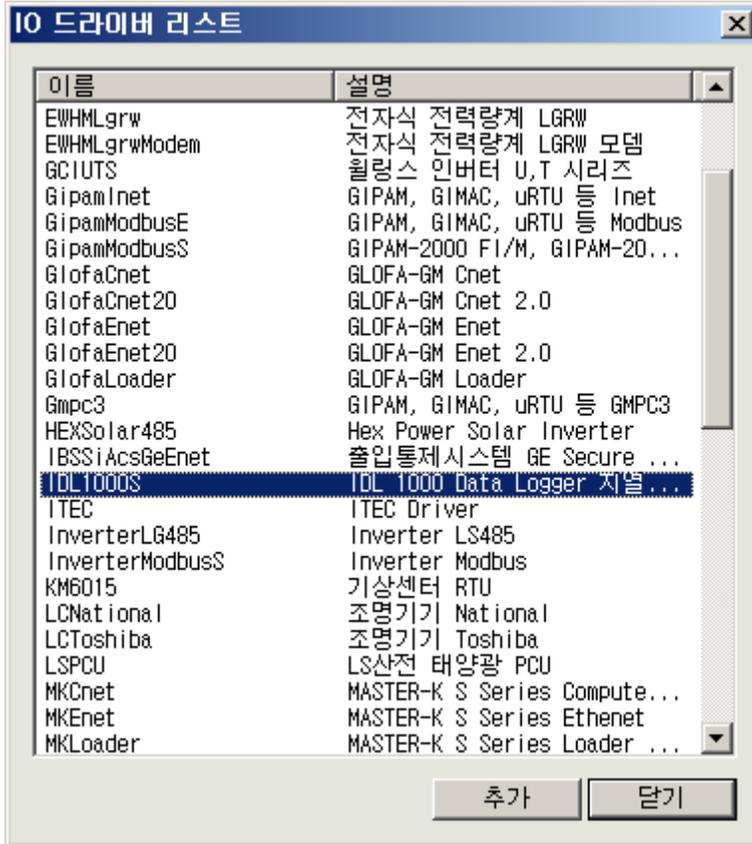
태그 종류 : 아날로그

데이터 타입: Float

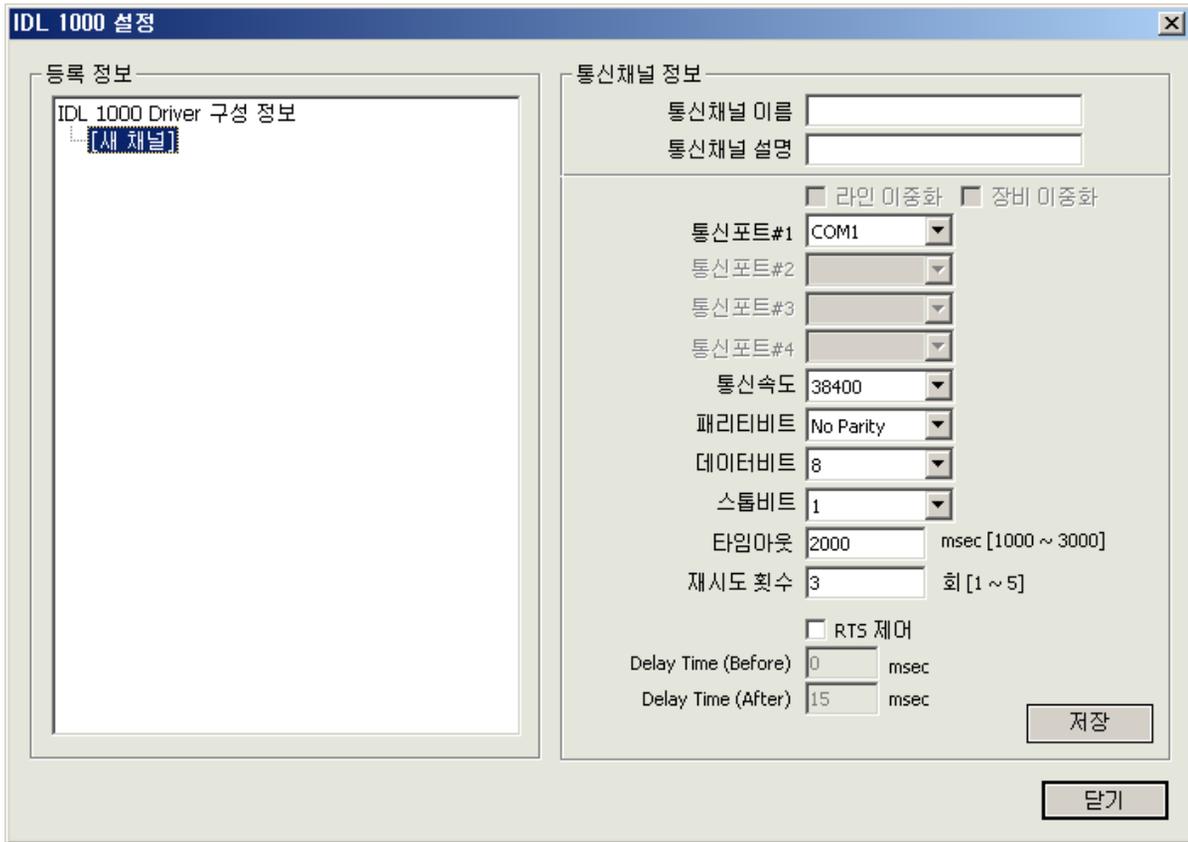
Address : TEST_4[0], TEST_4[1], ... TEST_4[9]

32. IDL1000S 드라이버

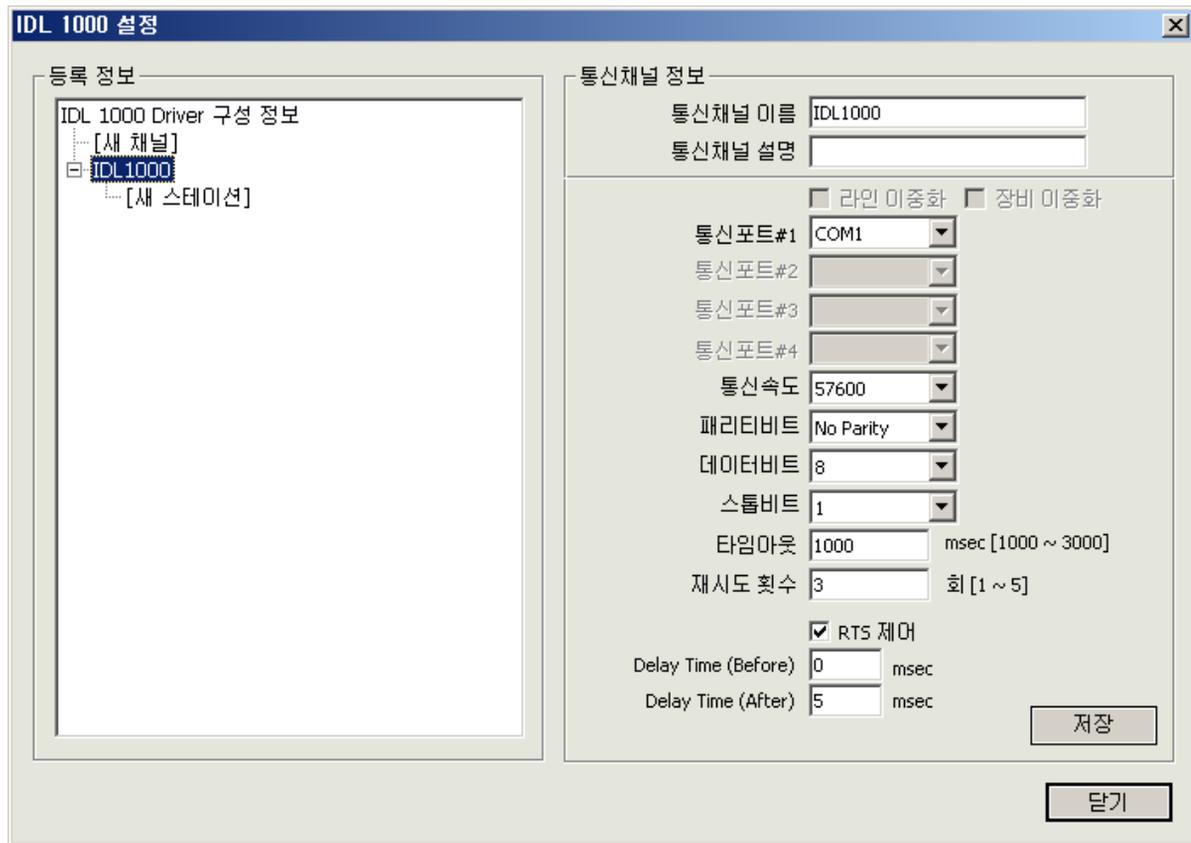
(1) 통신채널 추가



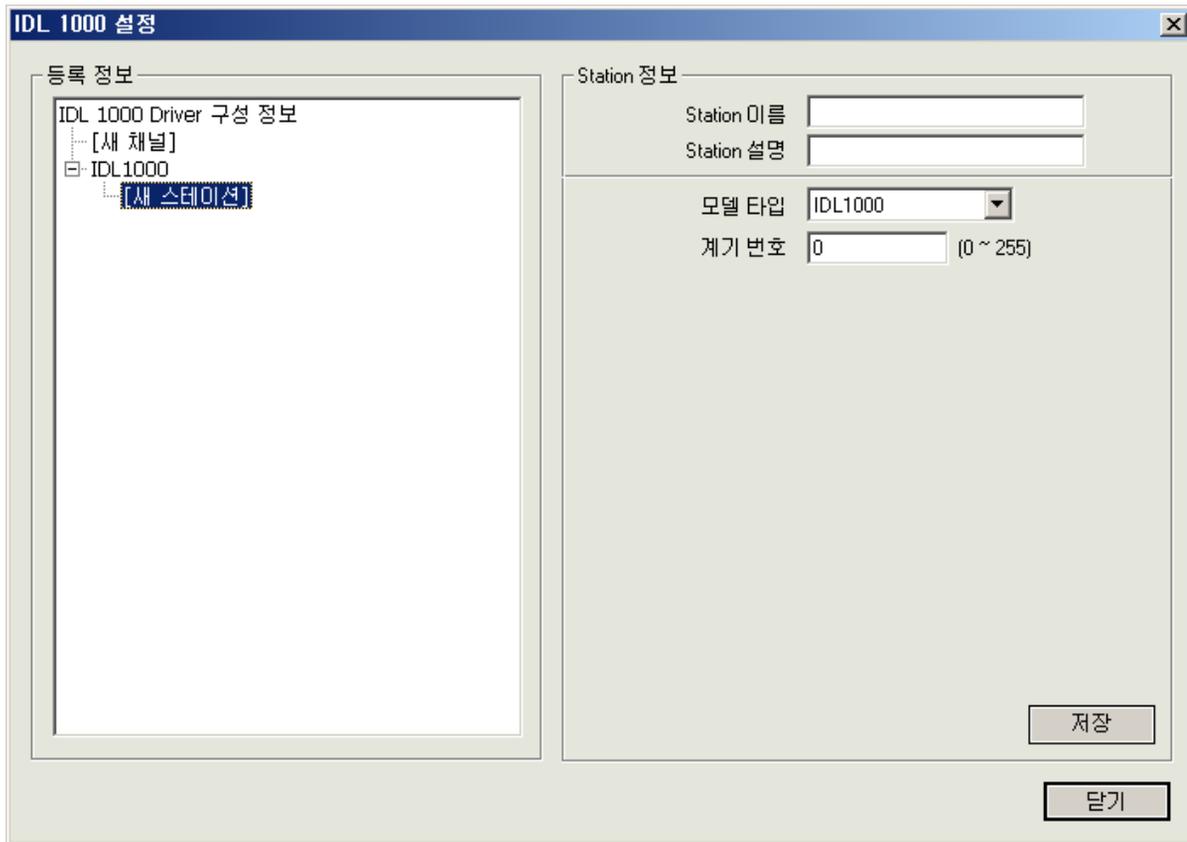
- IO 드라이버 리스트에서 “IDL1000S” 를 선택한 후 “추가” 버튼을 누릅니다.



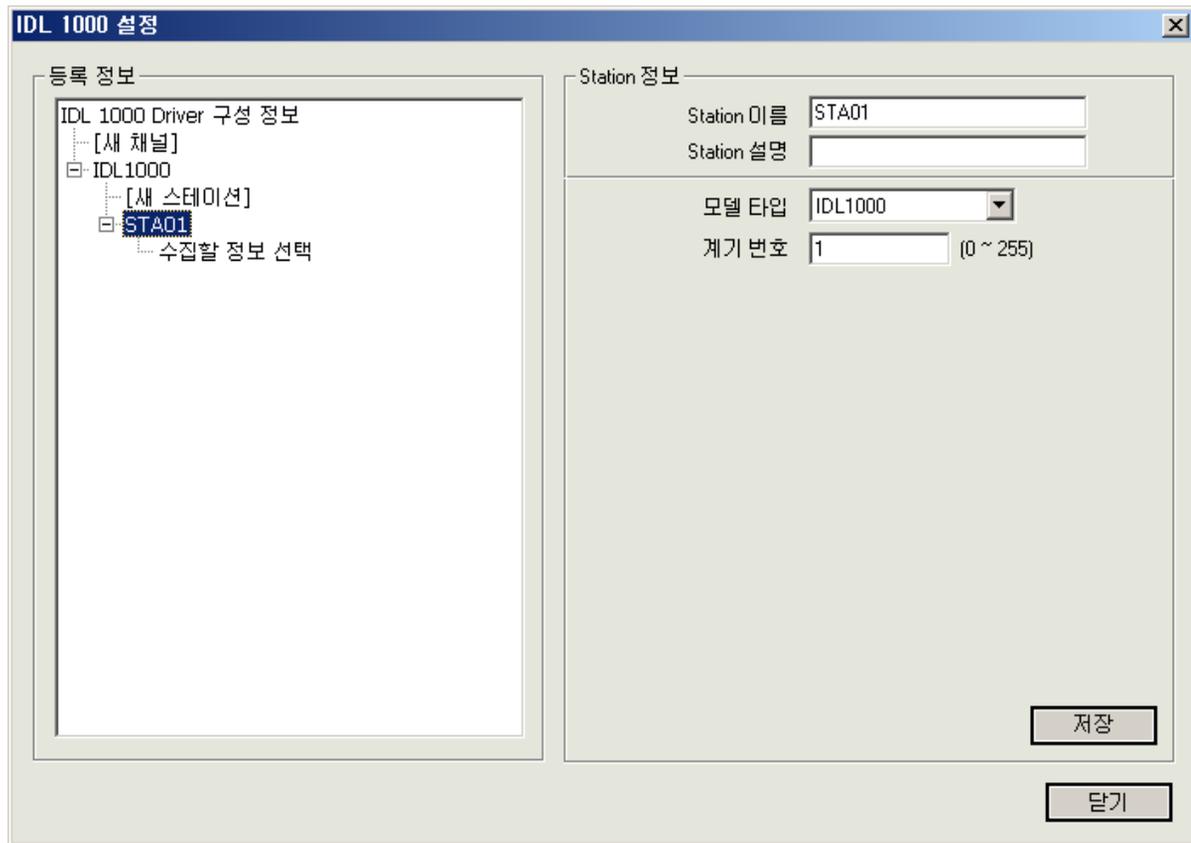
- (a) 통신채널 이름 : 통신채널 이름을 입력합니다.
- (b) 통신채널 설명 : 통신채널 설명을 입력합니다.
- (c) 통신포트 #1 : PC의 시리얼 포트를 선택합니다.
- (d) 통신속도 : 시리얼 통신 속도를 선택합니다.
- (e) 패리티비트 : 패리티비트를 선택합니다.
- (f) 데이터비트 : 데이터비트를 선택합니다.
- (g) 스톱비트 : 스톱비트를 선택합니다.
- (h) 타임아웃 : 장비에서 데이터를 요청한 후 Time Out으로 처리하게 되는 시간을 말합니다. 통신 오류를 판별하는 근거가 됩니다.
- (i) 재시도 횟수 : 통신 실패시 재시도하는 횟수를 설정합니다.
- (j) 저장 : 저장 버튼을 누르면, 설정된 통신 채널 정보가 저장되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.
- (k) RTS 제어 : 485 통신을 할 경우, RTS 제어를 사용할 것인지 선택합니다. RS232 to RS422/485 컨버터가 AUTO RTS Control 을 지원하지 않는 모델일 경우, 반드시 RTS 제어를 사용하여 통신 해야 합니다.
- (l) Delay Time : RTS 제어 타이밍을 조절하는 파라미터 입니다.



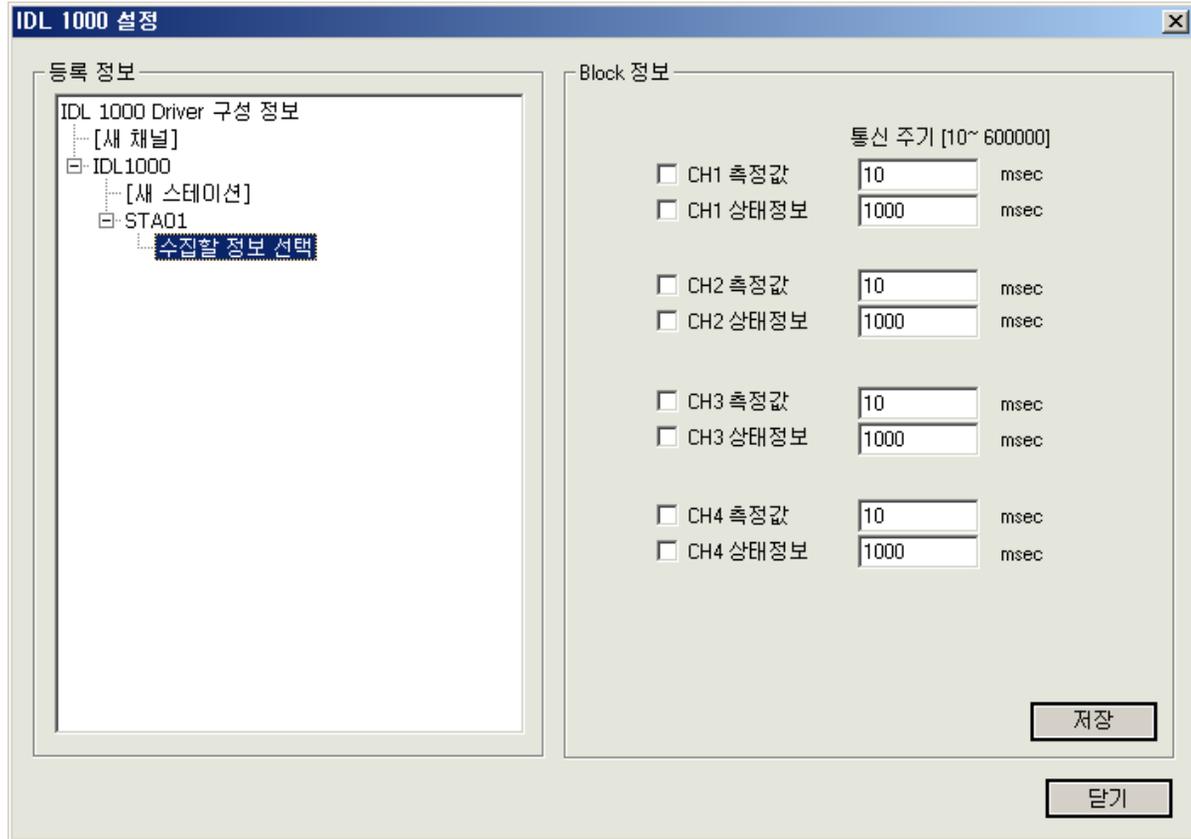
(2) Station 추가



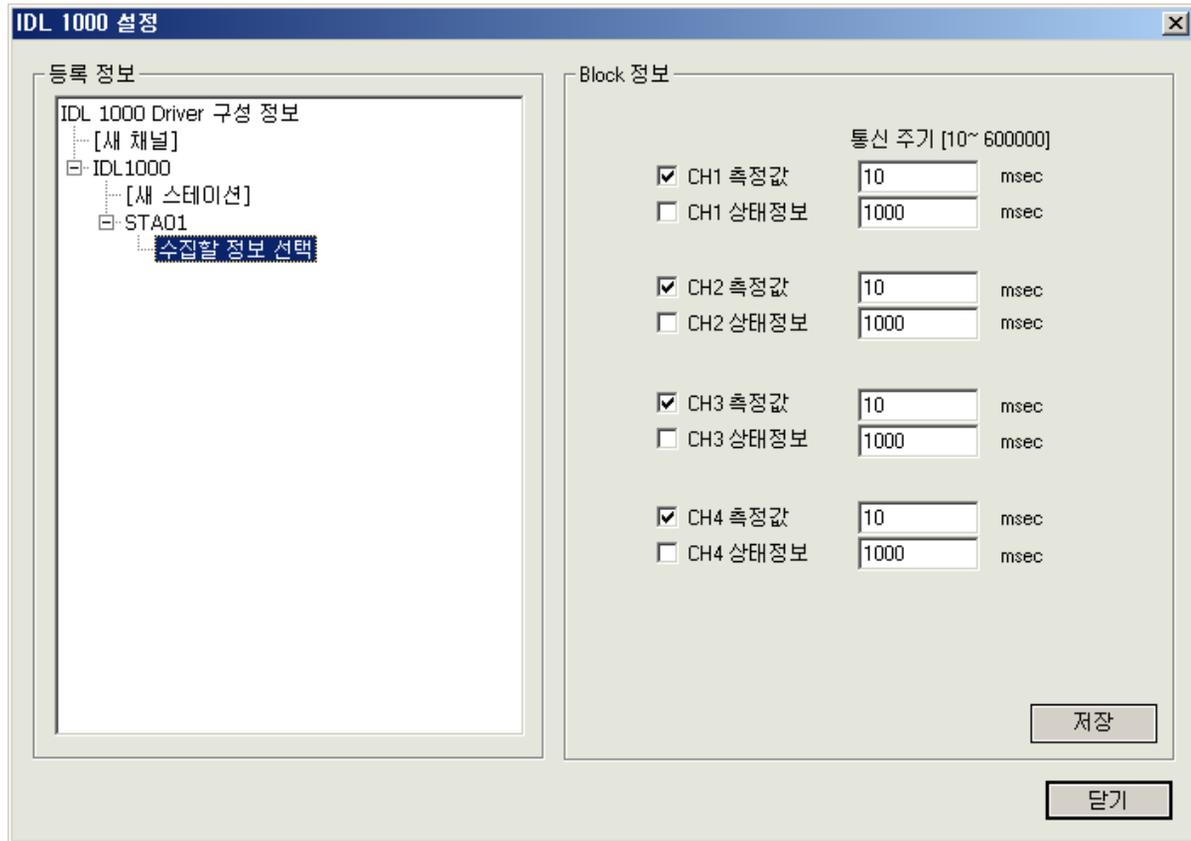
- (a) “등록 정보” 트리에서 [새 스테이션]을 선택합니다.
- (b) Station 이름 : Station 이름을 입력합니다.
- (c) Station 설명 : Station 설명을 입력합니다.
- (d) 모델 타입 : 모델 타입을 선택합니다. 기본값은 IDL 1000 입니다.
- (e) 계기 번호 : 통신할 장비의 ID를 입력합니다.
- (e) 저장 : 저장 버튼을 누르면, 설정된 Station 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(3) 수집할 정보 선택



- (a) “등록 정보” 트리에서 “수집할 정보 선택” 을 선택합니다.
- (b) CH1 측정값 : 채널1의 측정값을 수집할 것인지 선택 합니다.
- (c) CH1 상태값 : 채널1의 상태값을 수집할 것인지 선택 합니다.
- (d) CH2 측정값 : 채널2의 측정값을 수집할 것인지 선택 합니다.
- (e) CH2 상태값 : 채널2의 상태값을 수집할 것인지 선택 합니다.
- (f) CH3 측정값 : 채널3의 측정값을 수집할 것인지 선택 합니다.
- (g) CH3 상태값 : 채널3의 상태값을 수집할 것인지 선택 합니다.
- (h) CH4 측정값 : 채널4의 측정값을 수집할 것인지 선택 합니다.
- (i) CH4 상태값 : 채널4의 상태값을 수집할 것인지 선택 합니다.
- (j) 저장 : 저장 버튼을 누르면, 설정된 Block 정보가 저장이 되고, 왼쪽의 “등록 정보” 트리에 추가되어 나타납니다.



(4) 입출력 주소

IDL 1000				
구분	데이터 종류	ADDRESS	태그종류	비고
1	채널 1 측정값	CH1_PV0	Analog	Sign Integer(2 Byte) : INT16
2	채널 1 System Error Under	CH1_EUN	Digital	
3	채널 1 System Error Over	CH1_EOV	Digital	
4	채널 1 System Error Operation	CH1_EOP	Digital	
5	채널 2 측정값	CH2_PV0	Analog	Sign Integer(2 Byte) : INT16
6	채널 2 System Error Under	CH2_EUN	Digital	
7	채널 2 System Error Over	CH2_EOV	Digital	
8	채널 2 System Error Operation	CH2_EOP	Digital	
9	채널 3 측정값	CH3_PV0	Analog	Sign Integer(2 Byte) : INT16
10	채널 3 System Error Under	CH3_EUN	Digital	
11	채널 3 System Error Over	CH3_EOV	Digital	
12	채널 3 System Error Operation	CH3_EOP	Digital	
13	채널 4 측정값	CH4_PV0	Analog	Sign Integer(2 Byte) : INT16
14	채널 4 System Error Under	CH4_EUN	Digital	
15	채널 4 System Error Over	CH4_EOV	Digital	
16	채널 4 System Error Operation	CH4_EOP	Digital	
17	System Error EEPROM	COM_LERM	Digital	
18	System Error A/D	COM_LEAD	Digital	

(5) 참고 사항

- (a) 테스트시 사용한 RS232 to RS422/485 컨버터
 - . CS-428/9 (시스템 베이스)
 - . CS-428/9AT (시스템 베이스)
 - . CNV485SI (리얼시스)
- (b) 테스트 결과, AUTO RTS Control 을 지원하지 않는 컨버터를 사용할 경우, 다소 통신 성공율이 떨어지는 것을 확인 하였습니다. 가능한 AUTO RTS Control 을 지원하는 컨버터를 사용할 것을 추천합니다.
- (c) AUTO RTS Control 을 지원하지 않는 컨버터를 사용할 경우, 가능한 재시도 횟수를 높여 주시고, 아래 값의 범위 정도에서 통신 성공율이 가장 높은 수치를 사용합니다.
 - . Delay Time (Before) : 0
 - . Delay Time (After) : 1~5
- (d) 수집할 정보 선택시 각 채널의 상태 정보는, 꼭 필요한 경우에만 선택하여, 불필요한 정보를 수집하지 않도록 합니다.

부록.2. 내부함수 사용법	2-2
2.1 내부 함수 분류.....	2-2
2.1.1 항목별 리스트.....	2-2
2.1.2 사전순 리스트.....	2-5
2.2 내부 함수 설명 및 예제	2-10
2.2.1 프로젝트 관련 함수.....	2-10
2.2.2 화면 관련 함수.....	2-12
2.2.3 태그 관련 함수.....	2-15
2.2.4 스크립트 관련 함수.....	2-20
2.2.5 이벤트 관련 함수.....	2-21
2.2.6 로그 관련 함수.....	2-23
2.2.7 네트워크 관련 함수.....	2-27
2.2.8 시스템 관련 함수.....	2-31
2.2.9 레서피 관련 함수.....	2-35
2.2.10 데이터베이스 관련 함수.....	2-43
2.2.11 알람 관련 함수.....	2-43
2.2.12 외부 장치 관련 함수.....	2-55
2.2.13 사용자 관련 함수.....	2-56
2.2.14 보고서 관련 함수.....	2-58
2.2.15 이력데이터 관련 함수.....	2-60
2.2.16 I/O 드라이브 관련 함수.....	2-66
2.3 OLE AUTOMATION 사용 예	2-71

부록.2. 내부함수 사용법

2.1 내부 함수 분류

2.1.1 항목별 리스트

항목	관련 함수
프로젝트	LONG CloseProject ()
	BSTR GetProjectName ()
	BSTR GetProjectPath ()
	BSTR GetProjectType ()
	LONG OpenProject (BSTR szProjectName, BSTR szProjectPath)
	LONG QuitProject ()
화면	LONG ChangePage (BSTR szOldPage, BSTR szNewPage)
	LONG ClosePage (BSTR szPageName)
	VOID GetHwnd (ULONG* hwnd)
	VOID HardCopy ()
	BOOL IsRunning ()
	VOID OpenPage (BSTR szPageName, BSTR szPageParam)
	LONG OpenPageObject (BSTR szPageName, BSTR szObjectName)
	VOID Quit ()
태그	ULONG GetTagStatus (BSTR szTagName)
	BSTR GetTagStatusDesc (ULONG ITagStatus)
	VARIANT GetTagVal (BSTR szTagName)
	VARIANT GetTagValbyTagId (LONG ITagId)
	BSTR GetTagValnStatusByTagId (LONG ITagId, VARIANT* vtTagVal, LONG *IScanMode, LONG *IMainQ, LONG *ISubQCnt, LONG *ISubQ1, LONG *ISubQ2, LONG *ISubQ3, LONG *ISubQ4, LONG *ISubQ5)
	LONG GetTagValueStatus (BSTR szTagName, VARIANT* vtTagVal)
	LONG ResetStaticValue (BSTR szTagName)
	VOID ShowCrossReference (BSTR szTagName)
	VOID ShowRawDataInfo ()
	VOID ShowSIMCfgInfo ()
	VOID SetTagVal (BSTR szTagName, VARIANT vtTagVal)
	LONG SetTagValEx (SHORT nMode, BSTR szTagName, VARIANT vtTagVal, BSTR szClientName, ULONG hWndClientHandle)
스크립트	VOID RunScript (BSTR szScriptName)
이벤트	LONG NotifyAlarmEvent (DOUBLE lIoccur, USHORT nAlmLevel, BSTR szMessage, BSTR szArea, BSTR szUser, BSTR zInformation, BSTR szUserDefine1, BSTR szUserDefine2, BSTR szUserDefine3, BSTR szUserDefine4, BSTR szPrev, BSTR szCur, BSTR szObject)
	VOID SendEmail (BSTR szMailAddTo, BSTR szMailAddFrom, BSTR szMailTitle, BSTR szMailMsg)
	VOID SendsMS (BSTR szRecvPhoneNo, BSTR szCallBackPhoneNo, BSTR szSMSMsg)

항목	관련 함수
로그	VOID NotifyError (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
	VOID NotifyHostError (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
	VOID NotifyHostInfo (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
	VOID NotifyHostLog (USHORT nLogType, USHORT nLogCategory, BSTR szHostName, BSTR szMsg, BSTR szDesc)
	VOID NotifyHostWarn (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
	VOID NotifyInfo (USHORT nLogCategory, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
	VOID NotifyLog (USHORT nLogType, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
	VOID NotifyWarn (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
네트워크	BSTR ChangLine (BSTR szLineMode, BSTR szTargetLineIP)
	LONG ChangeProcessServer (BSTR szProcessName, BSTR szProcessIP)
	LONG ChangeServerMode (BSTR szServerMode)
	LONG GetCurLine (int* nCurLine)
	INT GetLineStatus (int nLineMode)
	BSTR GetNodeCurIP (BSTR szNodeName)
	BSTR GetServerMode ()
	BOOL IsDualLine ()
시스템	VARIANT GetSystemTime ()
	VOID ProcessSleep (LONG dwMilliseconds)
	VOID RunApp (BSTR szAppPath)
	VOID ShowAlarmPage (BSTR szAlarmTag)
	VOID Sleep (LONG dwMilliseconds)
	BSTR TimeConvert (DOUBLE dTimeValue, INT nTimeType)
레서피	LONG RecipeChangeData (BSTR ModelName, BSTR RecipeName)
	LONG RecipeChangeUnit (BSTR ModelName, BSTR UnitName)
	LONG RecipeControlSet (BSTR szModelName, BSTR szUnitName, BSTR szDataName)
	LONG RecipeControlSetEx (SHORT nMode, BSTR szClientProgName, ULONG hClientHandle, BSTR szModelName, BSTR szUnitName, BSTR szDataName, LONG* pITransNo, SHORT* pnResult)
	LONG RecipeControlWorkingSet ();
	LONG RecipeControlWorkingSetEx (SHORT nMode, BSTR szClientProgName, ULONG hClientHandle, LONG* pITransNo, SHORT* pnResult);
	LONG RecipeDeleteData (BSTR ModelName, BSTR RecipeName)
	LONG RecipeDeleteModel (BSTR ModelName)
	LONG RecipeDeleteUnit (BSTR ModelName, BSTR UnitName)
	LONG RecipeLoadDynamic (BSTR ModelName, BSTR UnitName)
	LONG RecipeLoadSet (BSTR ModelName, BSTR UnitName, BSTR RecipeName)
	LONG RecipeSaveAsArchive (BSTR ModelName, BSTR RecipeName)
	LONG RecipeSetData (BSTR ModelName, BSTR ItemName, VARIANT setValue)

부록2 내부함수 사용법

항목	관련 함수	
데이터베이스	VOID DataExportRunWork (BSTR szWorkName)	
알람	BOOL AckAllAlarm ()	
	BSTR GetAlarmCountErrInfo (LONG lCount)	
	LONG GetAlarmLogging (DOUBLE dtFrom, DOUBLE dtTo, BSTR szTag, BSTR szAlarmGr, LONG lAlarmType, LONG *lOutCount, SAFEARRAY **pszRstrTag, SAFEARRAY **pszRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **pIRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind)	
	LONG GetAlarmLoggingEx (DOUBLE dtFrom,DOUBLE dtTo,BSTR szTag, LONG nAlarmGroupCount, SAFEARRAY **pszAlarmGr, LONG lAlarmType, LONG lAlarmGrade, BSTR szAlarmKind, LONG *lOutCount, SAFEARRAY**pRstrTag,SAFEARRAY**pRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **pIRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind, SAFEARRAY **pdbRv0ccDValue, SAFEARRAY**pszRv0ccSValue, SAFEARRAY **pIAlarmGrade)	
	LONG GetAlmCount (LONG lkind, LONG lStatus)	
	LONG GetGroupUserInfoByTag (BSTR szTagName, LONG* , SAFEARRAY(BSTR) *pszUserID, SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(LONG) *pIMailCheck, SAFEARRAY(LONG) *pISMSCheck, SAFEARRAY(BSTR) *pszMailAdd, SAFEARRAY(BSTR) *pszHPhoneNo)	
	LONG GetHistAlmCount (LONG lkind, LONG lStatus)	
	LONG GetUserInfoList (LONG* lOutCount, SAFEARRAY(BSTR) *pszUserID, SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(BSTR) *pszDesc, SAFEARRAY(BSTR) *pszPassword, SAFEARRAY(LONG) *pIUserLevel, SAFEARRAY(BSTR) *pszHPhoneNo, SAFEARRAY(BSTR) *pszPhoneNo, SAFEARRAY(BSTR) *pszMailAdd)	
	LONG SetUserAlarm (BSTR szAlarmMessage, BSTR szAlarmGroup)	
	LONG SetUserAlarmEx (DOUBLE dOccur, LONG lAlarmLevel, BSTR szAlarmMessage ,BSTR szAlarmGroup, DOUBLE dCur, DOUBLE dPrev)	
	LONG SetUserEvent (BSTR szEventMessage)	
	VOID StopAlarmSound ()	
	외부 장치	VOID PlaySound (BSTR szWaveFile)
		VOID PlaySoundFile (BSTR szSoundFile)
사용자	BSTR GetUserID ()	
	LONG GetUserLevel ()	
	LONG UserLogOff ()	
	LONG UserLogOn (BSTR szUserID, BSTR szPassWord)	
보고서	VOID ReportDialog ()	
	VOID ReportListRefresh ()	
	VOID ReportPrint (BSTR szReportName)	
	VOID ReportPrintEx (BSTR szReportName, BSTR szReqTime)	
	VOID ReportSchedulerHide ()	
	VOID ReportSchedulerShow ()	

항목	관련 함수
이력데이터	VOID GetHistDataCount (BSTR szModelName, BSTR , BSTR szDataKind, DOUBLE dRefTime, LONG lCount, LONG lDir, LONG* lResultCount, SAFEARRAY(double) *psadbITimeStamp, SAFEARRAY(double) *psadbINumValue, SAFEARRAY(LONG) *psalStatus, SAFEARRAY(LONG) *psalResult)
	VOID GetHistDataPeriod (BSTR szModelName, BSTR szTagName, BSTR szDataKind, DOUBLE dFrom, DOUBLE dTo, LONG* lResultCount, SAFEARRAY(double) *psadbITimeStamp, SAFEARRAY(double) *psadbINumValue, SAFEARRAY(LONG) *psalStatus, SAFEARRAY(LONG) *psalResult)
	LONG GetLoggingModelList (LONG *lOutCount ,SAFEARRAY(BSTR) *psaszModelName, SAFEARRAY(BSTR) *psaszModelDesc, SAFEARRAY(BSTR) *psaszModelType, SAFEARRAY(BSTR) *psaszArchivingCycle, SAFEARRAY(BSTR) *psaszSegmentPeriod)
	LONG GetLoggingTag (BSTR szModelName, LONG *lOutCount ,SAFEARRAY(BSTR) *psaszTagName, SAFEARRAY(BSTR) *psaszTagType)
	LONG SetHistData (BSTR szModelName, BSTR szTagName, BSTR szDataKind, LONG lDataCount, SAFEARRAY(double) *psadbISaveTime, SAFEARRAY(double) *psadbINumValue, SAFEARRAY(LONG) *psalTagStatus, SAFEARRAY(LONG) *psalResult)
I/O 드라이브	VOID ClearStaStatusCount (BSTR szChannelName, BSTR szStaNamE)
	VOID EnableStation (BSTR szChannelName, BSTR szStaNamE, LONG lMode)
	LONG GetCountReadSuccess (BSTR szChannelName, BSTR szStaNamE)
	LONG GetCountReadTotal (BSTR szChannelName, BSTR szStaNamE)
	LONG GetCountWriteSuccess (BSTR szChannelName, BSTR szStaNamE)
	LONG GetCountWriteTotal (BSTR szChannelName, BSTR szStaNamE)
	LONG GetStatusRunMode (BSTR szChannelName, BSTR szStaNamE)
	LONG GetStatusStaOK (BSTR szChannelName, BSTR szStaNamE)
	VOID ShowNetworkStatus ()
	VOID ShowStationInfo ()

2.1.2 사전순 리스트

알파벳	함수명	항목
A	BOOL AckAllAlarm ()	알람
C	BSTR ChangLine (BSTR szLineMode, BSTR szTargetLineIP)	네트워크
	LONG ChangeProcessServer (BSTR szProcessName, BSTR szProcessIP)	네트워크
	LONG ChangeServerMode (BSTR szServerMode)	네트워크
	LONG ChangePage (BSTR szOldPage, BSTR szNewPage)	화면
	VOID ClearStaStatusCount (BSTR szChannelName, BSTR szStaNamE)	I/O 드라이브
	LONG ClosePage (BSTR szPageName)	화면 함수
	LONG CloseProject ()	프로젝트
D	VOID DataExportRunWork (BSTR szWorkName)	데이터 베이스
E	VOID EnableStation (BSTR szChannelName, BSTR szStaNamE, LONG lMode)	I/O 드라이브

부록2 내부함수 사용법

알파벳	함수명	항목
G	LONG GetAlmCount (LONG lkind, LONG lStatus)	알람
	BSTR GetAlarmCountErrInfo (LONG lCount)	알람
	LONG GetAlarmLogging (DOUBLE dtFrom, DOUBLE dtTo, BSTR szTag, BSTR szAlarmGr, LONG lAlarmType, LONG *lOutCount, SAFEARRAY **pszRstrTag, SAFEARRAY **pszRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **plRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind)	알람
	LONG GetAlarmLoggingEx (DOUBLE dtFrom,DOUBLE dtTo,BSTR szTag, LONG nAlarmGroupCount, SAFEARRAY **pszAlarmGr, LONG lAlarmType, LONG lAlarmGrade, BSTR szAlarmKind, LONG *lOutCount, SAFEARRAY**pRstrTag,SAFEARRAY**pRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **plRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind, SAFEARRAY **pdbRvOccDValue, SAFEARRAY**pszRvOccSValue, SAFEARRAY **plAlarmGrade)	알람
	LONG GetCountReadSuccess (BSTR szChannelName, BSTR szStaName)	I/O 드라이브
	LONG GetCountReadTotal (BSTR szChannelName, BSTR szStaName)	I/O 드라이브
	LONG GetCountWriteSuccess (BSTR szChannelName, BSTR szStaName)	I/O 드라이브
	LONG GetCountWriteTotal (BSTR szChannelName, BSTR szStaName)	I/O 드라이브
	LONG GetCurLine (int* nCurLine)	네트워크
	LONG GetGroupUserInfoByTag (BSTR szTagName, LONG* , SAFEARRAY(BSTR) *pszUserID, SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(LONG) *plMailCheck, SAFEARRAY(LONG) *plSMSMCheck, SAFEARRAY(BSTR) *pszMailAdd, SAFEARRAY(BSTR) *pszHPhoneNo)	알람
	LONG GetHistAlmCount (LONG lkind, LONG lStatus)	알람
	VOID GetHistDataCount (BSTR szModelName, BSTR , BSTR szDataKind, DOUBLE dRefTime, LONG lCount, LONG lDir, LONG* lResultCount, SAFEARRAY(double) *psadbITimeStamp, SAFEARRAY(double) *psadbINumValue, SAFEARRAY(LONG) *psalStatus, SAFEARRAY(LONG) *psalResult)	이력데이터
	VOID GetHistDataPeriod (BSTR szModelName, BSTR szTagName, BSTR szDataKind, DOUBLE dFrom, DOUBLE dTo, LONG* lResultCount, SAFEARRAY(double) *psadbITimeStamp, SAFEARRAY(double) *psadbINumValue, SAFEARRAY(LONG) *psalStatus, SAFEARRAY(LONG) *psalResul)	이력데이터
	VOID GetHMND (ULONG* hwnd)	화면
	INT GetLineStatus (int nLineMode)	네트워크

알파벳	함수명	항목
G	LONG GetLoggingModelList (LONG *lOutCount ,SAFEARRAY(BSTR) *psaszModelName, SAFEARRAY(BSTR) *psaszModelDesc, SAFEARRAY(BSTR) *psaszModelType, SAFEARRAY(BSTR) *psaszArchivingCycle, SAFEARRAY(BSTR) *psaszSegmentPeriod)	이력데이터
	LONG GetLoggingTag (BSTR szModelName, LONG *lOutCount ,SAFEARRAY(BSTR) *psaszTagName, SAFEARRAY(BSTR) *psaszTagType)	이력데이터
	BSTR GetNodeCurIP (BSTR szNodeName)	네트워크
	BSTR GetProjectName ()	프로젝트
	BSTR GetProjectPath ()	프로젝트
	BSTR GetProjectType ()	프로젝트
	BSTR GetServerMode ()	네트워크
	LONG GetStatusRunMode (BSTR szChannelName, BSTR szStaName)	I/O 드라이브
	LONG GetStatusStaOK (BSTR szChannelName, BSTR szStaName)	I/O 드라이브
	VARIANT GetSystemTime ()	시스템
	ULONG GetTagStatus (BSTR szTagName)	태그
	BSTR GetTagStatusDesc (ULONG lTagStatus)	태그
	VARIANT GetTagVal (BSTR szTagName)	태그
	VARIANT GetTagValbyTagId (LONG lTagId)	태그
	BSTR GetTagValInStatusByTagId (LONG lTagId, VARIANT* vtTagVal, LONG *lScanMode, LONG *lMainQ, LONG *lSubQCnt, LONG *lSubQ1, LONG *lSubQ2, LONG *lSubQ3, LONG *lSubQ4, LONG *lSubQ5)	태그
	LONG GetTagValueStatus (BSTR szTagName, VARIANT* vtTagVal)	태그
	BSTR GetUser ID ()	사용자
	LONG GetUserInfoList (LONG* lOutCount, SAFEARRAY(BSTR) *pszUserID, SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(BSTR) *pszDesc, SAFEARRAY(BSTR) *pszPassword, SAFEARRAY(LONG) *plUserLevel, SAFEARRAY(BSTR) *pszHPHoneNo, SAFEARRAY(BSTR) *pszPhoneNo, SAFEARRAY(BSTR) *pszMailAdd)	알람
	LONG GetUserLevel ()	사용자
	H	VOID HardCopy ()
I	BOOL IsDualLine ()	네트워크
	BOOL IsRunning ()	화면
	LONG IsHoliday (BSTR szDateTime)	시스템

부록2 내부함수 사용법

알파벳	함수명	항목
N	LONG NotifyAlarmEvent (DOUBLE lIoccur, USHORT nAlmLevel, BSTR szMessage, BSTR szArea, BSTR szUser, BSTR zInformation, BSTR szUserDefine1, BSTR szUserDefine2, BSTR szUserDefine3, BSTR szUserDefine4, BSTR szPrev, BSTR szCur, BSTR szObject)	이벤트 함수
	VOID NotifyError (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	로그
	VOID NotifyHostError (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	로그
	VOID NotifyHostInfo (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	로그
	VOID NotifyHostLog (USHORT nLogType, USHORT nLogCategory, BSTR szHostName, BSTR szMsg, BSTR szDesc)	로그
	VOID NotifyHostWarn (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	로그
	VOID NotifyInfo (USHORT nLogCategory (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	로그
	VOID NotifyLog (USHORT nLogType, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	로그
	VOID NotifyWarn (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)	로그
	O	VOID OpenPage (BSTR szPageName, BSTR szPageParam)
LONG OpenPageObject (BSTR szPageName, BSTR szObjectName)		화면
LONG OpenProject (BSTR szProjectName, BSTR szProjectPath)		프로젝트
P	VOID PlaySound (BSTR szWaveFile))	외부장치
	VOID PlaySoundFile (BSTR szSoundFile)	외부장치
	VOID ProcessSleep (LONG dwMilliseconds)	시스템
Q	VOID Quit ()	화면
	LONG QuitProject ()	프로젝트
R	LONG RecipeChangeData (BSTR ModelName, BSTR RecipeName)	레서피
	LONG RecipeChangeUnit (BSTR ModelName, BSTR UnitName)	레서피
	LONG RecipeControlSet (BSTR szModelName, BSTR szUnitName, BSTR szDataName)	레서피
	LONG RecipeControlSetEx (SHORT nMode, BSTR szClientProgName, ULONG hClientHandle, BSTR szModelName, BSTR szUnitName, BSTR szDataName, LONG* pITransNo, SHORT* pnResult)	레서피
	LONG RecipeControlWorkingSet ();	레서피
	LONG RecipeControlWorkingSetEx (SHORT nMode, BSTR szClientProgName, ULONG hClientHandle, LONG* pITransNo, SHORT* pnResult);	레서피
	LONG RecipeDeleteData (BSTR ModelName, BSTR RecipeName)	레서피
	LONG RecipeDeleteModel (BSTR ModelName)	레서피
	LONG RecipeDeleteUnit (BSTR ModelName, BSTR UnitName)	레서피
	LONG RecipeLoadDynamic (BSTR ModelName, BSTR UnitName)	레서피
	LONG RecipeLoadSet (BSTR ModelName, BSTR UnitName, BSTR RecipeName)	레서피
	LONG RecipeSaveAsArchive (BSTR ModelName, BSTR RecipeName)	레서피

알파벳	함수명	항목
R	LONG RecipeSetData (BSTR ModelName, BSTR ItemName, VARIANT setValue)	레서피
	VOID ReportDialog ()	보고서
	VOID ReportListRefresh ()	보고서
	VOID ReportPrint (BSTR szReportName)	보고서
	VOID ReportPrintEx (BSTR szReportName, BSTR szReqTime)	보고서
	VOID ReportSchedulerHide ()	보고서
	VOID ReportSchedulerShow ()	보고서
	LONG ResetStaticValue (BSTR szTagName)	태그
	VOID RunApp (BSTR szAppPath)	시스템
	VOID RunScript (BSTR szScriptName)	스크립트
S	VOID SendEmail (BSTR szMailAddTo, BSTR szMailAddFrom, BSTR szMailTitle, BSTR szMailMsg)	이벤트
	VOID SendSMS (BSTR szRecvPhoneNo, BSTR szCallBackPhoneNo, BSTR szSMSMsg)	이벤트
	LONG SetHistData (BSTR szModelName, BSTR szTagName, BSTR szDataKind, LONG IDataCount, SAFEARRAY(double) *psadbISaveTime, SAFEARRAY(double) *psadbINumValue, SAFEARRAY(LONG) *psalTagStatus, SAFEARRAY(LONG) *psalResult)	이력데이터
	VOID SetTagVal (BSTR szTagName, VARIANT vtTagVal)	태그
	LONG SetTagValEx (SHORT nMode, BSTR szTagName, VARIANT vtTagVal, BSTR szClientName, ULONG hClientHandle)	태그
	LONG SetUserAlarm (BSTR szAlarmMessage, BSTR szAlarmGroup)	알람
	LONG SetUserAlarmEx (DOUBLE dOccur, LONG IAlarmLevel, BSTR szAlarmMessage, BSTR szAlarmGroup, DOUBLE dCur, DOUBLE dPrev)	알람
	LONG SetUserEvent (BSTR szEventMessage)	알람
	VOID ShowAlarmPage (BSTR szAlarmTag)	시스템
	VOID ShowCrossReference (BSTR szTagName)	태그
	VOID ShowNetworkStatus ()	I/O 드라이브
	VOID ShowRawDataInfo ()	태그
	VOID ShowSIMCfgInfo ()	태그
	VOID ShowStationInfo ()	I/O 드라이브
	VOID Sleep (LONG dwMilliseconds)	시스템
	VOID StopAlarmSound ()	알람
	T	BSTR TimeConvert (DOUBLE dTimeValue, INT nTimeType)
U	LONG UserLogOff ()	사용자
	LONG UserLogOn (BSTR szUserID, BSTR szPassWord)	사용자

2.2 내부 함수 설명 및 예제

2.2.1 프로젝트 관련 함수

항목	내용
형식	LONG CloseProject ()
종류	프로젝트 관련 함수
내용	프로젝트를 종료합니다.
예제	<pre> Sub Main Dim lRet As Long lRet = ole.CloseProject() If lRet <> 0 Then MsgBox "CloseProject() False" Else MsgBox "CloseProject() OK" End If End Sub </pre>

