

Programmable Gateway System ETOS Series

ETOS 시리즈 사용자 설명서

AC&T System Co., Ltd.

2004-07-14

Revision 3.0

ETOS 시리즈 사용자 설명서

본 사용설명서의 일부 또는 전부를 무단으로 복제하여 사용 할 수 없습니다.
본 사용설명서의 내용은, 제품의 기능 향상을 위해 예고 없이 변경될 수 있습니다.

Revision History

Revision No.	Date	Change
001	2000-10-09	초판
002	2002-09-05	부분 개정
003	2004-07-14	ETOS 업그레이드에 따른 전면 개정

AC&T Systems Co., Ltd.
Gyeonggi IT Center
1591-9, Burim-dong, Dongan-gu
Anyang-city, Gyeonggi-do, KOREA
431-815
TEL: +82-31-386-7795
FAX: +82-31-386-7796

주) 에이씨앤티시스템
경기도 안양시 동안구 부림동 1591-9
경기지식산업안양센터 901,902호, 431-815
Tel : 031-386-7795
Fax : 031-386-7796

Website : <http://www.acnt-sys.co.kr>

Copyright © 2000~2004. All rights reserved.

ETOS 제품을 효율적이고, 안전하게 사용하기 위하여 다음 내용을 숙지한 후 사용하기 바랍니다.
제품의 설치,배선과 사용 환경 그리고 폐기 처리시 위험이나 주의 사항에 대한 내용입니다.

위험

- | |
|---|
| <ol style="list-style-type: none">1) 통신용 케이블 작업 시 전원용 케이블과는 최소 100mm 이상 분리하여 주십시오.2) 제품 별 정격 전원에 주의하여 입력하여 주십시오. 특히 DC 24V 입력을 받는 제품은 반드시 DC 24V 를 입력해주십시오. AC 110~220V 을 입력하면 고장의 원인이 됩니다.3) 제품 전원이 AC 입력인 경우 반드시 접지가 접속되어 있는 전원을 사용하기 바랍니다.4) 접지는 3 종 접지를 사용하기 바랍니다. |
|---|

주의

- | |
|---|
| <ol style="list-style-type: none">1) 개봉시 손상, 변형되어 있는 제품은 사용하지 말아 주십시오. 고장,오동작을 일으킬 수 있습니다.2) 제품을 떨어뜨리거나 넘어뜨리는 등의 충격을 주지 말아 주십시오. 제품의 손상, 고장을 일으킬 수 있습니다.3) 사용설명서에 기재되어 있는 환경 조건에서 보관 및 사용해주십시오. 특히 진동이나 먼지,부식성 가스, 이슬이 맺히는 환경에서는 고장의 원인이 될 수 있으므로 규격 내의 환경에서 사용하기 바랍니다.4) 통신 케이블 작업 시 본 사용설명서에 있는 핀 배치도를 정확히 인지한 후에 결선하시기 바랍니다. ETOS 제품과 통신 선이 연결될 상대방 기기도 통신선 결선 내용을 사전에 충분히 검토하신 후에 작업하시기 바랍니다.5) ETOS-PD 로 작성된 프로그램이나 파일은 잘 보관하시기 바랍니다.6) 제품을 폐기할 때는 산업폐기물로 취급해 주십시오. |
|---|

목 차

1. 개요	2
1.1. 제품 개요.....	2
1.2. 제품의 기능 및 특징	2
1.3. 시리즈 구성.....	4
1.3.1. 최대 지원 포트 수	5
1.3.2. 모델명 부여 방법.....	5
1.3.3. H/W 사양.....	6
1.3.4. S/W 사양	8
2. 제품 규격	10
2.1. 일반 규격.....	10
2.2. 전원 규격.....	11
2.3. 케이블 규격.....	12
2.3.1. RS232C / RS422 / RS485 케이블	12
2.3.2. 이더넷 케이블	13
2.4. 통신 방식 규격	14
2.4.1. 이더넷 (10/100 Base-T)	14
2.4.2. 시리얼 (RS232C, RS422/485)	14
2.4.3. 필드버스.....	14
2.5. LED 표시 내용.....	15
2.5.1. ETOS-50A.....	15
2.5.2. ETOS-100A.....	17
2.5.3. ETOS-150.....	19
2.5.4. ETOS-200.....	21
2.5.5. ETOS-1000.....	23
3. 설치 및 배선	28
3.1. ETOS-50A 전원 및 커넥터 규격	28
3.1.1. 전원 규격	28
3.1.2. ETOS-50A 시리얼 모델 커넥터 규격.....	28
3.1.3. ETOS-50A 이더넷 모델 커넥터 규격.....	29
3.2. ETOS-100A/150 전원 및 커넥터 규격	30
3.2.1. 전원 규격.....	30
3.2.2. 온라인 연결 포트.....	30
3.2.3. 외부 통신용 포트.....	31
3.2.4. ETOS-100A/150 옵션카드 설치 및 배선	32
3.3. ETOS-50A/100A/150 배선방법	34
3.3.1. ETOS-50A/100A/150 PGM포트	34
3.3.2. ETOS-50A/100A/150 이더넷 포트	34
3.3.3. ETOS-50A/100A/150 시리얼 포트.....	35
3.4. ETOS-200/1000 설치 및 배선	41
3.4.1. ETOS-200 전원 규격.....	41
3.4.2. ETOS-1000 전원 규격	41
3.4.3. ETOS-200/1000 CPU 모듈 커넥터 규격	41
3.4.4. ETOS-200/1000 시리얼 통신 모듈 커넥터 규격	43
4. ETOS-PD 사용법	52
4.1. 개요.....	52
4.1.1. ETOS-PD의 기능 및 특징	52
4.1.2. 파라미터, 프레임 및 프로세스	52
4.2. 시스템 환경 및 설치	54
4.2.1. 시스템 환경	54
4.2.2. 설치 방법	54