항목	내용
형식	BSTR GetProjectName ()
종류	프로젝트 관련 함수
내용	현재 실행되고 있는 프로젝트의 이름을 알려줍니다.
예제	<pre> Sub Main Dim PrjName As String PrjName = ole.GetProjectName() If lRet <> 0 Then MsgBox "GetProjectName()False" Else MsgBox "GetProjectName() OK " + PrjName End If End Sub </pre>

항목	내용
형식	BSTR GetProjectPath ()
종류	프로젝트 관련 함수
내용	현재 실행되고 있는 프로젝트의 경로를 알려줍니다.
예제	<pre> Sub Main Dim PrjName As String PrjPath = ole.GetProjectPath() If lRet <> 0 Then MsgBox "GetProjectPath() False" Else MsgBox "GetProjectPath() OK " + PrjPath End If End Sub </pre>

항목	내용
형식	BSTR GetProjectType ()
종류	프로젝트 관련 함수
내용	현재 실행되고 있는 프로젝트 타입을 알려줍니다.
예제	<pre>Sub Main Dim PrjType As String PrjType = ole.GetProjectType MsgBox "GetProjectType : " + PrjType End Sub</pre>

항목	내용
형식	LONG OpenProject (BSTR szProjectName, BSTR szProjectPath)
종류	프로젝트 관련 함수
내용	<p>프로젝트를 기동합니다.</p> <p><i>szProjectName</i> : 프로젝트 이름 <i>szProjectPath</i> : 프로젝트 경로</p>
예제	<pre>Sub Main Dim IRet As Long IRet = ole.OpenProject("프로젝트명", "프로젝트경로") If IRet <> 0 Then MsgBox "OpenProject() False" Else MsgBox "OpenProject() OK" End If End Sub</pre>

항목	내용
형식	LONG QuitProject ()
종류	프로젝트 관련 함수
내용	프로젝트를 종료합니다.
예제	<pre>Sub Main Dim IRet As Long IRet = ole.QuitProject() If IRet <> 0 Then MsgBox "QuitProject () False" Else MsgBox "QuitProject () OK" End If End Sub</pre>

2.2.2 화면 관련 함수

항목	내용
형식	LONG ChangePage (BSTR szOldPage, BSTR szNewPage)
종류	화면 관련 함수
내용	런타임시 지정된 화면으로 교체해 줍니다. szOldPage: 그래픽 화면 이전 페이지 szNewPage: 그래픽 화면 새로운 페이지
예제	<pre> Sub Main Dim lRet As Long lRet = ole.ChangePage("화면1.ivd", "화면2.ivd") If lRet <> 0 Then MsgBox "ChangePage() False" Else MsgBox "ChangePage() OK" End If End Sub </pre>

항목	내용
형식	LONG ClosePage (BSTR szPageName)
종류	화면 관련 함수
내용	런타임시 지정한 화면을 닫습니다. szPageName : 페이지 이름
예제	<pre> Sub Main Dim lRet As Long lRet = ole.ClosePage("화면.ivd") If lRet <> 0 Then MsgBox "ClosePage() False" Else MsgBox "ClosePage() OK" End If End Sub </pre>

항목	내용
형식	VOID GetHWND (ULONG* hwnd)
종류	화면 관련 함수
내용	런타임 프로그램의 핸들을 알려줍니다. <i>hwnd</i> : 런타임 프로그램의 핸들
예제	<pre> Sub Main Dim lRet As Long lRet = ole.GetHWND If lRet < 0 Then MsgBox "GetHWND()OK" + CStr(lRet) Else MsgBox "GetHWND() False => lRet : " + CStr(lRet) End If End Sub </pre>

항목	내용
형식	VOID HardCopy ()
종류	화면 관련 함수
내용	그래픽 런타임 프로그램의 메인 윈도우 영역의 모든 화면을 프린트로 출력합니다.
예제	<pre> Sub Main ole.HardCopy End Sub </pre>

항목	내용
형식	BOOL IsRunning ()
종류	화면 관련 함수
내용	현재 런타임 여부를 알려줍니다.
예제	<pre> Sub Main Dim bRet As Boolean bRet = ole.IsRunning If bRet = True Then MsgBox "IsRunning OK" Else MsgBox "IsRunning FALSE" End If End Sub </pre>

부록2 내부함수 사용법

항목	내용
형식	VOID OpenPage (BSTR szPageName, BSTR szPageParam)
종류	화면 관련 함수
내용	런타임시 지정된그래픽 화면을 보여줍니다. <i>szPageName</i> : 그래픽 화면 이름 <i>szPageParam</i> : 템플릿 화면에서 태그를 대체할 문자열을 지정.
예제	<pre> Sub Main Dim IRet As Long IRet = ole. OpenPage ("PAGE1.ivd", "\$GR=PLC1") If IRet <> 0 Then MsgBox "OpenPage() False" Else MsgBox "OpenPage () OK" End If End Sub </pre>

항목	내용
형식	LONG OpenPageObject (BSTR szPageName, BSTR szObjectName)
종류	화면 관련 함수
내용	런타임시 지정된그래픽 화면을 보여줍니다. <i>szPageName</i> : 그래픽 화면 이름 <i>szObjectName</i> : 그래픽 화면에 존재하는 객체의 이름
예제	<pre> Sub Main Dim IRet As Long IRet = ole.OpenPageObject("화면.ivd", "객체명") If IRet <> 0 Then MsgBox "OpenPageObject() False" Else MsgBox "OpenPageObject() OK" End If End Sub </pre>

항목	내용
형식	VOID Quit ()
종류	화면 관련 함수
내용	런타임 프로그램인 InfouX.exe 를 종료합니다. .
예제	<pre> Sub Main ole.Quit End Sub </pre>

2.2.3 태그 관련 함수

항목	내용
형식	ULONG GetTagStatus (BSTR szTagName)
종류	태그 관련 함수
내용	현재 태그 상태를 알려줍니다. <i>szTagName</i> : 태그 이름
예제	<pre> Sub Main Dim IRet As Long IRet = ole.GetTagStatus("SCAnaTag") If IRet <> 0 Then MsgBox "GetTagStatus() OK" Else MsgBox "GetTagStatus() False" End If End Sub </pre>

항목	내용
형식	BSTR GetTagStatusDesc (ULONG ITagStatus)
종류	태그 관련 함수
내용	태그 상태 값에 따른 설명을 문자열로 알려줍니다. <i>ITagStatus</i> : 태그 상태 값
예제	<pre> Sub Main Dim IRet As Long Dim szDesc As String IRet = ole.GetTagValueStatus("태그명") szDesc = ole.GetTagStatusDesc(IRet) MsgBox szDesc End Sub </pre>

항목	내용
형식	VARIANT GetTagVal (BSTR szTagName)
종류	태그 관련 함수
내용	태그값을 가지고 옵니다. <i>szTagName</i> : 태그 이름
예제	<pre> Sub Main Dim Value As Variant Value = 12.0 Value = ole.GetTagVal("FA0A_F_EHV02.AR") MsgBox "FA0A_F_EHV02.AR Value = " + CStr(Value) End Sub </pre>

부록2 내부함수 사용법

항목	내용
형식	VARIANT GetTagValbyTagId (LONG ITagId)
종류	태그 관련 함수
내용	태그 아이디(ID)로 태그값을 알려준다 <i>/TagId</i> : 태그 ID(인덱스)
예제	<pre> Sub Main Dim Value As Variant Dim Tagid As Long Value = ole.GetTagValbyTagId(Tagid) MsgBox "SCDigTag Value = " + Value End Sub </pre>

항목	내용
형식	BSTR GetTagValnStatusByTagId (LONG ITagId, VARIANT* vtTagVal, LONG *IScanMode, LONG *IMainQ, LONG *ISubQCnt, LONG *ISubQ1, LONG *ISubQ2, LONG *ISubQ3, LONG *ISubQ4, LONG *ISubQ5)
종류	태그 관련 함수
내용	태그 아이디(ID)로 태그 상태값을 알려준다. 리턴 값으로는 상태값에 대한 설명을 넘겨줍니다. <i>/TagId</i> : 태그 ID(인덱스) <i>vtTagVal</i> : 태그 값 <i>/ScanMode</i> : 태그 상태 대표 모드 (1: SCANSOP, 2 : SCANSTART) <i>/MainQ</i> : 대표 품질 코드 <i>/SubQCnt</i> : 상세 품질 코드 개수 <i>/SubQ1</i> : 상세 품질 코드 1 <i>/SubQ2</i> : 상세 품질 코드 2 <i>/SubQ3</i> : 상세 품질 코드 3 <i>/SubQ4</i> : 상세 품질 코드 4 <i>/SubQ5</i> : 상세 품질 코드 5

항목	내용
형식	LONG GetTagValueStatus (BSTR szTagName, VARIANT *vtTagVal)
종류	태그 관련 함수
내용	지정된 태그의 값과 상태값을 알려줍니다. 이때 리턴 값으로 상태값을 알려줍니다. <i>szTagName</i> : 태그 이름 <i>vtTagVal</i> : 태그값
예제	<pre> Sub Main Dim IRet As Long Dim TagValue As Variant IRet = ole.GetTagValueStatus("태그명" , TagValue) If IRet <> 0 Then MsgBox " GetTagValueStatus () : SCAN START , Value" + CStr(TagValue) Else MsgBox " GetTagValueStatus () : SCAN STOP False" End If End Sub </pre>

항목	내용
형식	LONG ResetStaticValue (BSTR szTagName)
종류	태그 관련 함수
내용	지정된 태그의 통계 값을 리셋합니다. <i>szTagName</i> : 태그 명
예제	<pre> Sub Main Dim IRet As Long IRet = ole.ResetStaticValue("SCAnaTag") If (IRet = 0) Then MsgBox "ResetStaticValue() OK " Else MsgBox "ResetStaticValue() FALSE " End If End Sub </pre>

항목	내용
형식	LONG SetTagValEx (SHORT nMode, BSTR szTagName, VARIANT vtTagVal, BSTR szClientName, ULONG hClientHandle)
종류	태그 관련 함수
내용	<p>지정된 태그값을 제어합니다. (nMode : 1: no return, 2: Return), 제어 결과를 받기위한 프로그램은 자신의 핸들값을 넘겨주어야 합니다. 함수의 리턴값은 제어 리턴 결과를 받기 위한 태그에 할당된 유일한 트랜잭션 번호입니다. (태그: 트랙잭션번호)</p> <p><i>nMode</i>: 제어 모드 nMode : 1: no return, 2: Return <i>szTagName</i>: 태그 이름 <i>vtTagVal</i>: 제어로 보내는 태그 값 <i>szClientName</i> : 제어를 내리는 클라이언트 프로그램 명 <i>hClientHandle</i>: 제어를 내리는 클라이언트의 윈도우 핸들</p>
예제	<pre> Sub Main Dim Value As Variant Dim TransNo As Long MsgBox " Before : SetTagValEx(SCDigTag) Value = " + ole.GetTagVal("태그명") Value = 1 TransNo = ole.SetTagValEx (1, "태그명", Value , "스크립트" , NULL) MsgBox " After : SetTagValEx(태그명) Value = " + ole.GetTagVal("태그명") End Sub </pre>

2.2.4 스크립트 관련 함수

항목	내용
형식	VOID RunScript (BSTR szScriptName)
종류	스크립트 관련 함수
내용	스크립트를 실행합니다. <i>szScriptName</i> : 스크립트 이름
예제	<pre> Sub Main Dim scTime As Variant ole.RunScript("스크립트명") End Sub </pre>

항목	내용
형식	VOID StopScript (BSTR szScriptName)
종류	스크립트 관련 함수
내용	스크립트를 중지합니다. <i>szScriptName</i> : 스크립트 이름
예제	<pre> Sub Main Dim scTime As Variant ole.StopScript("스크립트명") End Sub </pre>

항목	내용
형식	VOID TimeOutMsgBox (BSTR Msg, BSTR Title, Int nSec)
종류	스크립트 관련 함수
내용	설정된 시간만 표시되는 메시지 박스를 표시합니다. 스크립트에서 일반 메시지 박스를 사용하는 경우 스크립트함수가 멈추기 때문에 주의해야합니다. <i>Msg</i> : 메시지 내용 <i>Title</i> : 메시지박스의 제목 <i>nSec</i> : 메시지박스 유지시간
예제	<pre> Sub Main ole.TimeOutMsgBox "Today is Holiday", "광복절", 5 End Sub </pre>

2.2.5 이벤트 관련 함수

항목	내용
형식	LONG NotifyAlarmEvent (DOUBLE lIoccur, USHORT nAlmLevel, BSTR szMessage, BSTR szArea, BSTR szOUser, BSTR szInformation, BSTR szUserDefine1, BSTR szUserDefine2, BSTR szUserDefine3, BSTR szUserDefine4, BSTR szPrev, BSTR szCur, BSTR szObject)
종류	이벤트 관련 함수
내용	<p>알람 이벤트를 발생립니다.</p> <p><i>lIoccur</i>: 발생 시각, NULL인경우, 현재 시작을 자동 설정 해 줌</p> <p><i>nAlmLevel</i>: 알람등급 : 1~10 등급,</p> <p><i>szMessage</i>: 메시지</p> <p><i>szArea</i>: 발생 지역</p> <p><i>szOUser</i>: 발생 자</p> <p><i>szInformation</i>: 정보</p> <p><i>szUserDefine1</i>: 사용자 정의1</p> <p><i>szUserDefine2</i>: 사용자 정의2</p> <p><i>szUserDefine3</i>: 사용자 정의3</p> <p><i>szUserDefine4</i>: 사용자 정의4</p> <p><i>szPrev</i>: 메시지 이전 값</p> <p><i>szCur</i>: 메시지 새로운 값</p> <p><i>szObject</i>: 메시지 객체</p>
예제	<pre> Sub Main Dim lRet As Long Dim dDate As Double Dim iLevel As Integer ole.ChangeProcessServer("ALM_PROC", "150.150.32.60") If lRet <> 0 Then MsgBox "ChangeProcessServer() False : ret = " + CStr(lRet) Else MsgBox "ChangeProcessServer() OK : ret = " + CStr(lRet) End If dDate = NUL iLevel = 1 lRet = ole.NotifyAlarmEvent(dDate, iLevel, "NotifyAlarmEvent Test", _ "Script", "Script", "Interfun API Success", "", "", "", _ "", "before bun", "after Run", "") If lRet <> 0 Then MsgBox "NotifyAlarmEvent() False : ret = " + CStr(lRet) Else MsgBox "NotifyAlarmEvent() OK : ret = " + CStr(lRet) End If End Sub </pre>

부록2 내부함수 사용법

항목	내용
형식	VOID SendEmail (BSTR szMailAddTo, BSTR szMailAddFrom, BSTR szMailTitle , BSTR szMailMsg)
종류	이벤트 관련 함수
내용	E-Mail를 보냅니다. <i>szMailAddTo</i> : E-Mail를 보내는 사람 <i>szMailAddFrom</i> : E-Mail를 받는 사람 <i>szMailTitle</i> : E-Mail 제목 <i>szMailMsg</i> : E-Mail 본문
예제	Sub Main ole.SendEmail(("receiver@lisis.biz", "sender@lisis.biz", "제목", "내용: Script Test Good~!!") End Sub

항목	내용
형식	VOID SendSMS (BSTR szRecvPhoneNo, BSTR szCallBackPhoneNo, BSTR szSMSMsg)
종류	이벤트 관련 함수
내용	SMS 문자를 제공합니다. <i>szRecvPhoneNo</i> : SMS 문자를 받는 사람 전화번호 <i>szCallBackPhoneNo</i> : SMS 문자를 보내는 사람 전화번호 <i>szSMSMsg</i> : SMS 문자 메시지(영문:80, 한글:40)
예제	Sub Main ole.SendSMS("019485XXXX", "019485XXXX", "Script Test") End Sub

2.2.6 로그 관련 함수

항목	내용
형식	VOID NotifyError (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
종류	로그 관련 함수
내용	오류 로그를 기록합니다. <i>nLogCategory</i> : 로그카테고리 (기본 : LOG_CATE_SCRIPT= 1, 0) <i>szMsg</i> : 로그 메시지 <i>szDesc</i> : 로그 설명
예제	Sub Main ole.NotifyError(1, "ERROR", "Notify_ERROR") End Sub

항목	내용
형식	VOID NotifyHostError (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
종류	로그 관련 함수
내용	지정된 컴퓨터에 오류 로그를 기록합니다. <i>szHostName</i> : 로그를 기록하는 컴퓨터 명 <i>nLogCategory</i> : 로그카테고리 (기본 : LOG_CATE_SCRIPT= 1, 0) <i>szMsg</i> : 로그 메시지 <i>szDesc</i> : 로그 설명
예제	Sub Main ole.NotifyHostError("127.0.0.1",1, "HOST_ERROR", "Notify_HOST_ERROR") End Sub

부록2 내부함수 사용법

항목	내용
형식	VOID NotifyHostInfo (BSTR szHostName, USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
종류	로그 관련 함수
내용	<p>지정된 컴퓨터에 정보 로그를 기록합니다.</p> <p><i>szHostName</i>: 로그를 기록하는 컴퓨터 명 <i>nLogCategory</i>: 로그카테고리 (기본 : LOG_CATE_SCRIPT= 1, 0) <i>szMsg</i>: 로그 메시지 <i>szDesc</i>: 로그 설명</p>
예제	<pre>Sub Main ole.NotifyHostInfo ("127.0.0.1",1, "HOST_INFO", "Notify_HOST_INFO") End Sub</pre>

항목	내용
형식	VOID NotifyHostLog (USHORT nLogType, USHORT nLogCategory, BSTR szHostName, BSTR szMsg, BSTR szDesc)
종류	로그 관련 함수
내용	<p>지정된 컴퓨터에 로그를 기록합니다.</p> <p><i>nLogType</i>: 로그 타입(1 = INFO_LOG_TYPE(정보), 2 = WARN_LOG_TYPE (경보), 3 = ERRO_LOG_TYPE(오류)) <i>nLogCategory</i>: 로그카테고리 (기본 : LOG_CATE_SCRIPT= 1, 0) <i>szHostName</i>: 로그를 기록하는 컴퓨터 명 <i>szMsg</i>: 로그 메시지 <i>szDesc</i>: 로그 설명</p>
예제	<pre>Sub Main ole.NotifyHostLog(1, 1, "127.0.0.1", "HOST_INFO", "NotifyHostLog__ INFO") ole.NotifyHostLog(2, 1, "127.0.0.1", "HOST_WARN", "NotifyHostLog _ WARN") ole.NotifyHostLog(3, 1, "127.0.0.1", "HOST_ERROR", "NotifyHostLog _ ERROR") End Sub</pre>

부록2 내부함수 사용법

항목	내용
형식	VOID NotifyWarn (USHORT nLogCategory, BSTR szMsg, BSTR szDesc)
종류	로그 관련 함수
내용	경보 로그를 기록합니다. <i>nLogCategory</i> : 로그카테고리 (기본 : LOG_CATE_SCRIPT= 1, 0) <i>szMsg</i> : 로그 메시지 <i>szDesc</i> : 로그 설명
예제	Sub Main ole.NotifyWarn(1, "WARN", "Notify_WARN") End Sub

2.2.7 네트워크 관련 함수

항목	내용
형식	BSTR ChangLine (BSTR szLineMode, BSTR szTargetLineIP)
종류	네트워크 관련 함수
내용	네트워크의 라인을 지정된 네트워크 주소(IP주소)로 변경합니다. <i>szLineMode</i> : 라인 모드 (DEFAULT, SUB) <i>szTargetLineIP</i> : 라인 IP
예제	<pre> Sub Main Dim bRet As Boolean bRet = ole.ChangLine("DEFAULT", "xxx.xxx.xxx.xxx") If bRet <> 0 Then MsgBox "ChangLine FALSE" Else MsgBox "ChangLine OK" End If End Sub </pre>

항목	내용
형식	LONG ChangeProcessServer (BSTR szProcessName, BSTR szProcessIP)
종류	네트워크 관련 함수
내용	실행되고 있는 프로세스를 지정된 네트워크주소(IP 주소)의 프로세스로 교체합니다. . <i>szProcessName</i> : 다른 컴퓨터에서 실행하고자 하는 프로세스 명 <i>szProcessIP</i> : 프로세스를 실행되어질 컴퓨터 IP
예제	<pre> Sub Main Dim lRet As Long lRet = ole.ChangeProcessServer("ALM_PROC", "xxx.xxx.xxx.xxx") If lRet <> 0 Then MsgBox "ChangeProcessServer() False" Else MsgBox "ChangeProcessServer() OK " End If End Sub </pre>

부록2 내부함수 사용법

항목	내용
형식	LONG ChangeServerMode (BSTR szServerMode)
종류	네트워크 관련 함수
내용	프로젝트 타입이 이중화 서버로 등록된 경우 서버의 모드를 변경합니다. <i>szServerMode</i> : 프로젝트 실행 서버의 모드 상태 (NORMAL, RUN, STANDBY)
예제	<pre> Sub Main Dim IRet As Long IRet = ole.ChangeServerMode("NORMAL") If IRet <> 0 Then MsgBox "ChangeServerMode()False" Else MsgBox "ChangeServerMode() OK" End If End Sub </pre>

항목	내용
형식	LONG GetCurLine (int* nCurLine)
종류	네트워크 관련 함수
내용	네트워크 라인 이중화에서 현재 사용하고 있는 라인을 알려줍니다. <i>nCurLine</i> : 네트워크 라인(0: Default IP, 1: Sub IP, 2: 통신 중지)
예제	<pre> Sub Main Dim IRet As Long Dim ILineSatus As Long IRet = ole.GetCurLine(ILineSatus) If IRet <> 0 Then MsgBox "GetCurLine()False => Status " + CStr(ILineSatus) Else MsgBox "GetCurLine() OK => Status " + CStr(ILineSatus) End If End Sub </pre>

항목	내용
형식	INT GetLineStatus (INT nLineMode)
종류	네트워크 관련 함수
내용	지정된 라인의 현재 상태를 알려줍니다. <i>nLineMode</i> : 네트워크의 현재 라인 상태
예제	<pre> Sub Main Dim IRet As Long Dim ILineStyle As Long ILineStyle = 0 IRet = ole.GetLineStatus(ILineStyle) If (IRet = 1) Then MsgBox "GetLineStatus() OK: " + CStr(ILineStyle) Else MsgBox "GetLineStatus() FALSE: " + CStr(ILineStyle) End If End Sub </pre>

항목	내용
형식	BSTR GetNodeCur IP (BSTR szNodeName)
종류	네트워크 관련 함수
내용	지정된 노드에서 사용하고 있는 네트워크 주소(IP 주소)를 알려줍니다. <i>szNodeName</i> : 프로젝트를 접속되어 사용하며 IP를 알고자하는 컴퓨터의 노드명
예제	<pre> Sub Main Dim IP As String IP = ole.GetNodeCur IP("SERVER") MsgBox "GetNodeCur IP() => SERVER " + IP End Sub </pre>

항목	내용
형식	BSTR GetServerMode ()
종류	네트워크 관련 함수
내용	현재 서버의 모드를 알려줍니다. (모드: 실행, 대기)
예제	<pre> Sub Main Dim SvrMode As String SvrMode = ole.GetServerMode MsgBox "GetServerMode : " + SvrMode End Sub </pre>

항목	내용
형식	BOOL IsDualLine ()
종류	네트워크 관련 함수
내용	네트워크 라인 이중화 여부를 알려줍니다.
예제	<pre> Sub Main Dim bRet As Boolean bRet = ole.IsDualLine If bRet = True Then MsgBox "IsDualLine OK" Else MsgBox "IsDualLine FALSE" End If End Sub </pre>

2.2.8 시스템 관련 함수

항목	내용
형식	VARIANT GetSystemTime ()
종류	시스템 관련 함수
내용	현재 시스템 시각을 알려줍니다.
예제	<pre> Sub Main Dim scTime As Variant scTime = ole.GetSystemTime If IRet <> 0 Then MsgBox "GetSystemTime() False" Else MsgBox "GetSystemTime() OK" + scTime End If End Sub </pre>

항목	내용
형식	LONG IsHoliday (szDateTime)
종류	시스템 관련 함수
내용	<p>인자로 전달된 날짜의 휴일 여부를 알려줍니다.</p> <p>szDateTime: 날짜와 시간, YYYY/MM/DD HH:MM:SS 의 형식으로 전달합니다. Return Value: 0=평일, 1=토요일, 2=일요일, 3=공휴일</p>
예제	<pre> Sub Main Dim szDateTime Dim IRetn szDateTime = Format(Now(), "YYYY/MM/DD HH:MM:SS") IRetn = ole.IsHoliday(szDateTime) If IRetn = 0 Then Debug.Print "Today is a Weekday" ElseIf IRetn = 1 Then Debug.Print "Today is Saturday" ElseIf IRetn = 2 Then Debug.Print "Today is Sunday" ElseIf IRetn = 3 Then Debug.Print "Today is Holiday" End If End Sub </pre>

부록2 내부함수 사용법

항목	내용
형식	VOID ProcessSleep (LONG dwMilliseconds)
종류	시스템 관련 함수
내용	밀리세컨즈동안 Script를 중지합니다. dwMilliseconds: 밀리세컨즈
예제	Sub Main Dim scTime As Variant ole.ProcessSleep(1000) End Sub

항목	내용
형식	VOID RunApp (BSTR szAppPath)
종류	시스템 관련 함수
내용	응용 프로그램을 실행합니다. (경로포함) szAppPath: 응용 프로그램 경로
예제	Sub Main Dim scTime As Variant ole.RunApp("C:\WINDOWS\System32\Wmpaint.exe") End Sub

항목	내용
형식	VOID ShowAlarmPage (BSTR szAlarmTag)
종류	시스템 관련 함수
내용	알람 발생한 태그이름을 가진 알람 화면을 보여줍니다. szAlarmTag: 알람 발생하는 태그이름
예제	Sub Main Dim scTime As Variant ole. ShowAlarmPage ("태그명") End Sub

항목	내용
형식	VOID Sleep (LONG dwMilliseconds)
종류	시스템 관련 함수
내용	밀리세컨즈동안 시스템을 중지합니다. dwMilliseconds: 밀리세컨즈
예제	Sub Main Dim scTime As Variant ole. Sleep(1000) End Sub

항목	내용
형식	BSTR TimeConvert(DOUBLE dTimeValue, INT nTimeType)
종류	시스템 관련 함수
내용	<p>현재의 시각값을 지정된 시각표시형식에 따라 다음과 같은 문자열로 출력합니다.</p> <p><i>dTimeValue</i> ≤ 0 현재 시각 > 0 <i>dTimeValue</i> Time 시각 적용</p> <p><i>nTimeType</i> :</p> <ul style="list-style-type: none"> 0 : HH:MM 1 : HH:MM AM/PM 2 : HH:MM:SS 3 : HH:MM:SS AM/PM 4 : HH시MM분 5 : HH시MM분SS초 6 : mm월dd일 7 : mm/dd 8 : mm-dd 9 : yy년mm월dd일 10 : yy/mm/dd 11 : yy-mm-dd 12 : yyyy년mm월dd일 13 : yyyy/mm/dd 14 : yyyy-mm-dd 15 : mm/dd HH:MM 16 : mm/dd HH:MM AM/PM 17 : mm/dd HH:MM:SS 18 : mm/dd HH:MM:SS AM/PM 19 : mm-dd HH:MM 20 : mm-dd HH:MM AM/PM 21 : mm-dd HH:MM:SS 22 : mm-dd HH:MM:SS AM/PM 23 : yy/mm/dd HH:MM 24 : yy/mm/dd HH:MM AM/PM 25 : yy/mm/dd HH:MM:SS 26 : yy/mm/dd HH:MM:SS AM/PM 27 : yy-mm-dd HH:MM 28 : yy-mm-dd HH:MM AM/PM 29 : yy-mm-dd HH:MM:SS 30 : yy-mm-dd HH:MM:SS AM/PM 31 : yyyy/mm/dd HH:MM 32 : yyyy/mm/dd HH:MM AM/PM 33 : yyyy/mm/dd HH:MM:SS 34 : yyyy/mm/dd HH:MM:SS AM/PM 35 : yyyy-mm-dd HH:MM 36 : yyyy-mm-dd HH:MM AM/PM 37 : yyyy-mm-dd HH:MM:SS 38 : yyyy-mm-dd HH:MM:SS AM/PM 39 : mm월dd일 HH시MM분
항목	내용 (앞의 내용 이어서)

부록2 내부함수 사용법

	<p>40 : mm월dd일 HH시MM분SS초 41 : yy년mm월dd일 HH시MM분 42 : yy년mm월dd일 HH시MM분SS초 43 : yyyy년mm월dd일 HH시MM분 44 : yyyy년mm월dd일 HH시MM분SS초 45 : yy/mm 46 : yy년mm월 47 : yyyy/mm 48 : yyyy년mm월 49 : yy 50 : yyyy 51 : mm 52 : dd 53 : HH 54 : MM 55 : SS 56 : yy년 57 : yyyy년 58 : mm월 59 : dd일 60 : HH시 61 : MM분 62 : SS초 63 : “일” ~ “토” 64 : “일요일” ~ “토요일” 65 : “SUN” ~ “SAT”</p>
예제	<pre>Sub Main Dim szTime As String szTime = ole.TimeConvert (1, 33) MsgBox szTime End Sub</pre>

2.2.9 레서피 관련 함수

항목	내용
형식	LONG RecipeChangeData (BSTR szModelName, BSTR szRecipeName)
종류	레서피 관련 함수
내용	<p>워킹셋의 데이터를 변경합니다. (모델은 워킹셋의 모델이어야 한다.)</p> <p><i>szModelName</i>: 레서피 모델 이름 <i>szRecipeName</i>: 레서피 이름</p>
예제	<pre> Sub Main Dim Result As Long Result = ole.RecipeLoadSet("커피자판기2",NUL,"프림커피") If (Result <> 0) Then MsgBox("RecipeLoadSet : Error~~!!") Else Result = ole.RecipeChangeData("커피자판기2","커피라떼") If (Result <> 0) Then MsgBox("RecipeChangeData : Error~~!!") Else MsgBox("RecipeChangeData : OK~~!!") End If End If End Sub </pre>

부록2 내부함수 사용법

항목	내용
형식	LONG RecipeChangeUnit (BSTR szModelName, BSTR szUnitName)
종류	레서피 관련 함수
내용	<p>워킹셋의 유틸리티셋을 변경합니다. (모델은 워킹셋의 모델이어야 합니다.)</p> <p><i>szModelName</i>: 레서피 모델 이름 <i>szUnitName</i>: 레서피 유닛 이름</p>
예제	<pre> Sub Main Dim Result As Long Result = ole.RecipeLoadSet("커피자판기2",NUL,NUL) If (Result <> 0) Then MsgBox("RecipeLoadSet : Error~~!!") Else Result = ole.RecipeChangeUnit("커피자판기2","별관자판기") If (Result <> 0) Then MsgBox("RecipeChangeUnit : Error~~!!") Else MsgBox("RecipeChangeUnit : OK~~!!") End If End If End Sub </pre>

항목	내용
형식	<p>LONG RecipeControlSet (BSTR szModelName, BSTR szUnitName, BSTR szDataName)</p> <p>LONG RecipeControlSetEx (SHORT nMode, BSTR szClientProgName, ULONG hClientHandle, BSTR szModelName, BSTR szUnitName, BSTR szDataName, LONG* pITransNo, SHORT* pnResult)</p>
종류	레서피 관련 함수
내용	<p>모델의 정보, 유닛 정보, 레서피 데이터를 조합하여 제어를 수행하빈다. (워킹셋과 는 관련이 없습니다.)</p> <p><i>nMode</i>: 제어 모드 타임 nMode : 1: no return, 2: Return <i>szClientProgName</i>: 제어를 내리는 클라이언트 프로그램 명 <i>hClientHandle</i>: 제어를 내리는 클라이언트의 윈도우 핸들 <i>szModelName</i> : 레서피 모델 이름 <i>szUnitName</i> : 레서피 유닛 이름 <i>szDataName</i> : 레서피 데이터 이름 <i>pITransNo</i>: 제어 리턴의 결과를 다루기 위한 트랜잭션 번호 <i>pnResult</i> : 제어 요청의 결과</p>
예제	<pre> Sub Main Dim TransNo As Long Dim Result As Long Dim Result2 As Integer Result = ole.RecipeLoadSet("커피자판기2",NUL,"커피라떼") If (Result <> 0) Then MsgBox("RecipeLoadSet : Error~!!") Else Result2 = ole.RecipeControlSetEx(2,"스크립드",this, "커피자판기2","본관자판기","커피라떼",TransNo) 'Result2 = ole.RecipeControlSet("커피자판기2","본관자판기", "커피라떼") If (Result2 <> 0) Then MsgBox("RecipeControlSet : Error~!!") Else MsgBox("RecipeControlSet : OK~!! , TransNO = " + CStr(TransNo)) End If End If End Sub </pre>

항목	내용
형식	<p>LONG RecipeControlWorkingSet ();</p> <p>LONG RecipeControlWorkingSetEx (SHORT nMode, BSTR szClientProgName, ULONG hClientHandle, LONG* pITransNo, SHORT* pnResult);</p>
종류	레서피 관련 함수
내용	워킹셋의 데이터를 유닛 이름의 태그들과 매핑시켜 제어를 수행합니다.

	<p><i>nMode</i>: 제어 모드 타임 nMode : 1: no return, 2: Return <i>szClientProgName</i>: 제어를 내리는 클라이언트 프로그램 명 <i>hClientHandle</i>: 제어를 내리는 클라이언트의 윈도우 핸들 <i>pITransNo</i>: 제어 리턴의 결과를 다루기 위한 트랜잭션 번호 <i>pnResult</i> : 제어 요청의 결과</p>
<p>예제</p>	<pre> Sub Main Dim TransNo As Long Dim Result As Long Dim Result2 As Integer Result = ole.RecipeLoadSet("커피자판기2",NUL,"커피라떼") If (Result <> 0) Then MsgBox("RecipeLoadSet : Error~!!") Else Result2 = ole.RecipeControlWorkingSetEx(2,"스크립드",this, TransNo) Result2 = ole.RecipeControlWorkingSet() If (Result2 <> 0) Then MsgBox("RecipeControlWorkingSet : Error~!!") Else MsgBox("RecipeControlWorkingSet : OK~!! , TransNO = " + CStr(TransNo)) End If End If End Sub </pre>

항목	내용
형식	LONG RecipeDeleteData (BSTR szModelName, BSTR szRecipeName)
종류	레시피 관련 함수
내용	<p>설정된 모델의 레시피 데이터를 삭제합니다.</p> <p><i>szModelName</i> : 레시피 모델이름 <i>szRecipeName</i> : 레시피 이름</p>
예제	<pre> Sub Main Dim Result As Long Result = ole.RecipeDeleteData("커피자판기2", "아메리카노") If (Result <> 0) Then MsgBox("RecipeDeleteData : Error~~!!") Else MsgBox("RecipeDeleteData : OK~~!!") End If End Sub </pre>

항목	내용
형식	LONG RecipeDeleteModel (BSTR szModelName)
종류	레시피 관련 함수
내용	<p>레시피 모델을 삭제합니다. (모델 및, 모델에 속한 모든 유닛, 데이터가 전부 삭제됩니다.)</p> <p><i>szModelName</i>: 레시피 모델 이름</p>
예제	<pre> Sub Main Dim Result As Long Result = ole.RecipeDeleteModel("커피자판기3") If (Result <> 0) Then MsgBox("RecipeDeleteModel : Error~~!!") Else MsgBox("RecipeDeleteModel : OK~~!!") End If End Sub </pre>

항목	내용
형식	LONG RecipeDeleteUnit (BSTR szModelName, BSTR szUnitName)
종류	레시피 관련 함수
내용	<p>설정된 모델의 레시피 유닛을 삭제합니다.</p> <p><i>szModelName</i>: 레시피 모델 이름 <i>szUnitName</i> : 레시피 유닛 이름</p>
예제	<pre> Sub Main Dim Result As Long Result = ole.RecipeDeleteUnit("커피자판기2", "별관자판기") If (Result <> 0) Then MsgBox("RecipeDeleteUnit : Error~~!!") Else MsgBox("RecipeDeleteUnit : OK~~!!") End If End Sub </pre>

항목	내용
형식	LONG RecipeLoadDynamic (BSTR szModelName, BSTR szUnitName)
종류	레시피 관련 함수
내용	<p>유닛(태그)의 실시간 데이터를 불러 워킹셋에 레시피 모델과 데이터를 올려놓습니다.(기존의 워킹셋은 클리어됩니다.)</p> <p><i>szModelName</i>: 레시피 모델 이름 <i>szUnitName</i>: 레시피 유닛 이름</p>
예제	<pre> Sub Main Dim Result As Long Result = ole.RecipeLoadDynamic("커피자판기2", "본관자판기") If (Result <> 0) Then MsgBox("RecipeLoadDynamic : Error~~!!") Else MsgBox("RecipeLoadDynamic : OK~~!!") End If End Sub </pre>

항목	내용
형식	LONG RecipeLoadSet (BSTR szModelName, BSTR szUnitName, BSTR szRecipeName)
종류	레서피 관련 함수
내용	<p>워킹셋에 레서피 모델과 유닛 데이터를 올려놓습니다. (기존의 워킹셋은 클리어됩니다.)</p> <p><i>szModelName</i>: 레서피 모델 이름 <i>szUnitName</i>: 레서피 유닛 이름 , NULL 이어도 상관 없음. <i>szRecipeName</i> : 레서피 데이터 이름</p>
예제	<pre>Sub Main Dim Result As Long Result = ole.RecipeLoadSet("커피자판기2",NUL,"프림커피") If (Result <> 0) Then MsgBox("RecipeLoadSet : Error~~!!") Else MsgBox("RecipeLoadSet : OK~~!!") End If End Sub</pre>

항목	내용
형식	LONG RecipeSaveAsArchive (BSTR szModelName, BSTR szRecipeName)
종류	레서피 관련 함수
내용	<p>워킹셋의 데이터를 모델 이름의 레서피 이름으로 저장합니다. (모델은 워킹셋의 모델이어di 합니다.)</p> <p><i>szModelName</i>: 레서피 모델 이름 <i>szRecipeName</i>: 레서피 이름</p>
예제	<pre>Sub Main Dim Result As Long Result = ole.RecipeLoadSet("커피자판기2",NUL,"커피라떼") If (Result <> 0) Then MsgBox("RecipeLoadSet : Error~~!!") Else Result = ole.RecipeSaveAsArchive("커피자판기2","커피라떼") If (Result <> 0) Then MsgBox("RecipeSaveAsArchive : Error~~!!") Else MsgBox("RecipeSaveAsArchive : OK~~!!") End If End If End Sub</pre>

항목	내용
형식	LONG RecipeSetData (BSTR szModelName, BSTR szItemName, VARIANT szSetValue)
종류	레서피 관련 함수
내용	<p>워킹셋의 아이템의 위치의 값을 수정합니다. (모델은 워킹셋의 모델이어야 합니다.)</p> <p><i>szModelName</i> : 레서피 모델 이름 <i>szItemName</i> : 레서피 아이템 이름 <i>szSetValue</i> : 레서피 아이템의 위치 값</p>
예제	<pre> Sub Main Dim vtSetValue As Variant Dim Result As Long Result = ole.RecipeLoadSet("커피자판기2",NUL,NUL) vtSetValue = 55 If (Result <> 0) Then MsgBox("RecipeLoadSet : Error~!!") Else Result = ole.RecipeSetData("커피자판기2","커피",vtSetValue) If (Result <> 0) Then MsgBox("RecipeSetData : Error~!!") Else MsgBox("RecipeSetData : OK~!!") End If End If End Sub </pre>

2.2.10 데이터베이스 관련 함수

항목	내용
형식	VOID DataExportRunWork (BSTR szWorkName)
종류	데이터 베이스 관련 함수
내용	DataExport의 작업을 실행합니다. szWorkName : 작업이름
예제	Sub Main ole. DataExporRuntWork (“작업이름”) End Sub

2.2.11 알람 관련 함수

항목	내용
형식	BOOL AckAllAlarm ()
종류	알람 관련 함수
내용	그래픽 런타임에 접속된 사용자의 등급으로 발생한 알람을 인지하고 그 결과를 알려줍니다. .
예제	Sub Main ole. AckAllAlarm End Sub

항목	내용
형식	BSTR GetAlarmCountErrInfo (LONG ICount)
종류	알람 관련 함수
내용	알람 개수 요청시 에러 번호에 대해 설명합니다. ICount: 함수 GetAlmCount(), GetHistAlmCount()의 리턴값 < 0
예제	Sub Main Dim szErrDesc As String Dim AlmCount As Long AlmCount = ole. GetHistAlmCount (1, 0) If AlmCount < 0 Then szErrDesc = ole.GetAlarmCountErrInfo (AlmCount) MsgBox szErrDesc End If End Sub

항목	내용
형식	LONG GetAlarmLogging (DOUBLE dtFrom, DOUBLE dtTo, BSTR szTag, BSTR szAlarmGr, LONG lAlarmType, LONG *lOutCount, SAFEARRAY **pszRstrTag, SAFEARRAY **pszRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **pIRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind)
종류	알람 관련 함수
내용	이력알람 리스트를 조회 조건에 따라 조회하는 함수입니다. <i>dtFrom</i> : 조회하고자 하는 알람의 시작 시간 (이력의 처음부터 조회하고자 할때는 0 입력) <i>dtTo</i> : 조회하고자 하는 알람의 끝시간 (이력의 최근까지 조회하고자 할때는 0 입력) <i>szTag</i> : 태그 이름, 조회하고자하는 알람의 태그 이름(NULL: ALL) <i>szAlarmGr</i> : 조회하고자 하는 알람의 알람 그룹 (NULL: ALL) <i>lAlarmType</i> : 알람 타입, 0: ALL 1: 알람(발생) 2: 이벤트 3: 알람(해제) 4: 알람 (발생+해제) <i>lOutCount</i> : 조회 결과의 알람 리스트 개수 <i>pszRstrTag</i> : 조회 결과의 태그 이름 리스트 <i>pszRstrGr</i> : 조회 결과의 알람 그룹 리스트 <i>pdbRdtOccAlarm</i> : 조회 결과의 알람 발생 시간 리스트 <i>pdbRdtRevAlarm</i> : 조회 결과의 복구 시간 리스트 <i>pdbRdtAckAlarm</i> : 조회 결과의 알람 인지 시간 리스트 <i>pIRnAlarmType</i> : 조회 결과의 알람 타입 리스트 <i>pszRstrMsg</i> : 조회 결과의 알람 메시지 리스트 <i>pszRstrAlarmKind</i> : 조회 결과의 알람 종류 HH: 아날로그 경계 알람 HH 알람 HI: 아날로그 경계 알람 HI 알람 LO: 아날로그 경계 알람 LO 알람 LL: 아날로그 경계 알람 LL 알람 Achange: 아날로그 변화 알람 MGap: 아날로그 주이격 알람 SGap: 아날로그 보조이격 알람 On: 디지털 On 알람 Off: 디지털 Off 알람 OnOff: 디지털 On ->Off 알람 OffOn: 디지털 Off ->On 알람 DChagne: 디지털 변화 알람
항목	내용 (앞의 항목 이어서)

예제	<pre> Sub Main Dim dtFrom As Date Dim dtTo As Date Dim szTag As String Dim szAlarmGr As String Dim lAlarmType As Long Dim lOutCount As Long Dim aRstrTag() As String Dim aRstrGr() As String Dim aRdtOccAlarm() As Double Dim aRdtRevAlarm() As Double Dim aRdtAckAlarm() As Double Dim aRnAlarmType() As Long Dim aRstrMsg() As String Dim aRstrAlarmKind() As String dtFrom = Now() - 27 dtTo = Now() szTag = "" szAlarmGr = "AlmGrp_UT" lAlarmType = 0 lRet = ole.GetAlarmLogging(dtFrom ,dtTo , szTag , szAlarmGr, lAlarmType, lOutCount, aRstrTag, aRstrGr, aRdtOccAlarm, aRdtRevAlarm, aRdtAckAlarm, aRnAlarmType, aRstrMsg, aRstrAlarmKind) For i = 0 To lOutCount - 1 MsgBox (aRstrTag(i) + " " + aRstrGr(i) + " "+ CStr(CDate(aRdtOccAlarm(i))) + " "+ CStr(CDate(aRdtRevAlarm(i))) + " "+ CStr(CDate(aRdtAckAlarm(i))) + " "+ CStr(aRnAlarmType(i)) + " " + aRstrMsg(i) + " " + aRstrAlarmKind(i)) Next i End Sub </pre>
----	---

항목	내용
형식	LONG GetAlarmLoggingEx (DOUBLE dtFrom,DOUBLE dtTo,BSTR szTag, LONG nAlarmGroupCount, SAFEARRAY **pszAlarmGr, LONG lAlarmType, LONG lAlarmGrade, BSTR szAlarmKind, LONG *lOutCount, SAFEARRAY**pRstrTag,SAFEARRAY**pRstrGr, SAFEARRAY **pdbRdtOccAlarm, SAFEARRAY **pdbRdtRevAlarm, SAFEARRAY **pdbRdtAckAlarm, SAFEARRAY **plRnAlarmType, SAFEARRAY **pszRstrMsg, SAFEARRAY **pszRstrAlarmKind, SAFEARRAY **pdbRvOccDValue, SAFEARRAY**pszRvOccSValue, SAFEARRAY **plAlarmGrade)
종류	알람 관련 함수
내용	이력알람 리스트를 조회 조건에 따라 조회하는 함수로써 GetAlarmLogging() 함수의 확장 함수입니다. <i>dtFrom</i> : 조회하고자 하는 알람의 시작 시간 (이력의 처음부터 조회하고자 할때는 0 입력) <i>dtTo</i> : 조회하고자 하는 알람의 끝시간 (이력의 최근까지 조회하고자 할때는 0 입력) <i>szTag</i> : 태그 이름, 조회하고자하는 알람의 태그 이름(NULL: ALL) <i>nAlarmGroupCount</i> : 조회하고자 하는 알람 그룹의 개수 <i>pszAlarmGr</i> : 알람 그룹, 조회하고자 하는 알람의 알람그룹 리스트 (NULL: ALL) <i>lAlarmType</i> : 알람 타입 이벤트의 경우 발생만 있음 0: ALL 1: 알람(발생) 2: 이벤트 3: 알람(해제) 4: 알람 (발생+해제)) <i>lAlarmGrade</i> : 알람 등급, 조회한 알람 일상의 알람이 반환됨 1등급 - 10 등급 (1등급 > 10 등급) ex) 50이면 히이상 등급 (1~5) <i>szAlarmKind</i> : 조회하고자 하는 알람 종류 NULL: ALL HH: 아날로그 경계 알람 HH 알람 HI: 아날로그 경계 알람 HI 알람 LO: 아날로그 경계 알람 LO 알람 LL: 아날로그 경계 알람 LL 알람 Achange: 아날로그 변화 알람 MGap: 아날로그 주이격 알람 SGap: 아날로그 보조이격 알람 On: 디지털 On 알람 Off: 디지털 Off 알람 OnOff: 디지털 On ->Off 알람 OffOn: 디지털 Off ->On 알람 DChagne: 디지털 변화 알람 <i>lOutCount</i> : 조회 결과의 알람 리스트 개수 <i>pszRstrTag</i> : 조회 결과의 태그 이름 리스트 <i>pszRstrGr</i> : 조회 결과의 알람 그룹 리스트 <i>pdbRdtOccAlarm</i> : 조회 결과의 알람 발생 시간 리스트 <i>pdbRdtRevAlarm</i> : 조회 결과의 복구 시간 리스트 <i>pdbRdtAckAlarm</i> : 조회 결과의 알람 인지 시간 리스트
항목	내용 (앞의 항목 이어서)

<p>내용</p>	<p><i>pIRnAlarmType</i>: 조회 결과의 알람 타입 리스트 <i>pszRstrMsg</i>: 조회 결과의 알람 메시지 리스트 <i>pszRstrAlarmKind</i>: 조회 결과의 알람 종류 HH: 아날로그 경계 알람 HH 알람 HI: 아날로그 경계 알람 HI 알람 LO: 아날로그 경계 알람 LO 알람 LL: 아날로그 경계 알람 LL 알람 Achange: 아날로그 변화 알람 MGap: 아날로그 주이격 알람 SGap: 아날로그 보조이격 알람 On: 디지털 On 알람 Off: 디지털 Off 알람 OnOff: 디지털 On ->Off 알람 OffOn: 디지털 Off ->On 알람 <i>pdbrV0ccDValue</i> 조회 결과의 현재값 리스트 (알람인 경우만 있음) <i>pszRv0ccSValue</i>: 조회 결과의 현재값 리스트(이벤트의 경우만 있음) <i>pIAlarmGrade</i>: 조회 결과의 알람 등급 리스트</p>
<p>예제</p>	<pre> Sub Main Dim dtFrom As Double Dim dtTo As Double Dim szTag As String Dim nAlarmCount As Long Dim aAlarmGr(0) As String Dim lAlarmType As Long Dim lAlarmGrade As Long Dim szAlarmKind As String Dim lOutCount As Long Dim aRstrTag() As String Dim aRstrGr() As String Dim aRdt0ccAlarm() As Double Dim aRdtRevAlarm() As Double Dim aRdtAckAlarm() As Double Dim aRnAlarmType() As Long Dim aRstrMsg() As String Dim aRstrAlarmKind() As String Dim a0ccDvalue() As Double Dim a0ccSValue() As String Dim aAlarmLevel() As Long dtFrom = Now() - 27 dtTo = Now szTag = "" nAlarmCount = 1 </pre>
<p>항목</p>	<p>내용 (앞의 항목 이어서)</p>

<p>예제</p>	<pre> aAlarmGr(0) = "AlmGrp_UT" lAlarmType = 4 lAlarmGrade = 10 szAlarmKind = "L0" lRet = ole.GetAlarmLoggingEx (dtFrom ,dtTo , szTag , nAlarmCount, aAlarmGr,lAlarmType,lAlarmGrade, szAlarmKind, lOutCount, aRstrTag, aRstrGr, aRdtOccAlarm, aRdtRevAlarm,aRdtAckAlarm, aRnAlarmType, aRstrMsg,aRstrAlarmKind, aOccDvalue, aOccSValue, aAlarmLevel) For i = 0 To lOutCount MsgBox (aRstrTag(i) + " " + aRstrGr(i) + " "+ CStr(CDate(aRdtOccAlarm(i)))+ " "+ CStr(CDate(aRdtRevAlarm(i)))+ " "+ CStr(CDate(aRdtAckAlarm(i))) + " " + CStr(aRnAlarmType(i)) + " " + aRstrMsg(i) + " " + aRstrAlarmKind(i)) Next i End Sub </pre>
-----------	---

항목	내용
형식	LONG GetAlmCount (LONG IKind, LONG IStatus)
종류	알람 관련 함수
내용	<p>실시간 알람 개수를 알려줍니다.</p> <p><i>IKind</i>: 알람 종류 0 : All 1 : Process Alarm 2 : System Alarm 3 : Process + System 4 : Event 5 : Process + Event 6 : System + Event</p> <p><i>IStatus</i>: 알람 상태 0 : All 1 : 미복구 2 : 미인지 3 : 미복구 + 미인지</p>
예제	<pre> Sub Main Dim AlmCount As Long AlmCount = ole. GetAlmCount(1, 0) If AlmCount > 0 Then MsgBox " 실시간 Alarm Count = " + CStr(AlmCount) End If End Sub </pre>

부록2 내부함수 사용법

항목	내용
형식	LONG GetGroupUserInfoByTag (BSTR szTagName, LONG* lOutCount, SAFEARRAY(BSTR) *pszUserID, SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(LONG) *pMailCheck, SAFEARRAY(LONG) *pSMSCheck, SAFEARRAY(BSTR) *pszMailAdd, SAFEARRAY(BSTR) *pszHPhoneNo)
종류	알람 관련 함수
내용	<p>태그 이름을 가지고 태그가 소속된 알람그룹의 등록후 사용자 정보를 가져옵니다 태그 정보(알람정보)를 가지고 담당 사용자의 호출 등에 사용됩니다.</p> <p><i>szTagName</i>: 태그명 <i>lOutCount</i>: 반환되는 유저정보 개수 <i>pszUserID</i>: 사용자 ID <i>pszUserName</i>: 사용자 이름 <i>pMailCheck</i>: 사용자가 알람 발생시 메일통보를 받을지 여부 <i>pSMSCheck</i>: 사용자가 알람 발생시 SMS통보를 받을지 여부 <i>pszMailAdd</i>: 사용자 등록 메일 주소 <i>pszHPhoneNo</i>: 사용자 등록 핸드폰 번호</p>
예제	<pre> Sub Main Dim lOutCount As Long Dim aUserID() As String Dim aUserName() As String Dim aMailCheck() As Long Dim aSMSCheck() As Long Dim aMailAdd() As String Dim aHPhoneNo() As String ole.GetGroupUserInfoByTag("AlmGrpTestTag", lOutCount, aUserID, aUserName, aMailCheck, aSMSCheck, aMailAdd, aHPhoneNo) For i = 0 To lOutCount MsgBox (aUserID(i) + " " + aUserName(i) + " " + CStr(aMailCheck(i)) + " " + CStr(aSMSCheck(i)) + " " + aMailAdd(i) + " " + aHPhoneNo(i)) Next i End Sub </pre>

항목	내용
형식	LONG GetHistAlmCount (LONG IKind, LONG IStatus)
종류	알람 관련 함수
내용	<p>이력 알람 개수를 알려줍니다</p> <p><i>IKind</i>: 알람 종류 0 : All 1 : Process Alarm 2 : System Alarm 3 : Process + System 4 : Event 5 : Process + Event 6 : System + Event</p> <p><i>IStatus</i>: 알람 상태 0 : All 1 : 미복구 2 : 미인지 3 : 미복구 + 미인지</p>
예제	<pre> Sub Main Dim AlmCount As Long AlmCount = ole. GetHistAlmCount (1, 0) If AlmCount > 0 Then MsgBox "이력 Alarm Count = " + CStr(AlmCount) End If End Sub </pre>

부록2 내부함수 사용법

항목	내용
형식	LONG GetUserInfoList (LONG* lOutCount, SAFEARRAY(BSTR) *pszUserID, SAFEARRAY(BSTR) *pszUserName, SAFEARRAY(BSTR) *pszDesc, SAFEARRAY(BSTR) *pszPassword, SAFEARRAY(LONG) *plUserLevel, SAFEARRAY(BSTR) *pszHPhoneNo, SAFEARRAY(BSTR) *pszPhoneNo, SAFEARRAY(BSTR) *pszMailAdd)
종류	알람 관련 함수
내용	<p>모든 사용자 리스트 및 각 사용자의 정보를 얻어옵니다 사용자의 정보는 SMS 호출이나 메일 전송등에 사용됩니다.</p> <p><i>lOutCount</i>: 반환되는 유저정보 개수 <i>pszUserID</i>: 사용자 ID <i>pszUserName</i>: 사용자 이름 <i>pszDesc</i>: 사용자 설명 <i>pszPassword</i>: 사용자 비밀번호 <i>plUserLevel</i>: 사용자 제어 정보 <i>pszHPhoneNo</i>: 사용자 등록 핸드폰 번호 <i>pszPhoneNo</i>: 사용자 등록 전화 번호 <i>pszMailAdd</i>: 사용자 등록 메일 주소</p>
예제	<pre> Sub Main Dim lRet As Long Dim lOutCount As Long Dim aUserID() As String Dim aUserName() As String Dim aDesc() As String Dim aPassword() As String Dim aUserLevel() As Long Dim aMailAdd() As String Dim aHPhoneNo() As String Dim aPhoneNo() As String lRet = ole.GetUserInfoList(lOutCount, aUserID, aUserName, aDesc, aPassword, aUserLevel, aHPhoneNo, aPhoneNo, aMailAdd) For i = 0 To lOutCount - 1 MsgBox (aUserID(i) + " " + aUserName(i) + " " + CStr(aUserLevel(i)) + " " + aHPhoneNo(i) + " " + aPhoneNo(i) + " " + aMailAdd(i)) Next i End Sub </pre>

항목	내용
형식	LONG SetUserAlarm (BSTR szAlarmMessage, BSTR szAlarmGroup)
종류	알람 관련 함수
내용	<p>사용자 발생 알람 함수입니다. 사용자가 임의로 발생하는 알람으로서 복구가 어려운 이유로 발생과 동시에 복구 실행합니다.</p> <p><i>szAlarmMessage</i>: 알람메시지 <i>szAlarmGroup</i>: 알람그룹</p> <p>알람정보중 자동입력되는 항목은 <i>사용자</i>: 현재 로그인된 유저 입력 - 발생유저, 복구유저 <i>발생시간, 복구시간</i>: 실행된 시스템의 현재시간 입력 <i>알람의 종류</i>: 프로세스알람 <i>알람등급</i>: 1~10 등급 중 10등급으로 입력. 태그이름에 "\$Script"라고 기록한다</p>
예제	<pre>Sub Main ole.SetUserAlarm "사용자 알람 발생", "알람 그룹" End Sub</pre>

항목	내용
형식	LONG SetUserAlarmEx (DOUBLE dOccur, LONG lAlarmLevel, BSTR szAlarmMessage, BSTR szAlarmGroup, DOUBLE dCur, DOUBLE dPrev)
종류	알람 관련 함수
내용	<p>사용자 발생 알람 함수 - 확장버전 입니다. 사용자가 임의로 발생하는 알람으로서 자동입력 항목을 사용자가 직접 입력할 수 있습니다.</p> <p><i>dOccur</i>: 시간입력 - 0으로 입력시 현재시간 자동입력 <i>lAlarmLevel</i>: : 알람등급 - 1~10등급으로 입력시 입력되며, 그 이외의 값 입력되는 자동으로 10으로 입력됨 <i>szAlarmMessage</i>: 알람메시지 <i>szAlarmGroup</i>: 알람그룹 <i>dCur</i>: 현재값 <i>dPrev</i>: 이번값</p> <p>자동입력항목 <i>사용자</i>: 현재 로그인된 사용자 입력 - 발생유저, 복구유저 <i>알람의 종류</i>: 프로세스알람. 태그이름에 "\$Script"라고 기록한다</p>
예제	<pre>Sub Main ole.SetUserAlarmEx Now(), 2, "사용자 확장 알람 발생", "확장 그룹", 22.34, 44.56 End Sub</pre>

항목	내용
형식	LONG SetUserEvent (BSTR szEventMessage)
종류	알람 관련 함수
내용	<p>사용자 발생 이벤트 함수입니다. 사용자가 스크립트를 통하여 임의로 발생시키는 이벤트입니다.</p> <p><i>szEventMessage</i> : 이벤트 메시지</p> <p>자동입력 항목 <i>이벤트시간</i> : 실행된 시스템의 현재시간 입력 <i>이벤트 발생자</i> : 현재 로그인된 유저입력 <i>발생 컴퓨터(복구유저부분에 표시)</i> : 현재 시스템 이름 자동입력 <i>알람의 종류</i> : 이벤트 알람 <i>알람등급</i> : 1~10 등급 중 10등급으로 입력. 이벤트이름에 "\$Script"라고 기록한다</p>
예제	<pre>Sub Main ole.SetUserEvent "사용자 이벤트 발생" End Sub</pre>

항목	내용
형식	VOID StopAlarmSound ()
종류	알람 관련 함수
내용	현재 발생되고 있는 알람 사운드를 중지합니다.
예제	<pre>Sub Main ole.StopAlarmSound End Sub</pre>

2.2.12 외부 장치 관련 함수

항목	내용
형식	VOID PlaySound (BSTR szWaveFile)
종류	외부 장치 관련 함수
내용	<p>지정된 사운드를 들려줍니다. 절대 경로 상대 경로 모두 가능합니다. 지정 경로에서 찾지 못할 경우 프로젝트의 Wav 폴더, 프로젝트의 GraLS 폴더, 프로젝트 폴더 및 실행프로그램 폴더 내에 사운드를 들려줍니다.(확장자, 미포함도 가능합니다. 자동 Wav 확장자 포함하여 실행합니다. 사운드 파일은 비동기적으로 작동합니다.</p> <p><i>szSoundFile</i> : 사운드 파일</p>
예제	<pre>Sub Main ole.PlaySound ("사운드명.wav") End Sub</pre>

항목	내용
형식	VOID PlaySoundFile (BSTR szSoundFile)
종류	외부 장치 관련 함수
내용	<p>지정된 사운드를 출력합니다. 절대 경로 상대 경로 모두 가능합니다. 지정 경로에서 찾지 못할 경우, 프로젝트의 Wav 폴더, 프로젝트의 GraLS 폴더, 프로젝트 폴더 및 실행프로그램 폴더 내에 사운드를 출력합니다. (확장자, 미포함도 가능합니다. 자동 Wav 확장자 포함하여 실행합니다. 사운드 파일은 동기적으로 작동합니다.</p> <p><i>szSoundFile</i> : 사운드 파일</p>
예제	<pre>Sub Main ole.PlaySoundFile("사운드명.wav") End Sub</pre>

2.2.13 사용자 관련 함수

항목	내용
형식	BSTR GetUserID ()
종류	사용자 관련 함수
내용	그래픽 런타임에서 호출될 경우 현재 로그인한 사용자의 아이디를 반환합니다.
예제	<pre> Sub Main Dim szUserID As String szUserID = ole. GetUserID MsgBox " 사용자 아이디 = "szUserID End Sub </pre>

항목	내용
형식	LONG GetUserLevel ()
종류	사용자 관련 함수
내용	그래픽 런타임에서 호출될 경우 현재 로그인 사용자의 우선순위를 반환합니다.
예제	<pre> Sub Main Dim lRet As Long lRet = ole. GetUserLevel MsgBox " 사용자 우선 순위 = "+ CStr(lRet) End Sub </pre>

항목	내용
형식	LONG UserLogOff ()
종류	사용자 관련 함수
내용	런타임에 로그인된 사용자를 로그아웃합니다.
예제	<pre> Sub Main Dim lRet As Long lRet = ole.UserLogOff If lRet <> 0 Then MsgBox "UserLogOff() False" Else MsgBox "UserLogOff() OK" End If End Sub </pre>

항목	내용
형식	LONG UserLogOn (BSTR szUser ID, BSTR szPassWord)
종류	사용자 관련 함수
내용	지정된 사용자 아이디와 비밀번호로 런타임에 로그인 합니다. <i>szUser ID</i> : 사용자 아이디 <i>szPassWord</i> : 사용자 비밀번호
예제	<pre> Sub Main Dim lRet As Long lRet = ole.UserLogOn("사용자며 아이디", "비밀번호") If lRet <> 0 Then MsgBox "UserLogOn() False" Else MsgBox "UserLogOn() OK" End If End Sub </pre>

2.2.14 보고서 관련 함수

항목	내용
형식	VOID ReportDialog ()
종류	보고서 관련 함수
내용	보고서 창을 보여줍니다.
예제	Sub Main ole. ReportDialog End Sub

항목	내용
형식	VOID ReportListRefresh()
종류	보고서 관련 함수
내용	보고서 목록을 갱신합니다.
예제	Sub Main ole. ReportListRefresh End Sub

항목	내용
형식	VOID ReportPrint (BSTR szReportName)
종류	보고서 관련 함수
내용	지정된 보고서를 프린터로 출력합니다. <i>szReportName</i> : 보고서 이름
예제	Sub Main ole. ReportPrint (“보고서이름”) End Sub

항목	내용
형식	VOID ReportPrintEx (BSTR szReportName, BSTR szReqTime)
종류	보고서 관련 함수
내용	지정된 보고서를 지정된 시각에 프린터로 출력합니다. 만약 지정된 시간이 NULL 인 경우 ReportPrint() 함수와 동일하게 동작합니다. szReportName : 보고서 이름 szReqTime : 프린트 출력 요청 시각
예제	Sub Main ole. ReportPrintEx (“보고서이름” , NULL) End Sub

항목	내용
형식	VOID ReportSchedulerHide ()
종류	보고서 관련 함수
내용	보고서 스케줄 창을 숨깁니다.
예제	Sub Main ole.ReportSchedulerHide End Sub

항목	내용
형식	VOID ReportSchedulerShow ()
종류	보고서 관련 함수
내용	보고서 스케줄 창을 보여줍니다.
예제	Sub Main ole. ReportSchedulerShow End Sub

2.2.15 이력데이터 관련 함수

항목	내용
형식	VOID GetHistDataCount (BSTR szModelName, BSTR , BSTR szDataKind, DOUBLE dRefTime, LONG ICount, LONG IDir, LONG* IResultCount, SAFEARRAY(double) *psadbITimeStamp, SAFEARRAY(double) *psadbI NumValue, SAFEARRAY(LONG) *psaIStatus, SAFEARRAY(LONG) *psaIResult)
종류	이력데이터 관련 함수
내용	<p>지정된 이력모델과 태그에 대해서 요구한 시각에 발생된 이력데이터를 요청 갯수 만큼 이력 데이터(타임스탬프, 태그값, 상태값, 결과값)를 알려줍니다.</p> <p><i>szModelName</i> : 이력 모델 이름 <i>szTagName</i> : 태그 이름 <i>szDataKind</i>: 데이터 종류 (디폴트: 현재값) <i>dRefTime</i>: 요청 시각 <i>ICount</i>, : 이력 개수 <i>IDir</i>: 방향 (시간순서) <i>IResultCount</i>: 이력 정보 가지고 온 갯수 <i>psadbITimeStamp</i>: 기간내 타임 스탬프의 배열 <i>psadbI NumValue</i>: 기간내 태그 값의 배열 <i>psaIStatus</i>: 기간내 태그 상태의 배열 <i>psaIResult</i> 이력 데이터의 결과</p>

항목	내용(앞쪽에 이어서)
예제	<pre> Sub Main Dim ModelName As String Dim TagName As String Dim DataKind As String Dim RefTime As Date Dim ICount As Long Dim IDir As Long Dim ResultCount As Long Dim TimeStamp() As Double Dim NumValue() As Double Dim Status() As Long Dim Result() As Long ModelName = "INV1_SEC" TagName = "INV_1.1_AI_CT" DataKind = "현재값" RefTime = Now() ICount = 5 IDir = 1 ole.GetHistDataCount(ModelName, TagName, DataKind, RefTime, ICount, IDir, ResultCount, TimeStamp, NumValue, Status, Result) For count = 0 To ResultCount - 1 MsgBox("NumValue = " + CStr(NumValue(count)) + " Status = " + CStr(Status(count))) Next End Sub </pre>

항목	내용
형식	VOID GetHistDataPeriod (BSTR szModelName, BSTR szTagName, BSTR szDataKind, DOUBLE dFrom, DOUBLE dTo, LONG* lResultCount, SAFEARRAY(double) *psadbITimeStamp, SAFEARRAY(double) *psadbINumValue, SAFEARRAY(LONG) *psalStatus, SAFEARRAY(LONG) *psalResult)
종류	이력데이터 관련 함수
내용	지정된 이력모델과 태그에 대해서 설정된 기간내에 발생 이력 데이터 타임스탬프, 태그값, 상태값, 결과값)를 알려줍니다. <i>szModelName</i> : 이력 모델 이름 <i>szTagName</i> : 태그 이름 <i>szDataKind</i> : 데이터 종류 (디폴트: 현재값) <i>dFrom</i> : 기간 시작 시각 <i>dTo</i> : 기간 끝 시각 <i>lResultCount</i> : 이력 정보 가지고 온 갯수 <i>psadbITimeStamp</i> : 기간내 타임 스탬프의 배열 <i>psadbINumValue</i> : 기간내 태그 값의 배열 <i>psalStatus</i> : 기간내 태그 상태의 배열 <i>psalResult</i> 이력 데이터의 결과
예제	<pre> Sub Main Dim ModelName As String Dim TagName As String Dim DataKind As String Dim dFrom As Date Dim dTo As Date Dim ResultCount As Long Dim TimeStamp() As Double Dim NumValue() As Double Dim Status() As Long Dim Result() As Long ModelName = "INV1_SEC" TagName = "INV_1.1_AI_CT" DataKind = "현재값" dFrom = Now() -1 'To = Now() ole.GetHistDataPeriod(ModelName, TagName, DataKind, _ dFrom, dTo, ResultCount, TimeStamp, _ NumValue, Status, Result) For count = 0 To ResultCount -1 MsgBox("NumValue = " + CStr(NumValue(count)) + " Status = " + CStr(Status (count))) Next End Sub </pre>

항목	내용
형식	LONG GetLoggingModelList (LONG *lOutCount ,SAFEARRAY(BSTR) *psaszModelName, SAFEARRAY(BSTR) *psaszModelDesc, SAFEARRAY(BSTR) *psaszModelType, SAFEARRAY(BSTR) *psaszArchivingCycle, SAFEARRAY(BSTR) *psaszSegmentPeriod)
종류	이력데이터 관련 함수
내용	<p>이력에 등록된 모델의 리스트를 알려줍니다.</p> <p><i>lOutCount</i>: 모델 개수 <i>psaszModelName</i>: 모델명 리스트 <i>psaszModelDesc</i>: 모델 설명 리스트 <i>psaszModelType</i>: 모델 타입 리스트 <i>psaszArchivingCycle</i>: 모델 수집 기간 리스트 <i>psaszSegmentPeriod</i> 모델 저장 기간 리스트</p>
예제	<pre> Sub Main Dim lRet As Long Dim lOutCount As Long Dim aModelName() As String Dim aModelDesc() As String Dim aModelType() As String Dim aArchivingCycle() As String Dim aSegmentPeriod() As String lRet = ole.GetLoggingModelList(lOutCount, aModelName, aModelDesc, aModelType, aArchivingCycle, aSegmentPeriod) For i = 0 To lOutCount -1 MsgBox (aModelName(i) + " " + aModelDesc(i) + " "+ aModelType(i) + " " + aArchivingCycle(i) + " " + aSegmentPeriod(i)) Next i End Sub </pre>

항목	내용
형식	LONG GetLoggingTag (BSTR szModelName, LONG *lOutCount, SAFEARRAY(BSTR) *psaszTagName, SAFEARRAY(BSTR) *psaszTagType)
종류	이력데이터 관련 함수
내용	<p>주어진 이력 모델에 등록된 태그 리스트를 알려줍니다.</p> <p><i>szModelName</i>: 이력 모델명 <i>lOutCount</i>: 주어진 모델에 등록되어 있는 태그의 개수 <i>psaszTagName</i>: 주어진 모델에 등록되어 있는 태그명 리스트 <i>psaszTagType</i>: 주어진 모델에 등록되어 있는 태그의 타입 리스트</p>
예제	<pre> Sub Main Dim lRet As Long Dim lOutCount As Long Dim aTagName() As String Dim aTagType() As String lRet = ole.GetLoggingTag ("INV1_MON", lOutCount, aTagName, aTagType) For i = 0 To lOutCount -1 MsgBox ("INV1_MON " + aTagName(i) + " " + aTagType(i)) Next i End Sub </pre>

항목	내용
형식	VOID SetHistData (BSTR szModelName, BSTR szTagName, BSTR szDataKind, LONG IDataCount, SAFEARRAY(double) *psadbI SaveTime, SAFEARRAY(double) *psadbI NumValue, SAFEARRAY(LONG) *psalTagStatus, SAFEARRAY(LONG) *psalResult)
종류	이력데이터 관련 함수
내용	<p>지정된 이력모델과 태그에 대해서 각각의 지정한 시간에 이력 데이터(타임스탬프, 태그값, 상태값, 결과값)를 설정합니다.</p> <p><i>szModelName</i> : 이력 모델 이름 <i>szTagName</i> : 태그 이름 <i>szDataKind</i>: 데이터 종류 (디폴트: 현재값) <i>IDataCount</i> : 데이터 개수 <i>psadbI SaveTime</i> : 이력 데이터의 각각의 저장 시각 <i>psadbI NumValue</i> : 각 시각에서의 태그 값의 배열 <i>psalTagStatus</i> : 각 시각의 태그 상태 의 배열 <i>psalResult</i> 각 시각의 이력 데이터의 설정 결과</p>
예제	<pre> Sub Main Dim szModelName As String Dim szTagName As String Dim szDataKind As String Dim ICount As Long Dim dtSaveTime(3) As Double Dim dbI TagNumValue(3) As Double Dim ulTagStatus(3) As Long Dim IResult() As Long Dim IRet As Long szModelName = "INV1_SEC" szTagName = "INV_1.1_AI_CT" szDataKind = "현재값" ICount = 3 For count = 0 To ICount -1 dtSaveTime(count) = Now() + count*0.1 dbI TagNumValue(count) = 100 + count + 1 ulTagStatus(count) = 0 Next IRet = ole.SetHistData(szModelName ,szTagName , szDataKind, ICount , dtSaveTime, dbI TagNumValue, ulTagStatus, IResult) If IRet <> 0 Then MsgBox("SetHistData Error!!!1") Else For count = 0 To ICount -1 MsgBox("Result = " + CStr(IResult (count))) Next End If End Sub </pre>

2.2.16 I/O 드라이브 관련 함수

항목	내용
형식	VOID ClearStaStatusCount (BSTR szChannelName, BSTR szStaName)
종류	I/O 드라이브 관련 함수
내용	스테이션의 상태 갯수를 리셋합니다. szChannelName : 통신 채널 이름 szStaName : 스테이션 이름
예제	Sub Main ole. ClearStaStatusCount ("통신채널이름" , "스테이션이름") End Sub

항목	내용
형식	VOID EnableStation (BSTR szChannelName, BSTR szStaName, LONG IMode)
종류	I/O 드라이브 관련 함수
내용	I/O 드라이버의 기기 상태 창을 보여주거나 숨깁니다. szChannelName : 통신 채널 이름 szStaName : 스테이션 이름 IMode : 0 : 활성화 1: 비활성
예제	Sub Main Dim SuccessCount As Long SuccessCount = ole. EnableStation ("통신채널이름" , 스테이션이름") End Sub

항목	내용
형식	LONG GetCountReadSuccess (BSTR szChannelName, BSTR szStaName)
종류	I/O 드라이브 관련 함수
내용	통신 채널에 존재하는 스테이션의 읽은 성공 갯수를 알려줍니다. szChannelName : 통신 채널 이름 szStaName : 스테이션 이름
예제	Sub Main Dim SuccessCount As Long SuccessCount = ole. GetCountReadSuccess ("통신채널이름" , "스테이션이름") MsgBox "Read Success Count = " + CStr(SuccessCount)

End Sub

부록2 내부함수 사용법

항목	내용
형식	LONG GetCountReadTotal (BSTR szChannelName, BSTR szStaName)
종류	I/O 드라이브 관련 함수
내용	통신 채널에 존재하는 스테이션의 전체 읽은 갯수를 알려줍니다. <i>szChannelName</i> : 통신 채널 이름 <i>szStaName</i> : 스테이션 이름
예제	<pre> Sub Main Dim ReadCount As Long ReadCount = ole. GetCountReadTotal ("통신채널이름" , "스테이션이름") MsgBox "Read Count = " + CStr(ReadCount) End Sub </pre>

항목	내용
형식	LONG GetCountWriteSuccess (BSTR szChannelName, BSTR szStaName)
종류	I/O 드라이브 관련 함수
내용	통신 채널에 존재하는 스테이션의 쓴(Write) 성공 갯수를 알려줍니다. <i>szChannelName</i> : 통신 채널 이름 <i>szStaName</i> : 스테이션 이름
예제	<pre> Sub Main Dim SuccessCount As Long SuccessCount = ole. GetCountWriteSuccess ("통신채널이름" , "스테이션이름") MsgBox "Write Success Count = " + CStr(SuccessCount) End Sub </pre>

항목	내용
형식	LONG GetCountWriteTotal (BSTR szChannelName, BSTR szStaName)
종류	I/O 드라이브 관련 함수
내용	통신 채널에 존재하는 스테이션의 전체 쓴(Write) 갯수를 알려줍니다. <i>szChannelName</i> : 통신 채널 이름 <i>szStaName</i> : 스테이션 이름
예제	<pre> Sub Main Dim WriteCount As Long WriteCount = ole. GetCountWriteTotal ("통신채널이름" , "스테이션이름") MsgBox "Write Count = " + CStr(WriteCount) End Sub </pre>

항목	내용
형식	LONG GetStatusRunMode (BSTR szChannelName, BSTR szStaName)
종류	I/O 드라이브 관련 함수
내용	통신 채널에 존재하는 스테이션의 상태가 실행 모드인 갯수를 알려줍니다. szChannelName : 통신 채널 이름 szStaName : 스테이션 이름
예제	<pre> Sub Main Dim RunModeCount As Long RunModeCount = ole. GetStatusRunMode ("통신채널이름" , "스테이션이름") MsgBox "Station Run Mode Count = " + CStr(RunModeCount) End Sub </pre>

항목	내용
형식	LONG GetStatusStaOK (BSTR szChannelName, BSTR szStaName)
종류	I/O 드라이브 관련 함수
내용	통신 채널에 존재하는 스테이션의 상태가 정상인 갯수를 알려줍니다. szChannelName : 통신 채널 이름 szStaName : 스테이션 이름
예제	<pre> Sub Main Dim OKCount As Long OKCount = ole. GetStatusStaOK ("통신채널이름" , "스테이션이름") MsgBox "OK Count = " + CStr(OKCount) End Sub </pre>

항목	내용
형식	VOID ShowNetworkStatus ()
종류	I/O 드라이브 관련 함수
내용	스테이션 정보를 보여줍니다.
예제	<pre> Sub Main ole.ShowNetworkStatus End Sub </pre>

부록2 내부함수 사용법

항목	내용
형식	VOID ShowStationInfo ()
종류	I/O 드라이브 관련 함수
내용	스테이션 정보를 보여줍니다.
예제	<pre>Sub Main ole.ShowStationInfo End Sub</pre>

알아두기

내부함수 사용처

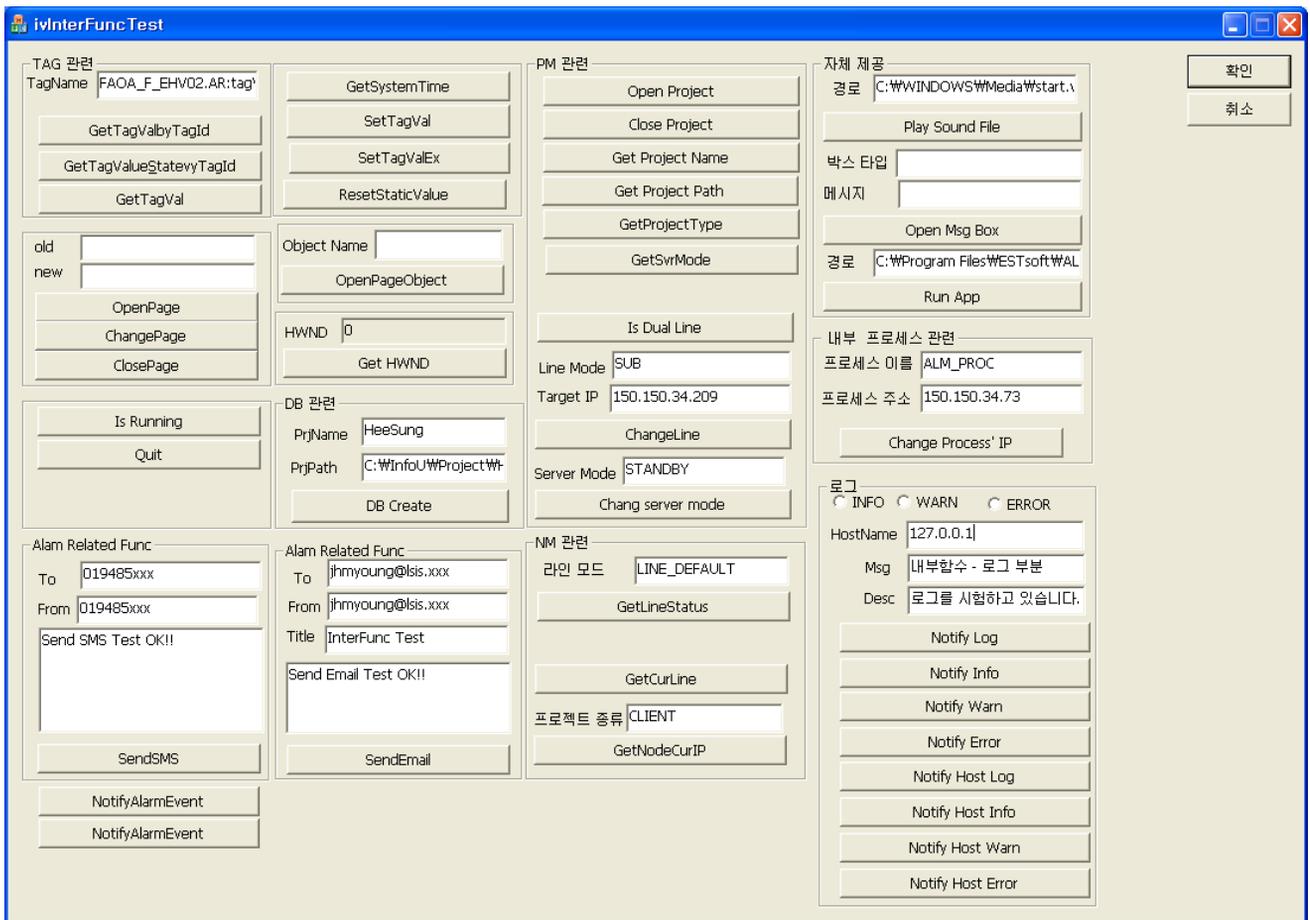
- (1) 스크립트
- (2) OLE Automation 을 사용할 수 있는 타 VC, VB 개발 프로그램

2.3 OLE Automation 사용 예

InfoU에서 지원하는 OLE Automation 을 사용하는 방법을 보여주는 예제로 아래와 같이 VB(Visual Basic)이나 VC++(Visual C++)를 사용하여 오브젝트를 생성하여 폼(Form)을 만들어서 사용할 수 있습니다..

프로그램	오브젝트 생성 방법
VB	<pre>Public iu01e as Object Set iu01e = CreateObject("iu01eSvr.iuInterFunc01eSvr")</pre>
VC++	<pre>#import "iu01eSvr\Debug\iu01eSvr.tlb" no_namespace IiuInterFunc01eSvrPtr m_piv01eSvr; m_piv01eSvr = IiuInterFunc01eSvrPtr(__uuidof(iuInterFunc01eSvr));</pre>

이렇게 생성된 오브젝트는 OLE Automation을 이용하여 InfoU에 등록된 태그 값을 가져오거나 제어를 수행 할 수 있을 뿐만 아니라, 그래픽런타임을 제어할 수 있도록 내부함수에서 제공하는 모든 함수를 사용할 수 있습니다.



부록3 비주얼베이직 스크립트 도움말	3-2
마이크로소프트 스크립트 엔진(Script Engine)의 스크립트 함수	3-2
<A>	3-7
<C>	3-9
<D>	3-15
<E>	3-21
<F>	3-22
<G>	3-26
<H>	3-27
<I>	3-28
<J>	3-34
<L>	3-35
<M>	3-38
<N>	3-41
<O>	3-41
<R>	3-42
<S>	3-45
<T>	3-52
<U>	3-55
<V>	3-56
<W>	3-57
<Y>	3-58

부록.3. 비주얼베이직 스크립트 도움말

3.1 마이크로소프트 스크립트 엔진(Script Engine)의 스크립트 함수

함수 명	형식
Abs	Function Abs(Number)
Array	Function Array(ArgList)
Asc	Function Asc(String As String)
Atn	Function Atn(Number As Double) As Double
CBool	Function CBool(Expression) As Double
CByte	Function CByte(Expression) As Byte
CCur	Function CCur(Expression) As Currency
CDate	Function CDate(Expression) As Date
CDbl	Function CDbl(Expression) As Double
Chr	Function Chr(CharCode As Long
CInt	Function CInt(Expression) As Integer
CLng	Function CLng(Expression) As Long
Cos	Function Cos(Number As Double) As Double
CreateObject	Function CreateObject(Class As String)
CSng	Function CSng(Expression) As Single
CStr	Function CStr(Expression) As String
Date	Function Date() As Date
DateAdd	Function DateAdd(Interval As String, Number As Double, Date)
DateDiff	Function DateDiff(Interval As String, Date1, Date2, [FirstDayOfWeek As VbDayOfWeek = vbSunday], [FirstWeekOfYear As VbFirstWeekOfYear = vbFirstJan1])
DatePart	Function DatePart(Interval As String, Date, [FirstDayOfWeek As VbDayOfWeek = vbSunday], [FirstWeekOfYear As VbFirstWeekOfYear = vbFirstJan1])
DateSerial	Function DateSerial(Year As Integer, Month As Integer, Day As Integer)
DateValue	Function DateValue(Date As String)
Day	Function Day(Date)
Eval	Function Eval(Expression As String)
Exp	Function Exp(Number As Double) As Double
Filter	Function Filter(InputStrings As Array, Value As String [, Include As Boolean [, Compare As VbCompareMethod = vbBinaryCompare]])
Fix	Function Int(Number)
FormatCurrency	Function FormatCurrency(Expression[, NumDigitsAfterDecimal [, UseParensForNegativeNumbers [, GroupDigits]])])
FormatDateTime	Function FormatDateTime(Date[, NamedFormat])
FormatNumber	Function FormatNumber(Expression[, NumDigitsAfterDecimal [, IncludeLeadingDigit [, UseParensForNegativeNumbers [, GroupDigits]])])

부록3 비주얼베이직 스크립트 도움말

함수 명	형식
FormatPercent	Function FormatPercent(Expression[, NumDigitsAfterDecimal [, IncludeLeadingDigit [, UseParensForNegativeNumbers [, GroupDigits]]]])
GetObject	Function GetObject([PathName], [Class])
Hex	Function Hex(Number)
Hour	Function Hour(Time)
InputBox	Function InputBox(Prompt, [Title], [Default], [XPos], [YPos], [HelpFile], [Context]) As String
InStr	Function InStr([Start], [String1], [String2], [Compare As VbCompareMethod = vbBinaryCompare])
InStrRev	Function InStrRev([Start], [String1], [String2], [Compare As VbCompareMethod = vbBinaryCompare])
Int	Function Int(Number)
IsArray	Function IsArray(VarName) As Boolean
IsDate	Function IsDate(Expression) As Boolean
IsEmpty	Function IsEmpty(Expression) As Boolean
IsNull	Function IsNull(Expression) As Boolean
IsNumeric	Function IsNumeric(Expression) As Boolean
IsObject	Function IsObject(Expression) As Boolean
Join	Function Join(List As Array[, Delimiter As String])
LBound	Function LBound(Arrayname[, Dimension As Number])
LCase	Function LCase(String)
Left	Function Left(String, Length As Long)
Len	Function Len(Expression)
LoadPicture	Function LoadPicture([filename], [widthDesired As Long], [heightDesired As Long], [flags As LoadPictureConstants = Default]) As IpictureDisp
Log	Function Log(Number As Double) As Double
LTrim	Function LTrim(String)
Mid	Function Mid(String, Start As Long, [Length])
Minute	Function Minute(Time)
Month	Function Month(Date)
MonthName	Function MonthName(Month As Number[, Abbreviate As Boolean])
MsgBox	Function MsgBox(Prompt, [Buttons As VbMsgBoxStyle = vbOKOnly], [Title], [HelpFile], [Context]) As VbMsgBoxResult
Now	Function Now()
Oct	Function Oct(Number)
Replace	Function Replace(Expression As String, Find As String, ReplaceWith As String [, Start As Number[, Count As Number[, Compare As VbCompareMethod = vbBinaryCompare]]])
Right	Function Right(String, Length As Long)
Rnd	Function Rnd([Number]) As Single
RGB	Function RGB(red, green, blue)
Round	Function Round(Expression[, NumDecimalPlaces])
RTrim	Function RTrim(String)
ScriptEngine	Function ScriptEngine() As String
ScriptEngineBuildVersion	Function ScriptEngineBuildVersion As Number

부록3 비주얼베이직 스크립트 도움말

함수 명	형식
ScriptEngineMajorVersion	Function ScriptEngineMajorVersion Function As Number
ScriptEngineMinorVersion	Function ScriptEngineMinorVersion Function As Number
Second	Function Second(Time)
Sgn	Function Sgn(Number)
Sin	Function Sin(Number As Double) As Double
Space	Function Space(Number As Long)
Split	Split(Expression[, Delimiter As String [, Count As Number[, Compare As Number]])
Sqr	Function Sqr(Number As Double) As Double
StrComp	Function StrComp(String1, String2, [Compare As VbCompareMethod = vbBinaryCompare])
String	Function String(Number As Long, Character)
StrReverse	Function StrReverse(String)
Tan	Function Tan(Number As Double) As Double
Time	Function Time() As Date
TimeSerial	Function TimeSerial(Hour As Integer, Minute As Integer, Second As Integer)
TimeValue	Function TimeValue(Time As String)
Trim	Function Trim(String)
TypeName	Function TypeName(VarName) As String
UBound	Function UBound(Arrayname[, Dimension As Number])
UCase	Function UCase(String)
VarType	Function VarType(VarName) As VbVarType
Weekday	Function WeekDay(Date, [FirstDayOfWeek As VbDayOfWeek = vbSunday])
WeekdayName	Function WeekdayName(Weekday As Number, Abbreviate As Boolean, FirstDayOfWeek As Number)
Year	Function Year(Date)



Abs 함수

설명

수의 절대값을 반환합니다.

구문

`Abs(number)`

`number` 인수는 유효한 수식이면 됩니다. `number`에 Null이 있으면 Null을 반환합니다. 그리고 초기화되지 않는 변수이면 0을 반환합니다.

참고

수의 절대값은 부호가 없는 크기만을 나타냅니다. 예를 들어 `Abs(-1)`과 `Abs(1)`은 모두 1을 반환합니다. 아래 예제에서 `Abs` 함수는 수의 절대값을 계산합니다.

Dim MyNumber

MyNumber = Abs(50.3) '50.3을 반환합니다.

MyNumber = Abs(-50.3) '50.3을 반환합니다.

Array 함수

설명

배열을 포함한 Variant를 반환합니다.

구문

`Array(arglist)`

`arglist` 인수는 쉼표로 구분된 값으로 구성되며, 이 값에 Variant에 포함된 배열 요소를 지정합니다. 인수를 지정하지 않은 경우 길이가 0인 배열이 만들어집니다.

참고

필요한 요소를 표시하는 인덱스 번호를 포함하고 있는 괄호 다음의 변수 이름으로 구성되는 변수 요소들을 참조할 때 사용하는 표기법입니다. 예를 들면 첫번째 문은 Variant와 같은 A라는 변수를 만듭니다. 두번째 문은 변수 A에 배열을 지정합니다. 마지막 문은 다른 변수의 두번째 배열 요소를 포함하는 값을 지정합니다.

Dim A As Variant

A = Array(10,20,30)

B = A(2)

Array 함수를 사용하여 만들어진 배열의 최소 범위는 Option Base 문이 지정한 최소 범위에 의해 결정됩니다.

Array 함수를 사용하여 만들어진 배열의 하한값은 Option Base 문으로 지정된 하한값에 의해 결정되고 그렇지 않으면 Array 함수의 이름은 형식 라이브러리의 이름(예: VBA.Array)으로 정해집니다. 형식 라이브러리로 이름을 지정되면 Array 함수는 Option Base 문의 영향을 받지 않습니다.

메모 배열로 선언하지 않은 Variant도 배열을 포함할 수 있습니다. Variant 변수는 고정 길이 문자열과 사용자 정의 형식을 제외한 모든 형식의 배열을 포함할 수 있습니다. 배열을 포함한 Variant 형식은 그 구성 요소가 Variant 형식인 배열과 개념적으로 다르지만 해당하는 배열의 요소들은 동일하게 액세스됩니다.

Asc 함수

설명

문자열에서 첫 글자에 해당하는 ANSI 문자 코드를 반환합니다.

구문

Asc(string)

string 인수는 유효한 문자식이면 됩니다. string에 문자가 전혀 없으면 런타임 오류가 발생합니다.

참고

Asc는 아래 예제에서 각 문자열 첫 자의 ANSI 문자 코드를 반환합니다.

```
Dim MyNumber
MyNumber = Asc("A")      '65를 반환합니다.
MyNumber = Asc("a")      '97을 반환합니다.
MyNumber = Asc("Apple") '65를 반환합니다.
```

메모 AscB 함수는 문자열이 가지는 바이트 데이터를 사용합니다. AscB는 첫 문자의 문자 코드가 아니라 그 문자의 첫 바이트를 반환합니다. AscW 함수는 유니코드(Unicode) 문자를 사용하는 32비트 플랫폼을 위해 제공되고 있습니다. 이 함수는 유니코드(Unicode) 문자 코드를 반환하므로 유니코드(Unicode)에서 ANSI 코드로 변환할 필요가 없습니다.

Atn 함수

설명

수의 아크 탄젠트를 반환합니다.

구문

Atn(number)

number 인수는 유효한 수식이면 됩니다.

참고

Atn 함수는 직각 삼각형에 있는 두 변의 길이에 의한 비(number)를 사용하여 라디안 단위의 동위각을 반환합니다. 이 비는 해당 각의 맞은 쪽에 있는 변의 길이를 이 각에 인접한 변의 길이로 나눈 것입니다. 결과의 범위는 pi/2에서 pi/2 라디안까지입니다.

도를 라디안으로 변환하려면 도에 pi/180를 곱하고, 라디안을 도로 변환하려면 라디안에 180/pi를 곱합니다.

아래 예제에서 Atn은 pi 값을 계산합니다.

```
Dim pi
pi = 4 * Atn(1)  ' pi 값을 계산합니다.
```

메모 Atn 함수는 인수로 각을 사용하여 직각 삼각형의 두 변의 비를 반환하는 Tan의 역 삼각 함수입니다. Atn을 단순히 탄젠트의 역(1/탄젠트)인 코탄젠트와 혼동하지 않도록 합니다.



CBool 함수

설명

Boolean 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

CBool(expression)
expression 인수는 유효한 식이면 됩니다.

참고

expression이 0이면 False를 반환하고 그렇지 않으면 True를 반환합니다. expression을 숫자 값으로 해석할 수 없으면 런타임 오류가 발생합니다.

아래 예제에서 CBool 함수는 식을 Boolean으로 변환합니다. 식이 0 값이 아니라면 CBool은 True를 반환하고 그렇지 않으면 False를 반환합니다.

```
Dim A, B, Check
A = 5: B = 5           ' 변수를 초기화합니다.
Check = CBool(A = B) ' Check는 True를 가집니다.

A = 0                 ' 변수를 정의합니다.
Check = CBool(A)     ' Check는 False를 가집니다.
```

CByte 함수

설명

Byte 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

CByte(expression)
expression 인수는 유효한 식이면 됩니다.

참고

대개 하위 형식의 변환 함수를 사용하여 코드를 작성하면 연산 결과가 기본 데이터 형식이 아닌 특정 데이터 형식으로 나타나는 것을 볼 수 있습니다. 예를 들어 통화, 단정밀도, 배정밀도 또는 정수 산술 연산을 할 경우에 CByte를 사용하면 바이트 산술 연산을 하게 됩니다.

CByte 함수를 사용하면 다른 형식의 데이터를 국제적으로 통용되는 Byte 하위 형식으로 변환합니다. 예를 들어 천 단위 구분 기호처럼 서로 다른 자릿수 구분 기호는 컴퓨터의 로케일 설정에 따라 적절하게 인식됩니다.

expression이 Byte 하위 형식의 허용 범위를 넘으면 오류가 발생합니다. 아래 예제에서 CByte함수는 식을 바이트로 변환합니다.

```
Dim MyDouble, MyByte
MyDouble = 125.5678   ' MyDouble은 Double입니다.
MyByte = CByte(MyDouble) ' MyByte는 값 126을 가집니다.
```

CCur 함수

설명

Currency 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

```
CCur(expression)  
expression 인수는 유효한 식이면 됩니다.
```

참고

대개 하위 형식의 변환 함수를 사용하여 코드를 작성하면 연산 결과가 기본 데이터 형식이 아닌 특정 데이터 형식으로 나타나는 것을 볼 수 있습니다. 예를 들어 정수 산술 연산을 할 경우에 CCur를 사용하면 통화 산술 연산이 실행되도록 할 수 있습니다.

CCur 함수를 사용하면 다른 형식의 데이터를 국제적으로 통용되는 Currency 하위 형식으로 변환합니다. 예를 들어 천 단위 구분 기호처럼 서로 다른 자릿수 구분 기호는 컴퓨터의 로케일 설정에 따라 적절하게 인식됩니다.

아래 예제에서 CCur 함수는 식을 Currency로 변환합니다.

```
Dim MyDouble, MyCurr  
MyDouble = 543.214588 ' MyDouble은 Double입니다.  
MyCurr = CCur(MyDouble * 2) ' MyDouble에 2 (1086.429176)를 곱한 값을 Currency (1086.4292)로 변환합니다.
```

CDate 함수

설명

Date 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

```
CDate(date)  
date 인수는 유효한 날짜식이면 됩니다.
```

참고

IsDate 함수를 사용하면 date 인수가 날짜나 시간으로 변환될 수 있는지 결정합니다. CDate 함수는 날짜 리터럴 및 시간 리터럴과 함께 날짜의 허용 범위 안에 있는 숫자를 인식합니다. 숫자를 날짜로 변환하면 정수 부분이 날짜로 변환합니다. 소수 부분은 자정을 시작으로 하는 하루 중의 시간으로 변환합니다.

CDate 함수는 컴퓨터의 로케일 설정에 따른 날짜 형식을 인식합니다. 인식하지 않는 날짜 형식을 제공하면 년, 월, 일 순서를 올바르게 결정할 수 없습니다. 또한 요일에 해당하는 문자열이 있을지라도 자세하게 표시 날짜 형식을 인식하지 않습니다.

아래 예제에서 CDate 함수는 문자열을 날짜로 변환합니다. 대개 이 예제에서와 같이 문자열로 하드 코드화된 날짜와 시간은 사용하기에 좋지 않다. 대신 #10/19/1962#, #4:45:23 PM#와 같은 날짜와 시간 리터럴을 사용합니다.

```
MyDate = "1962년 10월 19일" ' 날짜를 결정합니다.  
MyShortDate = CDate(MyDate) ' Date 데이터 형식으로 변환합니다.  
MyTime = "4:35:47 PM" ' 시간을 결정합니다.  
MyShortTime = CDate(MyTime) ' Date 데이터 형식으로 변환합니다.
```

CDBi 함수

설명

Double 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

`CDbl(expression)`
`expression` 인수는 유효한 식이면 됩니다.

참고

대개 하위 형식의 변환 함수를 사용하여 코드를 작성하면 연산 결과가 기본 데이터 형식이 아닌 특정 데이터 형식으로 나타나는 것을 볼 수 있습니다. 예를 들어 통화, 정수 산술 연산을 할 경우에 `CDbl` 또는 `CSng`를 사용하면 단정밀도나 배정밀도 산술 연산을 하게 됩니다.

`CDbl` 함수를 사용하면 다른 형식의 데이터를 국제적으로 통용되는 `Double` 하위 형식으로 변환합니다. 예를 들어 천 단위 구분 기호처럼 서로 다른 자릿수 구분 기호는 컴퓨터의 로케일 설정에 따라 적절하게 인식됩니다.

아래 예제에서 `CDbl` 함수는 식을 `Double`로 변환합니다.

```
Dim MyCurr, MyDouble
MyCurr = CCur(234.456784)           ' MyCurr는 Currency입니다.
MyDouble = CDbl(MyCurr * 8.2 * 0.01) ' 결과를 Double로 변환합니다.
```

Chr 함수

설명

지정한 ANSI 문자 코드와 연관된 문자를 반환합니다.

구문

`Chr(charcode)`
`charcode` 인수는 문자를 식별하는 번호입니다.

참고

0번에서 31번까지는 인쇄할 수 없는 ASCII 표준 코드와 같습니다. 예를 들어 `Chr(10)`은 라인 피드 문자를 반환합니다. 아래 예제에서 `Chr` 함수는 지정한 문자 코드와 연관된 문자를 반환합니다.

```
Dim MyChar
MyChar = Chr(65)   ' A를 반환합니다.
MyChar = Chr(97)   ' a를 반환합니다.
MyChar = Chr(62)   ' >를 반환합니다.
MyChar = Chr(37)   ' %를 반환합니다.
```

메모 `ChrB` 함수는 문자열이 가지는 바이트 데이터를 사용합니다. 1바이트나 2바이트 문자가 아니라 항상 1바이트를 반환합니다. `ChrW` 함수는 유니코드(Unicode) 문자를 사용하는 32비트 플랫폼을 위해 제공되고 있습니다. 이 함수의 인수는 유니코드(Unicode) 문자 코드이므로 ANSI에서 유니코드(Unicode)로 변환할 필요가 없습니다.