4.3.	화면 설명	54
4.3.1.	ETOS-PD 기본화면	54
4.3.2.	화면 구성	55
4.3.3.	메뉴 구성	56
4.3.4.	메뉴 상세 설명	59
4.3.5.	통신 모니터링	79
4.3.6.	메모리 모니터	81
5.	파라미터	84
5.1.	개요	84
5.2.	파라미터 설정 순서	85
5.2.1.	새 파일 시작	86
5.3.	시리얼 파라미터	89
5.3.1.	포트 종류 선택	90
5.4.	이더넷 파라미터	94
5.4.1.	이더넷 드라이버 설정	95
5.4.2.	이더넷 채널 설정	98
5.4.3.	HSMS 설정	102
6.	프레임	106
6.1.	개요	106
6.2.	프레임의 구성	107
6.3.	프레임 등록	107
6.3.1.	그룹 추가	108
6.3.2.	프레임 추가	110
6.3.3.	세그먼트 추가	112
6.3.4.	BCC 세그먼트	119
6.4.	프레임 등록 예제	123
6.4.1.	예제 프레임	123
6.4.2.	송신 프레임 등록	124
6.4.3.	수신 프레임 등록	129
6.4.4.	그룹 추가	129
6.5.	SECS 메시지 등록	134
6.5.1.	메시지 추가	134
6.6.	SECS 메시지 등록 예제	139
6.6.1.	헤더만 있는 S1F1	139
6.6.2.	고정 값을 갖는 S1F2	140
6.6.3.	가변 값을 갖는 S1F2	141
6.6.4.	가변 String을 갖는 S6F11	142
6.6.5.	크기 지정, 가변 String을 갖는 S6F11	143
6.6.6.	동일 format이 반복되는 S6F11	144
6.6.7.	동일 format이 반복되는 S6F11, 값이 가변인 경우	145
6.6.8.	List구조의 반복	146
7.	프로세스 및 아이템	154
7.1.	개요	154
7.1.1.	특징	154
7.2.	프로세스 추가/편집	155
7.3.	프로세스 등록 예제	157
7.3.1.	예제 프레임	157
7.3.2.	프로세스 추가	157
7.4.	아이템 분류	164
7.4.1.	조건 분기 아이템	164
7.4.2.	시작 아이템	165
7.4.3.	통신 아이템	166
7.4.4.	Queue 아이템	167

7.4.5.	전용 아이템	168
7.5.	아이템 일람	169
7.5.1.	아이템 일람 (기능별)	169
7.5.2.	아이템 일람 (철자순)	171
7.5.3.	데이터 형식	173
7.6.	아이템 설명	174
7.6.1.	ARITHMETIC	174
7.6.2.	ASC_TO_BIN	176
7.6.3.	ASC_TO_HEX	178
7.6.4.	BIN_TO_ASC	180
7.6.5.	BIT_MASK	182
7.6.6.	BIT_WRITE	184
7.6.7.	BIT_WRITE_EX	186
7.6.8.	BITS_CPY	189
7.6.9.	BITS_OPERATION	191
7.6.10.	DELAY	193
7.6.11.	ELSE	195
7.6.12.	ENCRYPTION	197
7.6.13.	END	198
7.6.14.	END_IF	200
7.6.15.	END_LOOP	202
7.6.16.	ENET_CLOSE	203
7.6.17.	ENET_DISABLE	204
7.6.18.	ENET_ENABLE	205
7.6.19.	ENET_OPEN	206
7.6.20.	EVENT_READ_WRITE	207
7.6.21.	FRAME_SND	209
7.6.22.	HEX_TO_ASC	211
7.6.23.	HEX_TO_BCD	213
7.6.24.	IF_3964R_READ	215
7.6.25.	IF_3964R_WRITE	216
7.6.26.	IF_BIT_FIND	217
7.6.27.	IF_BIT_TEST	220
7.6.28.	IF_EVENT_OCCUR	222
7.6.29.	IF_FRAME_RCV	224
7.6.30.	IF_GLOFA_READ	227
7.6.31.	IF_GLOFA_WRITE	229
7.6.32.	IF_IO_STATE_CHANGE	231
7.6.33.	IF_MELSEC_READ	233
7.6.34.	IF_MELSEC_WRITE	234
7.6.35.	IF_MEM_CMP	235
7.6.36.	IF_MEM_WRITE_RCV	237
7.6.37.	IF_MODBUS_READ	242
7.6.38.	IF_MODBUS_WRITE	245
7.6.39.	IF_NUM_CMP	247
7.6.40.	IF_POWER_ON	249
7.6.41.	IF_REQ_RESP	251
7.6.42.	IF_SECS_MC_FRAME_RCV	254
7.6.43.	IF_SECS_MC_REQ_RESP	256
7.6.44.	IF_STR_CMP	258
7.6.45.	IF_TIMER	260
7.6.46.	INT_TO_STR	262
7.6.47.	IO_READ_WRITE	265
7.6.48.	IP_SAVE	267
7.6.49.	JUMP	269
7.6.50.	LABEL	271
7.6.51.	LOOKUP_TABLE	272
7.6.52.	LOOP	275

7.6.53.	MEM_CONVERT	277
7.6.54.	MEM_CPY	279
7.6.55.	MEM_SET	281
7.6.56.	MEM_WRITE_RESP_SND	283
7.6.57.	MODEM_CONTROL	284
7.6.58.	QUEUE_CLEAR	286
7.6.59.	QUEUE_CREATE	287
7.6.60.	QUEUE_RCV_FROM	288
7.6.61.	QUEUE_SND_TO	289
7.6.62.	REMARK	290
7.6.63.	RESET	292
7.6.64.	SECS_MC_FRAME_SND	294
7.6.65.	SLAVE_MEM_READ_WRITE	296
7.6.66.	STR_TO_INT	298
7.6.67.	SWAP	300
7.6.68.	TIME_SETTING	302
7.6.69.	TIME_STR	304
7.6.70.	TYPE_CONVERT	306
7.7.	단종 아이템	308
8.	시리얼 통신	310
8.1.	포트 설정	310
8.1.1.	시리얼 통신 설정	312
8.1.2.	내장 프로토콜 설정	313
8.1.3.	시리얼 통신 고급 설정	314
8.2.	MODBUS MASTER	316
8.2.1.	프로토콜 상세 설정	316
8.2.2.	MODBUS Master 아이템	317
8.2.3.	읽기/쓰기 최대 길이	317
8.2.4.	프로그램 예제	318
8.3.	MODBUS SLAVE	319
8.3.1.	프로토콜 상세 설정	319
8.3.2.	ETOS Address Mapping	320
8.3.3.	지원 Function Code	321
8.3.4.	읽기/쓰기 최대 길이	321
8.4.	GLOFA MASTER	322
8.4.1.	프로토콜 상세 설정	322
8.4.2.	GLOFA Cnet 마스터 아이템	322
8.4.3.	읽기/쓰기 최대 길이	322
8.4.4.	프로그램 예제	323
8.5.	GLOFA SLAVE	324
8.5.1.	프로토콜 상세 설정	324
8.5.2.	지원 명령	325
8.6.	사용자 정의 프로토콜 (USER DEFINED)	326
8.6.1.	포트 설정	326
8.6.2.	프레임 정의	326
8.6.3.	사용자 정의 프로토콜 아이템	327
8.6.4.	프로그램 예제	328
9.	이더넷 통신	330
9.1.	개요	330
9.1.1.	이더넷 드라이버 설정	331
9.1.2.	이더넷 채널 설정	331
9.1.3.	이더넷 고속링크 설정	331
9.2.	이더넷 드라이버	332
9.2.1.	MODBUS/TCP	334
9.2.2.	GLOFA Enet	338