CInt 함수

설명

Integer 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

`CInt(expression)`
`expression` 인수는 유효한 식이면 됩니다.

참고

대개 하위 형식의 변환 함수를 사용하여 코드를 작성하면 연산 결과가 기본 데이터 형식이 아닌 특정 데이터 형식의

로 나타나는 것을 볼 수 있습니다. 예를 들어 통화, 단정밀도, 배정밀도 산술 연산을 할 경우에 CInt 또는 CLng 함수를 사용하면 정수 산술 연산을 하게 됩니다.

CInt 함수를 사용하면 다른 형식의 데이터를 국제적으로 통용되는 Integer 하위 형식으로 변환합니다. 예를 들어 천 단위 구분 기호처럼 서로 다른 자릿수 구분 기호는 컴퓨터의 로케일 설정에 따라 적절하게 인식됩니다.

expression이 Integer 하위 형식의 허용 범위를 넘으면 오류가 발생합니다.

아래 예제에서 CInt 함수는 값을 정수로 변환합니다.

```
Dim MyDouble, MyInt
MyDouble = 2345.5678      ' MyDouble은 Double입니다.
MyInt = CInt(MyDouble)   ' MyInt는 값 2346을 가집니다.
```

메모 CInt는 소수점 이하를 반올림하는 대신 잘라버리는 Fix, Int 함수 등과 다릅니다. 소수점 이하 부분이 정확히 0.5이면 CInt 함수는 항상 가장 가까운 짝수로 반올림합니다. 예를 들어 0.5는 0으로 반올림하고 1.5는 2로 반올림합니다.

CLng 함수

설명

Long 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

```
CLng(expression)
expression 인수는 유효한 식이면 됩니다.
```

참고

대개 하위 형식의 변환 함수를 사용하여 코드를 작성하면 연산 결과가 기본 데이터 형식이 아닌 특정 데이터 형식으로 나타나는 것을 볼 수 있습니다. 예를 들어 통화, 단정밀도, 배정밀도 산술 연산을 할 경우에 CInt 또는 CLng 함수를 사용하면 정수 산술 연산을 하게 됩니다.

CLng 함수를 사용하면 다른 형식의 데이터를 국제적으로 통용되는 Long 하위 형식으로 변환합니다. 예를 들어 천 단위 구분 기호처럼 서로 다른 자릿수 구분 기호는 컴퓨터의 로케일 설정에 따라 적절하게 인식됩니다.

expression이 Long 하위 형식의 허용 범위를 넘으면 오류가 발생합니다.

아래 예제에서 CLng 함수는 값을 Long으로 변환합니다.

```
Dim MyVal1, MyVal2, MyLong1, MyLong2
MyVal1 = 25427.45: MyVal2 = 25427.55      ' MyVal1과 MyVal2는 Doubles입니다.
MyLong1 = CLng(MyVal1)                   ' MyLong1은 값 25427을 가집니다.
MyLong2 = CLng(MyVal2)                   ' MyLong2은 값 25428을 가집니다.
```

메모 CLng 함수는 소수점 이하를 반올림하는 대신 잘라버리는 Fix와 Int 함수와 다릅니다. 소수점 이하 부분이 정확히 0.5이면 CLng 함수는 항상 가장 가까운 짝수로 반올림합니다. 예를 들어 0.5는 0으로 반올림하고 1.5는 2로 반올림합니다.

Cos 함수

설명

각의 코사인 값을 반환합니다.

구문

Cos(number)

number 인수는 각을 라디안으로 표시하는 유효한 수식이면 됩니다.

참고

Cos 함수는 각을 사용하고 직각 삼각형의 두 변의 비 즉, 각의 접변과 빗변 사이의 길이의 비를 반환합니다. 결과의 범위는 -1에서 1까지입니다.

각을 라디안으로 변환하려면 각에 pi/180을 곱하고 라디안을 각으로 변환하려면 라디안에 180/pi를 곱합니다. 아래 예제에서 Cos 함수는 각의 코사인 값을 반환합니다.

```
Dim MyAngle, MySecant
```

```
MyAngle = 1.3 ' 각을 라디안으로 정의합니다.
```

```
MySecant = 1 / Cos(MyAngle) ' 시컨트 값을 계산합니다.
```

CreateObject 함수

설명

자동화 개체에 대한 참조를 만들어 반환합니다.

구문

```
CreateObject(class)
```

class 인수는 servername.typename 구문을 사용하며, 아래 인수를 가집니다.

인수 설명

servername 개체를 제공하는 응용 프로그램의 이름입니다.

typename 만들 개체의 형식이나 클래스입니다.

참고

자동화 서버에서는 적어도 하나 이상의 개체 형식을 제공합니다. 예를 들어 워드 프로세싱 프로그램은 응용 프로그램 개체, 문서 개체 및 도구 모음 개체를 제공합니다.

자동화 개체를 만들려면 아래와 같이 CreateObject 함수에서 반환한 개체를 개체 변수로 지정합니다.

```
Dim ExcelSheet
```

```
Set ExcelSheet = CreateObject("Excel.Sheet")
```

이 코드는 해당 개체, 즉 Microsoft Excel 스프레드 시트를 만들면서 응용 프로그램을 시작합니다. 일단 개체를 만든 후에는 사용자가 정의한 개체 변수를 사용하여 코드 안에서 그 개체를 참조합니다. 아래 예제에서는 Application 개체와 ActiveSheet.Cells 컬렉션을 포함하여 ExcelSheet 개체 변수 및 Excel 개체를 사용하여 새 개체의 속성과 메서드를 액세스합니다. 사용 예는 아래와 같습니다.

```
' Application 개체를 통해 Excel 프로그램이 나타나도록 합니다.
```

```
ExcelSheet.Application.Visible = True
```

```
' 시트의 첫째 셀에 텍스트를 놓는다.
```

```
ExcelSheet.ActiveSheet.Cells(1,1).Value = "A열 1행입니다."
```

```
' 시트를 저장합니다.
```

```
ExcelSheet.SaveAs "C:\WDOCSWTEST.XLS"
```

```
' Application 개체의 Quit 메서드를 사용하여 Excel 프로그램을 닫습니다.
```

```
ExcelSheet.Application.Quit
```

```
' 개체 변수를 릴리즈합니다.
```

```
Set ExcelSheet = Nothing
```

CSng 함수

설명

Single 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

CSng(expression)
 expression 인수는 유효한 식이면 됩니다.

참고

대개 데이터 형식 변환 함수를 사용하여 코드를 작성하면 연산 결과가 기본 데이터 형식이 아닌 특정 데이터 형식으로 나타나는 것을 볼 수 있습니다. 예를 들어 일반적으로 통화나 정수 산술 연산을 할 경우에 CDbi 또는 CSng를 사용하면 단정밀도나 배정밀도 산술 연산을 하게 됩니다.

CSng 함수를 사용하면 다른 형식의 데이터를 국제적으로 통용되는 Single 하위 형식으로 변환합니다. 예를 들어 천 단위 구분 기호처럼 서로 다른 자릿수 구분 기호는 컴퓨터의 로케일 설정에 따라 적절하게 인식됩니다. expression이 Single 하위 형식의 허용 범위를 넘으면 오류가 발생합니다.

아래 예제에서 CSng 함수는 값을 Single로 변환합니다.

```
Dim MyDouble1, MyDouble2, MySingle1, MySingle2 ' MyDouble1과 MyDouble2는 Doubles입니다.
MyDouble1 = 75.3421115: MyDouble2 = 75.3421555
MySingle1 = CSng(MyDouble1) ' MySingle1은 값 75.34211을 가집니다.
MySingle2 = CSng(MyDouble2) ' MySingle2은 값 75.34216.을 가집니다.
```

CStr 함수

설명

String 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

CStr(expression)
 expression 인수는 유효한 식이면 됩니다.

참고

대개 데이터 형식 변환 함수를 사용하여 코드를 작성하면 연산 결과가 기본 데이터 형식이 아닌 특정 데이터 형식으로 나타나는 것을 볼 수 있습니다. 예를 들어 CStr를 사용하면 이 결과를 String으로 표시하게 됩니다.

Str 함수 대신 CStr 함수를 사용하면 다른 형식의 데이터를 국제적으로 통용되는 String 하위 형식으로 변환합니다. 예를 들어 서로 다른 자릿수 구분 기호는 컴퓨터의 로케일 설정에 따라 적절하게 인식됩니다.

expression에서 사용하는 데이터는 아래 표처럼 반환되는 결과를 결정합니다.

expression의 데이터	CStr 반환 결과
Boolean	True 또는 False를 포함하는 문자열
Date	컴퓨터에서 설정된 간단하게 표시 날짜 형식의 날짜를 포함하는 문자열
Null	런타임 오류
Empty	길이가 0인 문자열("")
Error	Error 단어 다음에 오류 번호가 나타나는 문자열
기타	숫자 번호를 포함하는 문자열

아래 예제에서 CStr 함수는 숫자 값을 String으로 변환합니다.

```
Dim MyDouble, MyString
```

```
MyDouble = 437.324      ' MyDouble은 Double입니다.
MyString = CStr(MyDouble) ' MyString은 값 "437.324"를 가집니다.
```



Date 함수

설명

현재 시스템의 날짜를 반환합니다.

구문

```
Date
```

참고

아래 예제에서 Date 함수는 현재 시스템 날짜를 반환합니다.

```
Dim MyDate
MyDate = Date      ' MyDate는 현재 시스템 날짜를 가집니다.
```

DateAdd 함수

설명

지정된 시간 간격을 추가한 날짜를 반환합니다.

구문

```
DateAdd(interval, number, date)
DateAdd 함수 구문은 아래 인수를 가집니다.
```

인수 설명

interval	필수적인 인수. interval을 추가한 날짜를 나타내는 문자식입니다. 그 값에 대해서는 아래의 설정을 참조합니다.
number	필수적인 인수. 추가할 간격 수의 수식입니다. 수식에서 양수는 미래의 날짜, 음수는 과거의 날짜입니다.
date	필수적인 인수. interval을 추가한 날짜를 나타내는 Variant 또는 리터럴입니다.

설정

interval 인수의 값은 아래와 같습니다.

설정	설명
yyyy	년
q	분기
m	월
y	일(일년 기준)
d	일
w	요일
ww	주(일년 기준)

h	시
n	분
s	초

참고

DateAdd 함수를 사용하면 날짜에서 지정한 시간 간격을 더하거나 뺄 수 있습니다. 예를 들어 DateAdd 함수를 사용하면 오늘부터 30일 뒤의 날짜 또는 현재 시간에서 45분 뒤의 시간을 계산할 수 있습니다. date에 어느 날을 더하려면 일년을 기준으로 한 일("y"), 일("d") 또는 요일("w")을 사용하면 됩니다.

DateAdd 함수에서는 잘못된 날짜는 반환하지 않습니다. 아래 예제는 1월 31일에 한 달을 더합니다.

```
NewDate = DateAdd("m", 1, "31-Jan-95")
```

이 경우 DateAdd 함수에서는 31-Feb-95가 아니라 28-Feb-95를 반환합니다. date가 31-Jan-96이면 1996년이 윤년이므로 29-Feb-96을 반환합니다.

계산되는 날짜가 서기 100년 이전이면 오류가 발생합니다.

Long 값의 수가 아니면 계산하기 전에 먼저 가장 가까운 정수로 반올림합니다.

DateDiff 함수

설명

두 날짜 사이의 간격 수를 반환합니다.

구문

```
DateDiff(interval, date1, date2 [,firstdayofweek[, firstweekofyear]])
```

DateDiff 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>interval</i>	필수적인 인수. 날짜1과 날짜2 사이의 차이를 계산하는 데 사용할 <i>interval</i> 의 문자식. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>date1, date2</i>	필수적인 인수. 날짜식에서 계산에 사용할 두 날짜입니다.
<i>firstdayofweek</i>	선택적인 인수. 요일을 지정하는 상수로 지정하지 않으면 일요일로 간주합니다. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>firstweekofyear</i>	선택적인 인수. 연도를 기준으로 한 첫째 주를 지정하는 상수입니다. 지정하지 않으면 1월 1일을 포함하는 주를 첫째 주로 간주합니다. 그 값에 대해서는 아래의 설정을 참조합니다.

인수 설명

interval 필수적인 인수. 날짜1과 날짜2 사이의 차이를 계산하는 데 사용할 *interval*의 문자식. 그 값에 대해서는 아래의 설정을 참조합니다.

date1, date2 필수적인 인수. 날짜식에서 계산에 사용할 두 날짜입니다.

firstdayofweek 선택적인 인수. 요일을 지정하는 상수로 지정하지 않으면 일요일로 간주합니다. 그 값에 대해서는 아래의 설정을 참조합니다.

firstweekofyear 선택적인 인수. 연도를 기준으로 한 첫째 주를 지정하는 상수입니다. 지정하지 않으면 1월 1일을

포함하는 주를 첫째 주로 간주합니다. 그 값에 대해서는 아래의 설정을 참조합니다.

설정

interval 인수의 값은 아래와 같습니다.

설정	설명
yyyy	년
q	분기
m	월
y	일(일년 기준)
d	일
w	요일
ww	주(일년 기준)
h	시
n	분
s	초

f firstweekofyear 인수의 값은 아래와 같습니다.

상수	값	설명
vbUseSystem	0	NLS(National Language Support) API 설정 사용
vbSunday	1	일요일(기본값)
vbMonday	2	월요일
vbTuesday	3	화요일
vbWednesday	4	수요일
vbThursday	5	목요일
vbFriday	6	금요일
vbSaturday	7	토요일

상수 값 설명

상수	값	설명
vbUseSystem	0	NLS(National Language Support) API 설정 사용
vbFirstJan1	1	1월 1일을 포함하는 주에서 시작(기본값)
vbFirstFourDays	2	새해의 처음 4일을 포함하는 주에서 시작
vbFirstFullWeek	3	새해의 처음 한 주일(7일)을 포함하는 주에서 시작

참고

DateDiff 함수를 사용하면 두 날짜 사이에 얼마만한 시간 간격이 있는 지를 알 수 있습니다. 예를 들어 DateDiff 함수를 사용하여 두 날짜 사이의 일 수 또는 오늘부터 올해 말까지의 주일 수를 계산할 수 있습니다.

date1과 date2 사이의 날짜 수를 계산하려면 연도를 기준으로 한 일("y")이나 일("d")을 사용할 수 있습니다. interval이 요일("w")이면 DateDiff 함수는 두 날짜 사이의 주일 수를 반환합니다. date1이 월요일인 경우 DateDiff 함수는 date2까지의 월요일 수를 계산합니다. 즉, date1이 아니라 date2를 계산합니다. 그러나 interval이 주("ww")인 경우 DateDiff 함수는 두 날짜 사이의 달력 상의 주일 수를 반환합니다. 즉, date1에서 date2 사이의 일요일 수를 계산합니다. DateDiff 함수에서는 date2가 일요일이면 date2를 계산하지, date1도 일요일인 경우 date1은 계산하지 않습니다.

date1이 date2보다 더 나중의 시점인 경우, DateDiff 함수는 음수를 반환합니다.

firstdayofweek 인수는 "w" 및 "ww" 간격 기호를 사용하는 계산에 영향을 미칩니다.

date1이나 date2가 날짜 리터럴인 경우 지정한 연도는 해당 날짜의 영구적인 요소가 됩니다. 그러나 date1이나 date2가 인용 부호(" ")로 둘러 싸이고 해당 연도를 생략할 경우 date1이나 date2 식을 계산할 때마다 사용자의 코드에 현재 연도가 삽입됩니다. 이런 방식으로 다음 연도에도 사용할 수 있는 코드를 작성할 수 있습니다.

12월 31일을 바로 다음 해의 1월 1일과 비교할 경우 단 하루가 경과했을지라도 DateDiff는 년("yyyy") 인수에 1을 반환합니다.

아래 예제에서 DateDiff 함수는 주어진 날짜로부터 현재까지의 경과일을 표시합니다.

```
Function DiffADate(theDate)
    DiffADate = "경과일: " & DateDiff("d", Now, theDate)
End Function
```

DatePart 함수

설명

주어진 날짜의 지정된 부분을 반환합니다.

구문

```
DatePart(interval, date[, firstdayofweek[, firstweekofyear]])
```

DatePart 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>interval</i>	필수적인 인수. 반환할 시간 간격의 문자식입니다. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>date</i>	필수적인 인수. 계산할 날짜식입니다.
<i>firstdayof week</i>	선택적인 인수. 요일을 지정하는 상수로서 지정하지 않으면 일요일로 간주합니다. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>firstweekofyear</i>	선택적인 인수. 연도를 기준으로 한 첫째 주를 지정하는 상수입니다. 지정하지 않으면 1월 1일을 포함하는 주를 첫째 주로 간주합니다. 그 값에 대해서는 아래의 설정을 참조합니다.

설정

interval 인수의 값은 아래와 같습니다.

설정	설명
<i>yyyy</i>	년
<i>q</i>	분기
<i>m</i>	월
<i>y</i>	일(일년 기준)
<i>d</i>	일
<i>w</i>	요일
<i>ww</i>	주(일년 기준)
<i>h</i>	시
<i>n</i>	분
<i>s</i>	초

firstweekofyear 인수의 값은 아래와 같습니다.

상수	값	설명
vbUseSystem	0	NLS(National Language Support) API 설정 사용

vbSunday	1	일요일(기본값)
vbMonday	2	월요일
vbTuesday	3	화요일
vbWednesday	4	수요일
vbThursday	5	목요일
vbFriday	6	금요일
vbSaturday	7	토요일

상수 값 설명

상수	값	설명
vbUseSystem	0	NLS(National Language Support) API 설정 사용
vbFirstJan1	1	1월 1일을 포함하는 주에서 시작(기본값)
vbFirstFourDays	2	새해의 처음 4일을 포함하는 주에서 시작
vbFirstFullWeek	3	새해의 처음 한 주일(7일)을 포함하는 주에서 시작

참고

DatePart 함수를 사용하면 날짜를 계산하고 특정 시간 간격을 반환합니다. 예를 들어 DatePart를 사용하여 요일이나 현재 시간을 계산할 수 있습니다.

firstdayofweek 인수는 "w" 및 "ww" 간격 기호를 사용하는 계산에 영향을 줍니다.

date가 날짜 리터럴인 경우 지정한 연도는 해당 날짜의 영구적인 요소가 됩니다. 그러나 date가 인용 부호(" ")로 둘러 싸이고 해당 연도를 생략할 경우 date식을 계산할 때마다 사용자의 코드에 현재 연도가 삽입됩니다. 이런 방식으로 다른 해에도 사용할 수 있는 코드를 작성할 수 있습니다.

아래 예제는 날짜를 가지고 DatePart 함수를 사용하여 해당 연도의 분기를 표시합니다.

```
Function GetQuarter(TheDate)
    GetQuarter = DatePart("q", TheDate)
End Function
```

DateSerial 함수

설명

지정된 년, 월, 일의 Date 하위 형식인 Variant를 반환합니다.

구문

```
DateSerial(year, month, day)
DateSerial 함수 구문은 아래 인수를 가집니다
```

인수	설명
<i>year</i>	100에서 9999까지의 수 또는 수식
<i>month</i>	모든 수식
<i>day</i>	모든 수식

참고

예를 들어 1991년 12월 31일처럼 어떤 날짜를 지정하려면 각 DateSerial 인수의 숫자는 해당 인수의 허용 범위 안에 있어야 합니다. 즉, 일은 1-31이며, 월은 1-12입니다. 그러나 특정 날짜의 앞 뒤에 년, 월, 일 등을 나타내는 수식을 사용하여 각 인수의 상대 날짜를 지정할 수 있습니다.

아래 예제는 절대 날짜 수 대신 수식을 사용합니다. 여기서 DateSerial 함수는 1990년의 10년 전(1990-10) 8월의 2

개월 전(8 -2) 첫째 날에서 하루를 뺀 날짜(1-1), 즉 1980년 5월 31일을 반환합니다.

```
DateSerial(1990 - 10, 8 - 2, 1 - 1)
```

year 인수의 값이 0에서 99까지의 값이면 1900년에서 1999년까지로 해석됩니다. 그 밖의 모든 year 인수에서는 1800년처럼 4자리수의 완전한 연도를 사용합니다.

인수가 허용되는 범위를 넘으면 그 다음으로 큰 단위로 넘어갑니다. 예를 들어 35일을 지정하면 적용 연도에 따라 1개월 며칠로 계산합니다. 그러나 어느 한 인수가 -32,768~32,767 범위를 넘는 인수가거나 세 인수를 사용하여 직접 또는 식으로 지정한 날짜가 허용 범위를 넘을 경우 오류가 발생합니다.

DateValue 함수

설명

Date 하위 형식의 Variant를 반환합니다.

구문

```
DateValue(date)
```

대개 date 인수는 100년 1월 1일에서 9999년 12월 31일까지를 나타내는 문자식으로 이 범위에 있는 날짜, 시간 또는 날짜/시간을 나타내는 식입니다.

참고

date 인수에 시간 정보가 있는 경우 DateValue 함수에서는 그 정보를 반환하지 않습니다. 그러나 date 인수에 "89:98"처럼 잘못된 시간 정보가 있으면 오류가 발생합니다.

date 인수가 유효한 날짜 구분 기호로 분리된 숫자만 있는 문자열인 경우 DateValue 함수에서는 컴퓨터에 설정한 간단한 표시 날짜 형식에 따라 년, 월, 일에 대한 순서로 인식하게 됩니다. 또한 정식이나 약식의 월 이름을 포함하고 있는 명확한 날짜도 인식합니다. 예를 들어 DateValue 함수는 2/30/1991과 12/30/91 뿐만 아니라 December 30, 1991과 Dec 30, 1991을 모두 인식합니다.

date 인수에서 연도 부분을 생략할 경우 DateValue 함수는 컴퓨터 시스템에 설정되어 있는 현재 연도를 사용합니다.

아래 예제에서 DateValue 함수는 문자열을 날짜로 변환합니다. 날짜 리터럴을 사용하여 직접 날짜를 Variant 변수에 할당할 수도 있습니다.(예를 들면 MyDate = #9/11/63#).

```
Dim MyDate
```

```
MyDate = DateValue("September 11, 1963") ' 날짜를 반환합니다.
```

Day 함수

설명

날짜를 나타내는 1에서 31까지의 정수를 반환합니다.

구문

```
Day(date)
```

date 인수는 날짜를 표현하는 식입니다. date 인수에 Null이 있으면 Null을 반환합니다.

아래 예제에서 Day 함수는 지정한 날짜로부터 그 달의 날짜를 가져옵니다.

```
Dim MyDay
```

```
MyDay = Day("1962년 10월 19일") 'MyDay는 값 19를 가집니다.
```



Eval 함수

설명

인자로 넘어온 문장을 계산하여 결과 값 또는 변수의 stack값을 반환합니다.

구문

```
Eval(Expr[, Depth])
```

인자:

Expr : 연산되어지는 식으로써 VB스크립트 구문이 스트링 값으로 입력되어야 합니다.

Depth: 숫자로 입력되어야 하며, 로컬 변수의 stack에 값의 위치를 입력한다. 만약 Depth = 0 이면 현재 연산단계의 값을 반환하며, 생략되면 Depth는 0

참고

아래 예제는 Eval 함수의 사용예 입니다.

예제1)

```
Dim Guess, RndNum
RndNum = Int((100) * Rnd(1) + 1)
Guess = CInt(InputBox("Enter your guess:",,0))
Do
    If Eval("Guess = RndNum") Then
        MsgBox "Congratulations! You guessed it!"
        Exit Sub
    Else
        Guess = CInt(InputBox("Sorry! Try again.",,0))
    End If
Loop Until Guess = 0
```

예제2)

```
Sub Main
    Dim X As String
    X = "Hello"
    Debug.Print Eval("X")      'Hello를 반환
    A
End Sub
Sub A
    Dim X As String
    X = "Bye"
    Debug.Print Eval("X")      'Bye를 반환
    Debug.Print Eval("X",1)    'Hello를 반환 이전값인
End Sub
```

Exp 함수

설명

승수로 거듭 제공하는 자연 로그의 밑인 e를 반환합니다.

구문

Exp(number)
number 인수는 유효한 수식이면 됩니다.

참고

number의 값이 709.782712893을 넘으면 오류가 발생합니다. 상수 e는 대략 2.718282입니다.

메모

Exp 함수는 Log 함수와 보완 관계에 있으며 때로 역대수라고 합니다.
아래 예제에서 Exp 함수는 멱에 대한 e를 반환합니다.

```
Dim MyAngle, MyHSin ' 각을 라디안으로 정의합니다.
MyAngle = 1.3      ' 쌍곡선의 싸인 값을 계산합니다.
MyHSin = (Exp(MyAngle) - Exp(-1 * MyAngle)) / 2
```



Filter 함수

설명

지정한 필터 기준에 따라 문자열 배열의 부분 집합을 가지고 있는 0 기준 배열을 반환합니다.

구문

Filter(InputStrings, value[, Include[, compare]])
Filter 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>InputStrings</i>	필수적인 인수. 검색할 문자열의 1차원 배열입니다.
<i>value</i>	필수적인 인수. 검색할 문자열입니다.
<i>Include</i>	선택적인 인수. <i>value</i> 을 포함하거나 제외할 하위 문자열을 반환할지 여부를 나타내는 부울 값입니다. True인 경우 Filter 함수는 <i>value</i> 에 해당하는 배열의 부분 집합을 하위 문자열로 반환합니다. False인 경우 Filter 함수는 <i>value</i> 에 해당하지 않는 배열의 부분 집합을 하위 문자열로 반환합니다.
<i>compare</i>	선택적인 인수. 사용할 비교 형식을 나타내는 숫자 값입니다. 그 값에 대해서는 아래의 설정을 참조합니다.

설정

compare 인수의 값은 아래와 같습니다.

상수	값	설명
vbBinaryCompare	0	이진 비교를 실행합니다.
vbTextCompare	1	텍스트 비교를 실행합니다.

참고

InputStrings에 일치하는 value가 없는 경우 Filter 함수에서는 빈 배열을 반환합니다. InputStrings가 Null이거나 1차원 배열이 아닌 경우 오류가 발생합니다.

Filter 함수에서 반환하는 배열은 일치하는 항목 수를 포함하는 데 충분한 요소만 가집니다.

아래 예제에서 Filter 함수는 "Mon" 검색 기준을 가지고 있는 배열을 반환합니다.

```
Dim MyIndex
Dim MyArray (3)
MyArray(0) = "일요일"
MyArray(1) = "월요일"
MyArray(2) = "화요일"
MyIndex = Filter(MyArray, "Mon") 'MyIndex(0) contains "월요일"
```

Fix 함수

설명

숫자의 정수 부분을 반환합니다.

구문

```
Int(number)
Fix(number)
```

number 인수는 유효한 수식이면 됩니다. number에 Null이 있으면 Null을 반환합니다.

참고

Int 함수와 Fix 함수는 모두 number의 소숫점 이하 부분을 없앤 그 결과로 나타나는 정수 값을 반환합니다.

Int 함수와 Fix 함수의 차이점은 number가 음수일 경우 Int 함수는 number보다 작거나 같은 첫째 음의 정수를 반환하지만, Fix 함수는 number보다 크거나 같은 첫째 음의 정수를 반환합니다. 예를 들어 Int 함수는 -8.4를 -9로, Fix 함수는 -8.4를 -8로 변환합니다.

Fix(number) 함수는 아래 식과 같습니다.

```
Sgn(number) * Int(Abs(number))
```

아래 예제는 Int와 Fix 함수가 수의 정수 부분을 어떻게 반환하는 지를 보여줍니다.

```
MyNumber = Int(99.8)      ' 99를 반환합니다.
MyNumber = Fix(99.2)     ' 99를 반환합니다.
MyNumber = Int(-99.8)    ' -100을 반환합니다.
MyNumber = Fix(-99.8)    ' -99를 반환합니다.
MyNumber = Int(-99.2)    ' -100을 반환합니다.
MyNumber = Fix(-99.2)    ' -99를 반환합니다.
```

FormatCurrency 함수

설명

컴퓨터의 [제어판]에 정의된 통화 기호를 사용하는 통화 값으로 형식화된 식을 반환합니다.

구문

```
FormatCurrency(expression[,NumDigitsAfterDecimal [,IncludeLeadingDigit [,UseParensForNegativeNumbers [,GroupDigits]]]])
```

FormatCurrency 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>expression</i>	필수적인 인수. 서식이 지정되는 식입니다.

부록3 비주얼베이직 스크립트 도움말

인수	설명
<i>NumDigitsAfterDecimal</i>	선택적인 인수. 소수점 이하의 자릿수를 나타내는 숫자 값입니다. 기본값은 -1로 컴퓨터의 국가별 설정 사용을 나타냅니다.
<i>IncludeLeadingDigit</i>	선택적인 인수. 소수점 이하의 유효 숫자 앞에 0을 표시할지 여부를 나타내는 Tristate 상수입니다. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>UseParensForNegativeNumbers</i>	선택적인 인수. 음수 값을 괄호 안에 넣을지 여부를 나타내는 Tristate 상수입니다. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>GroupDigits</i>	선택적인 인수. 컴퓨터의 국가별 설정에서 지정한 그룹 구분 기호로 숫자를 구분할지 여부를 나타내는 Tristate 상수입니다.

설정

IncludeLeadingDigit, UseParensForNegativeNumbers, GroupDigits 인수는 아래 설정을 가집니다.

상수	값	설명
TristateTrue	-1	True
TristateFalse	0	False
TristateUseDefault	-2	컴퓨터의 국가별 설정 값 사용

참고

하나 이상의 인수를 생략하면 컴퓨터의 국가별 설정 값을 생략된 인수 값으로 사용합니다. 통화 가치와 관련된 통화 기호의 위치는 컴퓨터의 국가별 설정값에 의해 결정됩니다.

메모 [숫자] 탭에 있는 선행 0(영)을 제외한 모든 설정 정보는 [국가별 설정]의 [통화] 탭에 있습니다.

FormatDateTime 함수

설명

날짜 또는 시간으로 형식화된 식을 반환합니다.

구문

FormatDateTime(date[,NamedFormat])

FormatDateTime 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>date</i>	필수적인 인수. 날짜식을 지정합니다.
<i>NamedFormat</i>	선택적인 인수. 사용한 날짜/시간 형식을 나타내는 숫자 값입니다. 이 인수를 생략하면 vbGeneralDate를 사용합니다.

설정

NamedFormat 인수는 아래 설정을 가집니다.

상수	값	설명
vbGeneralDate	0	날짜 또는 시간을 표시합니다. 날짜 부분이 있으면 간단하게 표시 날짜 형식으로 날짜를 표시하며 시간 부분이 있으면 자세하게 표시 시간 형식으로 시간을 표시합니다. 날짜와 시간이 같이 있으면 두 부분을 모두 표시합니다.
vbLongDate	1	컴퓨터의 국가별 설정에서 지정한 자세하게 표시 날짜 형식으로 날짜를 표시합니다.
vbShortDate	2	컴퓨터의 국가별 설정에서 지정한 간단하게 표시 날짜 형식으로 날짜를 표시합니다.
vbLongTime	3	컴퓨터의 국가별 설정에서 지정한 시간 형식으로 시간을 표시합니다.
vbShortTime	4	24시간 형식(hh:mm)의 시간을 표시합니다.

참고

아래 예제에서 FormatDateTime 함수는 식을 긴 날짜 형식으로 형식 지정하고 이를 MyDateTime에 보냅니다.

```
Function GetCurrentDate
    'FormatDateTime은 Date를 긴 날짜 형식으로 지정합니다.
    GetCurrentDate = FormatDateTime(Date, 1)
End Function
```

FormatNumber 함수

설명

숫자로 형식화된 식을 반환합니다.

구문

```
FormatNumber (expression[, NumDigitsAfterDecimal [, IncludeLeadingDigit [, UseParensForNegativeNumbers [, GroupDigits]]]])
FormatNumber 함수 구문은 아래 인수를 가집니다.
```

인수	설명
<i>expression</i>	필수적인 인수. 서식이 지정되는 식입니다.
<i>NumDigitsAfterDecimal</i>	선택적인 인수. 표시될 소수점 이하의 자릿수를 나타내는 숫자 값입니다. 기본값인 -1은 컴퓨터의 국가별 설정을 사용함을 나타냅니다.
<i>IncludeLeadingDigit</i>	선택적인 인수. 소수점 이하의 유효 숫자 앞에 0을 표시할지 여부를 나타내는 Tristate 상수입니다. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>UseParensForNegativeNumbers</i>	선택적인 인수. 음수 값을 괄호 안에 넣을지 여부를 나타내는 Tristate 상수입니다. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>GroupDigits</i>	선택적인 인수. [제어판]에 설정된 그룹 구분 기호를 사용하여 숫자를 구분할지 여부를 나타내는 Tristate 상수입니다. 그 값에 대해서는 아래의 설정을 참조합니다.

설정

IncludeLeadingDigit, UseParensForNegativeNumbers, GroupDigits 인수는 아래 설정을 가집니다.

상수	값	설명
TristateTrue	-1	True
TristateFalse	0	False
TristateUseDefault	-2	컴퓨터의 국가별 설정 값을 사용합니다.

참고

하나 이상의 선택적인 인수를 생략하면 컴퓨터의 국가별 설정 값을 생략된 인수 값으로 사용합니다.

메모 모든 설정 정보는 [국가별 설정]의 [숫자] 탭에 있습니다.

아래 예제에서 FormatNumber 함수는 십진법의 네 자릿수로 숫자를 형식 지정합니다.

```
Function FormatNumberDemo
    Dim MyAngle, MySecant, MyNumber
    MyAngle = 1.3 ' 각을 라디안으로 정의합니다.
    MySecant = 1 / Cos(MyAngle) ' 시컨트 값을 계산합니다.
    FormatNumberDemo = FormatNumber(MySecant,4) ' MySecant를 십진법 네 자릿수로 형식 지정합니다.
End Function
```

FormatPercent 함수

설명

끝에 % 문자가 있고 100으로 곱해진 백분율로 서식이 지정된 식을 반환합니다.

구문

```
FormatPercent(expression[,NumDigitsAfterDecimal [,IncludeLeadingDigit [,UseParensForNegativeNumbers [,GroupDigits]]]])
```

FormatPercent 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>expression</i>	필수적인 인수. 형식화된 식을 반환합니다.
<i>NumDigitsAfterDecimal</i>	선택적인 인수. 표시될 소수점 이하의 자릿수를 나타내는 숫자값입니다. 기본값인 -1은 컴퓨터의 국가별 설정을 사용함을 나타냅니다.
<i>IncludeLeadingDigit</i>	선택적인 인수. 소수점 이하의 유효 숫자 앞에 0을 표시할지 여부를 나타내는 Tristate 상수입니다. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>UseParensForNegativeNumbers</i>	선택적인 인수. 음수 값을 괄호 안에 넣을지 여부를 나타내는 Tristate 상수입니다. 그 값에 대해서는 아래의 설정을 참조합니다.
<i>GroupDigits</i>	선택적인 인수. [제어판]에 설정된 그룹 구분 기호로 숫자를 구분할지 여부를 나타내는 Tristate 상수입니다. 그 값에 대해서는 아래의 설정을 참조합니다.

설정

IncludeLeadingDigit, UseParensForNegativeNumbers, GroupDigits 인수는 아래 설정을 가집니다.

상수	값	설명
TristateTrue	-1	True
TristateFalse	0	False
TristateUseDefault	-2	컴퓨터의 국가별 설정 값을 사용합니다.

참고

하나 이상의 선택적인 인수를 생략하면 컴퓨터의 국가별 설정값을 생략한 인수 값으로 사용합니다.

메모 모든 설정 정보는 [국가별 설정]의 [숫자] 탭에 있습니다.

아래 예제에서 FormatPercent 함수는 식을 백분율로 형식 지정합니다.

```
Dim MyPercent
```

```
MyPercent = FormatPercent(2/32) 'MyPercent는 6.25%를 가집니다.
```



GetObject 함수

설명

파일에서 자동화 개체에 대한 참조를 반환합니다.

구문

```
GetObject([pathname] [, class])
```

GetObject 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>pathname</i>	선택적인 인수. 검색할 개체가 있는 파일 이름과 그 파일의 전체 경로를 나타내는 문자열입니다. <i>pathname</i> 을 생략하면 <i>class</i> 가 반드시 있어야 합니다.
<i>class</i>	선택적인 인수. 개체의 클래스 인 문자열입니다.

class 인수는 *appName.objecttype* 구문을 사용하며, 그 구성 요소는 아래와 같이 설명됩니다.

구성 요소	설명
<i>appName</i>	필수적인 인수. 개체를 제공하는 응용 프로그램의 이름인 문자열입니다.
<i>objecttype</i>	필수적인 인수. 만들 개체의 클래스나 형식을 나타내는 문자열입니다.



Hex 함수

설명

숫자의 16진수 값을 나타내는 문자열을 반환합니다.

구문

```
Hex(number)
```

number 인수는 유효한 식이면 됩니다.

참고

정수가 아닌 *number*는 계산하기 전에 가장 가까운 정수로 반올림됩니다.

number의 값	Hex 함수의 결과
Null	Null
Empty	0
위 두 값을 제외한 모든 수	8개 이하의 16진수 문자

유효한 범위의 숫자 앞에 &H를 붙여 16진수를 표현할 수 있습니다. 예를 들어 &H10은 십진수 16의 16진수 표기를 나타냅니다.

아래 예제에서 Hex 함수는 수의 16진 값을 반환합니다.

```
Dim MyHex
```

```
MyHex = Hex(5) ' 5를 반환합니다.
```

```
MyHex = Hex(10) ' A를 반환합니다.
```

`MyHex = Hex(459) ' 1CB를 반환합니다.`

Hour 함수

설명

하루의 시간을 나타내는 0에서 23까지의 정수를 반환합니다.

구문

`Hour(time)`
 time 인수는 시간을 나타내는 식입니다. time이 Null이면 Null을 반환합니다.

아래 예제에서 Hour 함수는 현재 시간으로부터 시(hour)를 가져옵니다.

```
Dim MyTime, MyHour
MyTime = Now
MyHour = Hour(MyTime) ' MyHour는 현재 시각을 나타내는 숫자를 가집니다.
```

<|>

InputBox 함수

설명

대화 상자에 프롬프트를 표시하거나, 텍스트 입력이나 단추 누르기를 기다리거나 입력란의 내용을 반환합니다.

구문

`InputBox(prompt[, title][, default][, xpos][, ypos][, helpfile, context])`

InputBox 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>prompt</i>	대화 상자에서 메시지로 나타나는 문자식 입니다. <i>prompt</i> 의 최대 길이는 사용되는 문자의 너비에 따라 다르지만 약 1,024자입니다. 두 줄 이상의 <i>prompt</i> 면 캐리지 리턴 문자(Chr(13)), 라인 피드 문자(Chr(10)) 또는 캐리지 리턴-라인 피드 문자의 조합인(Chr(13) & Chr(10))을 사용하여 줄을 구분할 수 있습니다.
<i>title</i>	대화 상자의 제목 표시줄에 나타나는 문자식입니다. <i>title</i> 을 생략하면 제목 표시줄에 응용 프로그램 이름이 나타납니다.
<i>default</i>	아무 것도 입력하지 않았을 때 기본 응답으로 입력란에 나타나는 문자식입니다. <i>default</i> 를 생략하면 빈 입력란이 나타납니다.
<i>xpos</i>	화면의 왼쪽 끝에서 대화 상자의 왼쪽 끝까지의 수평 길이를 트립 단위로 지정한 수식 입니다. <i>xpos</i> 를 생략하면 대화 상자가 화면의 수평을 기준으로 가운데 위치합니다.
<i>ypos</i>	화면의 위쪽 끝에서 대화 상자의 위쪽 끝까지의 수직 길이를 트립 단위로 지정한 수식입니다. <i>ypos</i> 를 생략하면 대화 상자가 화면 밑에서 수직으로 1/3 지점에 위치합니다.
<i>helpfile</i>	대화 상자의 상황에 따른 도움말을 제공하기 위해 사용할 도움말 파일을 식별하는 문자식입니다. <i>helpfile</i> 을 제공하면 <i>context</i> 도 제공해야 합니다.
<i>context</i>	도움말 작성자가 해당 도움말 항목으로 지정한 도움말 문맥 번호를 식별하는 수식입니다. <i>context</i> 를 제공하면 <i>helpfile</i> 도 제공해야 합니다.

참고

helpfile과 context를 모두 제공하면 대화 상자에 [도움말] 단추를 자동으로 추가합니다.

InputBox 함수에서는 [확인]을 누르거나 키를 누르면 입력란의 내용을 반환합니다. [취소]를 누르면 길이가 0인 문자열("")을 반환합니다.

아래 예제에서 InputBox 함수는 입력 상자를 표시하고 문자열을 변수 입력에 지정합니다.

```
Dim Input
Input = InputBox("이름을 입력합니다")
MsgBox ("다음 이름이 입력되었습니다: " & Input)
```

InStr 함수

설명

한 문자열이 다른 문자열에서 처음 나타나는 위치를 반환합니다.

구문

InStr([start,]string1, string2[, compare])

InStr 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>start</i>	선택적인 인수. 각 검색을 시작할 위치를 지정하는 수식 입니다. 생략하면 첫째 문자에서 검색합니다. <i>start</i> 인수가 Null 이면 오류가 발생합니다. <i>Compare</i> 를 지정하려면 <i>start</i> 인수가 필요합니다.
<i>string1</i>	필수적인 인수. 찾을 문자식 입니다.
<i>string2</i>	필수적인 인수. 찾기 대상이 되는 문자식입니다.
<i>compare</i>	선택적인 인수. 하위 문자열을 평가하는 데 사용할 비교 형식을 나타내는 숫자 값입니다. 그 값에 대해서는 아래의 설정을 참조합니다. 생략하면 이진 비교를 실행합니다.

설정

compare 인수의 값은 아래와 같습니다.

상수	값	설명
vbBinaryCompare	0	이진 비교를 실행합니다.
vbTextCompare	1	텍스트 비교를 실행합니다.

결과 값

InStr 함수는 아래 값을 반환합니다.

경우	InStr 함수의 결과
<i>string1</i> 의 길이가 0인 경우	0
<i>string1</i> 이 Null인 경우	Null
<i>string2</i> 가 0인 경우	<i>start</i>
<i>string2</i> 가 Null인 경우	Null
<i>string2</i> 가 없는 경우	0
<i>string1</i> 에 <i>string2</i> 가 있는 경우	<i>string1</i> 에서의 일치하는 해당 위치
<i>start</i> > Len(<i>string2</i>)	0

참고

아래 예제에서 InStr은 문자열을 검색합니다.

```
Dim SearchString, SearchChar, MyPos
SearchString = "XpXpXpXpXp" ' 검색할 문자열
SearchChar = "P" ' "P" 검색.

' 4가 있는 위치에서 텍스트 비교 시작. 6을 반환합니다.
MyPos = Instr(4, SearchString, SearchChar, 1)

' 10이 있는 위치에서 2진 비교 시작. 9를 반환합니다.
MyPos = Instr(1, SearchString, SearchChar, 0)

' Comparison is binary by default (last argument is omitted).
MyPos = Instr(SearchString, SearchChar) ' Returns 9.

' A binary comparison starting at position 1. Returns 0 ("W" is not found).
MyPos = Instr(1, SearchString, "W")
```

메모

InStrB 함수는 문자열이 가지는 바이트 데이터를 사용합니다. InStrB 함수는 한 문자열이 다른 문자열에서 처음 나타나는 위치를 반환하는 대신 바이트 위치를 반환합니다.

InStrRev 함수

설명

한 문자열이 다른 문자열에 나타나는 위치를 문자열 끝에서 반환합니다.

구문

```
InStrRev(string1, string2[, start[, compare]])
```

InStrRev 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>string1</i>	필수적인 인수. 찾을 범위가 되는 문자식 입니다.
<i>string2</i>	필수적인 인수. 찾기 대상이 되는 문자식입니다.
<i>start</i>	선택적인 인수. 각 검색의 시작 위치를 지정하는 수식 입니다. 생략하면 -1을 사용하는데 마지막 문자에서 검색함을 의미합니다. <i>start</i> 이 Null 이면 오류가 발생합니다.
<i>compare</i>	선택적인 인수. 하위 문자열을 평가하는 데 사용할 비교 형식을 나타내는 숫자 값입니다. 생략하면 이진 비교를 실행합니다. 그 값에 대해서는 아래의 설정을 참조합니다.

설정

compare 인수의 값은 아래와 같습니다.

상수	값	설명
vbBinaryCompare	0	이진 비교를 실행합니다.
vbTextCompare	1	텍스트 비교를 실행합니다.

결과 값

InStrRev 함수는 아래 값을 반환합니다.

경우	InStrRev 함수의 결과
<i>string1</i> 의 길이가 0인 경우	0
<i>string1</i> 이 Null인 경우	Null
<i>string2</i> 의 길이가 0인 경우	<i>start</i>

<i>string2</i> 가 Null인 경우	Null
<i>string2</i> 가 없는 경우	0
<i>string1</i> 에 <i>string2</i> 가 있는 경우	<i>string1</i> 에서의 일치하는 해당 위치
<i>start</i> > Len(<i>string2</i>)	0

참고

아래 예제에서 InStrRev 함수는 문자열을 검색합니다.

```
Dim SearchString, SearchChar, MyPos
```

```
SearchString = "XpXpXpXpXp" ' 검색할 문자열
```

```
SearchChar = "P" ' "P" 검색
```

```
' 10이 있는 위치에서 2진 비교 시작. 9를 반환합니다.
```

```
MyPos = InStrRev(SearchString, SearchChar, 10, 0)
```

```
' 마지막 위치에서 텍스트 비교 시작. 12를 반환합니다.
```

```
MyPos = InStrRev(SearchString, SearchChar, -1, 1)
```

```
' 비교는 기본값이 2진입니다(마지막 인수는 생략됨). 0을 반환합니다.
```

```
MyPos = InStrRev(SearchString, SearchChar, 8)
```

메모 InStrRev 함수 구문은 InStr 함수 구문과 다릅니다.

Int 함수

설명

숫자의 정수 부분을 반환합니다.

구문

```
Int(number)
```

```
Fix(number)
```

number 인수는 유효한 수식이면 됩니다. number에 Null이 있으면 Null을 반환합니다.

참고

Int 함수와 Fix 함수는 모두 number의 소숫점 이하 부분을 없앤 그 결과로 나타나는 정수 값을 반환합니다.

Int 함수와 Fix 함수의 차이점은 number가 음수일 경우 Int 함수는 number보다 작거나 같은 첫째 음의 정수를 반환하지만, Fix 함수는 number보다 크거나 같은 첫째 음의 정수를 반환합니다. 예를 들어 Int 함수는 -8.4를 -9로, Fix 함수는 -8.4를 -8로 변환합니다.

Fix(number) 함수는 아래 식과 같습니다.

```
Sgn(number) * Int(Abs(number))
```

아래 예제는 Int와 Fix 함수가 수의 정수 부분을 어떻게 반환하는 지를 보여줍니다.

```
MyNumber = Int(99.8) ' 99를 반환합니다.
```

```
MyNumber = Fix(99.2) ' 99를 반환합니다.
```

```
MyNumber = Int(-99.8) ' -100을 반환합니다.
```

```
MyNumber = Fix(-99.8) ' -99를 반환합니다.
```

```
MyNumber = Int(-99.2) ' -100을 반환합니다.
```

```
MyNumber = Fix(-99.2) ' -99를 반환합니다.
```

IsArray 함수

설명

변수가 배열인지 여부를 나타내는 부울 값을 반환합니다.

구문

```
IsArray(varname)  
varname 인수는 변수입니다.
```

참고

IsArray 함수는 변수가 배열이면 True를, 변수가 배열이 아니면 False를 반환합니다. IsArray 함수는 배열을 포함하는 변이형(Variant)을 사용할 경우 특히 유용합니다.

아래 예제에서 IsArray 함수는 MyVariable이 배열인 지를 테스트합니다.

```
Dim MyVariable  
Dim MyArray(3)  
MyArray(0) = "일요일"  
MyArray(1) = "월요일"  
MyArray(2) = "화요일"  
MyVariable = IsArray(MyArray) ' MyVariable "True"를 가집니다.
```

IsDate 함수

설명

한 식이 날짜로 변환될 수 있는지 여부를 나타내는 부울 값을 반환합니다.

구문

```
IsDate(expression)  
expression 인수는 날짜식 또는 날짜나 시간으로 인식될 수 있는 문자식입니다.
```

참고

IsDate 함수는 해당 식이 날짜이거나 유효한 날짜로 변환될 수 있는 경우 True를, 그렇지 않으면 False를 반환합니다. Microsoft Windows의 유효한 날짜 범위는 서기 100년 1월 1일에서 서기 9999년 12월 31일까지이지만 운영 체제에 따라 그 범위가 달라집니다.

아래 예제에서 IsDate 함수는 식이 날짜로 변환될 수 있는 지 여부를 결정합니다.

```
Dim MyDate, YourDate, NoDate, MyCheck  
MyDate = "1962년 10월 19일": YourDate = #10/19/62#: NoDate = "Hello"  
MyCheck = IsDate(MyDate) ' True를 반환합니다.  
MyCheck = IsDate(YourDate) ' True를 반환합니다.  
MyCheck = IsDate(NoDate) ' False를 반환합니다.
```

IsEmpty 함수

설명

변수가 초기화되었는지 여부를 나타내는 부울 값을 반환합니다.

구문

IsEmpty(expression)

expression 인수는 아무 식이면 됩니다. 그러나 IsEmpty 함수는 개별 변수가 초기화되었는지 여부를 확인하는 데 사용하므로 대개 expression 인수는 단일 변수의 이름입니다.

참고

IsEmpty 함수는 변수가 초기화되지 않았거나 명시적으로 변수를 Empty로 지정한 경우 True를, 그렇지 않으면 False를 반환합니다. expression에 둘 이상의 변수가 있으면 항상 False를 반환합니다.

아래 예제에서 IsEmpty 함수는 변수가 초기화되었는지 여부를 판단합니다.

```
Dim MyVar, MyCheck
MyCheck = IsEmpty(MyVar)      ' True를 반환합니다.

MyVar = Null                  ' Assign Null.
MyCheck = IsEmpty(MyVar)     ' False를 반환합니다.

MyVar = Empty                 ' Assign Empty.
MyCheck = IsEmpty(MyVar)     ' True를 반환합니다.
```

IsNull 함수

설명

Null식에 유효한 데이터가 없는지, 즉 Null인지 여부를 나타내는 부울 값을 반환합니다.

구문

IsNull(expression)
expression 인수는 아무 식이면 됩니다.

참고

IsNull 함수는 expression이 Null인 경우, 즉 유효한 데이터가 없는 경우 True를, 그렇지 않은 경우 False를 반환합니다. expression이 둘 이상의 변수로 이루어진 경우 모든 구성 변수에 들어 있는 Null은 전체 식에 대해 True를 반환하도록 합니다.

Null 값은 해당 변수에 유효한 데이터가 없음을 나타냅니다. Null은 변수가 아직 초기화되지 않았음을 나타내는 Empty와 다르며, 때때로 널 문자열이라고 하는 길이가 0인 문자열("")과도 다릅니다.

메모

IsNull 함수를 사용하여 식에 Null 값이 있는지 여부를 결정합니다. If Var = Null 및 If Var <> Null과 같은 식처럼 어떤 경우에는 True일 것으로 예상할 수도 있는 식조차도 항상 False입니다. 왜냐하면 Null을 포함하는 식 자체가 Null이므로 False가 되기 때문입니다.

아래 예제에서 IsNull 함수는 변수가 Null 값을 가지는지 여부를 결정합니다.

```
Dim MyVar, MyCheck
MyCheck = IsNull(MyVar)      ' False를 반환합니다.

MyVar = Null                  ' Null을 지정합니다.
MyCheck = IsNull(MyVar)     ' True를 반환합니다.

MyVar = Empty                 ' Empty를 지정합니다.
```

```
MyCheck = IsNull(MyVar) ' False를 지정합니다.
```

IsNumeric 함수

설명

식이 숫자로 평가될 수 있는지 여부를 나타내는 부울 값을 반환합니다.

구문

```
IsNumeric(expression)  
expression 인수는 아무 식이면 됩니다.
```

참고

expression이 날짜식이면 IsNumeric 함수는 False를 반환합니다.
아래 예제에서 IsNumeric 함수는 변수가 숫자로 평가될 수 있는 지 여부를 결정합니다.

```
Dim MyVar, MyCheck  
MyVar = 53 ' 값을 지정합니다.  
MyCheck = IsNumeric(MyVar) ' True를 반환합니다.  
  
MyVar = "459.95" ' 값을 지정합니다.  
MyCheck = IsNumeric(MyVar) ' True를 반환합니다.  
  
MyVar = "45 Help" ' 값을 지정합니다.  
MyCheck = IsNumeric(MyVar) ' False를 반환합니다.
```

IsObject 함수

설명

식이 유효한 자동화 개체를 참조하는지 여부를 나타내는 부울 값을 반환합니다.

구문

```
IsObject(expression)  
expression 인수는 아무 식이면 됩니다.
```

참고

IsObject 함수는 해당 expression이 Object 하위 형식의 변수이거나 사용자 정의 개체인 경우 True를 그렇지 않으면 False를 반환합니다.
아래 예제에서 IsObject 함수는 식별자가 개체 변수를 나타낼 지 여부를 결정합니다.

```
Dim MyInt, MyCheck, MyObject  
Set MyObject = Me  
MyCheck = IsObject(MyObject) ' True를 반환합니다.  
MyCheck = IsObject(MyInt) ' False를 반환합니다.
```



Join 함수

설명

배열에 포함되어 있는 여러 하위 문자열을 결합하여 만든 문자열을 반환합니다.

구문

```
Join(list[, delimiter])
```

Join 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>list</i>	필수적인 인수. 결합할 하위 문자열을 포함하는 1차원 배열입니다.
<i>delimiter</i>	선택적인 인수. 반환되는 문자열의 하위 문자열을 분리하는 데 사용하는 문자입니다. 생략하면 공백 문자(" ")를 사용합니다. <i>delimiter</i> 가 길이가 0인 문자열이면 목록에 있는 모든 항목을 구분 기호없이 연결합니다.



LBound 함수

설명

지정한 배열 차원에 사용할 수 있는 가장 작은 첨자를 반환합니다.

구문

```
LBound(arrayname[, dimension])
```

LBound 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>arrayname</i>	배열 변수의 이름으로 일반 변수 이름 짓기 규칙을 따릅니다.
<i>dimension</i>	반환될 차원의 하한값을 나타내는 정수를 반환합니다. 1차원은 1, 2차원은 2, 등등으로 사용합니다. <i>dimension</i> 을 생략하면 1로 간주합니다.

참고

LBound 함수는 UBound 함수와 함께 사용하여 배열의 크기를 결정합니다. UBound 함수를 사용하면 배열 차원의 상한값을 알 수 있습니다.

모든 차원의 하한값은 항상 0입니다.

LCase 함수

설명

소문자로 변환된 문자열을 반환합니다.

구문

```
LCase(string)
```

string 인수는 유효한 문자식이면 됩니다. string에 Null이 있으면 Null을 반환합니다.

참고

대문자만 소문자로 변환하고 소문자와 글자가 아닌 문자는 변환하지 않고 그대로 둡니다.

아래 예제에서 LCase 함수는 대문자를 소문자로 변환합니다.

```
Dim MyString
```

```
Dim LCaseString
```

```
MyString = "VBScript"
LCaseString = LCase(MyString) ' LCaseString은 "vbscript" 를 가집니다.
```

Left 함수

설명

문자열의 왼쪽부터 지정한 수만큼의 문자를 반환합니다.

구문

```
Left(string, length)
```

Left 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>string</i>	왼쪽 끝 문자부터 반환될 문자식 입니다. <i>string</i> 에 Null 이 있으면 Null을 반환합니다.
<i>length</i>	반환할 문자 개수를 나타내는 수식 입니다. 0이면 길이가 0인 문자열("")을 반환합니다. <i>string</i> 에 있는 문자 개수보다 크거나 같으면 전체 문자열을 반환합니다.

참고

*string*의 문자 개수를 결정하려면 Len 함수를 사용합니다.

아래 예제에서 Left 함수는 MyString의 처음 세 문자를 반환합니다.

```
Dim MyString, LeftString
MyString = "VBScript"
LeftString = Left(MyString, 3) 'LeftString은 "VBS"를 가집니다.
```

메모

LeftB 함수는 문자열이 가지는 바이트 데이터를 사용합니다. length 인수에 반환할 문자 개수를 지정하는 대신 바이트 수를 지정합니다.

Len 함수

설명

문자열에 있는 문자수 또는 변수를 저장하는 데 필요한 바이트 수를 반환합니다.

구문

```
Len(string | varname)
```

Len 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>string</i>	문자식 입니다. <i>string</i> 에 Null 이 있으면 Null을 반환합니다.
<i>varname</i>	유효한 변수 이름이면 됩니다. <i>varname</i> 에 Null이 있으면 Null을 반환합니다.

참고

아래 예제에서 Len 함수는 한 문자열의 문자 개수를 반환합니다.

```
Dim MyString
MyString = Len("VBSCRIPT") 'MyString은 값 8을 가집니다.
```

메모

LenB 함수는 문자열이 가지는 바이트 데이터를 사용합니다. 문자열에 있는 문자수를 반환하는 대신 LenB는 이 문자

열을 표현하는 데 사용된 바이트 수를 반환합니다.

LoadPicture 함수

설명

그림 개체를 반환하며 32비트 플랫폼에서만 사용할 수 있습니다.

구문

```
LoadPicture(picturename)
picturename 인수는 로드할 그림 파일의 이름을 나타내는 문자식입니다.
```

참고

LoadPicture 함수에서 인식하는 그래픽 형식은 비트맵 파일(.bmp), 아이콘 파일(.ico), 실행 길이 부호화 파일(.rle), 메타 파일(.wmf), 향상된 메타 파일(.emf), GIF 파일(.gif) 및 JPEG 파일(.jpg)입니다.

Log 함수

설명

수의 자연 로그를 반환합니다.

구문

```
Log(number)
number 인수는 0보다 큰 유효한 수식이면 됩니다.
```

참고

자연 로그는 밑이 e인 로그입니다. 상수 e는 약 2.718282입니다.

아래와 같이 자연 로그 $\log(x)$ 를 자연 로그 $\log(n)$ 으로 나누면 밑이 n이고 값이 x인 자연 로그 $\log_n(x)$ 를 계산할 수 있습니다.

$$\text{Log}_n(x) = \text{Log}(x) / \text{Log}(n)$$

아래 예제는 밑이 10인 로그 함수를 계산하는 사용자 정의 함수(Function)입니다.

```
Function Log10(X)
```

```
    Log10 = Log(X) / Log(10)
```

```
End Function
```

LTrim 함수

설명

LTrim 함수는 선행 공백이 없는, RTrim 함수는 후미 공백이 없는, Trim 함수는 선행 공백과 후미 공백이 모두 없는 문자열 사본을 반환합니다.

구문

```
LTrim(string)
RTrim(string)
```

```
Trim(string)
```

string 인수는 유효한 문자식이면 됩니다. string에 Null이 있으면 Null을 반환합니다.

참고

아래 예제에서 LTrim 함수는 선행 공백을, RTrim함수는 후미 공백을, Trim 함수는 선행 공백과 후미 공백 모두를 각각 잘라냅니다.

```
Dim MyVar
MyVar = LTrim(" vbscript ") 'MyVar는 "vbscript"를 가집니다.
MyVar = RTrim(" vbscript ") 'MyVar는 " vbscript"를 가집니다.
MyVar = Trim(" vbscript ") 'MyVar는 "vbscript"를 가집니다.
```



Mid 함수

설명

문자열에서 지정한 개수의 문자를 반환합니다.

구문

```
Mid(string, start[, length])
```

Mid 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>string</i>	반환될 문자가 들어 있는 문자열입니다. <i>string</i> 에 Null 이 있으면 Null을 반환합니다.
<i>start</i>	<i>string</i> 에서 반환을 시작하는 문자 위치입니다. <i>start</i> 인수가 <i>string</i> 의 문자 개수보다 크면, Mid 함수는 길이가 0인 문자열("")을 반환합니다.
<i>length</i>	반환할 문자 개수입니다. 생략하거나 실제의 문자 개수가 <i>start</i> 인수의 문자를 포함하여 <i>length</i> 인수의 문자보다 작으면, <i>start</i> 위치에서 해당 문자열의 마지막 문자까지 모두 반환합니다.

참고

Len 함수를 사용하면 *string*의 문자 개수를 결정합니다.

아래 예제에서 Mid 함수는 한 문자열의 4번째 문자에서부터 시작하는 6개의 문자를 반환합니다.

```
Dim MyVar
MyVar = Mid("VB Script is fun!", 4, 6) 'MyVar는 "Script"를 반환합니다.
```

메모

MidB 함수는 문자열이 가지는 바이트 데이터를 사용합니다. 이 함수에서는 문자 개수를 지정하는 대신 바이트 수를 지정합니다.

Minute 함수

설명

시간의 분을 나타내는 0에서 59까지의 정수를 반환합니다.

구문

```
Minute(time)
```

time 인수는 시간을 나타내는 식입니다. *time*에 Null이 있으면 Null을 반환합니다.

참고

아래 예제에서 Minute 함수는 분을 반환합니다.

```
Dim MyVar
MyVar = Minute(Now)
```

Month 함수

설명

년의 월을 나타내는 1에서 12까지의 정수를 반환합니다.

구문

```
Month(date)
```

date 인수는 날짜를 나타내는 식입니다. date에 Null이 있으면 Null을 반환합니다.

참고

아래 예제에서 Month 함수는 현재의 월을 반환합니다.

```
Dim MyVar
MyVar = Month(Now) ' MyVar는 현재의 월에 해당하는 숫자를 가집니다.
```

MonthName 함수

설명

지정한 월을 나타내는 문자열을 반환합니다.

구문

```
MonthName(month[, abbreviate])
```

MonthName 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>month</i>	필수적인 인수. 1월은 1, 2월은 2, 등등으로 월을 나타내는 숫자입니다.
<i>abbreviate</i>	선택적인 인수. 월 이름을 약어 형태로 표시할지 여부를 나타내는 부울 값입니다. 생략하면 기본값인 False가 되며, 월 이름을 완전한 형태로 표시합니다.

참고

아래 예제에서 MonthName 함수는 날짜식에 들어갈 월 이름의 약어를 반환합니다.

```
Dim MyVar
MyVar = MonthName(10, True) ' MyVar는 "Oct"를 가집니다.
```

MsgBox 함수

설명

대화 상자에 메시지를 표시하거나 단추 누르기를 기다리거나, 사용자가 누른 단추를 가리키는 값을 반환합니다.

구문

```
MsgBox(prompt[, button][, title][, helpfile, context])
```

MsgBox 함수 구문은 아래 인수를 가집니다.

인수	설명
----	----

부록3 비주얼베이직 스크립트 도움말

인수	설명
<i>prompt</i>	대화 상자에서 메시지로 나타나는 문자식입니다. <i>prompt</i> 의 최대 길이는 사용되는 문자의 너비에 따라 다르지만 약 1,024자입니다. 두 줄 이상의 <i>prompt</i> 이면 캐리지 리턴 문자(Chr(13)), 라인 피드 문자(Chr(10)) 또는 캐리지 리턴-라인 피드 문자의 조합인(Chr(13) & Chr(10))을 사용하여 줄을 구분할 수 있습니다.
<i>button</i>	표시할 단추의 종류와 번호, 사용할 아이콘 유형, 기본 단추의 확인 및 메시지 상자의 모달성을 지정하는 값들의 합계인 <u>수식</u> 입니다. 그 값에 대해서는 아래의 설정을 참조합니다. 생략하면 단추의 기본값은 0입니다.
<i>title</i>	대화 상자의 제목 표시줄에 나타나는 문자식입니다. <i>title</i> 을 생략하면 제목 표시줄에 응용 프로그램 이름이 나타납니다.
<i>helpfile</i>	대화 상자의 문맥 인식 도움말을 제공하기 위해 사용할 도움말 파일을 식별하는 문자식입니다. <i>helpfile</i> 을 제공하면 <i>context</i> 도 제공해야 합니다. 16 비트 플랫폼에서는 사용할 수 없습니다.
<i>context</i>	도움말 작성자가 해당 도움말 항목으로 지정한 도움말 문맥 번호를 식별하는 수식입니다. <i>context</i> 을 제공하면 <i>helpfile</i> 파일도 제공해야 합니다. 16 비트 플랫폼에서는 사용할 수 없습니다.

설정

button 인수 설정은 아래와 같습니다.

상수	값	설명
<i>vbOKOnly</i>	0	확인 단추만 표시합니다.
<i>vbOKCancel</i>	1	확인 및 취소 단추를 표시합니다.
<i>vbAbortRetryIgnore</i>	2	중단, 다시 하기 및 무시 단추를 표시합니다.
<i>vbYesNoCancel</i>	3	예, 아니오 및 취소 단추를 표시합니다.
<i>vbYesNo</i>	4	예 및 아니오 단추를 표시합니다.
<i>vbRetryCancel</i>	5	다시 하기 및 취소 단추를 표시합니다.
<i>vbCritical</i>	16	중대 오류 메시지 아이콘을 표시합니다.
<i>vbQuestion</i>	32	경고 쿼리 아이콘을 표시합니다.
<i>vbExclamation</i>	48	경고 메시지 아이콘을 표시합니다.
<i>vbInformation</i>	64	정보 메시지 아이콘을 표시합니다.
<i>vbDefaultButton1</i>	0	첫째 단추가 기본 단추입니다.
<i>vbDefaultButton2</i>	256	둘째 단추가 기본 단추입니다.
<i>vbDefaultButton3</i>	512	셋째 단추가 기본 단추입니다.
<i>vbDefaultButton4</i>	768	넷째 단추가 기본 단추입니다.
<i>vbApplicationModal</i>	0	응용 프로그램 모달입니다. 사용자는 현재 응용 프로그램에서 작업을 계속하기 전에 메시지 상자에 응답해야 합니다.
<i>vbSystemModal</i>	4096	시스템 모달입니다. 사용자가 메시지 상자에 응답할 때까지 모든 응용 프로그램은 정지하고 있습니다.

첫째 그룹의 값(0-5)은 대화 상자에 표시되는 단추의 유형과 번호를 나타냅니다. 둘째 그룹(16, 32, 48, 64)은 아이콘 유형을 나타냅니다. 셋째 그룹(0, 256, 512, 768)은 기본 단추를 결정하고, 넷째 그룹(0, 4096)은 메시지 상자의 모달성을 결정합니다. 값들을 합산하여 단추 인수의 최종 값을 만들 때 각 그룹으로부터 하나의 숫자만 사용합니다.

결과 값

MsgBox 함수는 아래 값을 반환합니다.

상수	값	단추
vbOK	1	확인
vbCancel	2	취소
vbAbort	3	중단
vbRetry	4	다시 하기
vbIgnore	5	무시
vbYes	6	예
vbNo	7	아니오

참고

helpfile과 context를 모두 제공하면 F1 키를 눌러 상황에 따른 도움말 항목을 볼 수 있습니다.

대화 상자에 [취소] 단추가 있으면 Esc 키를 눌러도 [취소] 단추를 누르는 것과 같은 결과를 얻습니다. 대화 상자에 [도움말] 단추가 있는 경우 해당 대화 상자에 대한 상황에 따른 도움말이 제공됩니다. 그러나 다른 단추 중 하나를 누를 때까지는 어떤 값도 반환하지 않습니다.

아래 예제에서 MsgBox 함수는 메시지 상자를 표시하고 선택한 단추를 설명하는 값을 반환합니다.

```
Dim MyVar
MyVar = MsgBox ("안녕하십니까?", 65, "MsgBox Example") ' MyVar는 선택한 단추에 따라 값 1이나 2를 가집니다.
```

**Now 함수****설명**

컴퓨터의 시스템 날짜와 시간에 따른 현재 날짜와 시간을 반환합니다.

구문

```
Now
```

참고

아래 예제에서 Now 함수는 현재 날짜와 시간을 반환합니다.

```
Dim MyVar
MyVar = Now ' MyVar는 현재 날짜와 시간을 가집니다.
```

**Oct 함수****설명**

숫자의 8진수 값을 반환합니다.

구문

```
Oct(number)
number 인수는 유효한 식이면 됩니다.
```

참고

정수가 아닌 number는 계산하기 전에 가장 가까운 정수로 반올림됩니다.

수의 값 종류	Oct 함수의 결과
Null	Null
Empty	0
위 두 값을 제외한 모든 수	11개 이하의 8진수 문자

유효한 범위의 숫자 바로 앞에 &0를 붙여 8진수를 표현할 수 있습니다. 예를 들어 &010은 십진수 8의 8진수 표기를 나타냅니다.

아래 예제에서 Oct 함수는 8진수 값을 반환합니다.

```
Dim MyOct
MyOct = Oct(4)      ' 4를 반환합니다.
MyOct = Oct(8)      ' 10을 반환합니다.
MyOct = Oct(459)    ' 713을 반환합니다.
```



Replace 함수

설명

지정한 하위 문자열을 지정한 횟수만큼 다른 하위 문자열로 바꾼 문자열을 반환합니다.

구문

```
Replace(expression, find, replacewith[, start[, count[, compare]])
```

Replace 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>expression</i>	필수적인 인수. 바꿀 하위 문자열이 들어 있는 문자식 입니다.
<i>find</i>	필수적인 인수. 찾을 하위 문자열입니다.
<i>replacewith</i>	필수적인 인수. 바꿀 하위 문자열입니다.
<i>start</i>	선택적인 인수. <i>expression</i> 에서의 하위 문자열 검색을 시작하는 위치입니다. 생략하면 1로 간주합니다. 반드시 <i>count</i> 와 함께 사용해야 합니다.
<i>number</i>	선택적인 인수. 하위 문자열의 바꾸기 횟수입니다. 생략하면 기본값이 -1이 되며 해당하는 문자열을 모두 바꿉니다. 반드시 <i>start</i> 과 함께 사용해야 합니다.
<i>compare</i>	선택적인 인수. 사용할 하위 문자열 비교 형식을 나타내는 숫자 값입니다. 그 값에 대해서는 아래의 설정을 참조합니다. 생략하면 기본값은 0이 되며 이진 비교를 수행합니다.

설정

compare 인수의 값은 아래와 같습니다.

상수	값	설명
vbBinaryCompare	0	이진 비교를 실행합니다.
vbTextCompare	1	텍스트 비교를 실행합니다.

결과 값

Replace 함수는 아래 값을 반환합니다.

경우	Replace 함수의 결과
----	----------------

길이가 0인 <i>expression</i> 인 경우	길이가 0인 문자열("")
<i>expression</i> 이 Null인 경우	오류
길이가 0인 <i>find</i> 인 경우	<i>expression</i> 의 사본
길이가 0인 <i>replacewith</i> 인 경우	<i>find</i> 에 대한 모든 검색 생각과 함께 <i>expression</i> 의 사본
<i>start</i> > Len(<i>expression</i>)	길이가 0인 문자열
<i>number</i> 가 0인 경우	<i>expression</i> 사본

참고

Replace 함수는 start로 지정한 위치에서 시작하여 *expression*의 문자열 끝에서 바꾼 다음 문자열을 반환합니다. 즉, 결과로 반환되는 문자열은 원래의 전체 문자열의 사본이 아닙니다.

아래 예제에서 Replace 함수는 문자열을 반환합니다.

```
Dim MyString
```

'문자열의 처음부터 시작하는 이진 비교. "XXYXXPXXY"를 반환합니다.

```
MyString = Replace("XpXpXpXp", "p", "Y")
```

'3 위치에서 시작하는 텍스트 비교. "YXXYXXY"를 반환합니다.

```
MyString = Replace("XpXpXpXp", "p", "Y", 3, -1, 1)
```

Rnd 함수

설명

난수를 반환합니다.

구문

```
Rnd[(number)]
```

number 인수는 유효한 수식이면 됩니다.

참고

Rnd 함수는 0보다 크거나 같고 1보다 작은 값을 반환합니다. number의 값은 Rnd 함수가 난수를 발생시키는 방식을 결정합니다.

number의 값	Rnd 함수의 결과
0보다 작을 경우	<i>number</i> 를 <u>시드</u> 로 하여 매번 같은 수
0보다 클 경우	순차적으로 생성되는 그 다음의 난수
0과 같을 경우	가장 최근에 생성한 난수
생략한 경우	순차적으로 생성되는 그 다음의 난수

주어진 임의의 초기 시드의 경우 Rnd 함수에 대한 매번의 연속적인 호출에서 순차적인 다음 숫자를 위해 이전 숫자를 시드로 사용하기 때문에 같은 순차에 따른 숫자가 발생하게 됩니다.

Rnd를 호출하기 전에 Randomize문을 사용하여 시스템 타이머에 기반한 시드로 난수 발생기를 초기화합니다.

주어진 범위의 무작위 정수를 만들려면 아래 공식을 사용합니다.

```
Int((upperbound - lowerbound + 1) * Rnd + lowerbound)
```

upperbound는 범위에서 가장 큰 수이고 lowerbound는 범위에서 가장 작은 수입니다.

메모

난수 발생 순차를 반복하려면 숫자 인수로 Randomize를 사용하기 바로 전에 Rnd 함수를 음수 인수로 호출합니다. 같은 값의 number로 Randomize를 사용하면 이전의 순차를 반복하지 않습니다.

RGB 함수

설명

RGB 색상 값을 나타내는 정수를 반환합니다.

구문

RGB(red, green, blue)

RGB 함수는 아래 인수를 가집니다.

인수	설명
<i>red</i>	필수적인 인수. 빨간색 성분을 나타내는 0-255 범위의 숫자입니다.
<i>green</i>	필수적인 인수. 녹색 성분을 나타내는 0-255 범위의 숫자입니다.
<i>blue</i>	필수적인 인수. 파란색 성분을 나타내는 0-255 범위의 숫자입니다.

참고

색상 지정을 받아들이는 응용 프로그램의 메서드와 속성에는 RGB 색상 값을 나타내는 숫자가 필요합니다. RGB 색상 값은 빨강, 녹색 및 파랑의 상대적 농도를 지정하여 특정 색을 나타내도록 합니다.

하위 바이트에는 빨강 값, 중간 바이트에는 녹색 값, 상위 바이트에는 파랑 값이 포함됩니다.

아래 함수는 거꾸로 된 바이트 순서이지만 그 순서가 필요한 응용 프로그램에 동일한 정보를 제공합니다.

Function RevRGB(red, green, blue)

RevRGB= CLng(blue + (green * 256) + (red * 65536))

End Function

255를 초과하는 RGB 함수의 인수 값은 255로 간주됩니다.

Round 함수

설명

지정한 소수점 자리에서 반올림한 수를 반환합니다.

구문

Round(expression[, numdecimalplaces])

Round 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>expression</i>	필수적인 인수. 반올림할 수식 입니다.
<i>numdecimalplaces</i>	선택적인 인수. 소수점 이하의 자릿수를 나타내는 수입니다. 생략하면 Round 함수는 정수를 반환합니다.

참고

아래 예제에서 Round 함수는 하나의 수를 두 자리의 십진수로 반올림합니다.

Dim MyVar, pi

pi = 3.14159

MyVar = Round(pi, 2) 'MyVar는 3.14를 가집니다.

RTrim 함수

설명

LTrim 함수는 선행 공백이 없는, RTrim 함수는 후미 공백이 없는, Trim 함수는 선행 공백과 후미 공백이 모두 없는 문자열 사본을 반환합니다.

구문

```
LTrim(string)
RTrim(string)
```

```
Trim(string)
```

string 인수는 유효한 문자식이면 됩니다. string에 Null이 있으면 Null을 반환합니다.

참고

아래 예제에서 LTrim 함수는 선행 공백을, RTrim함수는 후미 공백을, Trim 함수는 선행 공백과 후미 공백 모두를 각각 잘라냅니다.

```
Dim MyVar
MyVar = LTrim(" vbscript ") 'MyVar는 "vbscript"를 가집니다.
MyVar = RTrim(" vbscript ") 'MyVar는 " vbscript"를 가집니다.
MyVar = Trim(" vbscript ") 'MyVar는 "vbscript"를 가집니다.
```



ScriptEngine 함수

설명

사용 중인 스크립트 언어를 나타내는 문자열을 반환합니다.

구문

```
ScriptEngine
```

결과 값

ScriptEngine 함수는 아래 문자열을 반환합니다.

문자열	설명
<i>VBScript</i>	Microsoft(R) Visual Basic(R) Scripting Edition이 현재의 스크립트 엔진임을 나타냅니다.
<i>JScript</i>	Microsoft JScript(R)가 현재의 스크립트 엔진임을 나타냅니다.
<i>VBA</i>	Microsoft Visual Basic for Applications가 현재의 스크립트 엔진임을 나타냅니다.

참고

아래 예제에서 ScriptEngine 함수는 사용 중인 스크립트 언어를 설명하는 문자열을 반환합니다.

```
Function GetScriptEngineInfo
Dim s
s = "" ' 필요한 정보로 문자열을 작성합니다.
s = ScriptEngine & " 버전 "
```

```
s = s & ScriptEngineMajorVersion & "."
s = s & ScriptEngineMinorVersion & "."
s = s & ScriptEngineBuildVersion
GetScriptEngineInfo = s '결과를 반환합니다.
End Function
```

ScriptEngineBuildVersion 함수

설명

사용 중인 스크립트 엔진의 구성 버전 번호를 반환합니다.

구문

```
ScriptEngineBuildVersion
```

참고

반환되는 결과 값은 사용 중인 스크립트 언어의 DLL에 들어 있는 버전 정보와 같습니다.
아래 예제에서 ScriptEngineBuildVersion 함수는 스크립트 엔진의 구성 버전 번호를 반환합니다.

```
Function GetScriptEngineInfo
Dim s
s = "" '필요한 정보로 문자열을 작성합니다.
s = ScriptEngine & " 버전 "
s = s & ScriptEngineMajorVersion & "."
s = s & ScriptEngineMinorVersion & "."
s = s & ScriptEngineBuildVersion
GetScriptEngineInfo = s '결과를 반환합니다.
End Function
```

ScriptEngineMajorVersion 함수

설명

사용 중인 스크립트 엔진의 주 버전 번호를 반환합니다.

구문

```
ScriptEngineMajorVersion
```

참고

반환되는 결과 값은 사용 중인 스크립트 언어의 DLL에 들어 있는 버전 정보와 같습니다.
아래 예제에서 ScriptEngineMajorVersion 함수는 스크립트 엔진의 버전 번호를 반환합니다.

```
Function GetScriptEngineInfo
Dim s
s = "" '필요한 정보로 문자열을 작성합니다.
s = ScriptEngine & " 버전 "
s = s & ScriptEngineMajorVersion & "."
s = s & ScriptEngineMinorVersion & "."
```

```
s = s & ScriptEngineBuildVersion
GetScriptEngineInfo = s '결과를 반환합니다.
End Function
```

ScriptEngineMinorVersion 함수

설명

사용 중인 엔진 스크립트의 보조 버전 번호를 반환합니다.

구문

```
ScriptEngineMinorVersion
```

참고

반환되는 결과 값은 사용 중인 스크립트 언어의 DLL에 들어 있는 버전 정보와 같습니다. 아래 예제에서 ScriptEngineMinorVersion 함수는 스크립트 엔진의 보조 버전 번호를 반환합니다.

```
Function GetScriptEngineInfo
Dim s
s = "" '필요한 정보로 문자열을 작성합니다.
s = ScriptEngine & " 버전 "
s = s & ScriptEngineMajorVersion & "."
s = s & ScriptEngineMinorVersion & "."
s = s & ScriptEngineBuildVersion
GetScriptEngineInfo = s '결과를 반환합니다.
End Function
```

Second 함수

설명

분의 초를 나타내는 0에서 59까지의 정수를 반환합니다.

구문

```
Second(time)
time 인수는 시간을 나타내는 식입니다. time 인수에 Null이 있으면 Null을 반환합니다.
```

참고

아래 예제에서 Second 함수는 현재 시간의 초를 반환합니다.

```
Dim MySec
MySec = Second(Now) 'MySec는 현재 몇 초인 지에 대한 정보를 가집니다.
```

Sgn 함수

설명

수의 부호를 나타내는 정수를 반환합니다.

구문

Sgn(number)
number 인수는 유효한 수식이면 됩니다.

결과 값

Sgn 함수는 아래 값을 반환합니다.
number의 값 Sgn 함수의 결과
0보다 큰 경우 1
0과 같은 경우 0
0보다 작은 경우 -1

참고

number 인수의 부호가 Sgn 함수의 결과 값을 결정합니다.
아래 예제에서 Sgn 함수는 수의 부호를 결정합니다.

```
Dim MyVar1, MyVar2, MyVar3, MySign
MyVar1 = 12: MyVar2 = -2.4: MyVar3 = 0
MySign = Sgn(MyVar1) ' 1을 반환합니다.
MySign = Sgn(MyVar2) ' -1을 반환합니다.
MySign = Sgn(MyVar3) ' 0을 반환합니다.
```

Sin 함수

설명

각도의 싸인 값을 반환합니다.

구문

Sin(number)
number 인수는 각을 라디안으로 나타내는 유효한 수식이면 됩니다.

참고

Sin 함수는 직각 삼각형의 동위각을 사용하여 두 변의 길이에 의한 비를 반환합니다. 이 비는 해당 각의 맞은 쪽 변의 길이를 직각 삼각형의 사변의 길이로 나눈 비율입니다. 반환되는 결과의 범위는 -1에서 1까지입니다.
각을 라디안으로 변환하려면 각에 pi/180를 곱하고, 라디안을 각으로 변환하려면 라디안에 180/pi를 곱합니다.

아래 예제에서 Sin 함수는 각의 싸인 값을 반환합니다.

```
Dim MyAngle, MyCosecant
MyAngle = 1.3 ' 각을 라디안으로 정의합니다.
MyCosecant = 1 / Sin(MyAngle) ' 코씨컨트 값을 계산합니다.
```

Space 함수

설명

지정된 수만큼의 공백이 들어있는 문자열을 반환합니다.

구문

Space(number)
number 인수는 문자열에 채워 넣을 공백의 개수입니다.

참고

아래 예제에서 Space 함수는 지정된 수만큼의 공백이 들어 있는 문자열을 반환합니다.

```
Dim MyString
MyString = Space(10) ' 10개의 공백이 있는 문자열을 반환합니다.
MyString = "안녕" & Space(10) & "하십니까?" ' 두 문자열 사이에 10개의 공백을 삽입합니다.
```

Split 함수

설명

지정한 개수의 하위 문자열을 포함하는 0 기준 1차원 배열을 반환합니다.

구문

```
Split(expression[, delimiter[, number[, compare]])
```

Split 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>expression</i>	필수적인 인수. 하위 문자열과 구분 기호를 포함하는 문자식 입니다. <i>expression</i> 이 길이가 0인 문자열이면 Split는 빈 배열, 즉 구성 요소와 데이터가 없는 배열을 반환합니다.
<i>delimiter</i>	선택적인 인수. 하위 문자열의 한계를 식별하는 데 사용하는 문자열 문자입니다. 생략하면 공백 문자(" ")를 구분 기호로 간주합니다. <i>Delimiter</i> 의 길이가 0인 문자열이면 전체 <i>expression</i> 문자열을 포함하는 단일 요소 배열을 반환합니다.
<i>number</i>	선택적인 인수. 반환할 하위 문자열의 수입니다. -1은 모든 하위 문자열을 반환한다는 것을 나타냅니다.
<i>compare</i>	선택적인 인수. 사용할 하위 문자열 비교 형식을 나타내는 숫자 값입니다. 그 값에 대해서는 아래의 설정을 참조합니다.

설정

compare 인수의 값은 아래와 같습니다.

상수	값	설명
vbBinaryCompare	0	이진 비교를 실행합니다.
vbTextCompare	1	텍스트 비교를 실행합니다.

참고

아래 예제에서 Split 함수는 문자열의 배열을 반환합니다. 이 함수는 구분 문자의 텍스트 비교를 실행하고 하위 문자열을 모두 반환합니다.

```
Dim MyString, MyArray
MyString = Split("VBScriptXisXfun!", "x", -1, 1)
' MyString(0)은 "VBScript"를 가집니다.
' MyString(1)은 "is"를 가집니다.
' MyString(2)은 "fun!"을 가집니다.
```

Sqr 함수

설명

숫자의 제곱근을 반환합니다.

구문

Sqr(number)
number 인수는 0보다 크거나 같은 유효한 수식입니다.

참고
아래 예제에서 Sqr 함수는 숫자의 제곱근을 계산합니다.

```
Dim MySqr
MySqr = Sqr(4)      ' 2를 반환합니다.
MySqr = Sqr(23)    ' 4.79583152331272를 반환합니다.
MySqr = Sqr(0)     ' 0을 반환합니다.
MySqr = Sqr(-4)    ' 런타임 오류를 일으킵니다.
```

StrComp 함수

설명
문자열 비교의 결과를 나타내는 값을 반환합니다.

구문
StrComp(string1, string2[, compare])
StrComp 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>string1</i>	필수적인 인수. 유효한 문자식.
<i>string2</i>	필수적인 인수. 유효한 문자식.
<i>compare</i>	선택적인 인수. 문자열을 평가하는 데 사용할 비교 형식을 나타내는 숫자 값입니다. 생략하면 이진 비교를 실행합니다. 그 값에 대해서는 아래의 설정을 참조합니다.

설정
compare 인수의 값은 아래와 같습니다.

상수	값	설명
vbBinaryCompare	0	이진 비교를 실행합니다.
vbTextCompare	1	텍스트 비교를 실행합니다.

결과 값
StrComp 함수는 아래 값을 반환합니다.

경우	StrComp 함수의 결과
<i>string1</i> 이 <i>string2</i> 보다 작은 경우	-1
<i>string1</i> 이 <i>string2</i> 와 같은 경우	0
<i>string1</i> 이 <i>string2</i> 보다 큰 경우	1
<i>string1</i> 이나 <i>string2</i> 가 Null인 경우	Null

참고
아래 예제에서 StrComp 함수는 문자열 비교의 결과를 반환합니다. 세번째 인수가 1이라면 텍스트 비교가 실행됩니다. 세번째 인수가 0이거나 생략된다면 이진 비교가 실행됩니다.

```
Dim MyStr1, MyStr2, MyComp
MyStr1 = "ABCD": MyStr2 = "abcd"      ' 변수를 정의합니다.
MyComp = StrComp(MyStr1, MyStr2, 1)  ' 0을 반환합니다.
MyComp = StrComp(MyStr1, MyStr2, 0)  ' -1을 반환합니다.
```

```
MyComp = StrComp(MyStr2, MyStr1) ' 1을 반환합니다.
```

String 함수

설명

String 하위 형식의 Variant로 변환된 식을 반환합니다.

구문

```
CStr(expression)
expression 인수는 유효한 식이면 됩니다.
```

참고

대개 데이터 형식 변환 함수를 사용하여 코드를 작성하면 연산 결과가 기본 데이터 형식이 아닌 특정 데이터 형식으로 나타나는 것을 볼 수 있습니다. 예를 들어 CStr를 사용하면 이 결과를 String으로 표시하게 됩니다.

Str 함수 대신 CStr 함수를 사용하면 다른 형식의 데이터를 국제적으로 통용되는 String 하위 형식으로 변환합니다. 예를 들어 서로 다른 자릿수 구분 기호는 컴퓨터의 로케일 설정에 따라 적절하게 인식됩니다.

expression에서 사용하는 데이터는 아래 표처럼 반환되는 결과를 결정합니다.

<i>expression의 데이터</i>	CStr 반환 결과
Boolean	True 또는 False를 포함하는 문자열
Date	컴퓨터에서 설정된 간단하게 표시 날짜 형식의 날짜를 포함하는 문자열
Null	런타임 오류
Empty	길이가 0인 문자열("")
Error	Error 단어 다음에 오류 번호가 나타나는 문자열
기타 숫자	번호를 포함하는 문자열

아래 예제에서 CStr 함수는 숫자 값을 String으로 변환합니다.

```
Dim MyDouble, MyString
MyDouble = 437.324 ' MyDouble은 Double입니다.
MyString = CStr(MyDouble) ' MyString은 값 "437.324"를 가집니다.
```

StrReverse 함수

설명

지정한 문자열의 순서가 거꾸로 된 문자열을 반환합니다.

구문

```
StrReverse(string1)
string1 인수는 그 문자들이 역순인 문자열입니다. string1이 길이가 0인 문자열("")일 경우 길이가 0인 문자열을 반환하며, Null일 경우 오류가 발생합니다.
```

참고

아래 예제에서 StrReverse 함수는 역순의 문자열을 반환합니다.

```
Dim MyStr
MyStr = StrReverse("VBScript") 'MyStr은 "tpircSBV"를 가집니다.
```



Tan 함수

설명

각도의 탄젠트 값을 반환합니다.

구문

Tan(number)

number 인수는 각을 라디안으로 나타내는 유효한 수식입니다.

참고

Tan 함수는 인수로 각을 사용하여 직각 삼각형에 있는 두 변의 길이에 의한 비를 반환합니다. 이 비는 해당 각의 맞은 쪽 변의 길이를 이 각에 인접한 변의 길이로 나눈 비율입니다.

각을 라디안으로 변환하려면 각에 $\pi/180$ 를 곱하고, 라디안을 각으로 변환하려면 라디안에 $180/\pi$ 를 곱합니다.

아래 예제에서 Tan 함수는 각의 탄젠트 값을 반환합니다.

```
Dim MyAngle, MyCotangent
MyAngle = 1.3           ' 각을 라디안으로 정의합니다.
MyCotangent = 1 / Tan(MyAngle) ' 코탄젠트 값을 계산합니다.
```

Time 함수

설명

현재 시스템 시간을 나타내는 Date 하위 형식의 Variant를 반환합니다.

구문

Time

참고

아래 예제에서 Time 함수는 현재 시스템 시간을 반환합니다.

```
Dim MyTime
MyTime = Time ' 현재 시스템 시간을 반환합니다.
```

TimeSerial 함수

설명

지정한 년, 월, 일의 Date 하위 형식인 Variant를 반환합니다.

구문

TimeSerial(hour, minute, second)

TimeSerial 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>time</i>	0(오전 12:00)에서 23(오후 11:00) 사이의 수(0과 23 포함) 또는 수식
<i>minute</i>	모든 수식
<i>second</i>	모든 수식

참고

11시 59분 59초처럼 어떤 시간을 지정하려면 각 TimeSerial 인수의 숫자는 해당 인수의 허용 범위 안에 있어야 합니다. 즉, 시간은 0-23이며, 분과 초는 0-59입니다. 그러나 특정 시간의 앞뒤에 시, 분, 초를 나타내는 수식을 사용하여 각 인수의 상대적인 시간을 지정할 수도 있습니다.

아래 예제는 절대 시간 수 대신 식을 사용하고 있습니다. 여기서 TimeSerial 함수는 정오의 6시간 15분 전, 즉 오전 5시 45분을 반환합니다

```
TimeSerial(12 - 6, -15, 0)
```

인수가 해당되는 허용 범위를 넘으면 그 다음으로 큰 단위로 넘어갑니다. 예를 들어 75분을 지정하면 1시간 15분으로 계산합니다. 그러나 어느 한 인수가 -32,768~32,767 범위를 넘는 인수이거나 세 인수를 사용하여 직접 또는 식으로 지정한 날짜가 허용 범위를 넘을 경우 오류가 발생합니다.

TimeValue 함수

설명

시간을 포함하는 Date 하위 형식의 Variant를 반환합니다.

구문

```
TimeValue(time)
```

대개 time 인수는 0:00:00(오전 12:00:00)에서 23:59:59(오후 11:59:59)까지의 시간을 나타내는 문자식으로 이 범위에 있는 시간을 나타내는 식도 time 인수가 될 수 있습니다. time에 Null이 있으면 Null을 반환합니다.

참고

12시간이나 24시간 짜리 시계에서 유효한, 시간 단위를 입력할 수 있습니다. 즉, "2:24PM"와 "14:24" 모두 유효한 time 인수입니다. time 인수에 날짜 정보가 있는 경우 TimeValue 함수에서는 그 정보를 반환하지 않습니다. 그러나 time 인수에 잘못된 날짜 정보가 있으면 오류가 발생합니다.

아래 예제에서 TimeValue 함수는 문자열을 시간으로 변환합니다. 날짜 리터럴을 사용하면 시간을 직접 Variant에 지정할 수 있습니다(예를 들면, MyTime = #4:35:17 PM#).

```
Dim MyTime
```

```
MyTime = TimeValue("4:35:17 PM") ' MyTime은 "4:35:17 PM"을 가집니다.
```

Trim 함수

설명

LTrim 함수는 선행 공백이 없는, RTrim 함수는 후미 공백이 없는, Trim 함수는 선행 공백과 후미 공백이 모두 없는 문자열 사본을 반환합니다.

구문

```
LTrim(string)
```

```
RTrim(string)
```

```
Trim(string)
```

string 인수는 유효한 문자식이면 됩니다. string에 Null이 있으면 Null을 반환합니다.

참고

아래 예제에서 LTrim 함수는 선행 공백을, RTrim함수는 후미 공백을, Trim 함수는 선행 공백과 후미 공백 모두를 각각 잘라냅니다.

```
Dim MyVar
MyVar = LTrim(" vbscript ") 'MyVar는 "vbscript"를 가집니다.
MyVar = RTrim(" vbscript ") 'MyVar는 " vbscript"를 가집니다.
MyVar = Trim(" vbscript ") 'MyVar는 "vbscript"를 가집니다.
```

TypeName 함수

설명

변수에 관한 Variant 하위 형식 정보를 제공하는 문자열을 반환합니다.

구문

```
TypeName(varname)
필수적인 인수. varname 인수는 아무 변수이면 됩니다.
```

결과 값

TypeName 함수는 아래 값들을 반환합니다.

값	설명
Byte	바이트 값
Integer	정수 값
Long	Long 정수 값
Single	단정밀도 부동 소수점 숫자 값
Double	배정밀도 부동 소수점 숫자 값
Currency	통화 값
Decimal	십진 값
Date	날짜 또는 시간 값
String	문자열 값
Boolean	부울 값 True 또는 False
Empty	초기화되지 않음
Null	유효한 데이터 없음
<object type>	개체의 실제 형식 이름
Object	일반 개체
Unknown	알 수 없는 개체 형식
Nothing	개체 인스턴스를 아직 참조하지 않은 개체 변수
Error	오류

참고

아래 예제에서 TypeName 함수는 변수 관련 정보를 반환합니다.

```
Dim ArrayVar(4), MyType
NullVar = Null ' Null 값을 지정합니다.

MyType = TypeName("VBScript") ' "String"을 반환합니다.
MyType = TypeName(4) ' "Integer"를 반환합니다.
MyType = TypeName(37.50) ' "Double"을 반환합니다.
MyType = TypeName(NullVar) ' "Null"을 반환합니다.
MyType = TypeName(ArrayVar) ' "Variant()"를 반환합니다.
```



UBound 함수

설명

지정한 배열 차원에 사용할 수 있는 가장 큰 첨자를 반환합니다.

구문

UBound(arrayname[, dimension])

UBound 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>arrayname</i>	필수적인 인수. 배열 변수의 이름으로 일반 변수 이름 짓기 규칙을 따릅니다.
<i>dimension</i>	선택적인 인수. 반환될 차원의 상한값을 나타내는 정수입니다. 1차원은 1, 2차원은 2, 등등으로 정수를 사용합니다. <i>dimension</i> 을 생략하면 1로 간주합니다.

참고

UBound 함수는 LBound 함수와 함께 사용하여 배열의 크기를 결정합니다. LBound 함수를 사용하면 배열 차원의 하한 값을 알 수 있습니다.

모든 차원의 하한값은 항상 0입니다. 따라서 UBound 함수는 각 차원에 따른 배열의 값을 아래와 같이 반환합니다.

Dim A(100,3,4)

문	결과 값
UBound(A, 1)	100
UBound(A, 2)	3
UBound(A, 3)	4

UCase 함수

설명

대문자로 변환된 문자열을 반환합니다.

구문

UCase(string)

string 인수는 유효한 문자식이면 됩니다. string에 Null이 있으면 Null을 반환합니다.

참고

소문자만 대문자로 변환하고 대문자와 글자가 아닌 문자는 변환하지 않고 그대로 둡니다.

아래 예제에서 UCase 함수는 문자열의 대문자 버전을 반환합니다.

Dim MyWord

MyWord = UCase("Hello World") ' "HELLO WORLD"를 반환합니다.



VarType 함수

설명

변수의 하위 형식을 나타내는 값을 반환합니다.

구문

VarType(varname)

varname 인수는 아무 변수이면 됩니다.

결과 값

VarType 함수는 아래 값들을 반환합니다.

상수	값	설명
vbEmpty	0	Empty (초기화되지 않음)
vbNull	1	Null (유효한 데이터 없음)
vbInteger	2	정수
vbLong	3	Long 정수
vbSingle	4	단정밀도 부동 소수점 숫자
vbDouble	5	배정밀도 부동 소수점 숫자
vbCurrency	6	통화
vbDate	7	날짜
vbString	8	문자열
vbObject	9	자동화 개체
vbError	10	오류
vbBoolean	11	부울 값
vbVariant	12	Variant(변이형 배열에만 사용)
vbDataObject	13	데이터 액세스 개체
vbByte	17	바이트
vbArray	8192	배열

메모

VBScript에서 지정한 상수들이므로 그 이름들은 코드의 어느 곳에서나 실제 값 대신 사용할 수 있습니다.

참고

VarType 함수 자체는 해당 배열 값을 결코 반환하지 않으며 항상 다른 값에 합해져서 특정 형식의 배열을 나타냅니다. 배열의 특정 값에 합해져서 VarType 함수의 인수가 배열임을 나타낼 때에만 Variant에 대한 값을 반환합니다. 예를 들어 배열에 반환되는 정수 값은 2+8192 또는 8194로 계산됩니다. 기본 속성을 가진 개체일 경우 VarType(object) 함수는 해당 개체에 대한 기본 속성의 형식을 반환합니다.

아래 예제에서 VarType 함수는 변수의 하위 형식을 결정합니다.

```
Dim MyCheck
MyCheck = VarType(300)           ' 2를 반환합니다.
MyCheck = VarType(#10/19/62#)   ' 7을 반환합니다.
MyCheck = VarType("VBScript")   ' 8을 반환합니다.
```



Weekday 함수

설명

요일을 나타내는 정수를 반환합니다.

구문

Weekday(date, [firstdayofweek])

Weekday 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>date</i>	날짜를 나타내는 식입니다. <i>date</i> 에 <u>Null</u> 이 있으면 Null을 반환합니다.
<i>firstdayofweek</i>	첫째 요일을 지정하는 상수입니다. 생략하면 vbSunday로 간주합니다.

설정

firstdayofweek 인수는 아래와 같이 설정됩니다.

상수	값	설명
vbUseSystem	0	NLS(National Language Support) API 설정 사용
vbSunday	1	일요일
vbMonday	2	월요일
vbTuesday	3	화요일
vbWednesday	4	수요일
vbThursday	5	목요일
vbFriday	6	금요일
vbSaturday	7	토요일

결과 값

대개 Weekday 함수는 아래 값을 반환합니다.

상수	값	설명
vbSunday	1	일요일
vbMonday	2	월요일
vbTuesday	3	화요일
vbWednesday	4	수요일
vbThursday	5	목요일
vbFriday	6	금요일
vbSaturday	7	토요일

참고

아래 예제에서 Weekday 함수는 지정된 날짜로부터 요일을 가져옵니다.

```
Dim MyDate, MyWeekDay
MyDate = #October 19, 1962# ' 날짜를 지정합니다.
MyWeekDay = Weekday(MyDate) ' MyDate가 Friday를 나타내므로 MyWeekDay는 값 6을 가집니다.
```

WeekdayName 함수

설명

지정된 요일을 나타내는 문자열을 반환합니다.

구문

WeekDayName(weekday, abbreviate, firstdayofweek)

WeekDayName 함수 구문은 아래 인수를 가집니다.

인수	설명
<i>weekday</i>	필수적인 인수. 요일을 나타내는 숫자입니다. <i>firstdayofweek</i> 설정에 따라 각 요일의 값이 다릅니다.
<i>abbreviate</i>	선택적인 인수. 요일 이름을 약어로 나타낼지 여부를 나타내는 부울 값입니다. 생략하면 기본값인 <i>False</i> 가 되며 요일 이름을 완전한 형태로 나타냅니다.
<i>firstdayofweek</i>	선택적인 인수. 첫째 요일을 나타내는 숫자 값입니다. 그 값에 대해서는 아래의 설정을 참조합니다.