- 9.2.3. ETOS 전용 드라이버339
- 9.3. 이더넷 채널 340
 - 9.3.1. MODBUS/TCP 마스터342
 - 9.3.2. GLOFA Enet 마스터344
 - 9.3.3. MELSEC 마스터346
 - 9.3.4. 사용자 정의 프로토콜 마스터/슬레이브 (User Defined)348
- 9.4. 이더넷 고속링크 353
 - 9.4.1. 데이터 송, 수신 원리353
 - 9.4.2. 이더넷 고속링크 설정354
- 10. SECS 통신 358**
 - 10.1. 개요 358
 - 10.2. SECS1 통신 358
 - 10.2.1. 10초마다 S1F1을 송신하고, S1F2를 받기358
 - 10.2.2. S1F1을 받으면 S1F2로 응답하기367
 - 10.2.3. S1F1을 받는 프로세스와 S1F2로 응답하는 프로세스 분리하기374
 - 10.3. HSMS 통신 378
 - 10.3.1. 10초마다 S1F1을 송신하고, S1F2를 받기378
 - 10.4. BRIDGE 통신 381
 - 10.4.1. 시스템의 구성381
 - 10.4.2. ETOS-PD를 이용한 파라미터의 설정382
 - 10.5. 진단 시스템 통신 386
 - 10.5.1. SECS1을 HSMS로 모니터와 진단하기386
 - 10.5.2. HSMS를 HSMS로 모니터와 진단하기388
- 11. 이중화 394**
 - 11.1. 개요 394
 - 11.2. H/W 구성 394
 - 11.2.1. 시스템 구성394
 - 11.2.2. DIP Switch 설정396
 - 11.2.3. LED 정의397
 - 11.3. CPU 이중화 398
 - 11.4. 네트워크 이중화 401
 - 11.5. 전원 이중화 401
- 부록 404**
 - A. 외형도 404
 - A.1. ETOS-50404
 - A.2. ETOS-100A/150405
 - A.3. ETOS-200406
 - A.4. ETOS-1000407
 - B. 에러코드 408
 - B.1. MELSEC 에러 코드408
 - B.2. GLOFA 에러 코드409
 - B.3. MODBUS 에러 코드410
 - B.4. ETOS 에러 코드411

그림 목차

그림 2.5-1 ETOS-100A 전면 패널	17
그림 2.5-2 ETOS-150 전면 패널	19
그림 3.1-1 ETOS-50A 시리얼 모델 후면 패널	28
그림 3.1-2 ETOS-50A 이더넷 모델 후면 패널	29
그림 3.2-1 ETOS-100A/150 후면 패널	31
그림 3.3-1 RS232C 전용모뎀 결선도 (2선식)	36
그림 3.3-2 RS232C 전용모뎀 결선도 (4선식)	36
그림 3.3-3 RS232C 널모뎀 결선도	37
그림 3.3-4 ETOS-50A/100A/150 RS422 결선도	38
그림 3.3-5 RS422 종단저항 연결 방법	40
그림 3.3-6 RS485 종단저항 연결 방법	40
그림 3.4-1 ETOS-200/1000 CPU 전면 패널	41
그림 3.4-2 ETOS-200/1000 시리얼 통신 모듈 전면 패널	43
그림 3.4-3 RS232C 전용모뎀 결선도 (2선식)	46
그림 3.4-4 RS232C 전용모뎀 결선도 (4선식)	46
그림 3.4-5 RS232C 널모뎀 결선도	47
그림 3.4-6 ETOS-200/1000 RS422 결선도	48
그림 3.4-7 ETOS-200/1000 RS485 결선도	49
그림 4.1-1 ETOS-PD 작업 순서도	53
그림 4.3-1 ETOS-PD의 기본 화면	54
그림 4.3-2 ETOS-PD 화면 구성	55
그림 4.3-3 프레임 보기	59
그림 4.3-4 편집 옵션 대화상자	60
그림 4.3-5 온라인 연결 설정 화면	61
그림 4.3-6 리모트 연결 설정	62
그림 4.3-7 원격지 ETOS의 리모트 연결 설정	63
그림 4.3-8 편집내용 검사	64
그림 4.3-9 편집내용 검사 결과 화면	64
그림 4.3-10 파라미터 쓰기 메뉴	65
그림 4.3-11 파라미터 쓰기 결과 화면	65
그림 4.3-12 프로그램 쓰기 메뉴	66
그림 4.3-13 프로그램 쓰기 결과 화면	66
그림 4.3-14 파라미터 읽기 메뉴	67
그림 4.3-15 파라미터 읽기 결과 화면	67
그림 4.3-16 프로그램 읽기 화면	68
그림 4.3-17 프로그램 읽기 결과 화면	68
그림 4.3-18 시스템 정보 화면	69
그림 4.3-19 메모리 저장 화면	69
그림 4.3-20 OS 업데이트 화면	70
그림 4.3-21 파라미터, 프로그램 지우기 결과 화면	71
그림 4.3-22 암호설정 및 변경 화면	74
그림 4.3-23 암호 지우기 화면	74
그림 4.3-24 Shared Memory 설정	75
그림 4.3-25 IFOR Direct Pass 설정	76
그림 4.3-26 IFOR Staus 창	77
그림 4.3-27 IFOR Staus 세부정보	77
그림 4.3-28 HSMS Diagnostic Status 읽기 화면	78
그림 4.3-29 프레임 모니터링	79
그림 4.3-30 메모리 출력 창	81
그림 4.3-31 ETOS 메모리 체계	81
그림 5.1-1 파라미터 설정 화면	84
그림 5.2-1 파라미터 설정 순서	85
그림 5.2-2 새파일 대화상자	86
그림 5.3-1 Serial Only 시스템 설정	89
그림 5.3-2 포트 종류 선택 화면	90
그림 5.3-3 COM 포트 설정 예	91
그림 5.3-4 I/O 파라미터 설정 예	92