설정

firstdayofweek 인수의 값은 아래와 같습니다.

상수	값	설명
<i>vbUseSystem</i>	0	NLS(National Language Support) API 설정 사용
<i>vbSunday</i>	1	일요일(기본값)
<i>vbMonday</i>	2	월요일
<i>vbTuesday</i>	3	화요일
<i>vbWednesday</i>	4	수요일
<i>vbThursday</i>	5	목요일
<i>vbFriday</i>	6	금요일
<i>vbSaturday</i>	7	토요일

참고

아래 예제에서 WeekDayName 함수는 지정한 날을 반환합니다.

Dim MyDate

MyDate = WeekDayName(6, True) 'MyDate는 Fri를 가집니다.



Year 함수

설명

연도를 나타내는 정수를 반환합니다.

구문

Year(date)

date 인수는 날짜를 나타내는 식입니다. date에 Null이 exist한다면 Null을 반환합니다.

참고

아래 예제에서 Year 함수는 지정한 날짜로부터 연도를 가져옵니다.

Dim MyDate, MyYear

MyDate = #October 19, 1962# ' 날짜를 지정합니다.

MyYear = Year(MyDate) ' MyYear는 1962를 가집니다

부록 4	LS산전 INFOU OPC 서버 도움말	4-2
4.1	개요	4-2
4.2	프로그램 설치	4-2
4.2.1	프로그램 설치.....	4-2
4.2.2	프로그램 수정/ 복구/ 제거.....	4-6
4.3	기본 화면 구성	4-11
4.3.1	설정(Configuration).....	4-11
4.3.2	모니터링(Monitoring).....	4-14
4.4	OPC 프로젝트 설정.....	4-17
4.4.1	새 OPC 프로젝트 생성.....	4-17
4.4.2	기존 OPC 프로젝트 불러오기 / 수정하기.....	4-19
4.5	모니터링(MONITORING)	4-20
4.5.1	LS InfoU OPC 서버 연결/ 모니터링(Monitoring) 시작.....	4-20
4.5.2	LS InfoU OPC 서버 끊기 / 모니터링(Monitoring) 중지.....	4-22
4.6	DCOM CONFIGURATION.....	4-22
4.6.1	방화벽 설정.....	4-22
4.6.2	DCOM 설정.....	4-27

부록.4. LS산전 InfoU OPC 서버 도움말

4.1 개요

OPC는 마이크로소프트사의 윈도우 운영체제 하에서 각종 산업용 기기들과의 통신을 표준화하는 목적으로 개발된 방식입니다. 이것은 윈도우 운영체제의 COM/DCOM 기능을 기반으로 동작되며, InfoU과 같은 OPC Client 소프트웨어와 OPC 서버 소프트웨어의 두 가지 소프트웨어로 구성됩니다. 이 절에서는 **OPC (OLE for Process Control)** 방식의 통신으로 데이터를 교환할 수 있는 자사 OPC 서버 사용 방법에 대해 설명합니다. LS InfoU OPC 서버는 런타임 시 데이터를 제공할 LSIS OPC DA 서버 와 OPC DA 서버에서 수집할 태그를 선택 하기 위한 설정(Configuration) 프로그램을 제공하고 있습니다. 설정(Configuration)프로그램의 경우, 자체 InfoU OPC 서버와의 로컬 및 원격 진단을 위한 모니터링(Monitoring) 기능을 포함하고 있습니다.

4.2 프로그램 설치

4.2.1 프로그램 설치

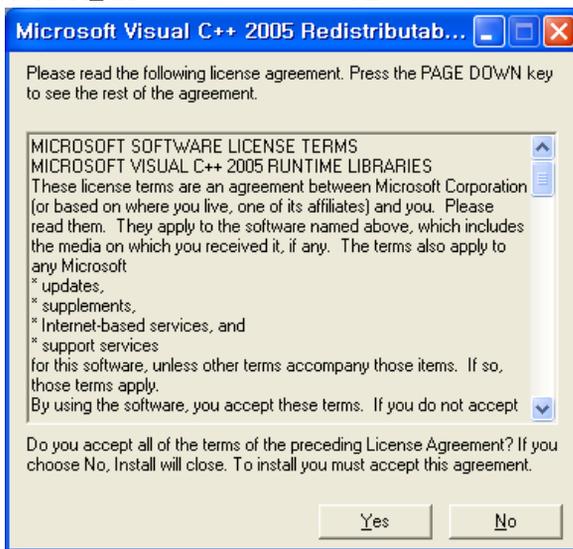
다음의 순서를 따라 프로그램을 설치합니다.

- (1) .NET Framework 2.0 설치

“dotnetfx.exe” 파일을 실행합니다.

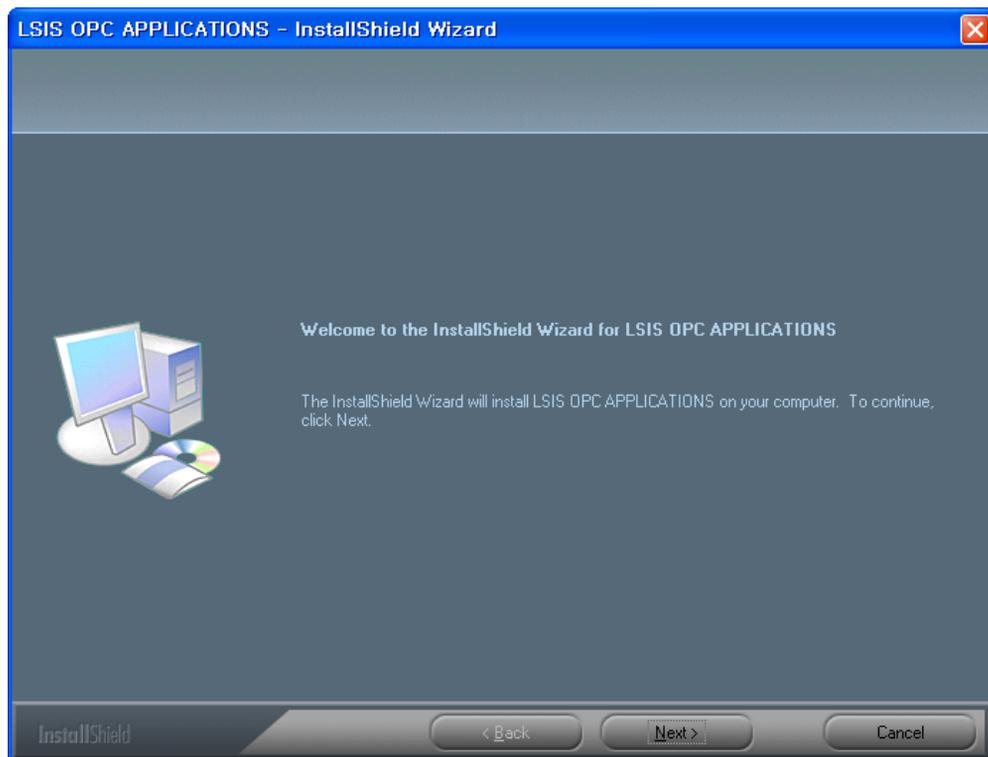
- (2) 마이크로소프트 Visual C++ 2005 Run Time library” 설치

“vcredist_x86.exe” 파일을 실행합니다.

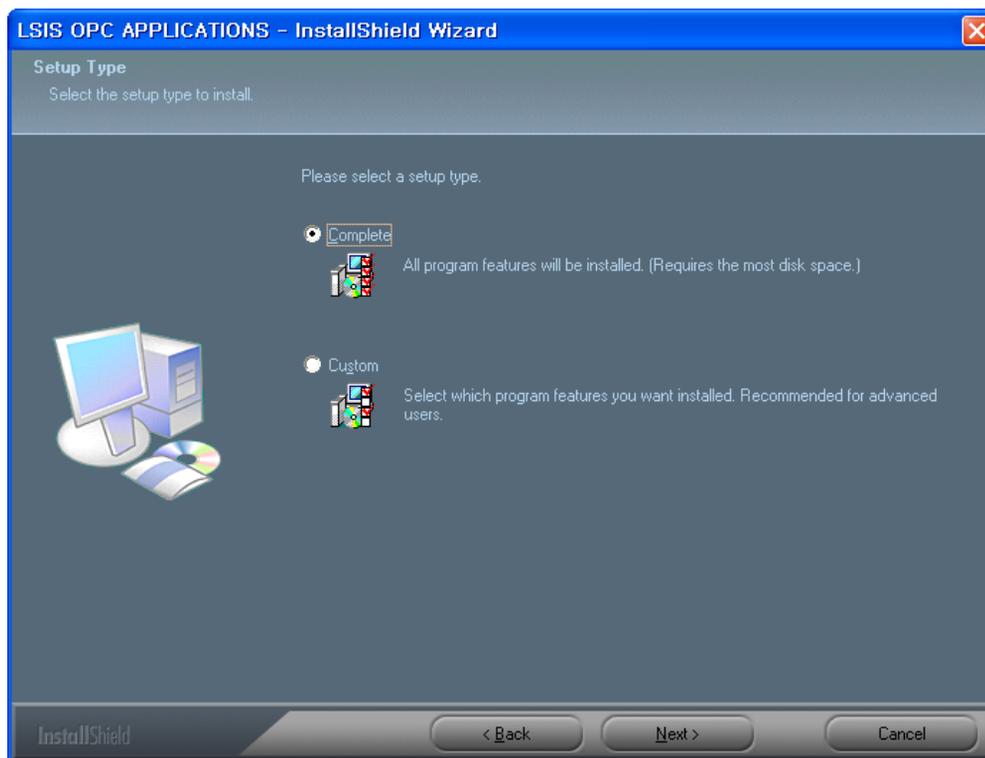


- (3) LS InfoU OPC DA 서버” 설치

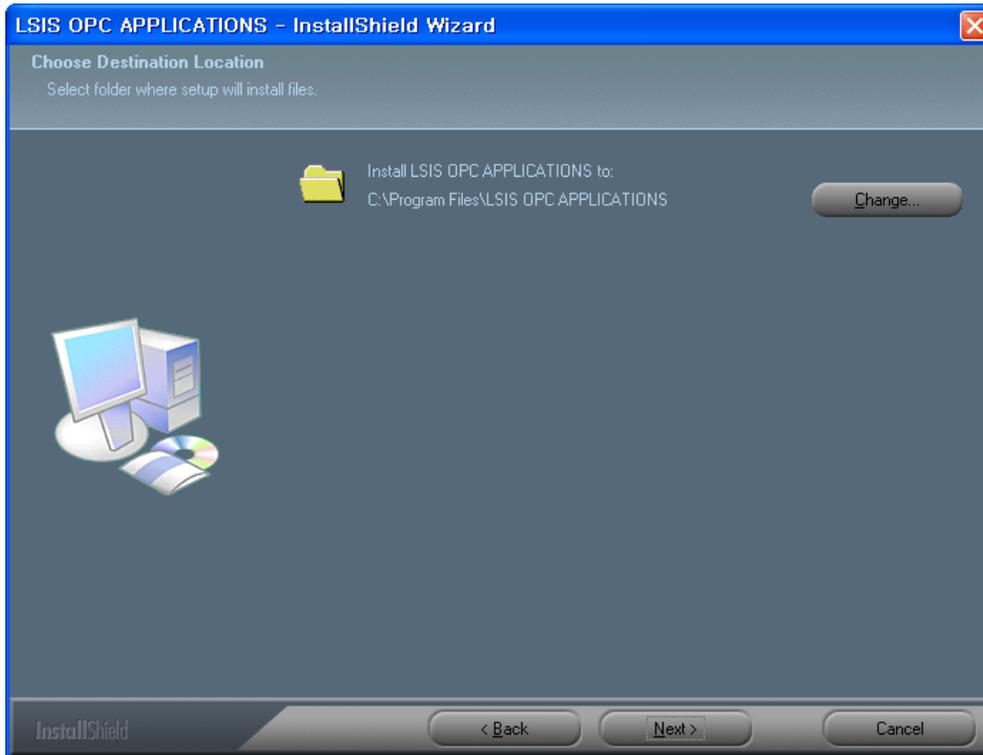
- ① “LSISOPC 서버 For InfoU_V_1_4.exe” 파일을 실행합니다.
- ② 마이크로소프트 Visual C++ 2005 Run Time library” 사용을 위한 사용권 계약에 “ 동의”를 위해 “Yes” 을 선택합니다.



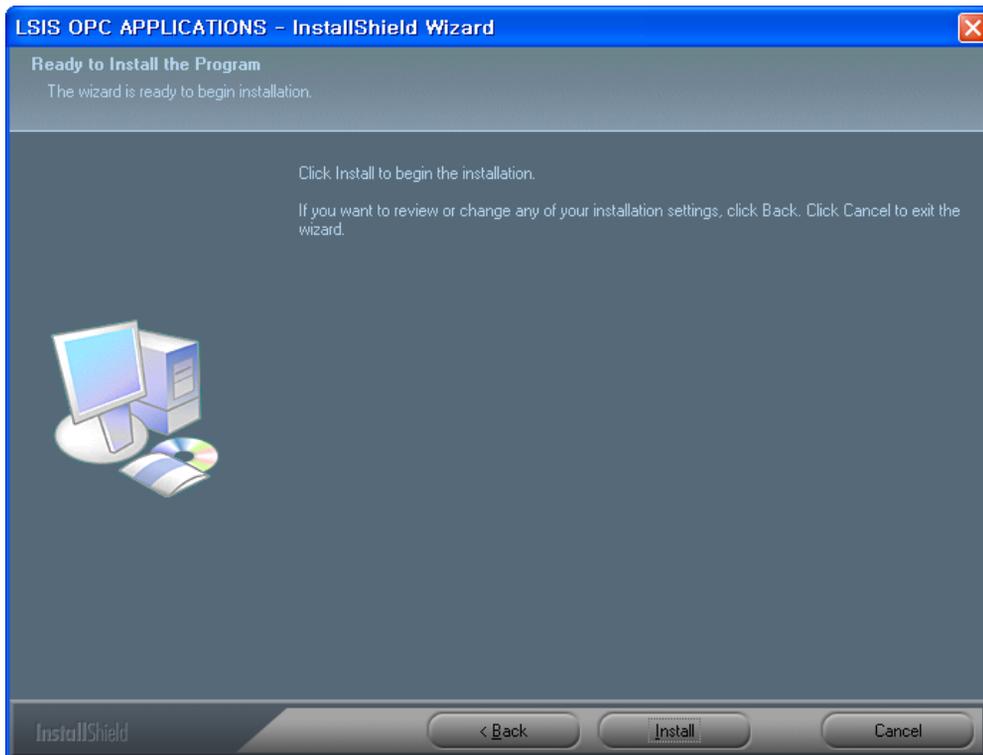
- ③ “설치” 타입을 선택합니다
- Complete: 모든 기능을 설치합니다.
 - Custom: 사용자가 선택한 기능만을 설치합니다.

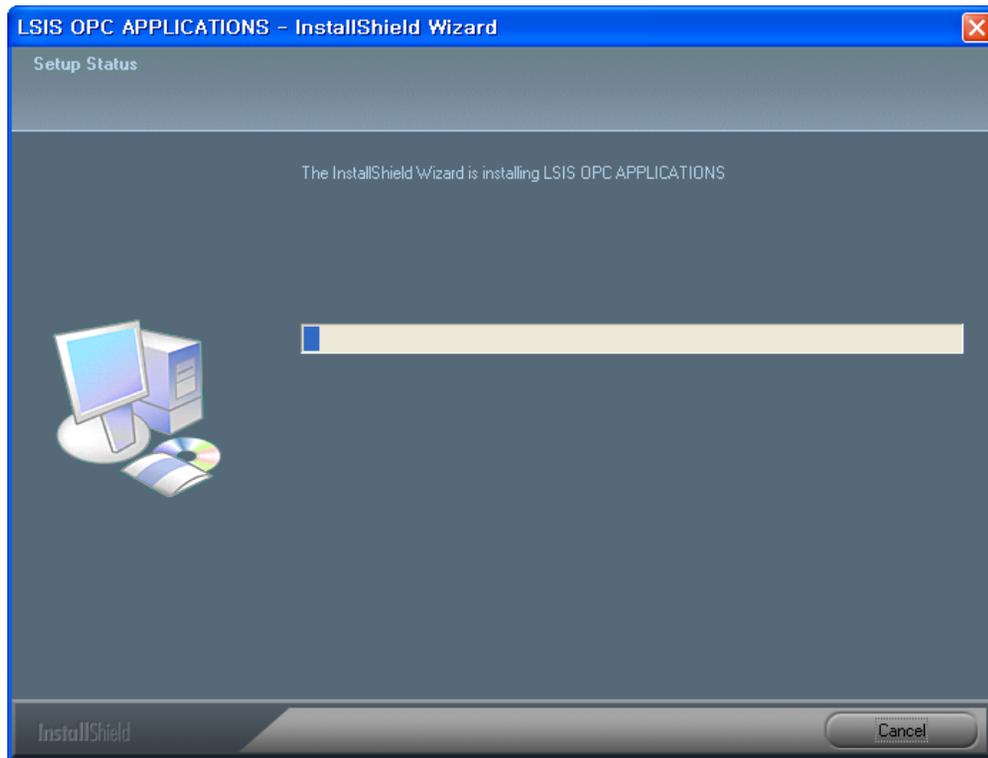


- ④ 프로그램을 설치할 위치를 설정합니다. 디폴트 폴더를 변경할 시에는 “Change” 버튼을 폴더를 새로 선택합니다.

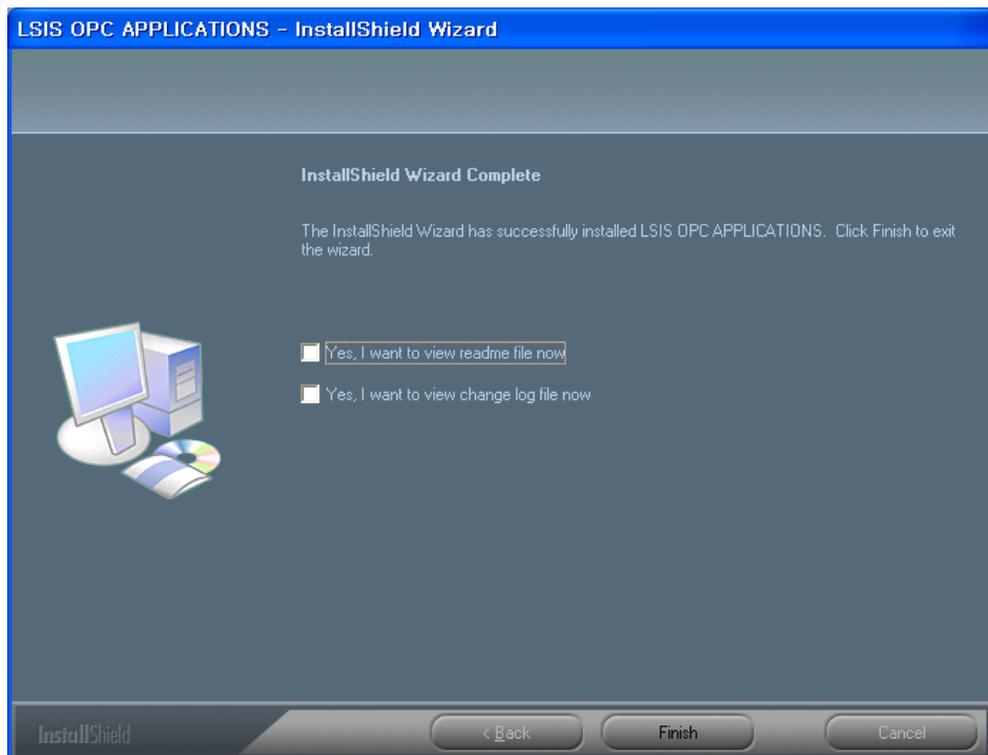


- ⑤ “Install” 버튼을 눌러 설치를 시작합니다





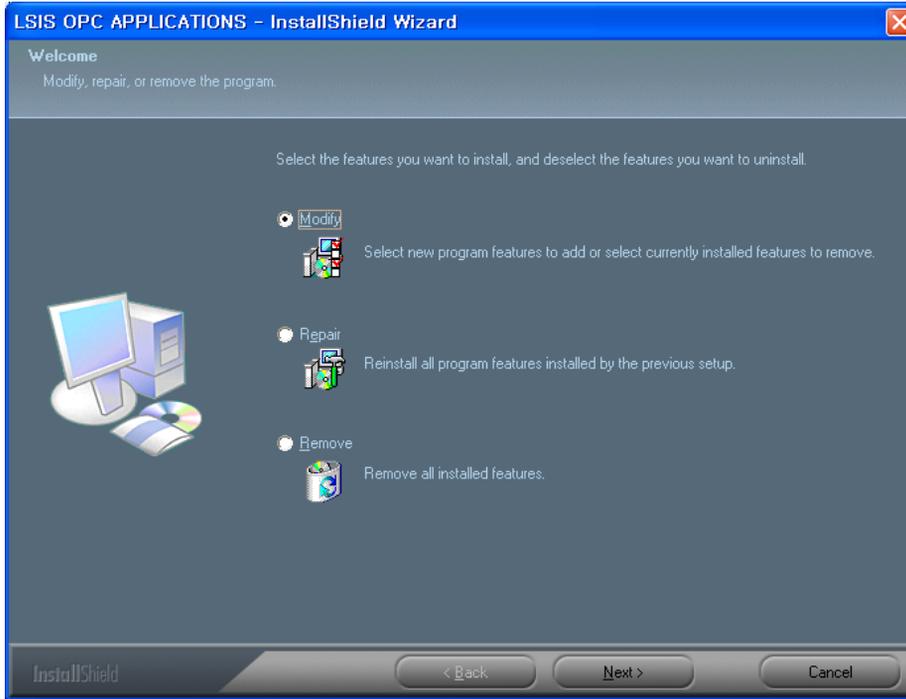
- ⑥ “Finish” 버튼을 누르고 설치를 종료합니다.



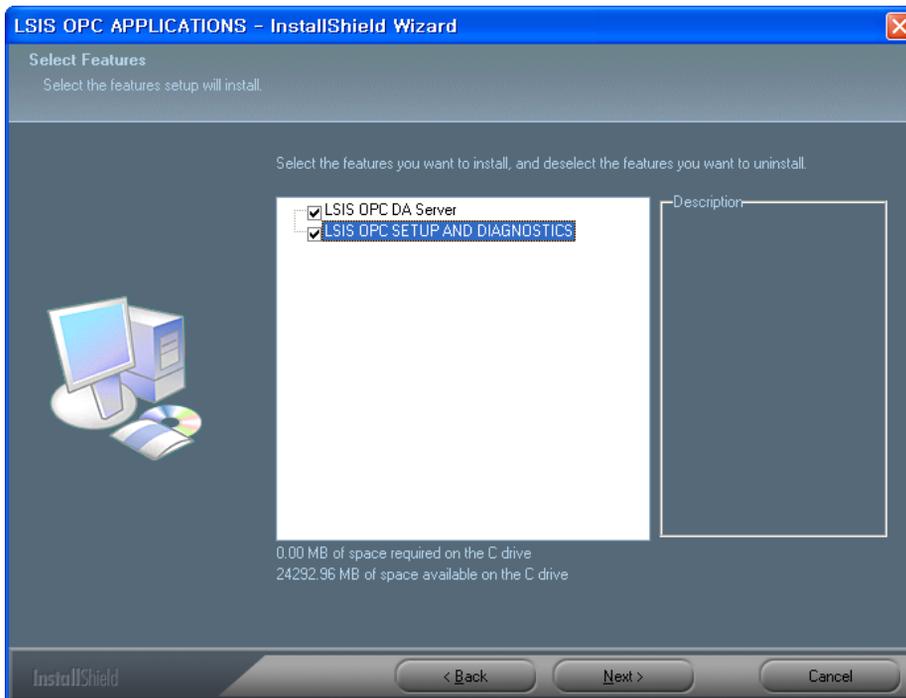
4.2.2 프로그램 수정/ 복구/ 제거

(1) 수정

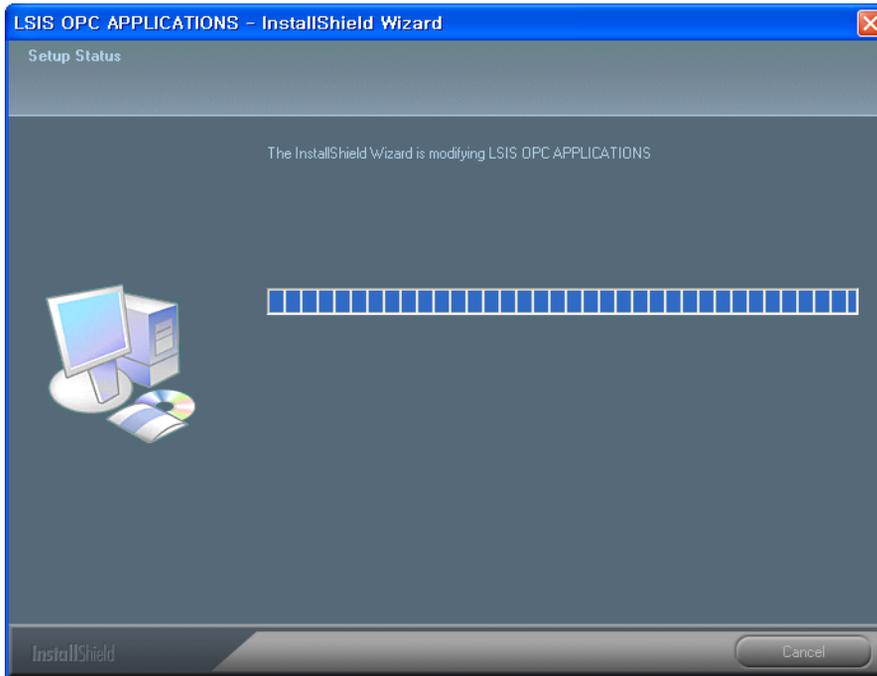
- ① “LSISOPC 서버 For InfoU_V_1_4.exe” 파일을 실행합니다.



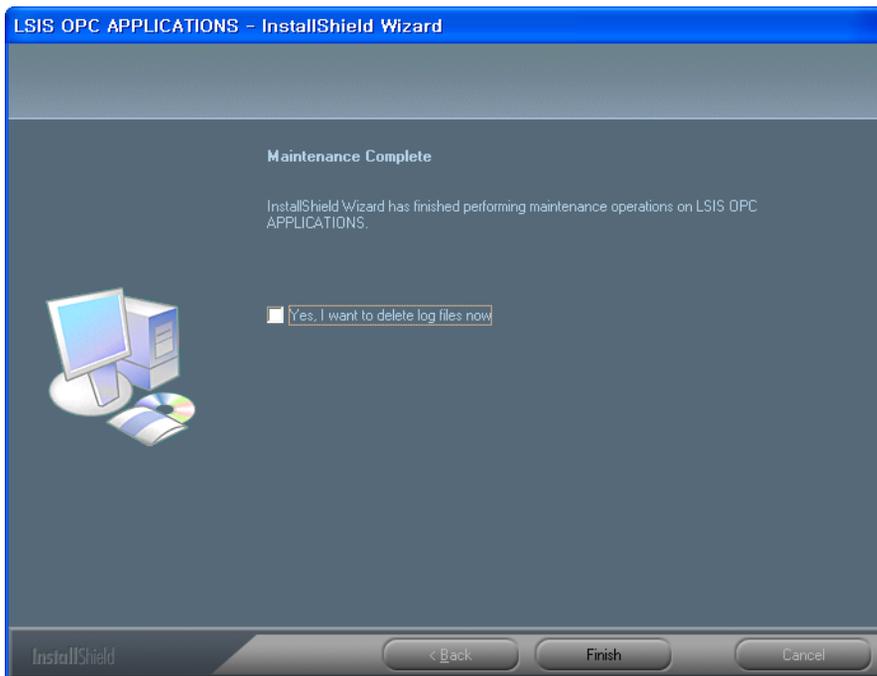
- ② 각각의 프로그램에 설치 및 제거를 결정합니다. 체크박스를 선택 시 프로그램 설치이며, 비 선택 시 제거를 실행합니다.



- ③ “Next” 를 클릭하여 수정을 실행 및 완료 합니다.

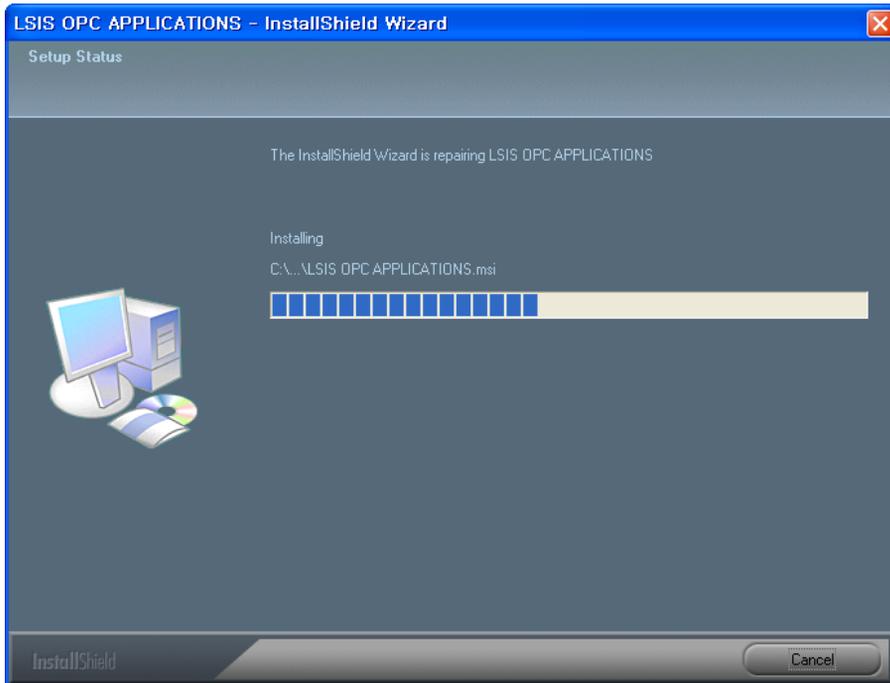
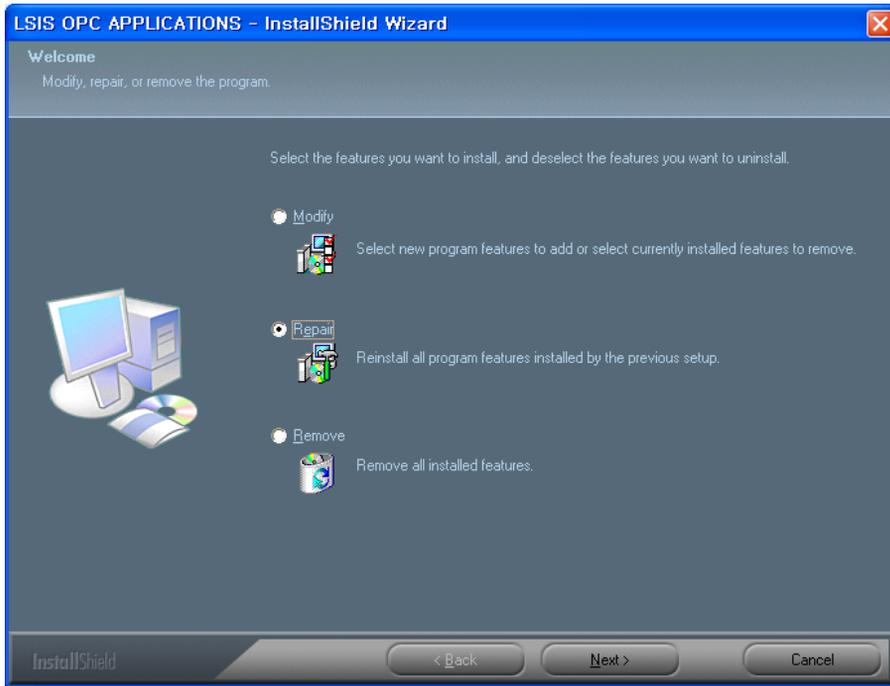


- ④ “Finish” 버튼을 누르고 설치를 종료합니다.

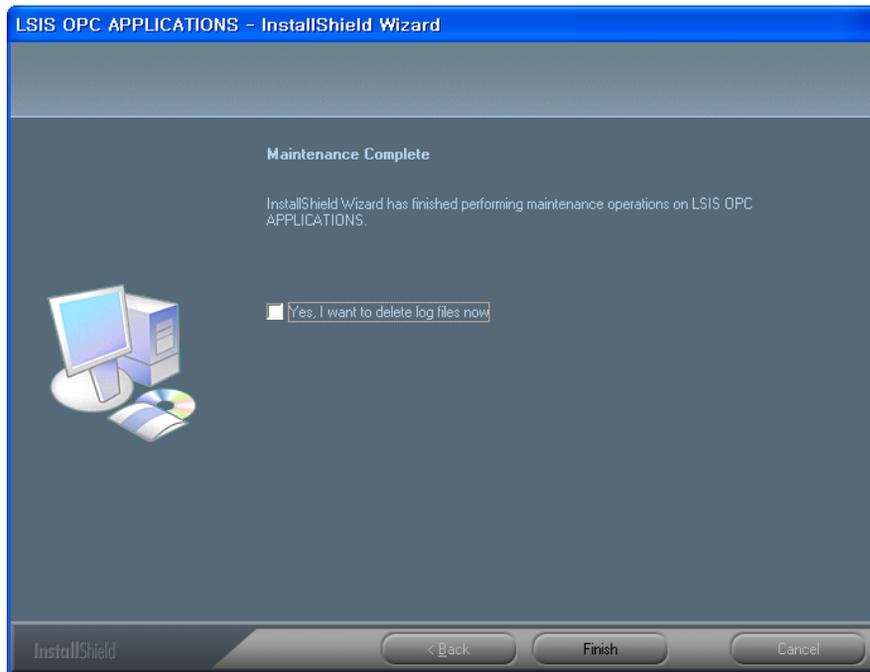


(2) 복구

- ① “LSISOPC 서버 For InfoU_V_1_4.exe” 파일을 실행합니다.
- ② 이전 설치 프로그램으로 설치된 프로그램을 재 설치를 위해 “Repair” 를 선택합니다.
- ③ “Next” 를 클릭하여 복구를 실행 및 완료 합니다.

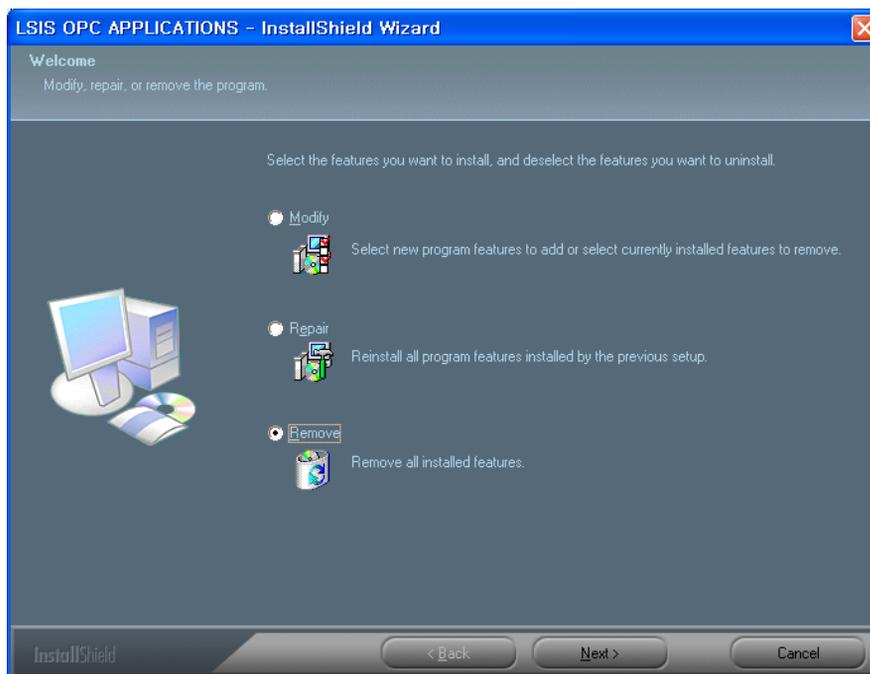


- ④ “Finish” 버튼을 누르고 설치를 종료합니다.

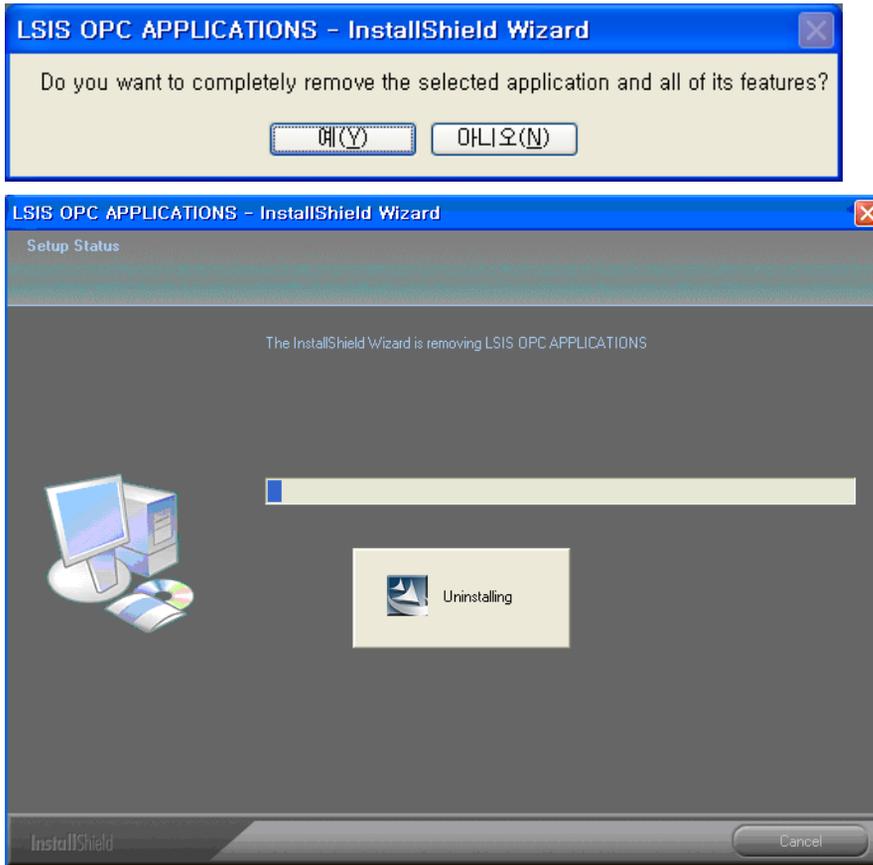


(3) 제거

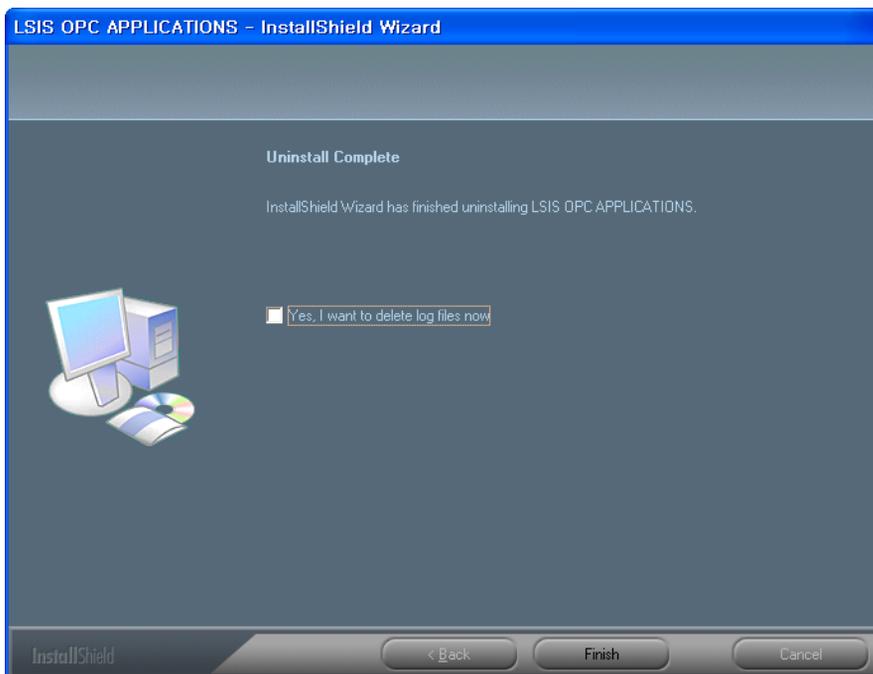
- ① “LSISOPC 서버 For InfoU_V_1_4.exe” 파일을 실행합니다.
 ② 설치된 관련 OPC 모든 프로그램을 제거를 위해 “Remove” 를 선택합니다.



- ③ 제거를 위해 “예” 버튼을 클릭합니다.

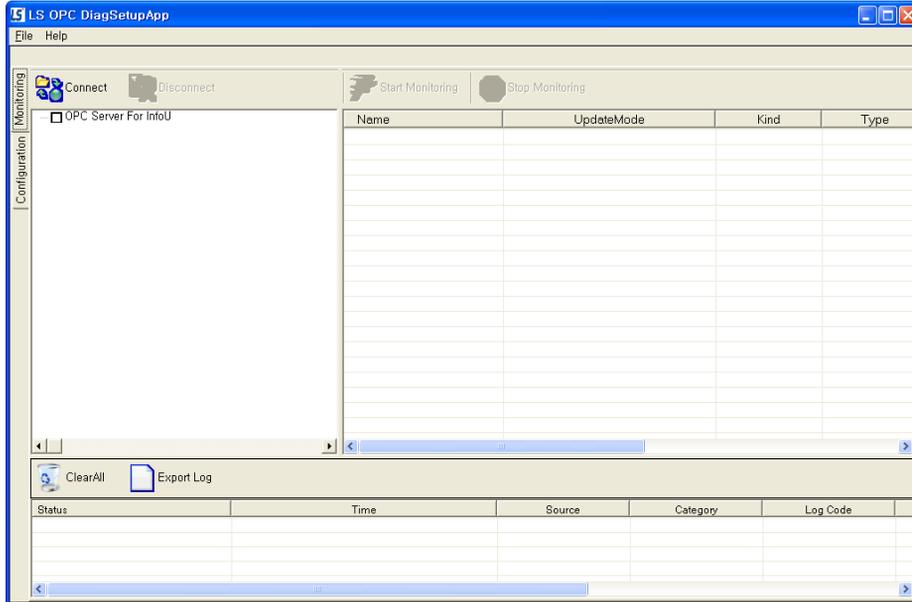


- ④ “Finish” 버튼을 누르고 설치를 종료합니다.



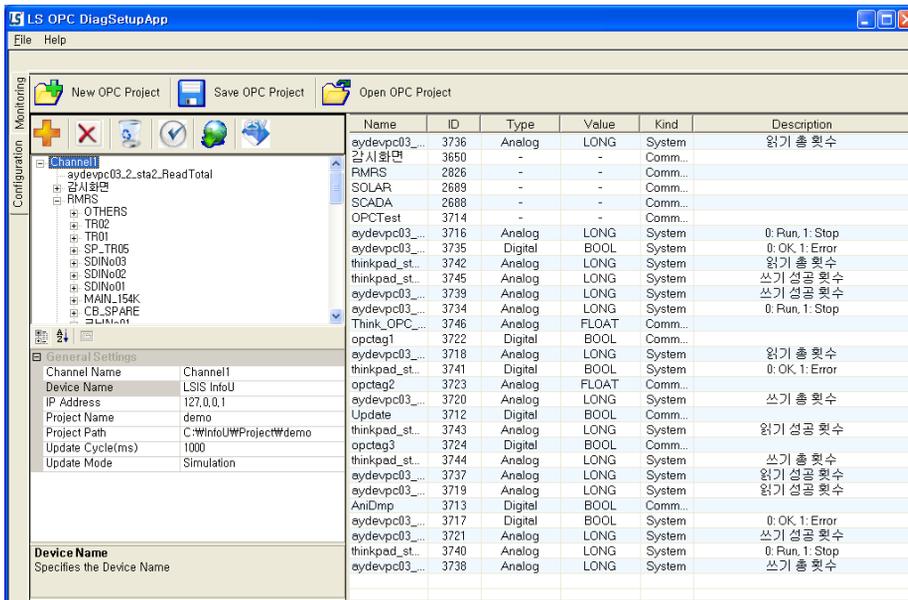
4.3 기본 화면 구성

작업 표시줄에서 “시작 -> 모든 프로그램 -> LSIS OPC 서버 APPLICATION-> LSIS OPC SETUP AND DIAGNOSTICS” 을 선택 합니다.



위의 프로그램은 OPC DA 서버의 설정(Configuration)과 OPC DA 서버 의 모니터링(Monitoring)기능을 가지고 있습니다. 그 중, 실행 시 화면으로는 모니터링 화면을 보여주고 있습니다. 위의 프로그램에서 “Configuration” 탭을 선택하면, OPC DA 서버를 위한 설정화면으로 서버에서 제공할 태그 목록들을 선택할 수 있습니다.

4.3.1 설정(Configuration)



(1) 툴바 (OPC 프로젝트)



- ① New OPC Project: OPC DA 서버가 런타임 시 수집해야 할 새로운 태그의 정보를 담고 OPC 프로젝트를 생성함

니다.

- ② Save OPC Project: 새롭게 생성되거나 변경된 OPC 프로젝트를 저장합니다.
- ③ Open OPC Project: OPC DA 서버 가 런타임 시 수집해야 할 기존 OPC 프로젝트를 로드 합니다.

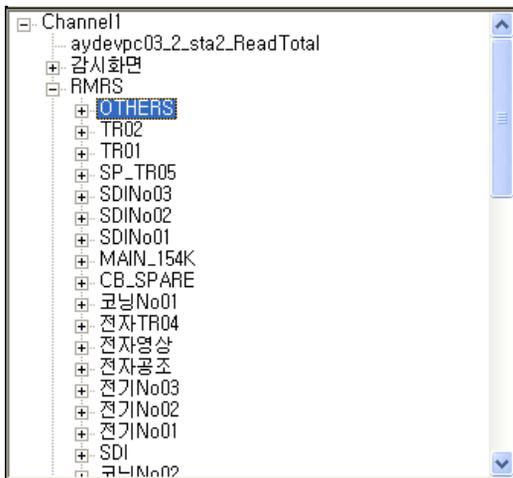
(2) 툴바 (채널)



- ①  : 채널 추가, 즉, 채널 하나 당 각기 다른 컴퓨터에서 실행되는 InfoU 프로젝트 하나를 설정합니다.
- ②  : 선택된 채널을 삭제합니다.
- ③  : OPC 프로젝트에 등록된 모든 채널을 삭제합니다.
- ④  : 선택된 채널의 통신 상태를 체크합니다.
- ⑤  : 선택된 채널의 InfoU 프로젝트로부터 태그 정보를 읽어드립니다.
- ⑥  : InfoUD 에서 생성한 CSV 을 읽어와 태그들을 설정합니다.

(3) 채널 창

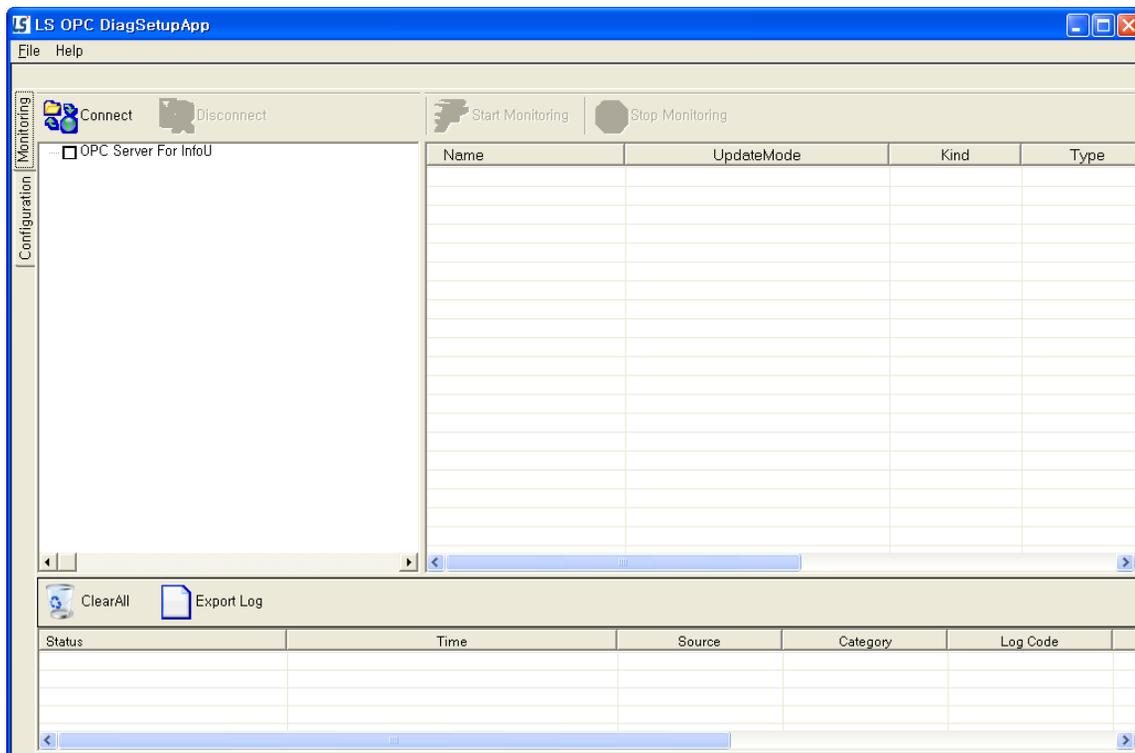
선택된 채널의 InfoU 프로젝트로부터 읽어 온 태그 DB를 트리 형식으로 보여줍니다.



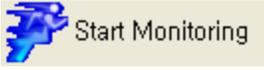
(6) 설명 창: 속성 창의 각각 필드에 대한 설명을 보여줍니다.



4.3.2 모니터링(Monitoring)

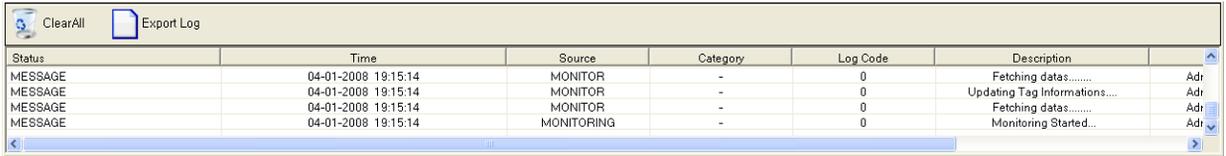


(1) 툴바

- ①  **Connect** : LSIS InfoU OPC 서버 에 연결합니다.
- ②  **Disconnect** : LSIS InfoU OPC 서버와의 연결을 끊습니다.
- ③  **Start Monitoring** : 모니터링을 시작합니다.
- ④  **Stop Monitoring** : 모니터링을 중단 합니다.

(4) 로그 창

- ① 모니터링 시 발생한 로그 정보를 보여줍니다.

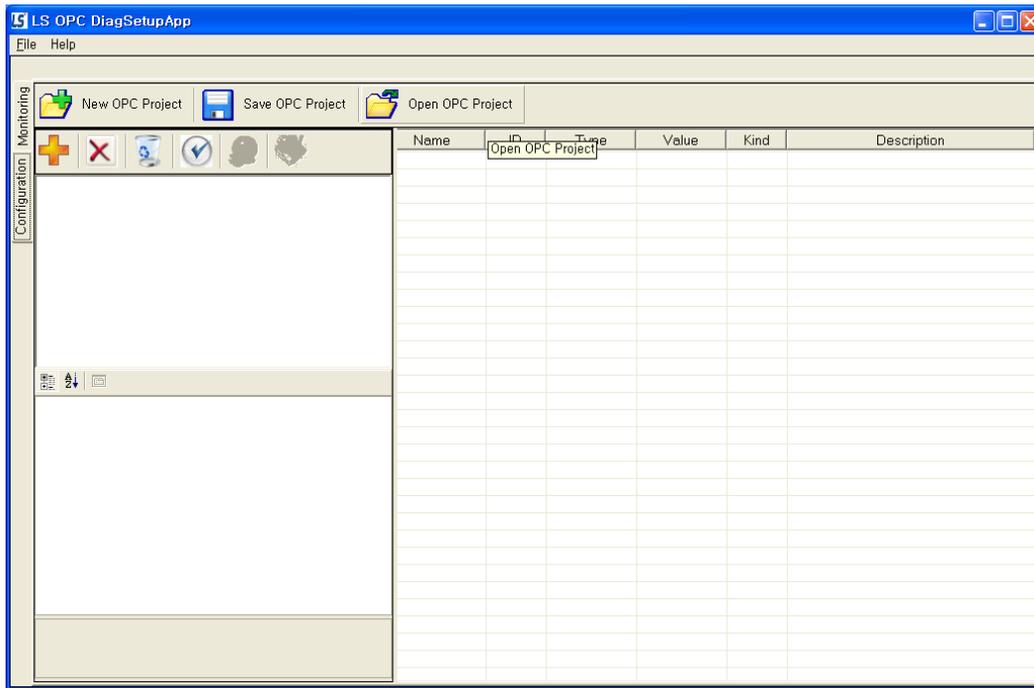


Status	Time	Source	Category	Log Code	Description
MESSAGE	04-01-2008 19:15:14	MONITOR	-	0	Fetching datas.....
MESSAGE	04-01-2008 19:15:14	MONITOR	-	0	Updating Tag Informations...
MESSAGE	04-01-2008 19:15:14	MONITOR	-	0	Fetching datas.....
MESSAGE	04-01-2008 19:15:14	MONITORING	-	0	Monitoring Started...

- ②  **ClearAll** : 로그 리스트의 내용을 삭제합니다.

- ③  **Export Log** : 로그 정보를 파일로 내보냅니다.

4.4 OPC 프로젝트 설정



4.4.1 새 OPC 프로젝트 생성

(1) New OPC Project

새로운 OPC 프로젝트를 생성하기 위해  New OPC Project 를 클릭합니다.

(2) 채널 추가

OPC 프로젝트에 채널 추가를 위해  을 클릭합니다.

(3) 채널속성 설정

생성된 채널 속성 창에, LS InfoU OPC DA 서버에서 수집해야 할 InfoU 프로젝트의 기본 정보와 데이터 수집 주기를 선택합니다.

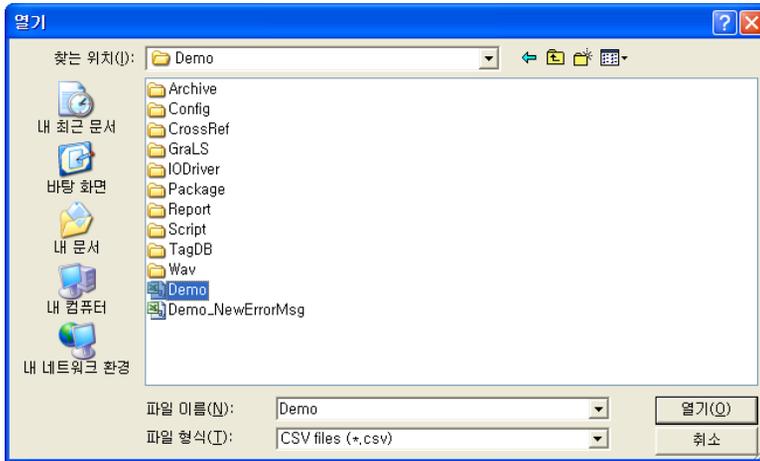
General Settings	
Channel Name	Channel1
Device Name	LSIS InfoU
IP Address	127.0.0.1
Project Name	Demo
Project Path	C:\InfoU\Project\
Update Cycle(ms)	1000
Update Mode	Realtime

- ① Channel Name: Channel 이름으로 유일한 이름을 가져야 합니다. InfoU 프로젝트를 구분할 수 있게 해줍니다.

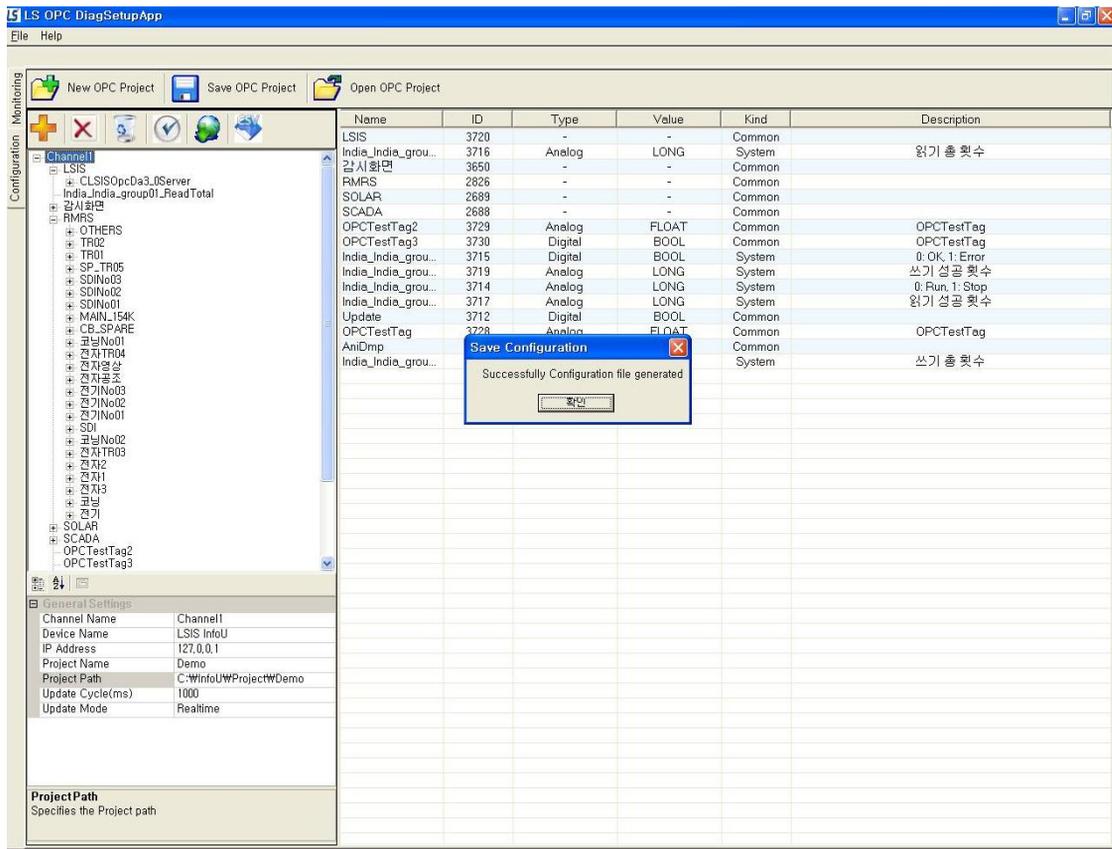
- ② Device Name : LSIS InfoU 임을 알려줍니다.
- ③ IP Address : InfoU 프로젝트가 실행되고 컴퓨터의 IP 주소를 설정합니다.
- ④ Project Name : InfoU 에서 현재 실행되고 있는 프로젝트 명을 설정합니다.
- ⑤ Project Path : InfoU 프로젝트가 존재하는 위치를 설정합니다.
- ⑥ Update Cycle(ms) : InfoU로부터 태그 값을 가져오는 주기를 설정합니다.
- ⑦ Update Mode : 실제 태그 값을 InfoU 에서 제공할지 OPC DA 서버 에서 시뮬레이션으로 제공할지를 설정합니다.
- ⑧ 채널의 통신 상태를 체크를 하기 위해,  를 클릭합니다.



- ⑨ 채널로부터 태그 리스트를 읽어 들여와 등록하기 위해  를 클릭합니다. 또는  를 클릭하여 InofUD 에서 생성한 해당 CSV File 을 선택하여 가져옵니다.



- ⑩  Save OPC Project 를 클릭하여 OPC 프로젝트인 Configuration File(LSIS0pcDaServer.config.xml)을 생성합니다. Configuration File 은 프로그램이 설치된 위치에 생성되어 LSIS InfoU OPC DA 서버가 기동 시 읽어 들여 태그 데이터를 가져오게 됩니다.



4.4.2 기존 OPC 프로젝트 불러오기 / 수정하기

(1)Open OPC Project

 Open OPC Project 를 선택하여 기존에 만들어 놓은 OPC 프로젝트인 Configuration File (LSISOpcDaServer.config.xml)를 가져옵니다.

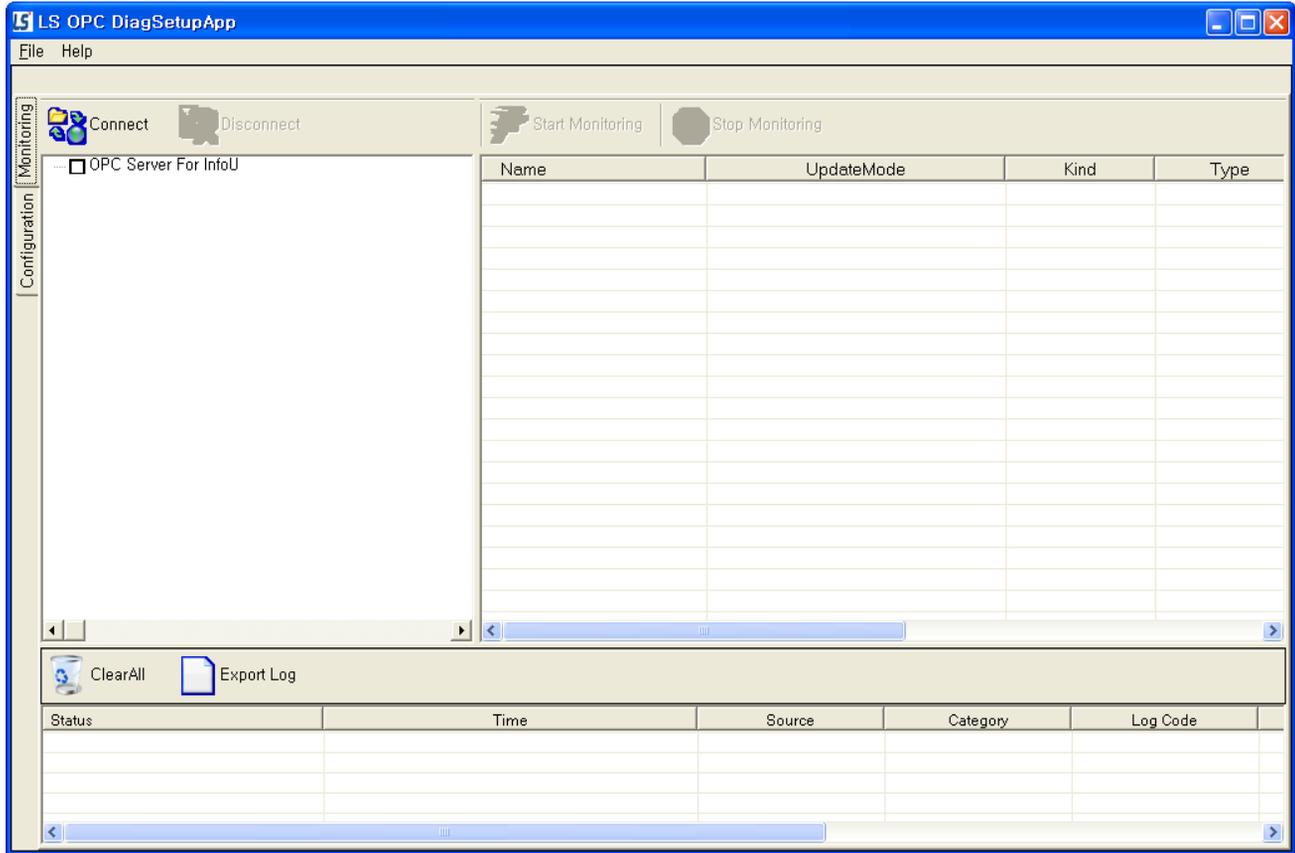
(2)채널 속성설정

채널 속성설정 창에서 Project Name, Project path, IP Address를 업데이트합니다.

(3)Save OPC Project

 Save OPC Project 를 클릭하여 변경된 OPC 프로젝트의 정보를 Configuration File(LSISOpcDaServer.config.xml)을 저장합니다.

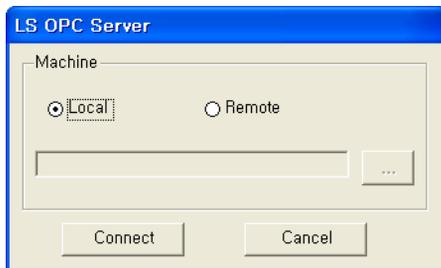
4.5 모니터링(Monitoring)



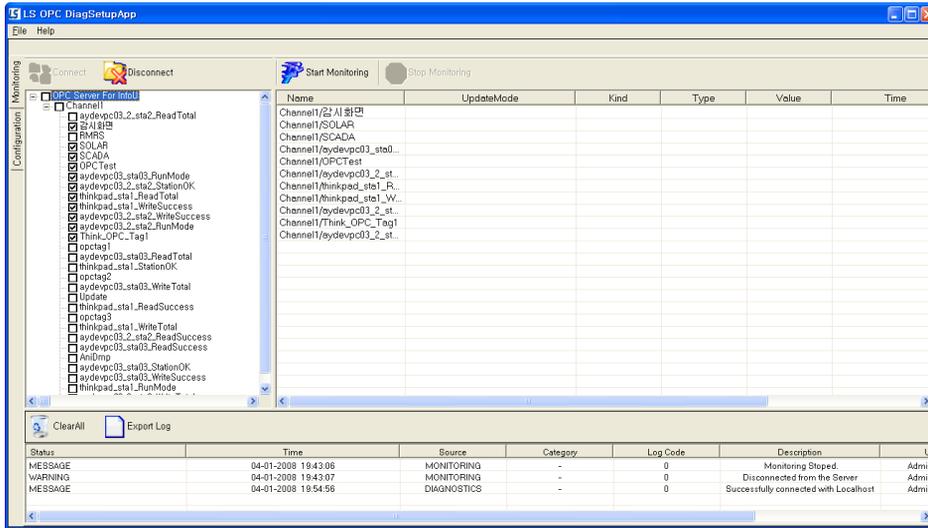
4.5.1 LS InfoU OPC 서버 연결/ 모니터링(Monitoring) 시작

(1) OPC 서버 연결

- ①  버튼을 클릭합니다. 아래 그림과 같이 연결하고자 하는 OPC 서버를 선택하는 창이 나옵니다.

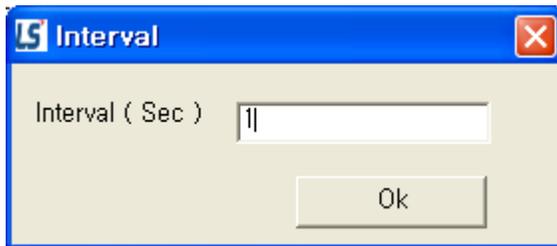


- ② 로컬인 경우, “Connect” 버튼을 클릭합니다. 원격인 경우 “Remote” 라디오 버튼을 선택한 후 LS InfoU OPC 서버가 실행될 컴퓨터의 이름을 직접 입력하거나 찾아보기를 통하여 입력한 후 “Connect” 버튼을 클릭합니다.

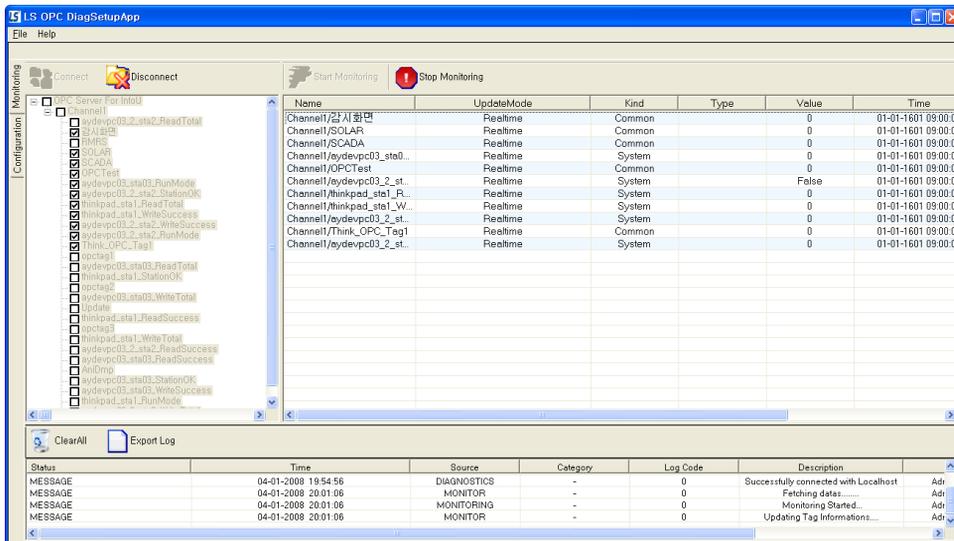


③ 태그 창으로부터 모니터링을 원하는 태그를 선택합니다. 선택된 태그는 모니터링 창에 등록됩니다.

④  Start Monitoring 버튼을 클릭합니다. LS InfoU OPC 서버로부터 태그 데이터들을 업데이트 하기 위한 주기를 선택하는 창이 나옵니다. 원하는 주기를 초단위로 입력 후 OK 버튼을 선택합니다.



⑤ 더 이상 모니터링을 위한 추가적인 태그 선택은 할 수 없도록 태그 창을 비활성화 한 후 모니터링이 시작됩니다. 모니터링 창에 태그 데이터들을 모니터링 결과를 볼 수 있습니다.



4.5.2 LS InfoU OPC 서버 끊기 / 모니터링(Monitoring) 중지

- ① 모니터링이 중지하기 위해  을 클릭합니다.
- ② LSIS InfoU OPC 서버 연결을 중지하기 위해  을 클릭합니다.

4.6 DCOM Configuration

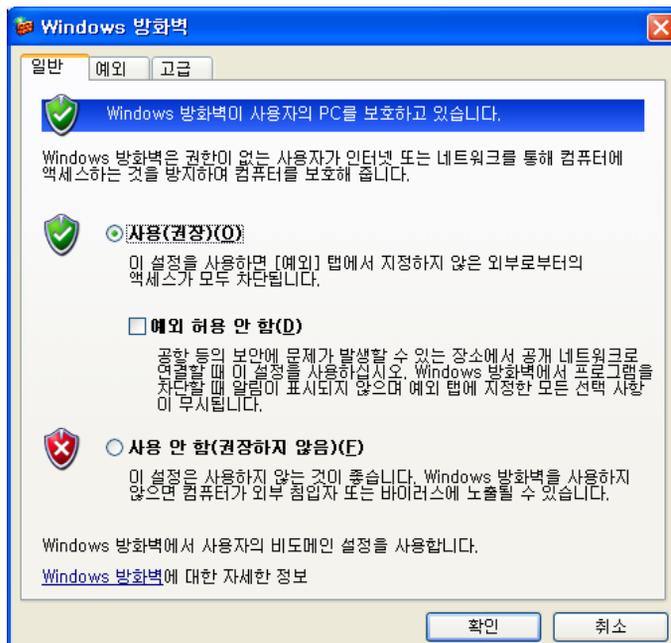
원격 컴퓨터상에서 OPC 서버 에 연결하기 위해서 DCOM 설정을 해 줘야 합니다. DCOM 설정은 컴퓨터 설정과 LS InfoU OPC DA 서버를 위한 설정이 있습니다. 이 설정은 윈도우 XP 서비스 팩 2 에서 DCOM 설정을 예를 들어 설명하였으며 다른 OS에서도 이와 비슷한 절차를 해 주어야 합니다.

기본 사항)

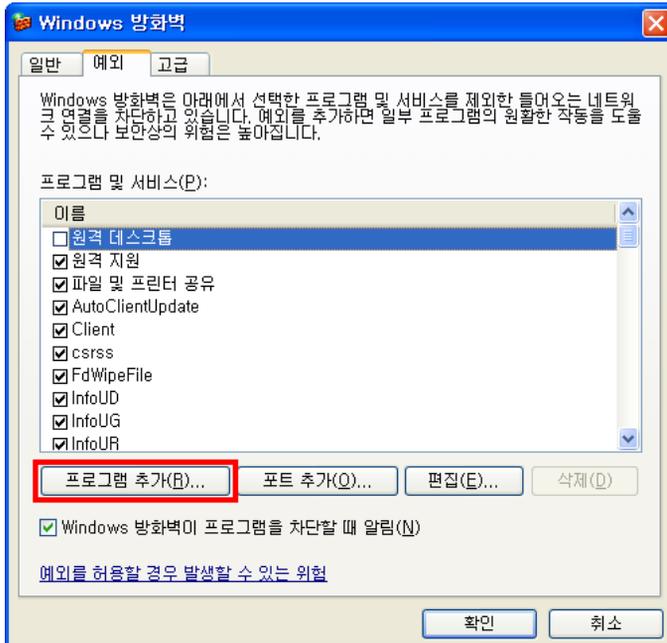
- 로컬 컴퓨터와 원격 컴퓨터는 동일한 계정으로 등록되어 있어야 하며 동일한 암호를 사용하여야 합니다.
- 로컬 컴퓨터와 원격 컴퓨터는 동일한 계정으로 로그인 되어야 합니다.
- 원격 컴퓨터상에서 OPC DA 서버를 사용하기 위해서는 로컬 컴퓨터상에 기본적인 OPC 파일들이 설치되어 있어야 합니다.

4.6.1 방화벽 설정

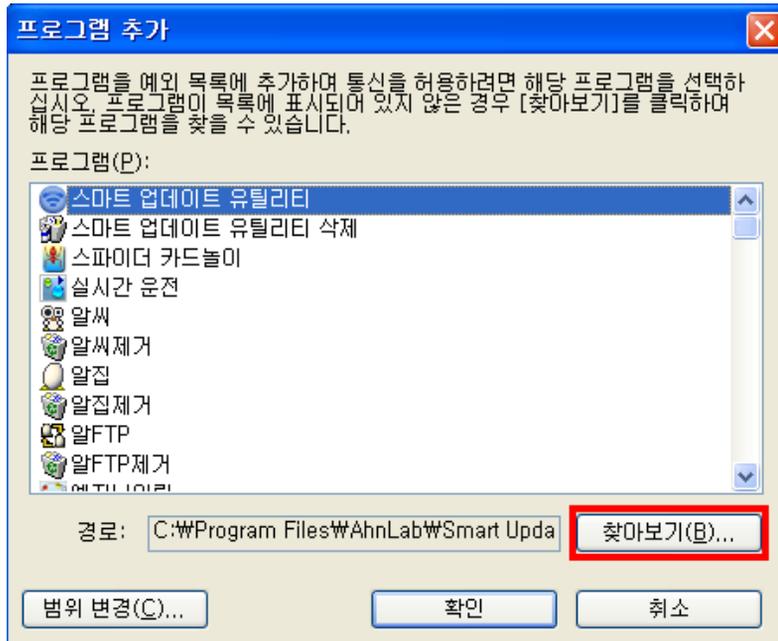
- ① “시작->제어판->보안센터->윈도우 방화벽” 을 선택합니다. 기본 설정으로 “사용(권장)” 이 설정되어 있습니다. 이 설정은 마이크로소프트 와 OPC로부터, 컴퓨터를 보호하기 위해 추천됩니다. 그러나, OPC 통신을 위해 “사용 안 함” 을 선택하거나 다음 아래의 사항을 따르도록 합니다.

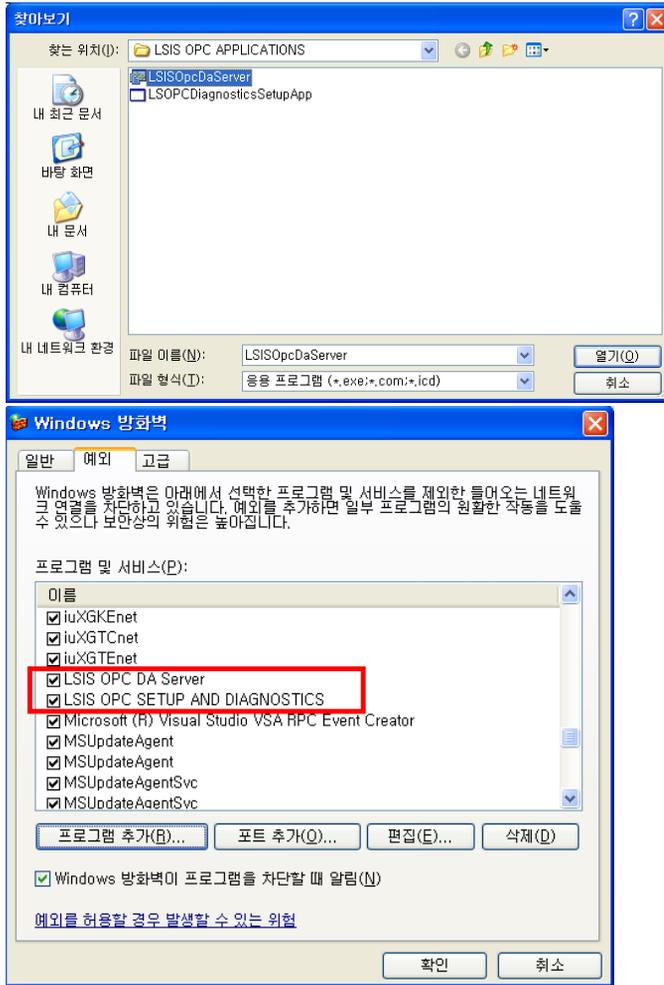


- ② “예외” 탭을 선택합니다. 아래 그림과 같이 “프로그램 추가” 를 선택합니다.

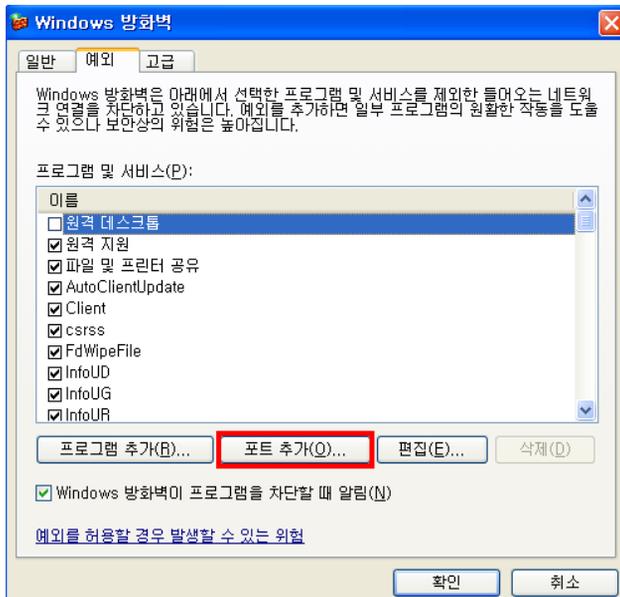


- ③ “찾아보기” 버튼을 선택하여 LSI InfoU OPC DA 서버 런타임 프로그램과 설정 프로그램인 LSISOpDaServer.exe 와 LSOPCDiagnosticSetupApp.exe를 등록합니다.

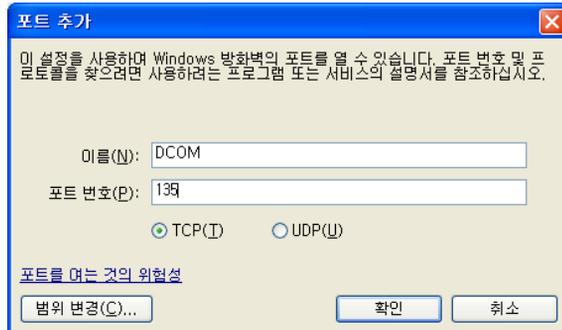




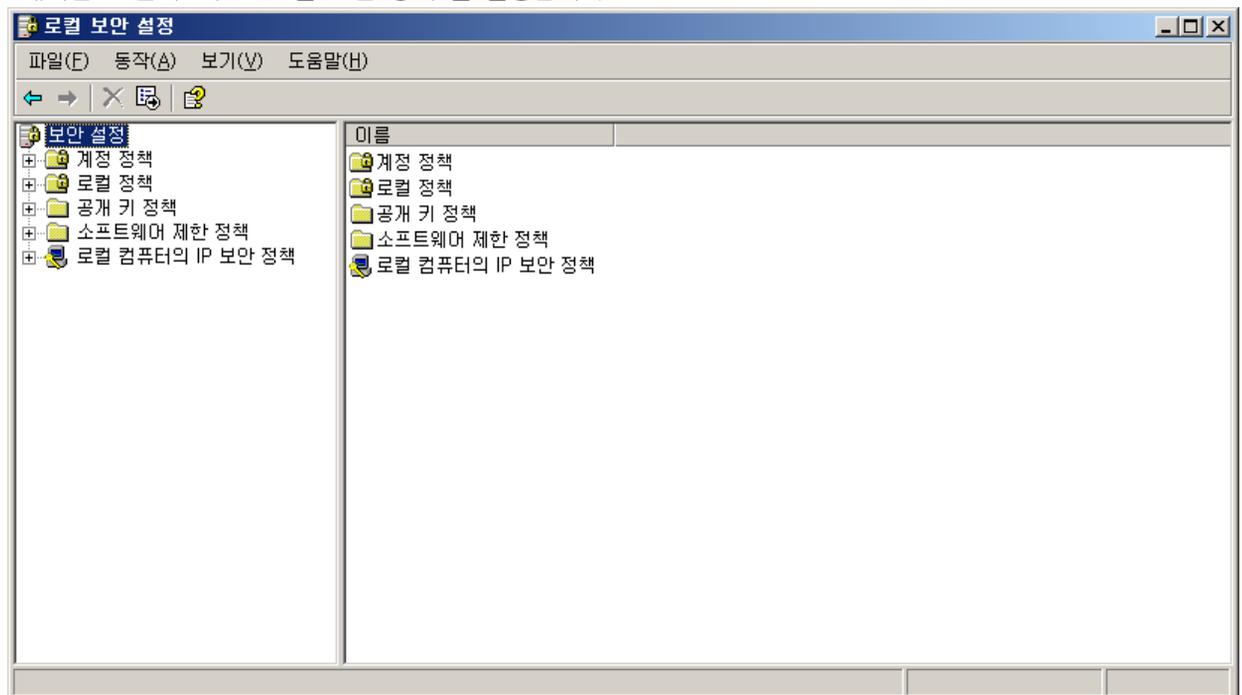
- ④ DCOM 통신을 초기화 하기 위해 TPC 포트 135를 추가해야 합니다. 윈도우 방화벽의 “예외” 탭에서 “포트 추가” 버튼을 클릭합니다.



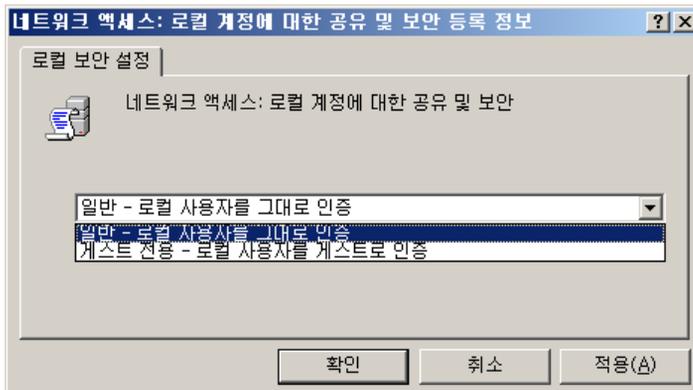
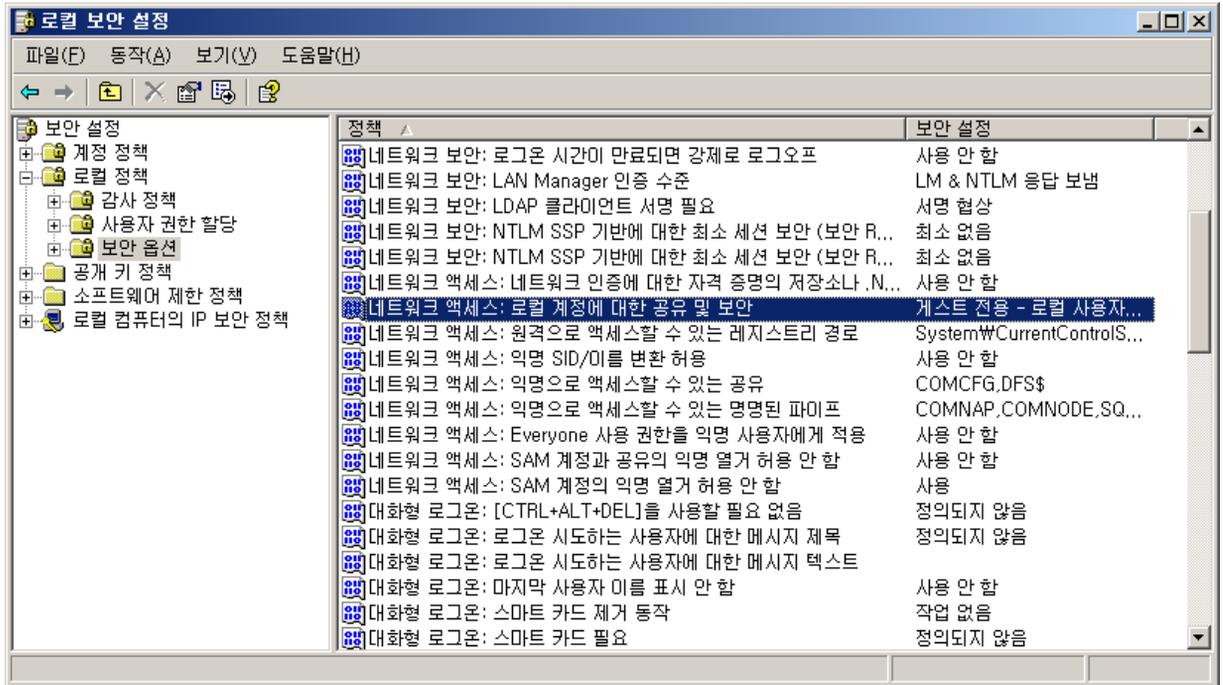
- ⑤ “ 포트 추가 ” 창에 다음과 같은 항목을 입력합니다.
 - 이름: DCOM
 - 포트번호: 135
 - “TCP “ 라디오 버튼 선택



- ⑥ 로컬 보안 설정
 - 제어판 - 관리도구 - 로컬 보안 정책 을 실행합니다.



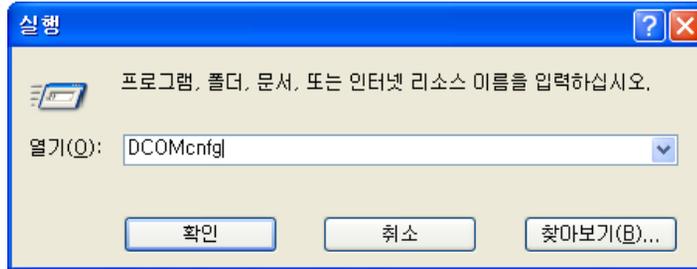
- 로컬 정책 - 보안 옵션 - “네트워크 액세스 : 로컬 계정에 대한 공유 및 보안”항목을 실행합니다.



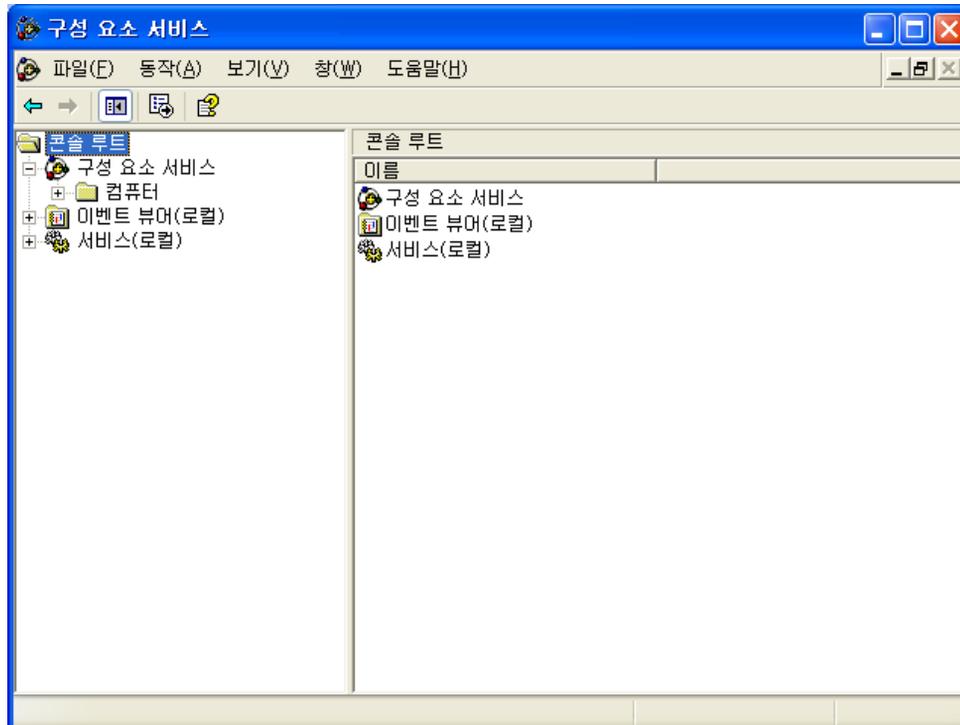
- “일반 - 로컬 사용자를 그대로 인증”을 선택 합니다.

4.6.2 DCOM 설정

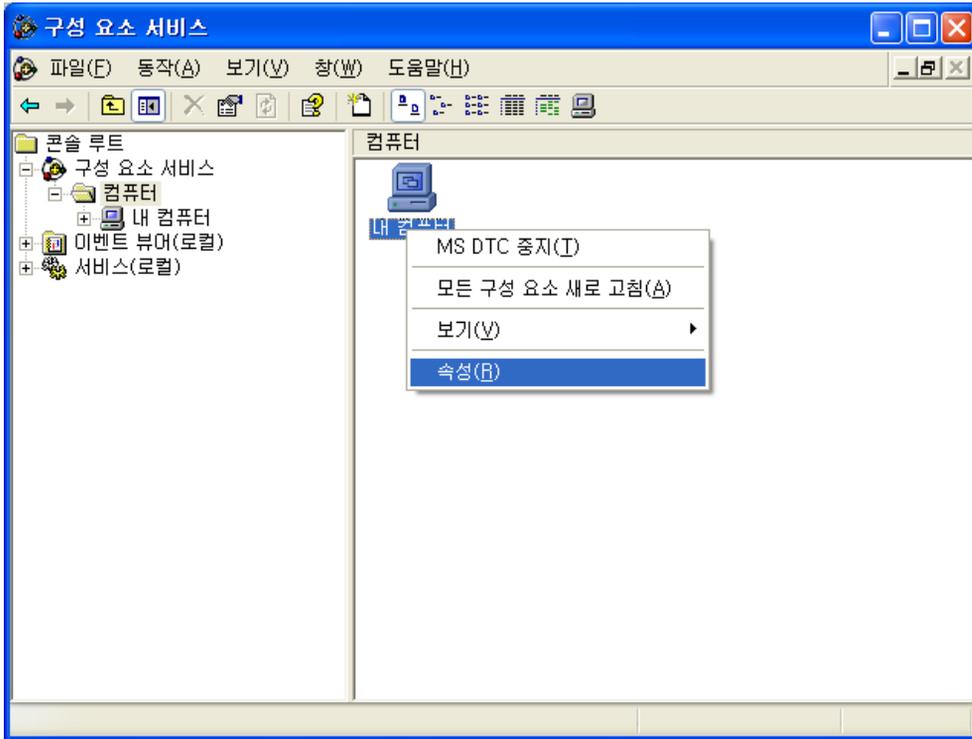
- 1) DCOM 컴퓨터 설정
 - “시작->실행->DCOMcng” 을 입력 후 “확인” 버튼을 누릅니다.



- 2) 아래와 같이 “구성 요소 서비스” 창이 표시됩니다. 콘솔 루트를 클릭하여 구성요소 서비스를 확장시킨 후 컴퓨터를 클릭합니다.

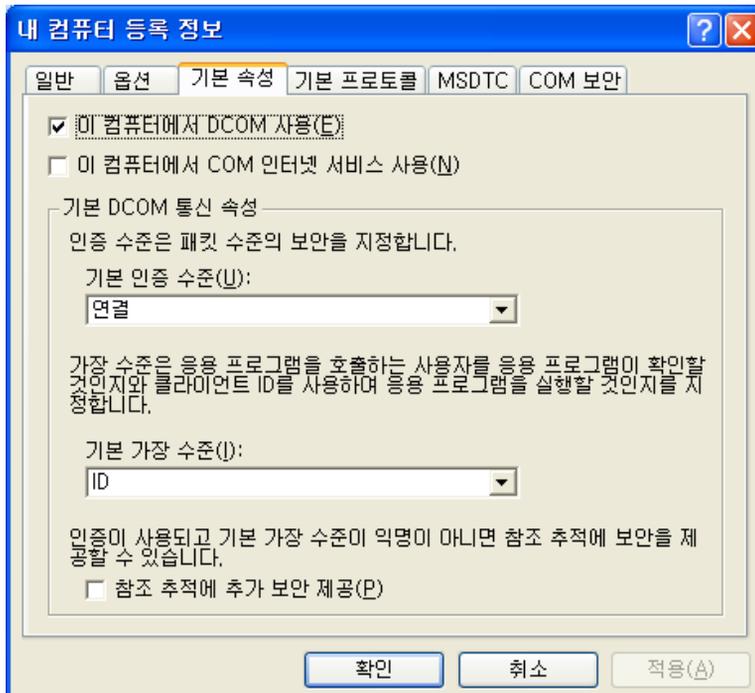


3) “내 컴퓨터” 항목을 찾아서 “속성” 메뉴를 실행합니다.

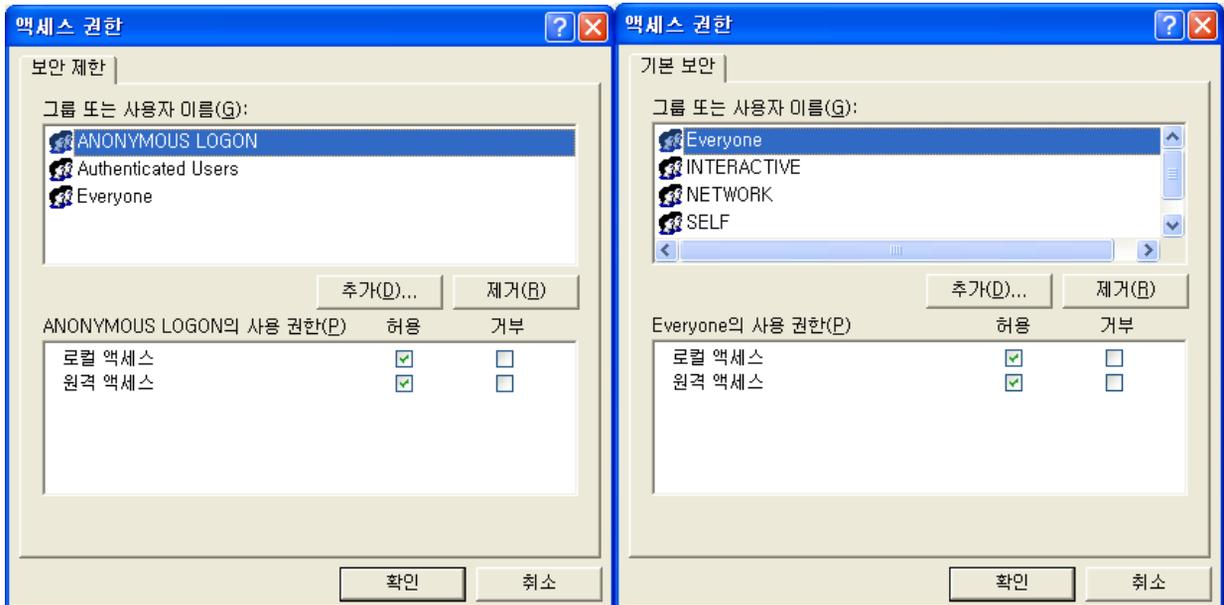


4) 아래 와 같이 “기본 속성” 탭에 아래와 항목에 지정된 값을 설정 합니다.

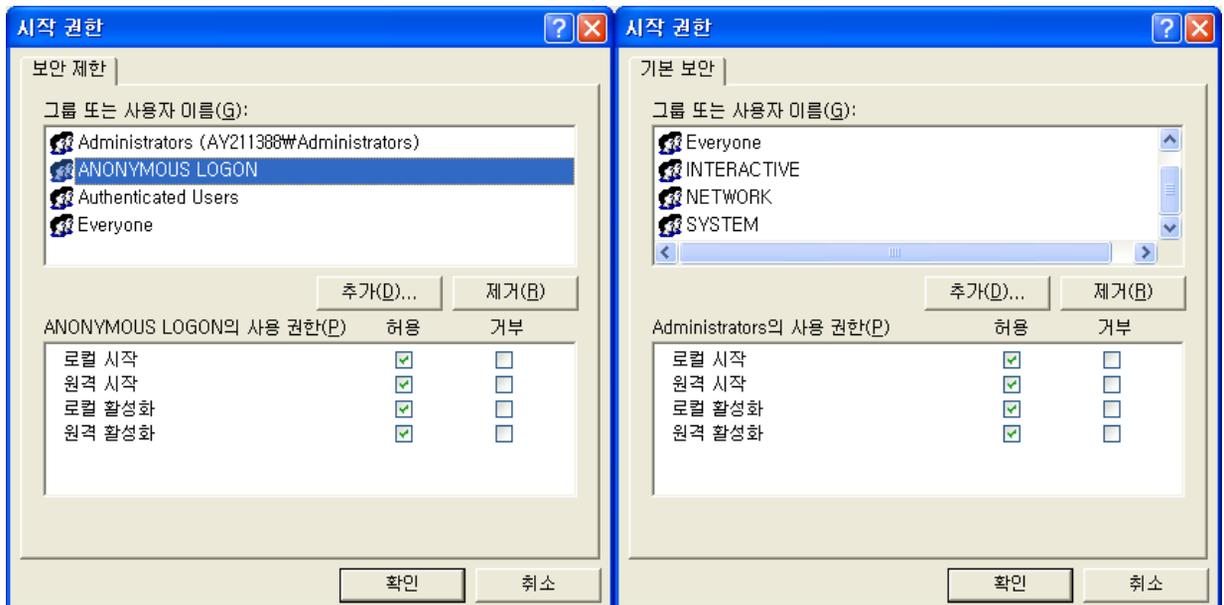
- "이 컴퓨터에서 DCOM 사용": 사용”으로 체크
- 기본 인증 수준: 연결.
- 기본 가장 수준: ID



- 5) 아래 와 같이 “COM 보안” 탭을 선택 후” 액세스 권한” 과 “ 시작 및 활성화 권한” 을 설정합니다.
- 6) “액세스 권한” 을 위해 “제한값 편집” 과 “기본값 편집” 아래와 같이 설정합니다.
 - 제한값 편집: 그룹 또는 사용자 이름 항목에 ANONYMOUS LOGIN, Everyone, Authenticated Users를 추가한 후 “로컬 액세스” 와 “원격 액세스” 항목을 모두 “허용”으로 체크합니다.
 - 기본값 편집: Everyone, INTERACTIVE, NETWORK, SYSTEM 사용자 각각에 대해 “로컬 액세스”와 “원격 액세스” 항목을 모두 “허용”으로 체크합니다.

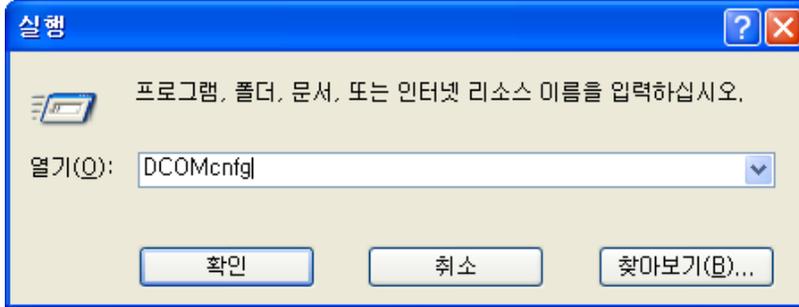


- 7) “시작 및 활성화 권한” 을 위해 “제한값 편집” 과 “ 기본값 편집” 을 아래와 같이 설정합니다.
 - 제한값 편집: ANONYMOUS LOGON, Authenticated Users 사용자 각각에 대해 “로컬 시작”, “원격 시작”, “로컬 활성화”, “원격 활성화” 항목을 모두 “허용”으로 체크합니다.
 - 기본값 편집: Everyone, INTERACTIVE, NETWORK, SYSTEM 사용자 각각에 대해 “로컬 시작”, “원격 시작”, “로컬 활성화”, “원격 활성화” 항목을 모두 “허용”으로 체크합니다.