그림 5.3-5 SECS-I 설정	93
그림 5.4-1 이더넷 파라미터	94
그림 5.4-2 이더넷 드라이버 사용 예	95
그림 5.4-3 이더넷 드라이버 설정 창	95
그림 5.4-4 드라이버 포트 설정 예	97
그림 5.4-5 이더넷 채널 사용 예	98
그림 5.4-6 이더넷 채널 설정 화면	98
그림 5.4-7 이더넷 채널 포트 설정 화면	100
그림 5.4-8 HSMS 설정 화면	102
그림 5.4-9 HSMS Port 설정 화면	103
그림 6.2-1 프레임 구성 예	107
그림 6.3-1 프레임 창	107
그림 6.3-2 그룹 추가 (ETOS-200/1000)	108
그림 6.3-3 그룹 추가 (ETOS-50A/100A/150)	108
그림 6.3-4 그룹 편집	109
그림 6.3-5 수신 그룹 등록 예	109
그림 6.3-6 프레임 추가	110
그림 6.3-7 프레임 편집 창	110
그림 6.3-8 프레임 등록 예	111
그림 6.3-9 실제 수신 프레임	111
그림 6.3-10 세그먼트 추가	112
그림 6.3-11 수신 프레임의 수치 상수 세그먼트 추가	113
그림 6.3-12 송신 프레임의 수치 상수 세그먼트 추가	113
그림 6.3-13 수신 프레임의 문자열 상수 세그먼트 추가	114
그림 6.3-14 송신 프레임의 문자열 상수 세그먼트 추가	114
그림 6.3-15 수신 프레임의 고정 크기 변수 세그먼트 추가	115
그림 6.3-16 송신 프레임의 고정 크기 변수 세그먼트 추가	115
그림 6.3-17 수신 프레임에 가변 크기 변수 세그먼트 추가	116
그림 6.3-18 송신 프레임에 가변 크기 변수 세그먼트 추가	116
그림 6.3-19 메모리 지정 옵션	117
그림 6.3-20 BCC 추가 메뉴	119
그림 6.3-21 BCC 설정 창	119
그림 6.3-22 BCC 영역 설정	121
그림 6.3-23 BCC 끝 위치 설정	121
그림 6.3-24 영역 끝을 끝 위치로 설정	122
그림 6.3-25 세그먼트 번호로 끝 위치를 설정	122
그림 6.3-26 세그먼트 번호	122
그림 6.4-1 송신 그룹 추가 및 편집	124
그림 6.4-2 송신 프레임 HEAD입력	124
그림 6.4-3 송신 세그먼트 추가	125
그림 6.4-4 송신 프레임 HEAD	125
그림 6.4-5 송신 TAIL 프레임 등록	125
그림 6.4-6 송신 TAIL 세그먼트 추가	126
그림 6.4-7 송신 프레임 BCC 설정	126
그림 6.4-8 송신 프레임 TAIL	126
그림 6.4-9 송신 프레임 BODY	127
그림 6.4-10 송신 BODY 프레임 등록	127
그림 6.4-11 세그먼트 추가 (국번, BODY)	127
그림 6.4-12 세그먼트 추가 (명령어, BODY)	128
그림 6.4-13 세그먼트 추가 (데이터, BODY)	128
그림 6.4-14 송신 프레임 BODY	128
그림 6.4-15 수신 그룹 추가 및 편집	129
그림 6.4-16 수신 프레임 HEAD입력	129
그림 6.4-17 수신 세그먼트 추가	130
그림 6.4-18 수신 TAIL 세그먼트 추가	130
그림 6.4-19 수신 프레임 BCC 설정	131
그림 6.4-20 수신 프레임 TAIL	131
그림 6.4-21 수신 프레임 BODY	132

그림 6.4-22 세그먼트 추가 (국번, BODY).....	132
그림 6.4-23 세그먼트 추가 (명령어, BODY).....	132
그림 6.4-24 세그먼트 추가 (데이터, BODY).....	133
그림 6.4-25 수신 프레임 BODY.....	133
그림 6.4-26 프레임 등록 결과.....	133
그림 6.5-1 SECS 메시지 편집 창.....	134
그림 6.5-2 SECS 메시지 추가.....	134
그림 6.5-3 SECS 메시지.....	135
그림 6.5-4 SECS-II Message Definition.....	136
그림 6.5-5 SECS 메시지 편집 팝업 창.....	138
그림 6.5-6 주소 수정.....	138
그림 6.6-1 SECS 메시지 추가.....	139
그림 6.6-2 SECS-II Message Definition.....	139
그림 6.6-3 정의된 SECS 메시지.....	139
그림 6.6-4 SECS 메시지 추가.....	140
그림 6.6-5 SECS-II Message Definition.....	140
그림 6.6-6 SECS-II Message Definition.....	141
그림 6.6-7 SECS-II Message Definition.....	142
그림 6.6-8 SECS-II Message Definition.....	143
그림 6.6-9 SECS-II Message Definition.....	144
그림 6.6-10 SECS-II Message Definition.....	145
그림 6.6-11 SECS-II Message Definition.....	147
그림 6.6-12 SECS-II Message Definition.....	148
그림 6.6-13 SECS-II Message Definition.....	149
그림 6.6-14 SECS-II Message Definition.....	150
그림 6.6-15 ETOS 메모리.....	151
그림 7.1-1 프로세스 등록 창.....	154
그림 7.2-1 프로세스 추가.....	155
그림 7.2-2 프로세스 편집 창.....	155
그림 7.2-3 프로세스 편집 메뉴.....	156
그림 7.3-1 예제 프로세스 추가.....	157
그림 7.3-2 시작 아이템 추가.....	158
그림 7.3-3 정주기 프로세스.....	158
그림 7.3-4 ARITHMETIC 아이템 삽입.....	159
그림 7.3-5 ARITHMETIC 아이템을 이용한 국번 증가.....	159
그림 7.3-6 IF_NUM_CMP 아이템 삽입.....	160
그림 7.3-7 IF_NUM_CMP 아이템을 이용한 국번 비교.....	160
그림 7.3-8 MEM_SET 아이템 삽입.....	161
그림 7.3-9 MEM_SET 아이템을 이용한 국번 초기화.....	161
그림 7.3-10 IF_REQ_RESP 아이템 추가.....	162
그림 7.3-11 IF_REQ_RESP아이템을 이용한 송,수신 프레임 설정.....	162
그림 7.4-1 IF 조건문 구성 예.....	164
그림 7.4-2 IF 조건문 구성 예.....	164
그림 7.4-3 IF 아이템 중복사용 예.....	164
그림 7.6-1 ARITHMETIC입력 예.....	174
그림 7.6-2 ASC_TO_BIN 입력 예.....	176
그림 7.6-3 ASC_TO_HEX 입력 예.....	178
그림 7.6-4 ASC_TO_BIN 입력 예.....	180
그림 7.6-5 BIT_MASK 입력 예.....	182
그림 7.6-6 BIT_WRITE 입력 예.....	184
그림 7.6-7 BIT_WRITE_EX 입력 예.....	186
그림 7.6-8 주소 간접 지정.....	187
그림 7.6-9 BITS_CPY 입력 예.....	189
그림 7.6-10 BITS_OPERATION 입력 예.....	191
그림 7.6-11 DELAY 입력 예.....	193
그림 7.6-12 ELSE 아이템.....	195
그림 7.6-13 ENCRYPTION 입력 예.....	197
그림 7.6-14 END 아이템.....	198

그림 7.6-15 END_IF 아이템	200
그림 7.6-16 END_LOOP 아이템	202
그림 7.6-17 ENET_CLOSE 입력 예	203
그림 7.6-18 ENET_DISABLE 입력 예	204
그림 7.6-19 ENET_ENABLE 입력 예	205
그림 7.6-20 ENET_OPEN 입력 예	206
그림 7.6-21 EVENT_READ_WRITE 입력 예	207
그림 7.6-22 FRAME_SND 입력 예	209
그림 7.6-23 HEX_TO_ASC 입력 예	211
그림 7.6-24 HEX_TO_BCD 입력 예	213
그림 7.6-25 IF_3964R_READ 입력 예	215
그림 7.6-26 IF_3964R_WRITE 입력 예	216
그림 7.6-27 IF_BIT_FIND 입력 예	218
그림 7.6-28 IF_BIT_TEST 입력 예	220
그림 7.6-29 IF_EVENT_OCCUR 입력 예	222
그림 7.6-30 IF_FRAME_RCV 입력 예	224
그림 7.6-31 IF_GLOFA_READ 입력 예	227
그림 7.6-32 IF_GLOFA_WRITE	229
그림 7.6-33 IF_IO_STATE_CHANGE	231
그림 7.6-34 IF_MELSEC_READ 입력 예	233
그림 7.6-35 IF_MELSEC_WRITE 입력 예	234
그림 7.6-36 IF_MEM_CMP 입력 예	235
그림 7.6-37 IF_MEM_WRITE_RCV 동작 원리	238
그림 7.6-38 IF_MEM_WRITE_RCV 입력 예	239
그림 7.6-39 IF_MODBUS_READ 입력 예	242
그림 7.6-40 IF_MODBUS_WRITE 입력 예	245
그림 7.6-41 IF_NUM_CMP 입력 예	247
그림 7.6-42 IF_POWER_ON 아이템	249
그림 7.6-43 IF_REQ_RESP 입력 예	251
그림 7.6-44 IF_SECS_MC_FRAME_RCV 입력 예	254
그림 7.6-45 IF_SECS_MC_REQ_RESP 입력 예	256
그림 7.6-46 IF_STR_CMP 입력 예	258
그림 7.6-47 IF_TIMER 입력 예	260
그림 7.6-48 INT_TO_STR 입력 예	262
그림 7.6-49 IO_READ_WRITE 입력 예	265
그림 7.6-50 IP_SAVE 입력 예	267
그림 7.6-51 JUMP 입력 예	269
그림 7.6-52 LABEL 입력 예	271
그림 7.6-53 LOOKUP_TABLE 동작 개념	272
그림 7.6-54 LOOKUP_TABLE 입력 예	273
그림 7.6-55 LOOP 입력 예	275
그림 7.6-56 MEM_CONVERT 입력 예	277
그림 7.6-57 MEM_CPY 입력 예	279
그림 7.6-58 MEM_SET 입력 예	281
그림 7.6-59 MEM_WRITE_RESP_SND 아이템	283
그림 7.6-60 MODEM_CONTROL 입력 예	284
그림 7.6-61 QUEUE_CLEAR 입력 예	286
그림 7.6-62 QUEUE_CREATE 입력 예	287
그림 7.6-63 QUEUE_RCV_FROM 입력 예	288
그림 7.6-64 QUEUE_SND_TO 입력 예	289
그림 7.6-65 REMARK 입력 예	290
그림 7.6-66 RESET 아이템	292
그림 7.6-67 SECS_MC_FRAME_SND 입력 예	294
그림 7.6-68 SLAVE_MEM_READ_WRITE 입력 예	296
그림 7.6-69 STR_TO_INT 입력 예	298
그림 7.6-70 SWAP 입력 예	300
그림 7.6-71 TIME_SETTING 입력 예	302
그림 7.6-72 TIME_STR 입력 예	304

그림 7.6-73 TYPE_CONVERT 입력 예	306
그림 8.1-1 시리얼 통신 시스템 구성 예	310
그림 8.1-2 포트 종류 선택(Serial)	311
그림 8.1-3 COM 포트 설정 예	312
그림 8.1-4 COM 포트 프로토콜 설정	313
그림 8.1-5 시리얼 통신 고급 설정 예	314
그림 8.1-6 RS422/485 제어방법 선택	315
그림 8.2-1 MODBUS Master 포트 사용 예	316
그림 8.2-2 MODBUS Master 상세설정	316
그림 8.2-3 MODBUS ASCII 모드 프레임 구조	317
그림 8.2-4 MODBUS RTU 모드 프레임 구조	317
그림 8.2-5 프로그램 예	318
그림 8.3-1 MODBUS Slave 포트 사용 예	319
그림 8.3-2 MODBUS Slave 상세설정	319
그림 8.4-1 GLOFA Master 포트 사용 예	322
그림 8.4-2 프로그램 예	323
그림 8.5-1 GLOFA Slave 포트 사용 예	324
그림 8.5-2 GLOFA Slave 상세설정	324
그림 8.6-1 사용자정의 프로토콜 사용 예	326
그림 8.6-2 그룹 등록 예	326
그림 8.6-3 프레임 등록	327
그림 8.6-4 시리얼 프레임 등록 예	327
그림 8.6-5 프로그램 예	328
그림 9.1-1 새파일 설정	330
그림 9.1-2 파라미터 설정 화면	330
그림 9.2-1 이더넷 드라이버 사용 예	332
그림 9.2-2 이더넷 드라이버 설정 대화상자	333
그림 9.2-3 이더넷 드라이버 설정 예	333
그림 9.2-4 MODBUS 드라이버 포트 설정	334
그림 9.2-5 MODBUS Slave 상세설정 예	335
그림 9.2-6 GLOFA Enet 드라이버 포트 설정	338
그림 9.2-7 ETOS 이더넷 전용 드라이버 포트 설정	339
그림 9.3-1 이더넷 채널 사용 예	340
그림 9.3-2 이더넷 채널 설정 대화상자	340
그림 9.3-3 MODBUS 채널 포트 설정	342
그림 9.3-4 MODBUS Master 상세 설정	343
그림 9.3-5 GLOFA 채널 포트 설정	344
그림 9.3-6 MELSEC 채널 포트 설정	346
그림 9.3-7 MELSEC Master 상세설정	347
그림 9.3-8 사용자 정의 프로토콜 채널 포트 설정	348
그림 9.3-9 그룹 등록 예	349
그림 9.3-10 프레임 등록	349
그림 9.3-11 이더넷 프레임 등록 예	350
그림 9.3-12 TCP/IP 데이터 통신	351
그림 9.3-13 프로그램 예	352
그림 9.4-1 이더넷 파라미터	354
그림 9.4-2 이더넷 고속링크 설정 대화상자	354
그림 9.4-3 고속링크 항목 설정	356
그림 10.2-1 ETOS 종류 및 드라이버 선택	358
그림 10.2-2 파라미터 설정	359
그림 10.2-3 포트 종류 선택	359
그림 10.2-4 COM 포트 설정 (SECS-I)	360
그림 10.2-5 파라미터 설정	360
그림 10.2-6 SECS 메시지 등록	361
그림 10.2-7 SECS-II Message Definition	361
그림 10.2-8 SECS 메시지	362
그림 10.2-9 SECS-II Message Definition	362
그림 10.2-10 SECS-II Message Definition	363