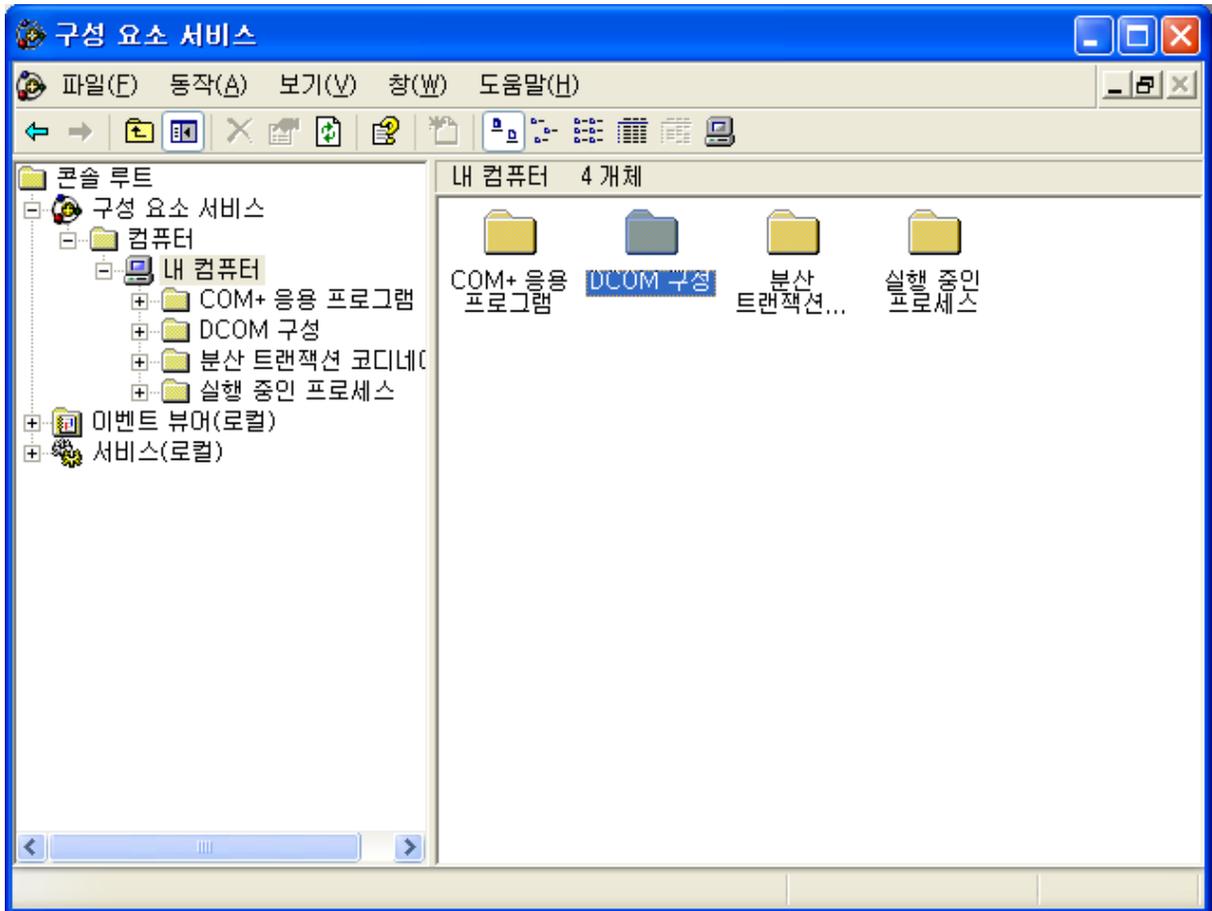


(1) LSI InfoU OPC 서버의 DCOM 설정

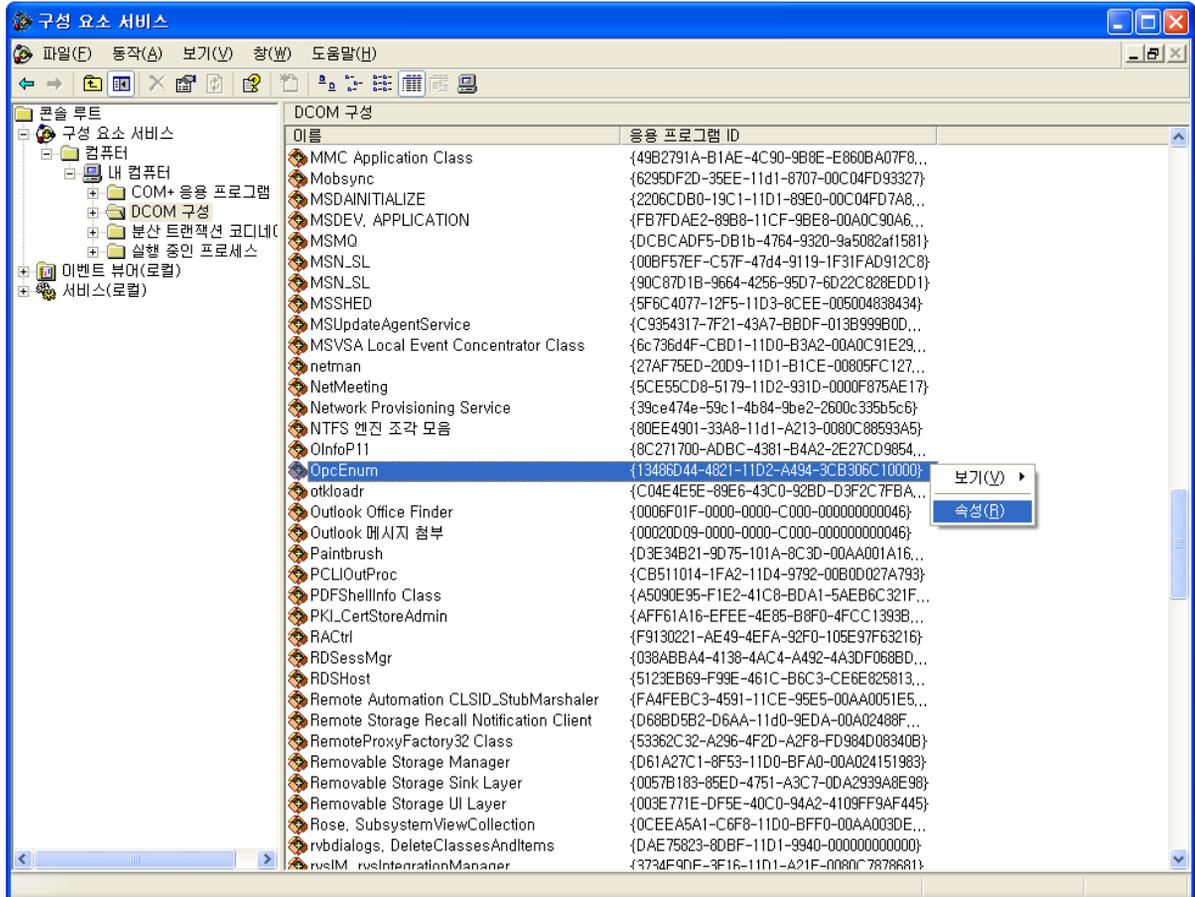
- 1) “시작->실행->DCOMcnfg” 을 입력 후 확인 버튼을 누릅니다.



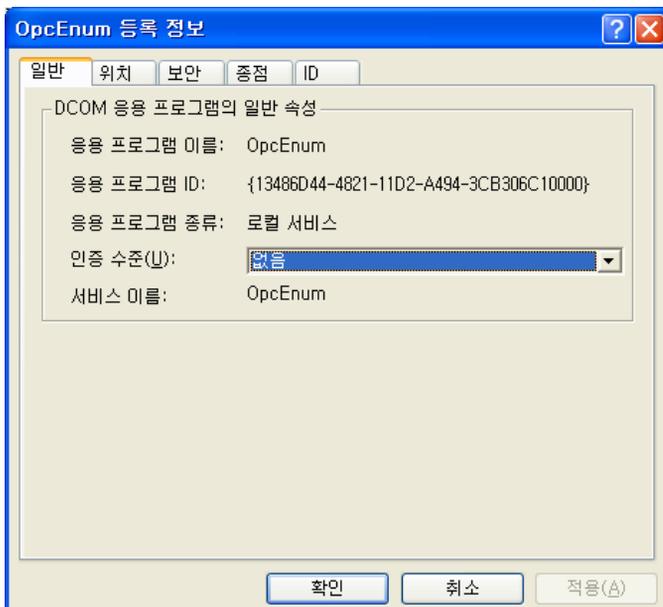
- 2) 아래와 같이 “구성 요소 서비스” 창이 표시됩니다. 콘솔 루트를 클릭하여 구성요소 서비스를 확장시킨 후 컴퓨터를 클릭합니다. 내 컴퓨터를 찾아 속성을 설정합니다.
- 3) “DCOM 구성” 을 더블 클릭합니다.



- 4) 익명의 연결을 허락하기 위해 “OpcEnum” 에 대한 속성을 설정해야 합니다.
(참고로 만약 “OpcEnum” 을 사용하지 않으면 익명의 사용자로부터의 원격 액세스를 허락할 수 없게 됩니다.)
- 5) “OpcEnum ” 를 리스트에서 찾아 “속성” 메뉴를 실행합니다.



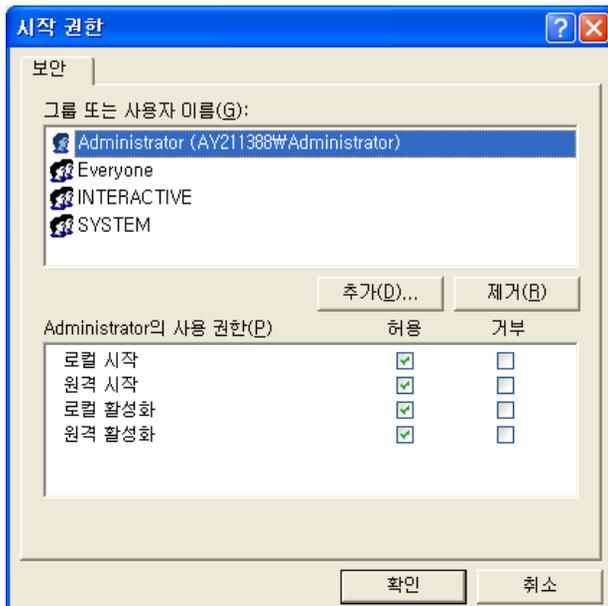
- 6) “OpcEnum 등록 정보” 창의 “일반” 탭에서의 “인증 수준” 항목을 “없음” 으로 설정합니다.



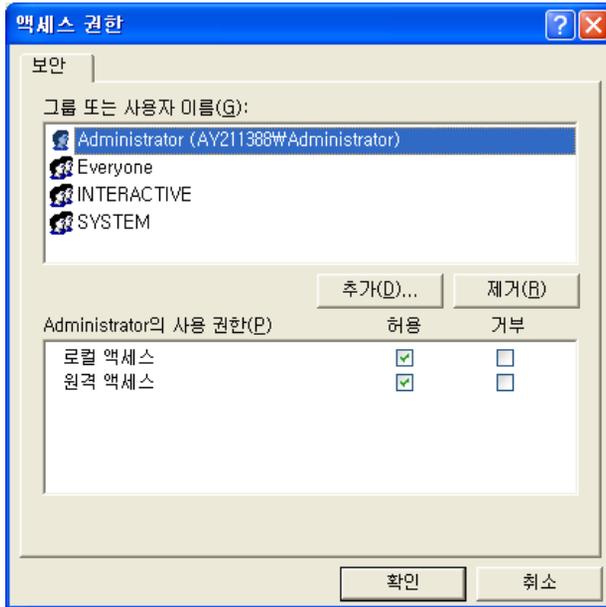
- 7) “보안” 탭을 선택합니다.



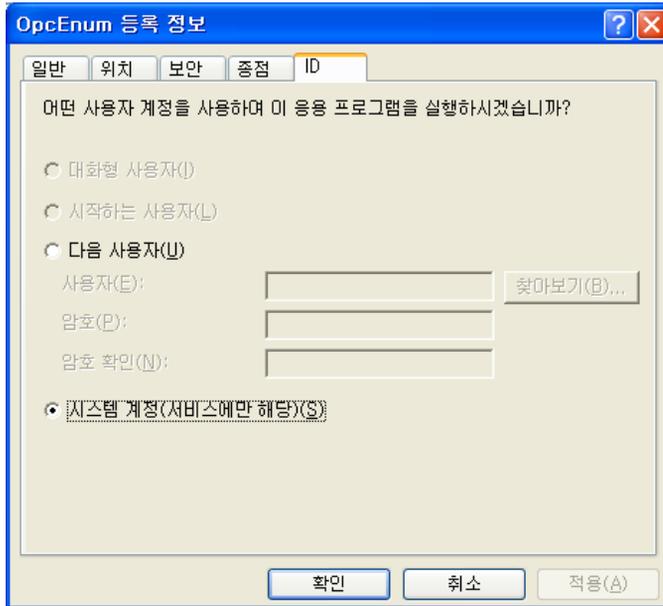
- 8) “시작 및 활성화 권한” 을 “사용자 지정” 을 선택 후 “편집” 버튼을 선택 후, 아래와 같이 설정합니다.
- “그룹 또는 사용자 이름” 모든 사용자 각각에 대해 “로컬 시작”, “원격 시작”, “로컬 활성화”, “원격 활성화” 항목을 모두 “허용”으로 체크합니다.



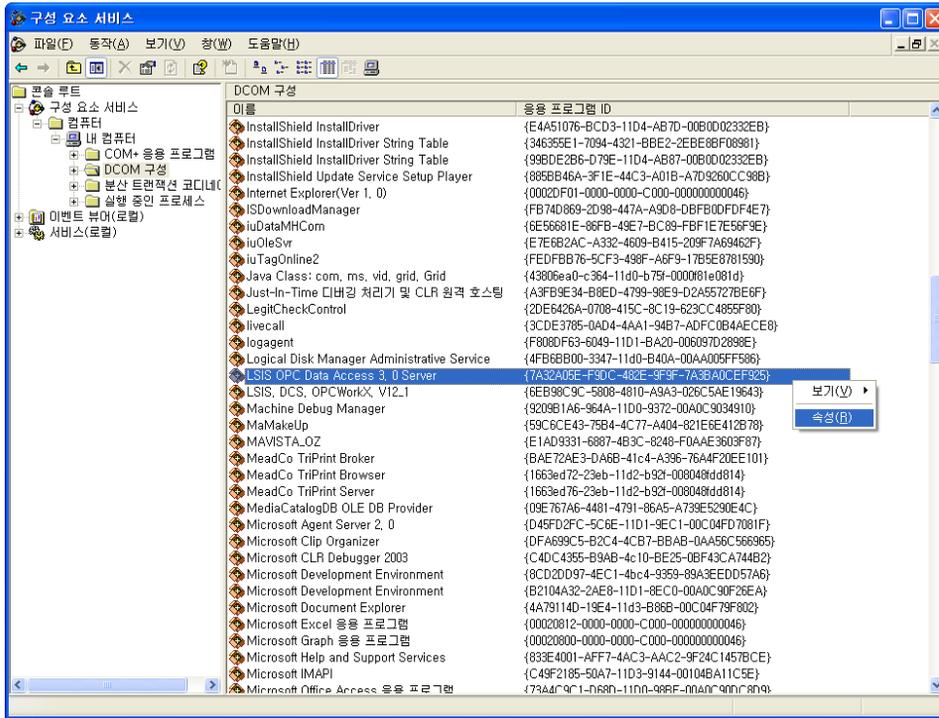
- 9) “액세스 권한” 을 설정한 후 “편집” 버튼을 선택하면 아래와 같이 설정합니다.
- “그룹 또는 사용자 이름”에 모든 사용자 각각에 대해 “로컬 액세스”와 “원격 액세스” 항목을 모두 “허용”으로 체크합니다.



- 10) “ID” 탭을 클릭한 후 “시스템 계정(서비스만 해당)” 항목을 선택합니다.

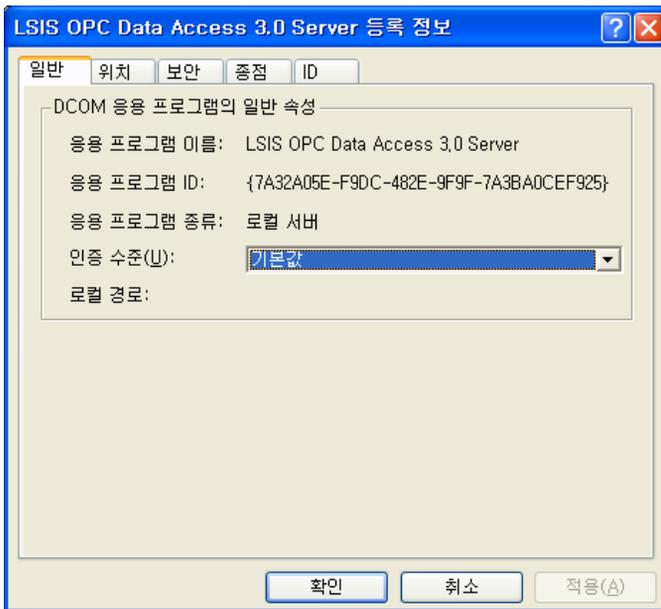


11) “LSIS OPC Data Access 3.0 서버” 를 리스트에서 찾아 “속성” 메뉴를 실행합니다.



12) “LSIS OPC Data Access 3.0 서버 등록 정보” 창이 표시됩니다.

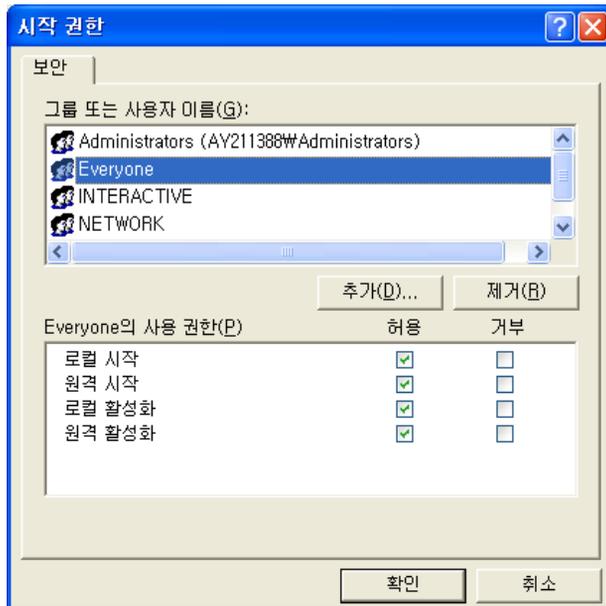
13) “일반” 탭에서 “인증 수준” 항목을 “기본값” 으로 유지합니다.



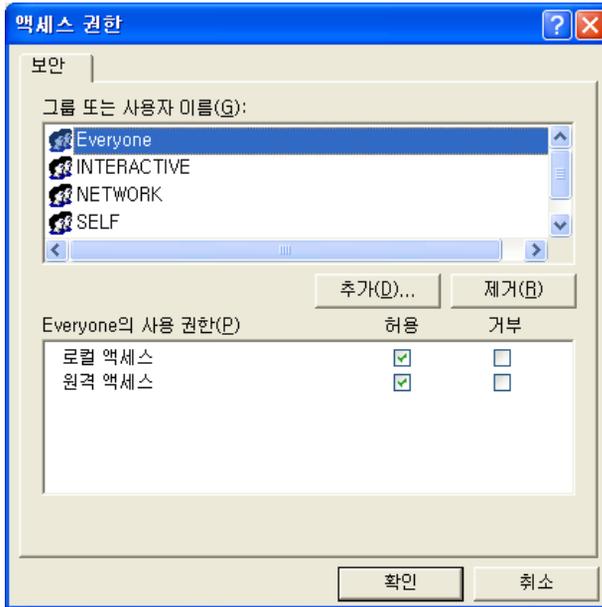
14) “보안” 탭을 선택합니다.



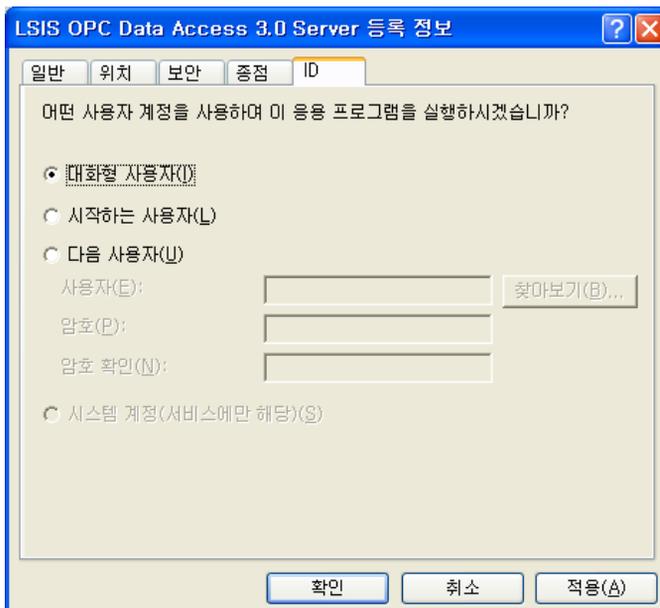
15) “시작 및 활성화 권한” 을 “사용자 지정” 을 선택 후 “편집” 버튼을 선택하면 아래와 같이 설정합니다.
 - “그룹 또는 사용자 이름” 모든 사용자 각각에 대해 “로컬 시작”, “원격 시작”, “로컬 활성화”, “원격 활성화” 항목을 모두 “허용”으로 체크합니다.



- 16) “액세스 권한” 을 설정한 후 “편집” 버튼을 선택 후 아래와 같이 설정합니다.
- “그룹 또는 사용자 이름”에 모든 사용자 각각에 대해 “로컬 액세스”와 “원격 액세스” 항목을 모두 “허용”으로 체크합니다.



- 17) “구성권한” 은 “기본값 사용” 으로 설정합니다.
 18) “ID” 탭을 선택 후 “대화형 사용자” 를 선택합니다.



- 19) “적용” 버튼 클릭 후 “확인” 버튼을 누릅니다.

부록.5. VISTA 및 WINDOWS7에서 INFOU 사용 위한 설정방법 5-2

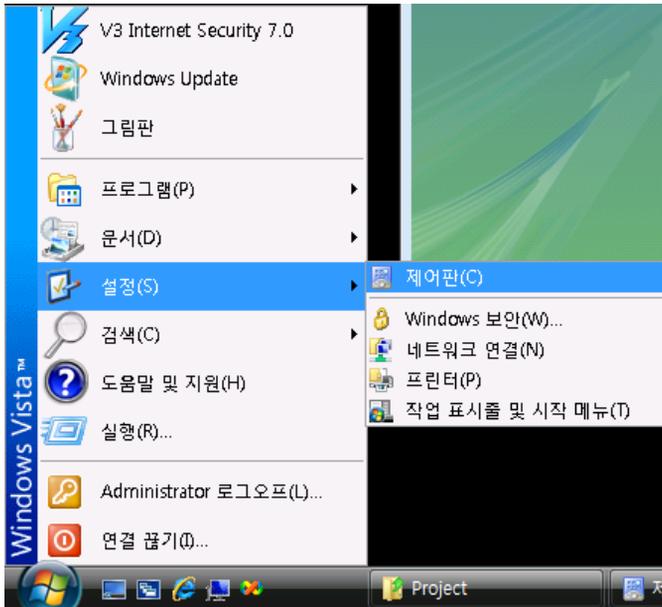
 5.1.1 Vista 보안설정.....5-2

 5.1.2 Windows7 보안설정.....5-4

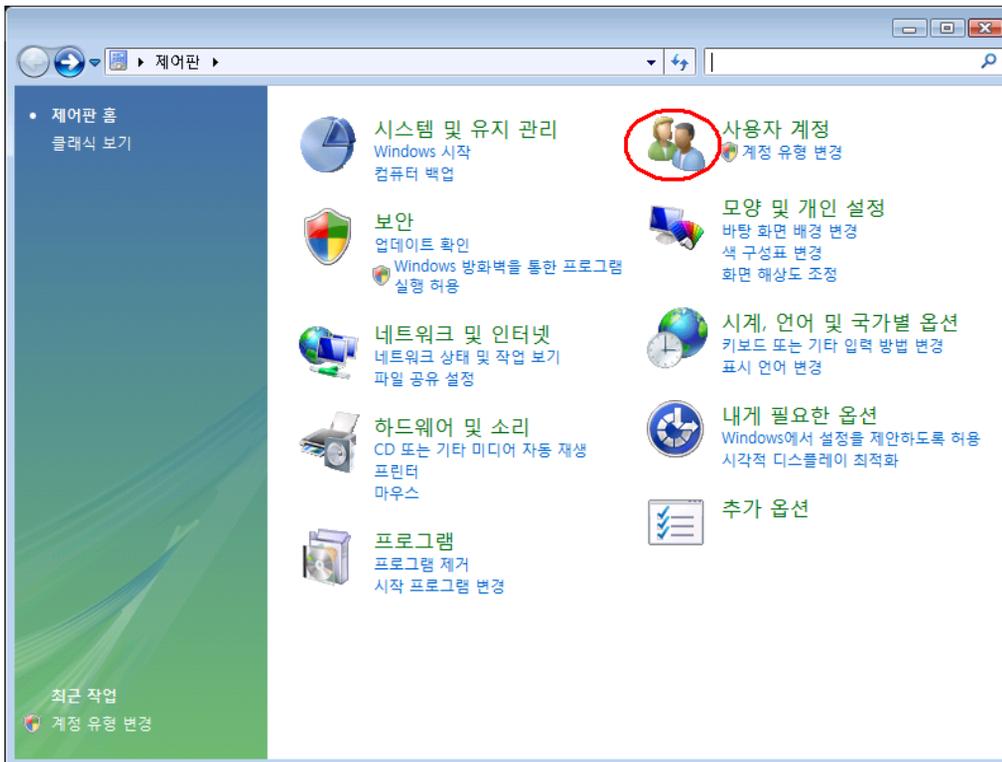
부록.5. Vista 및 Windows7에서 InfoU 사용 위한 설정방법

5.1.1 Windows Vista 보안설정

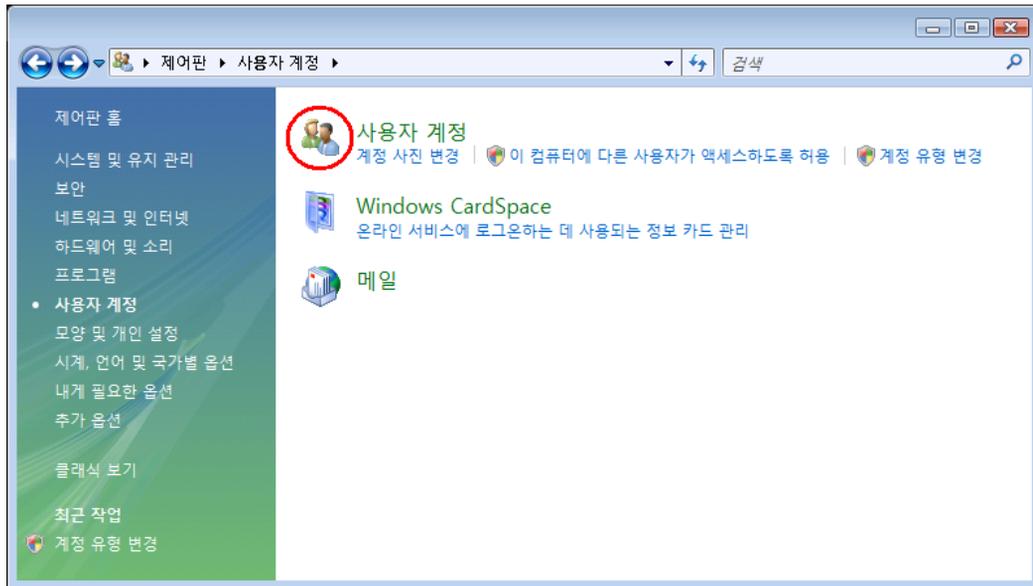
(1) 시작→설정을 클릭하여 제어판을 실행합니다.



(2) 제어판→사용자 계정 아이콘을 클릭합니다.



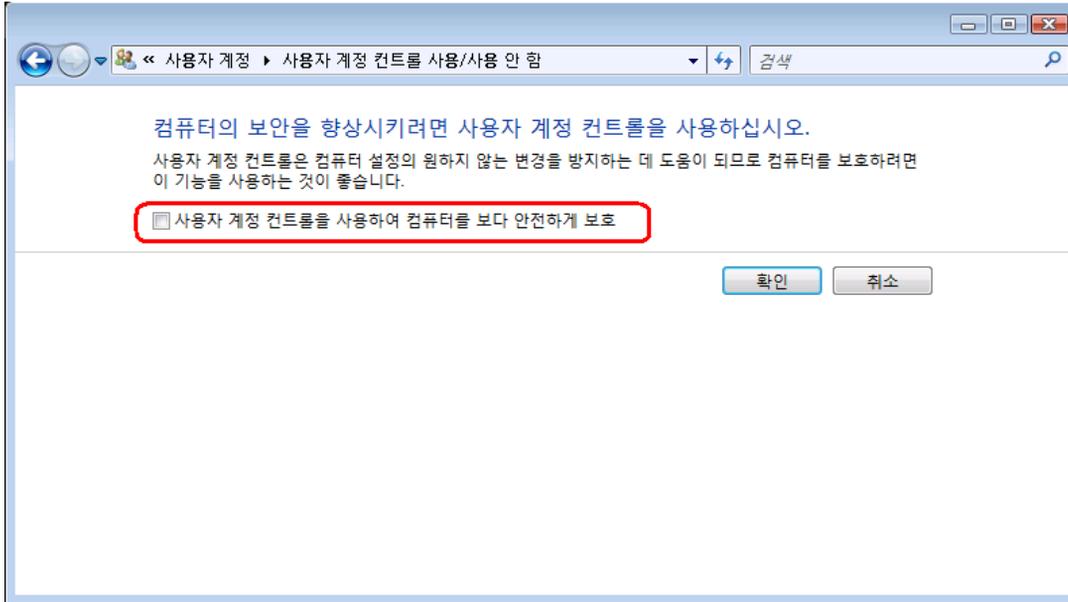
(3) 사용자 계정 설정 화면에서 사용자 계정 아이콘을 클릭합니다.



(4) 사용자 계정→사용자 계정 설정 화면에서 사용자 계정 컨트롤 사용/사용 안 함 아이콘을 클릭합니다.

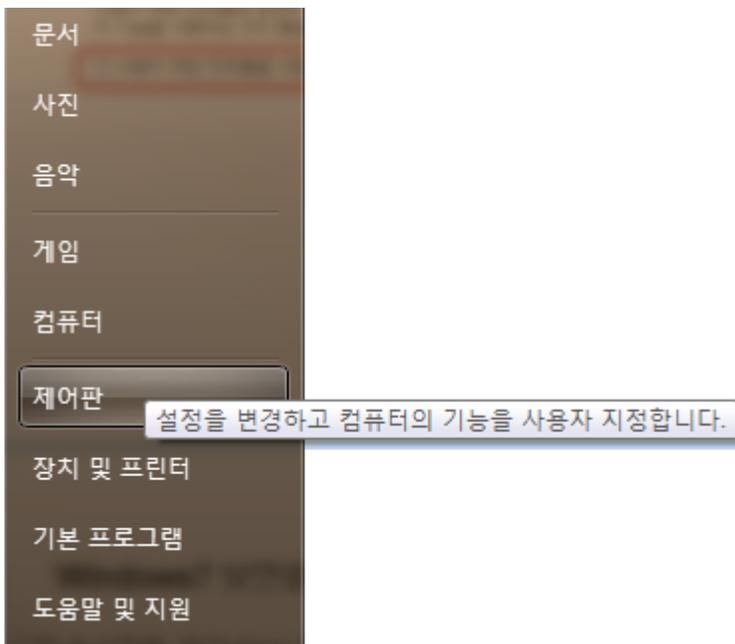


(5) 사용자 계정→사용자 계정 컨트롤 사용/사용 안 함 화면에서 사용자 계정 컨트롤 사용여부 설정을 해
제합니다..

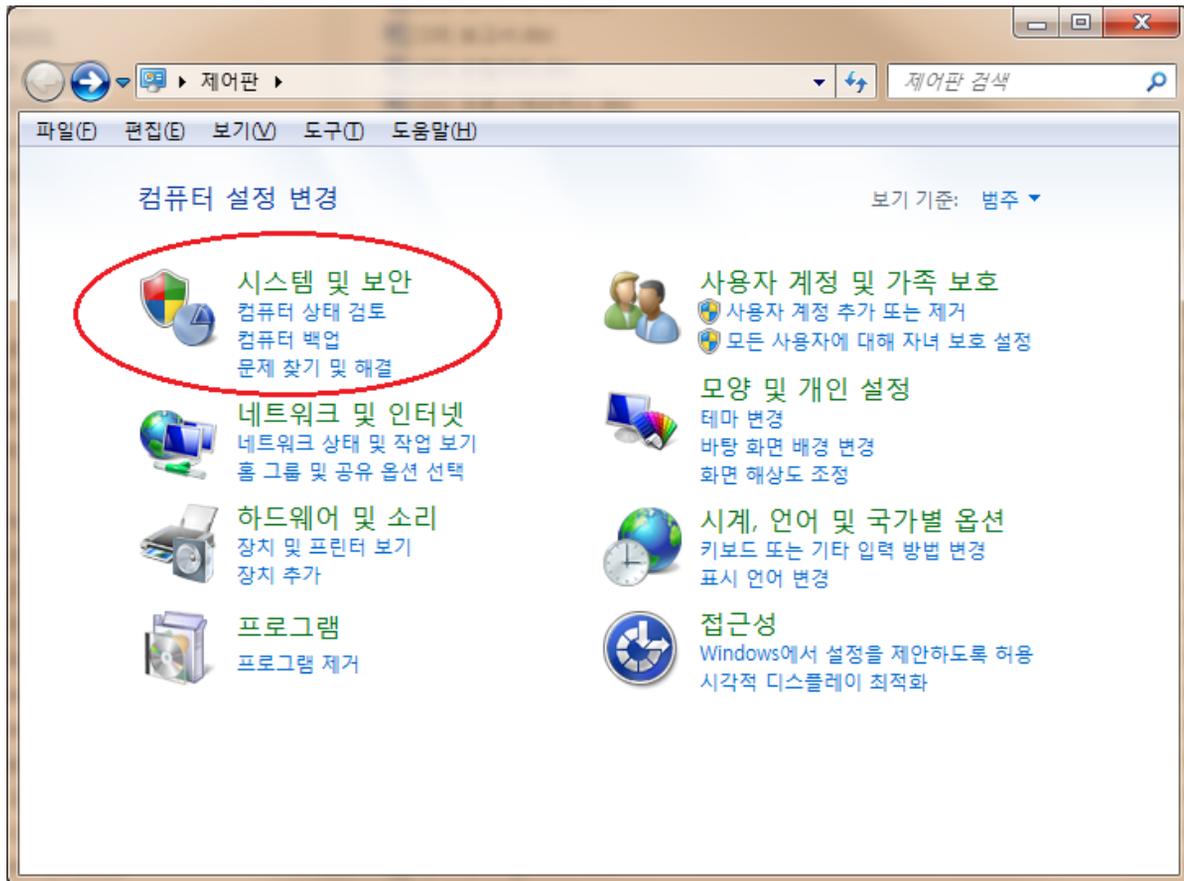


5.1.2 Windows7 보안설정

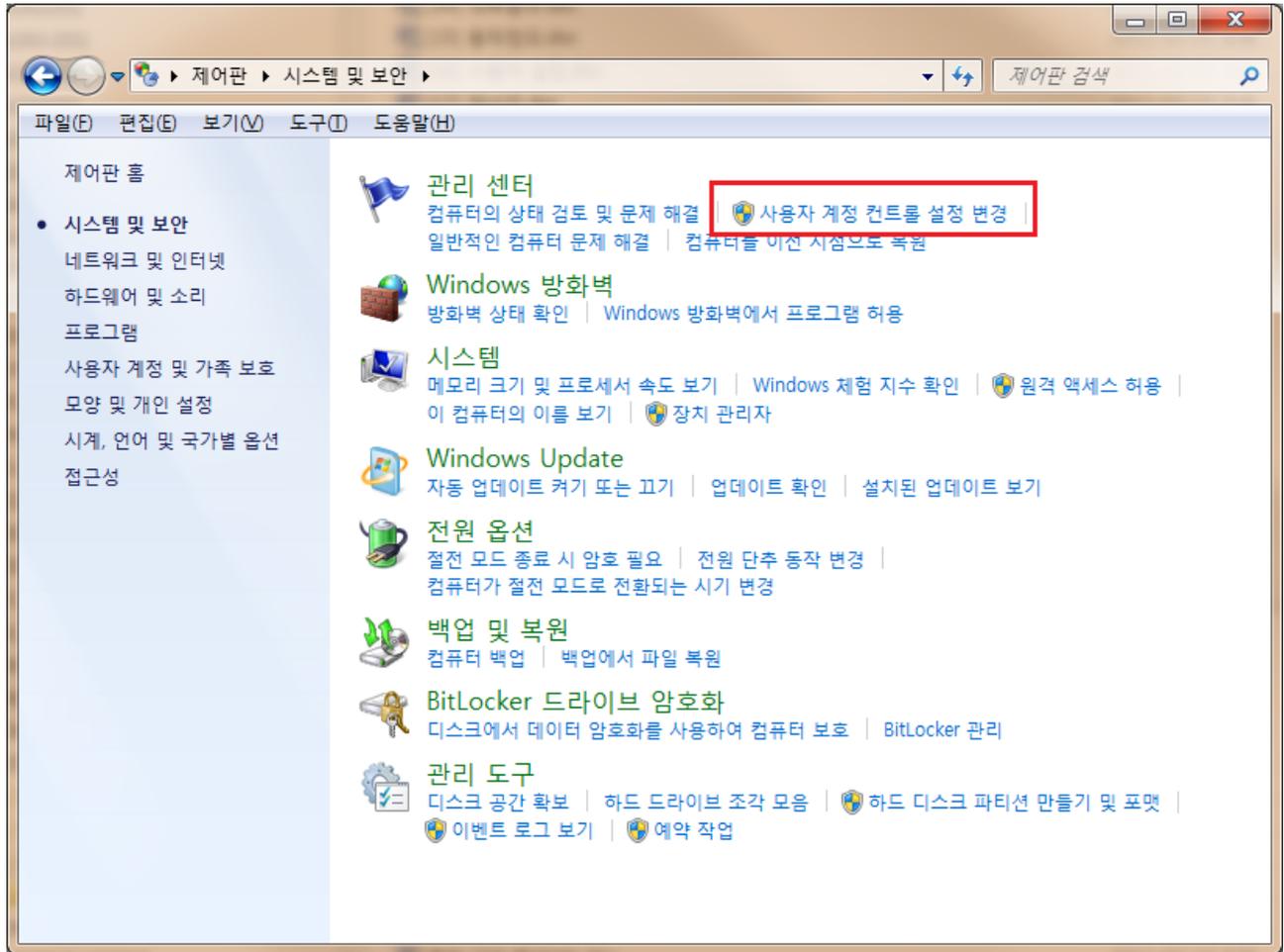
(1) 윈도우 시작을 클릭하여 제어판을 실행합니다.



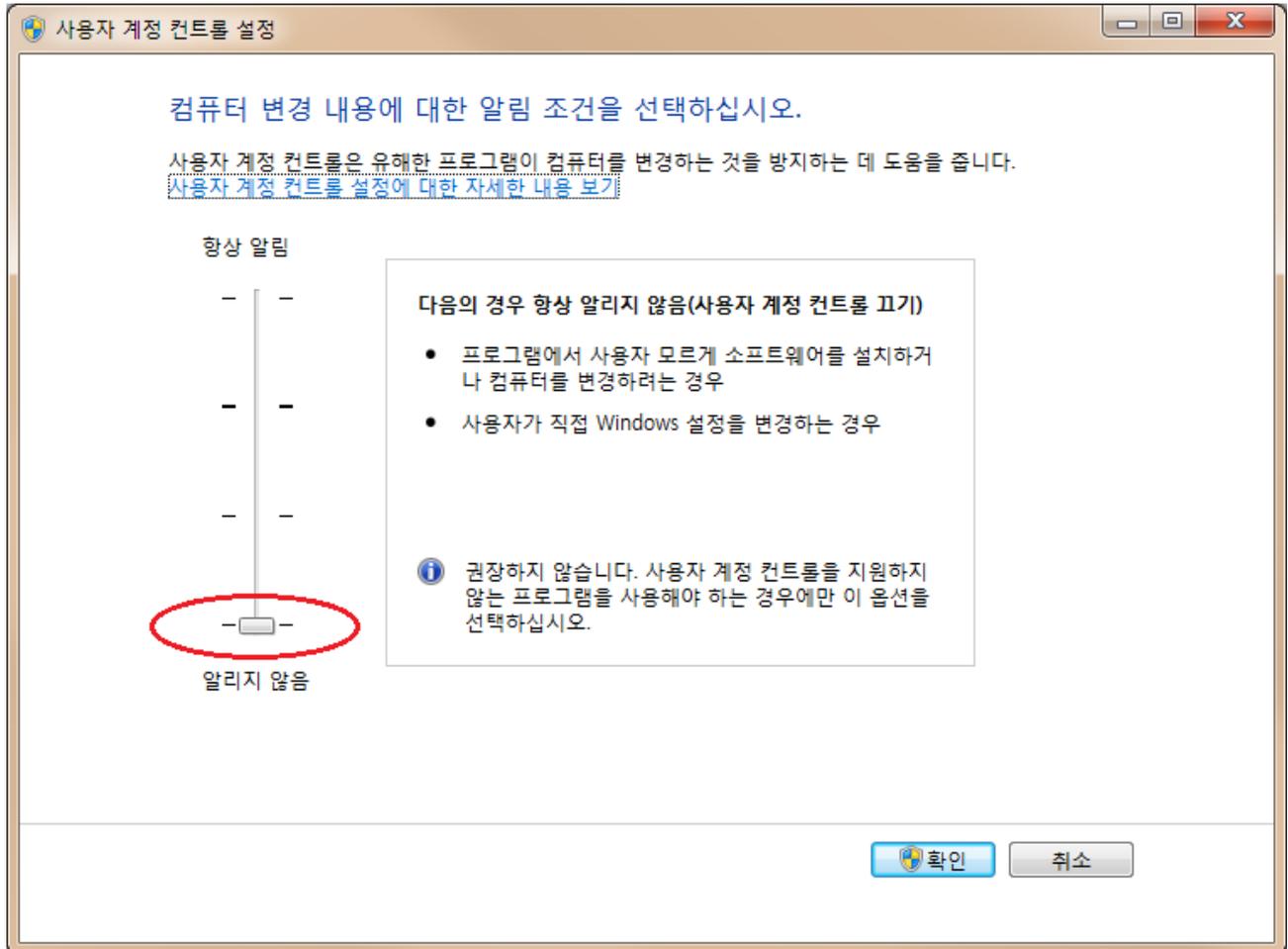
(2) 제어판→시스템 및 보안 아이콘을 클릭합니다.



(3) 관리 센터 - 사용자 계정 컨트롤 변경을 클릭합니다.



(4) 사용자 계정 컨트롤 설정을 알리지 않음으로 지정합니다.



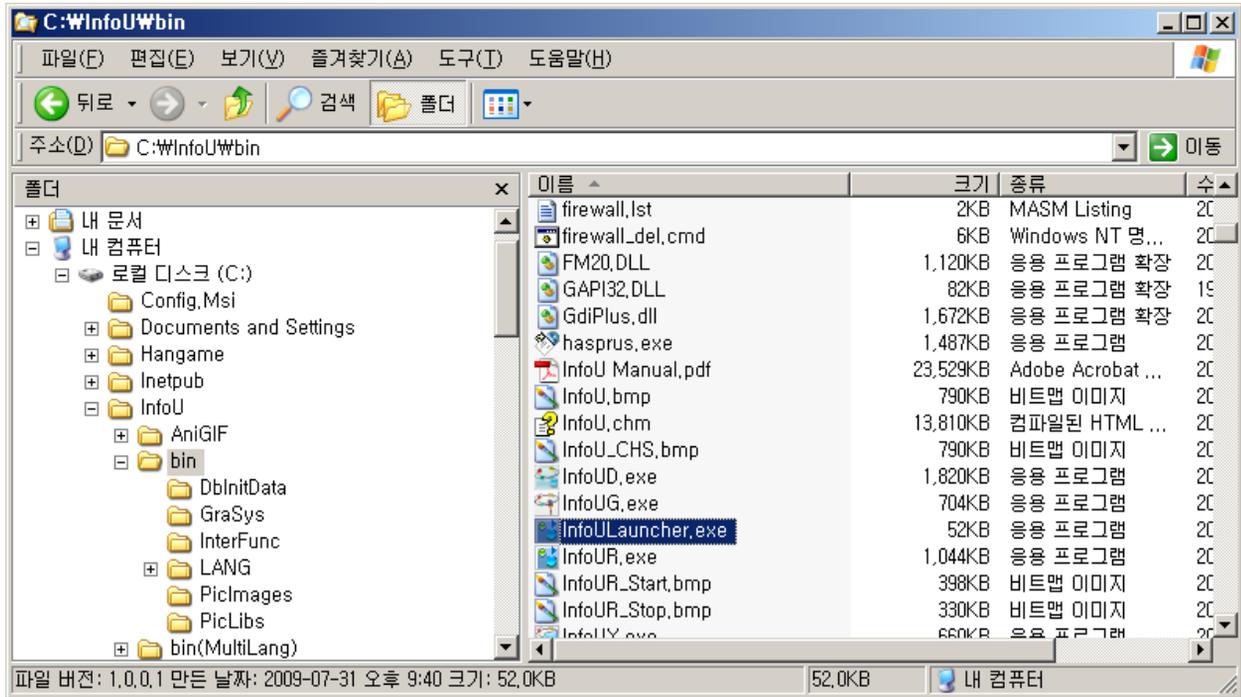
부록.6. 일반사용자 계정에서 INFOU 사용방법 6-3

 6.1.1 설치 및 실행.....6-3

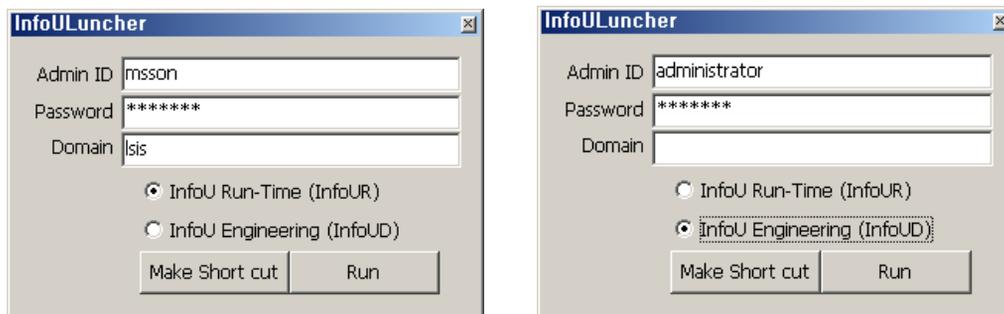
부록.6. 일반사용자 계정에서 InfoU 사용방법

6.1.1 설치 및 실행

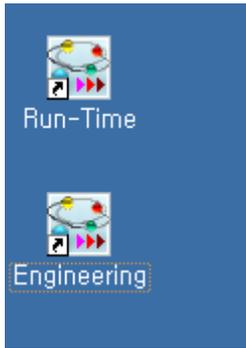
1. 설치는 반드시 Admin 권한을 가진 사용자로 로그인하여 설치합니다.
2. 일반 사용자 계정으로 로그인 합니다.
3. InfoU 설치 폴더의 bin 폴더에서 “InfoULauncher.exe”를 실행합니다.



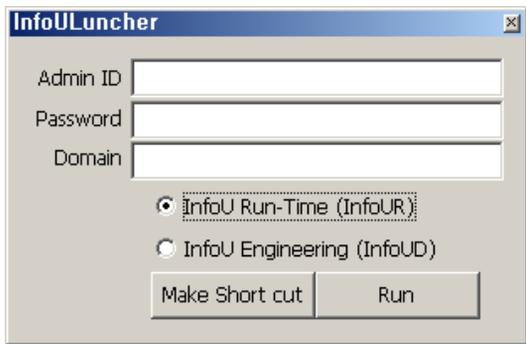
4. 아래화면이 표시되면 관리자(admin)권한을 가진 사용자 Id와 암호를 입력합니다.



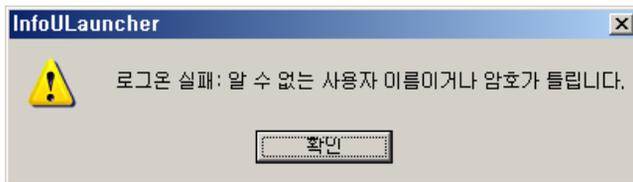
5. “InfoU Run-Time” 또는 “InfoU Engineering” 을 선택하고 Run버튼을 클릭합니다.
6. 사용자 ID와 암호가 맞는 경우에는 해당 프로그램이 실행되고 바탕화면에 아래와 같이 새로운 단축아이콘이 생성됩니다.
7. 또는 “Make Short cut” 버튼 클릭시 바탕화면에 단축아이콘이 생성됩니다.



8. 이후에는 바탕화면의 단축 아이콘을 클릭하면, 사용자와 암호 재입력 없이 바로 실행됩니다.
9. 단축아이콘을 클릭시 로그온이 실패했거나, 처음 실행하는 경우에는 입력화면이 표시됩니다.



10. 로그온 오류가 나온 경우에는 아래와 같은 오류 메시지가 표시됩니다.



11. 입력된 정보는 파일로 저장되지만 Password는 암호화되어 보안이 유지됩니다.