그림 10.2-11 등록된 SECS 메시지	363
그림 10.2-12 프로세스 등록	364
그림 10.2-13 아이템 추가	364
그림 10.2-14 아이템 추가	365
그림 10.2-15 아이템 삽입	365
그림 10.2-16 IF_SECS_MC_REQ_RESP	366
그림 10.2-17 추가된 아이템	366
그림 10.2-18 파라미터 설정	367
그림 10.2-19 SECS 메시지 추가	368
그림 10.2-20 SECS 메시지 추가	368
그림 10.2-21 SECS-II Message Definition	369
그림 10.2-22 SECS-II Message Definition	370
그림 10.2-23 추가된 SECS 메시지	370
그림 10.2-24 프로세스 편집	371
그림 10.2-25 IF_SECS_MC_FRAME_RCV	371
그림 10.2-26 SECS_MC_FRAME_SND	372
그림 10.2-27 등록된 프로세스	372
그림 10.2-28 ETOS 메모리	373
그림 10.2-29 프레임 모니터	373
그림 10.2-30 프로세스 복사	374
그림 10.2-31 프로세스 이름 변경	374
그림 10.2-32 프로세스 편집	375
그림 10.2-33 프로세스 Disable	375
그림 10.2-34 아이템 삭제	375
그림 10.2-35 IF_SECS_MC_FRAME_RCV 편집	376
그림 10.2-36 순서도	376
그림 10.2-37 S1F1_RCV_PRS	377
그림 10.2-38 S1F2_SND_PRS	377
그림 10.2-39 ETOS 메모리	377
그림 10.3-1 시스템 설정	378
그림 10.3-2 HSMS 설정	378
그림 10.3-3 HSMS Port 설정	379
그림 10.3-4 HSMS 설정	379
그림 10.3-5 프로세스의 수정	380
그림 10.4-1 Bridge 통신	381
그림 10.4-2 Bridge 통신의 개념	381
그림 10.4-3 ETOS 종류 및 드라이버 선택	382
그림 10.4-4 파라미터 설정	382
그림 10.4-5 HSMS 설정	383
그림 10.4-6 HSMS Port 설정	383
그림 10.4-7 SECS-I 포트 추가	384
그림 10.4-8 SECS-I 설정	385
그림 10.5-1 SECS-I 시스템의 진단	386
그림 10.5-2 SECS-I 시스템의 진단 개념	386
그림 10.5-3 HSMS 시스템의 진단	388
그림 10.5-4 HSMS 시스템의 진단 개념	388
그림 10.5-5 HSMS Diagnostic	390
그림 10.5-6 HSMS Diagnostic Info	390
그림 10.5-7 HSMS Diagnostic Status	391
그림 11.2-1 ETOS-1000 이중화 모델 정면도	394
그림 11.2-2 ETOS-1000 이중화 모델의 후면 기구 구성도	395
그림 11.3-1 이중화 파라미터 설정 예	399
그림 11.3-2 이중화 시스템 정보 저장 예	399
그림 11.3-3 동기 영역 설정 예	400
그림 11.4-1. ETOS-1000 네트워크 이중화 시스템 구성도	401

Notations

ETOS 사용설명서에서 사용하는 기호의 종류와 의미입니다.

{프로세스 등록 창} - 프로그램/창(Window)/대화상자(Dialog box)의 제목을 표시합니다.

[온라인], [파라미터 쓰기], [확인], [취소] - 메뉴 또는 버튼 명령과 같이 마우스로 선택을 해야 하는 경우를 표시합니다.

<통신형태>, <통신속도>, <데이터 비트> - 사용자가 설정해야 하는 입력값의 제목을 나타냅니다.

“사용자 설정값” - 사용자가 키보드로 입력한 값을 나타냅니다.

‘사용자 선택값’ - 사용자가 여러 선택값 중 입력값으로 선택한 값을 나타냅니다.

/ - 사용자가 선택할 수 있는 입력값을 나열할 때 사용합니다.

내용소개

[1. 개요](#)

ETOS 시리즈의 특징을 위주로 제품 전반에 대해 소개합니다.

[2. 제품 규격](#)

일반 성능, 전원, 통신 케이블 규격 등을 다룹니다.

[3. 설치 및 배선](#)

ETOS 시리즈의 전원 규격 및 통신 포트 규격, 통신 포트 핀 규격을 소개합니다.

[4. ETOS-PD 사용법](#)

시스템 설정 및 프로그래밍을 위해 제공되는 Windows software 인 ETOS-PD(Protocol Designer) 사용법에 대해 다룹니다.

[5. 파라미터](#)

ETOS 프로그램의 세가지 요소인 파라미터 설정, 프레임 등록, 프로세스의 개념을 설명합니다.

[6. 프레임](#)

사용자 정의 프레임 등록 방법을 살펴봅니다.

[7. 프로세스 및 아이템](#)

프로세스의 개념과 프로세스를 구성하는 단위인 아이템에 대해 설명합니다.

[8. 시리얼 통신](#)

ETOS의 시리얼 통신 기능 사용법을 살펴봅니다.

[9. 이더넷 통신](#)

ETOS의 이더넷 통신 기능 사용법을 살펴봅니다.

[10. SECS 통신](#)

SECS 통신 기능 사용법을 살펴봅니다.

[11. 이중화](#)

ETOS의 이중화 기능에 대해 살펴봅니다.

부록

제품 외형도 및 에러코드

개 요

제 1 장

이장은 ETOS 의 기능적 특징 등을 소개하고 있습니다.

	내용	위치
1.1	제품 개요	2
1.2	제품의 기능 및 특징	2
1.3	시리즈 구성	4

1. 개요

1.1. 제품 개요

ETOS (Ethernet To Serial Gateway)는 이더넷(Ethernet), 시리얼(Serial), 필드버스(Fieldbus)를 사용하는 다양한 기기종 장비들간의 데이터 통신을 가능하게 하는 프로그래머블 통신중계장치 (Programmable Gateway, 이하: 게이트웨이)입니다. 이더넷 대 시리얼, 이더넷 대 필드버스, 시리얼 대 시리얼, 이더넷 대 이더넷 등의 미디어 변환 및 프로토콜 변환 기능을 통해 산업 현장의 다양한 기기종 장비들을 효과적으로 통합할 수 있습니다.

1.2. 제품의 기능 및 특징

ETOS 는 다음과 같은 통신 기능을 가지고 있습니다.

▶ 이더넷 통신 기능(10/100Base-T)

- * TCP/IP, UDP/IP, DHCP 지원
- * 이더넷 전용 프로토콜 내장 (MODBUS/TCP, GLOFA ENET, MELSEC-Q)
- * 이더넷 사용자 정의 프로토콜 편집 기능

▶ 범용 시리얼 통신 기능(RS232C/RS422/RS485)

- * 시리얼 전용 프로토콜 내장 (MODBUS RTU/ASCII, GLOFA CNET, Siemens 3964(R))
- * 시리얼 사용자 정의 프로토콜 편집 기능

▶ 필드버스 통신 기능

- * Profibus
- * GLOFA-Fnet

▶ SECS 통신 기능

- * 반도체 장비용 통신 규격인 SECS-I,II,HSMS 지원.

통신 기능 이외에도 다음과 같은 기능 및 특징을 가지고 있습니다.

- * 통신 데이터의 저장 및 변환을 위한 사용자 메모리 제공
- * 통신 프레임 모니터 및 사용자 메모리 모니터 기능
- * 기기종 장비간의 프로토콜 변환을 위한 프로그램 기능
- * 다양한 프레임 에러 확인 기능(Checksum/CRC/LRC 등)
- * 데이터 형변환 기능 및 swap 기능
- * 널 모뎀/전용 모뎀/다이얼 업 모뎀/무선 모뎀/CDMA 모뎀 연결 기능
- * 이더넷/CPU/Power 이중화 (ETOS-1000 이중화 모델)
- * 상용 Real Time O/S(RTOS) 채택하여 신뢰성 확보.

사용자는 한가지 이상의 통신 기능을 조합하여 서로 다른 미디어 및 프로토콜을 사용하는 기기종 장비들을 통합할 수 있습니다. 시리얼 및 이더넷 내장 프로토콜은 사용자로 하여금 간단한 시스템 설정만으로 해당 프로토콜을 사용하는 장비와 통신이 가능하도록 하며, 오픈 프로토콜이 아닌 장비 제작사 고유의 프로토콜을 사용하는 장비들도 ETOS 의 강력한 프로토콜 편집기능에 의해 통신이 가능합니다.

또한 ETOS 는 프로토콜 편집 기능뿐 아니라, 이기종 프로토콜 간의 데이터 교환을 위해 유연한 프로그래밍 언어를 제공합니다.

ETOS 는 이와 같이 파라미터 설정, 프레임 및 프로세스 편집 기능을 통해 사용자가 요구하는 다양한 통합 통신 시스템 구축을 가능하게 합니다.

이러한 시스템 설정 및 프로그래밍을 위해 ETOS-PD(Protocol Designer)라는 Windows Software 가 제공됩니다. ETOS-PD 의 기능 및 사용 방법 등의 세부 사항은 [4. ETOS-PD 사용법](#)을 참고하시기 바랍니다.

1.3. 시리즈 구성

ETOS 시리즈는 50A/100A/150/200/1000 로 구성되어 있으며 각 제품에는 지원하는 통신 기능에 따라 다음표와 같이 여러 가지의 모델이 있습니다.

제품명	모델명	구분	규격
ETOS-1000 (*1)	ETOS-1000-RCP	CPU	이중화 CPU, 이더넷 1 포트
	ETOS-1000-SCP	CPU	단중화 CPU, 이더넷 1 포트
	ETOS-1000-S40	옵션카드	RS232 4 포트
	ETOS-1000-S80	옵션카드	RS232 8 포트
	ETOS-1000-S04	옵션카드	RS422 4 포트
	ETOS-1000-S08	옵션카드	RS422 8 포트
	ETOS-1000-S44	옵션카드	RS232 4 포트 + RS422 4 포트
	ETOS-1000-PSR	전원	이중화 전원
	ETOS-1000-PSS	전원	단중화 전원
	ETOS-1000-BAS	케이스	19" Rack Case
ETOS-200 (*1)	ETOS-200-E88	본체 + 옵션카드	이더넷 1 포트, RS232 8 포트, RS422 8 포트
	ETOS-200-E44	본체 + 옵션카드	이더넷 1 포트, RS232 4 포트, RS422 4 포트
	ETOS-200-EA6	본체 + 옵션카드	이더넷 1 포트, RS232 10 포트, RS422 6 포트
	ETOS-200-HSMS	본체 (+ 옵션카드)	이더넷 1 포트, RS232/RS422 최대 16 포트, SECS-I to HSMS 변환,
	ETOS-200-Sgate	본체 (+ 옵션카드)	
ETOS-150	ETOS-150-HSMS	본체	이더넷 1 포트, RS232 2 포트(최대 4 포트), SECS-I to HSMS 변환
	ETOS-150-Sgate	본체	
ETOS-100A (*1)	ETOS-100A-S20	본체	RS232 2 포트
	ETOS-100A-S40	본체	RS232 4 포트
	ETOS-100A-S22	본체	RS232 2 포트, RS422 2 포트
	ETOS-100A-E20	본체	이더넷 1 포트, RS232 2 포트
	ETOS-100A-E40	본체	이더넷 1 포트, RS232 4 포트
	ETOS-100A-E22	본체	이더넷 1 포트, RS232 2 포트, RS422 2 포트
ETOS-50A (*1)	ETOS-50A-S20	본체	RS232 2 포트
	ETOS-50A-S02	본체	RS422 2 포트
	ETOS-50A-S11 ^(*2)	본체	RS232 1 포트(COM2), RS422 1 포트(COM1)
	ETOS-50A-E10	본체	이더넷 1 포트, RS232 1 포트
	ETOS-50A-E01	본체	이더넷 1 포트, RS422 1 포트
	ETOS-50A-Sgate	본체	이더넷 1 포트, RS232 1 포트, SECS-I to HSMS
Option Module (*3)	OPT-232	옵션카드	RS232 2 포트
	OPT-422	옵션카드	RS422 2 포트
	OPT-Fnet	옵션카드	GLOFA Fnet
	OPT-Pnet	옵션카드	Profibus-DP master
	OPT-IN-16	옵션카드	DC 24V DI 16 점
	OPT-Out-12	옵션카드	Relay DO 12 점
	OPT-HYB-14	옵션카드	DC 24V DI 8 점, Relay DO 6 점

(*1) RS232 포트와 RS422/485 포트의 수는 최대 지원 포트 수 내에서 자유롭게 구성할 수 있습니다

(*2) ETOS-50A-S11 은 COM1 포트가 RS422/485 로 COM2 포트가 RS232 로 고정됩니다. 제품 전면의 라벨을 참고하십시오.

(*3) 옵션 모듈은 ETOS-100A/150 시리즈에만 장착 가능합니다

1.3.1. 최대 지원 포트 수

제품 별로 지원하는 최대 시리얼 및 이더넷 포트 수는 다음과 같습니다.

제품	이더넷(10/100Base-T)	시리얼(RS232C/RS422)
ETOS-1000	1	64
ETOS-200	1	16
ETOS-150	1	4
ETOS-100A	1	4
ETOS-50A	1	2

- * ETOS-50A 는 이더넷 1 포트와 시리얼 2 포트의 구성은 불가능하며, 시리얼 2 포트 또는 시리얼 1 포트, 이더넷 1 포트의 구성만 가능합니다.
- * ETOS-100A/150/200/1000 은 이더넷 1 포트와 시리얼 최대 포트구성이 가능합니다.
- * 이더넷 포트 수는 물리적인 포트 수를 말하며, 연결 가능한 connection 수와는 다른 개념입니다.

1.3.2. 모델명 부여 방법

모델명은 다음과 같이 부여합니다.

ETOS-[A]-[B][C][D]

[A] : 제품명 예) 50A,100A,150,200,1000

[B] : S (시리얼 포함) , E (이더넷 포함)

[C] : RS232C 포트 수, 16 진수로 표기 예) A - 10 포트, C - 12 포트

[D] : RS422/485 포트 수, 16 진수로 표기

- * ETOS-50A 는 시리얼 포트가 최대 2 포트까지 지원됩니다. ETOS-50 을 RS232C 1 포트와 RS422 1 포트로 구성하면, 모델명 부여 방법에 따라 ETOS-50A-S11 이 됩니다.

1.3.3. H/W 사양

ETOS 제품 별로 주요 H/W 사양을 비교합니다.

1.3.3.1. ETOS-50A

ETOS-50A 는 통신 포트 구성에 따라 다음과 같은 모델이 있습니다. ETOS-50A 는 시리얼 2 포트 또는 시리얼 1 포트, 이더넷 1 포트의 구성이 가능하며 옵션 카드는 추가 장착할 수 없습니다.

항 목	ETOS-50A		
	Serial	Ethernet	SECS gate
모델명	ETOS-50A-SXX	ETOS-50A-EXX	ETOS-50A-Sgate
사용자 메모리 (*1)	64KBytes	64KBytes	64KBytes
사용자 메모리 백업	-	-	-
프로그램 메모리(FLASH) (*2)	1.0MBytes	1.0MBytes	1.0Mbytes
로그 메모리(*3)	256KBytes	256KBytes	256KBytes
로그 메모리 백업	-	-	-
로그 메모리 위치	0x1740000	0x1740000	0x1740000
이더넷	-	10/100Base-T 1 포트	10/100Base-T 1 포트
시리얼	2 포트	1 포트	1 포트
Profibus	-	-	-
I/O	-	-	-
WatchDog	-	-	-
RTC	-	-	-
이중화 구성	-	-	-
드라이버(*4)	Serial Only	지원	지원
	Ethernet	-	지원
	SECS	-	지원

(*1) 사용자 메모리 : 사용자가 작성한 프레임 및 프로세스가 사용하는 영역입니다. 사용자가 프로그램 내에서 참조하는 영역입니다

(*2) 프로그램 메모리 : 사용자가 설정한 파라미터와 프레임, 프로세스 정의가 저장되는 시스템 메모리 영역입니다. 사용자가 프로그램 내에서 참조할 수 없습니다

(*3) ETOS 동작 중에 발생한 중요 이벤트를 텍스트로 저장해두는 메모리입니다. [온라인] -> [메모리 저장]에서 시스템 메모리를 체크하면 파일로 저장할 수 있습니다.

(*4) ETOS 에서 사용하는 프로그램 파일은 ETOS 가 지원하는 통신 기능에 따라 몇 가지의 드라이버로 구분됩니다

1.3.3.2. ETOS-100A / 150 / 200 / 1000

ETOS-100A/150/200/1000 의 H/W 사양을 비교합니다.

제품		ETOS-100A	ETOS-150	ETOS-200	ETOS-1000
항 목					
사용자 메모리 ^(*1) (백업 영역)		256KBytes (128KBytes)	256KBytes (128KBytes)	512KBytes (256KBytes)	512KBytes (256KBytes)
사용자 메모리 백업 ^(*2)		지원	지원	지원	지원
프로그램 메모리(FLASH) ^(*3)		1.0MBytes	1.0MBytes	2.0MBytes	2.0MBytes
로그 메모리 ^(*4)		256KBytes	256KBytes	512KBytes	512KBytes
로그 메모리 백업 ^(*5)		지원	지원	지원	지원
로그 메모리 위치 ^(*6)		0x640000	0x640000	CPU: 0xC80000 S28: 0x1700000	CPU : 0xC80000 S28 : 0x1700000
이더넷		10/100Base-T	10/100Base-T	10/100Base-T	10/100Base-T
시리얼 (+옵션 카드) ^(*7)		2 (+2) 포트	2 (+2) 포트	0 (+2~16) 포트	0 (+2~64) 포트
Profibus (optional)		1 포트	1 포트	-	-
I/O (optional)		입력 8 + 출력 6	입력 8 + 출력 6	-	-
WatchDog		지원	지원	-	-
RTC		지원	지원	지원	지원
이중화 구성		-	-	-	option
드라이버 ^(*8)	Serial Only	지원	지원	지원	지원
	Ethernet	지원	지원	지원	지원
	SECS	-	지원	지원 ^(*9)	-

- (*1) 사용자 메모리 : 사용자가 작성한 프레임 및 프로세스가 사용하는 영역입니다. 사용자가 프로그램 내에서 참조하는 영역입니다.
- (*2) 배터리에 의해 백업되는 영역입니다. 백업 영역은 ETOS-100/150 은 상위 128Kbytes. 즉, 131072 ~ 262143 번지이며, ETOS-200/1000 은 상위 256Kbytes, 즉, 262144 ~ 524287 번지까지 입니다.
- (*3) 프로그램 메모리 : 사용자가 설정한 파라미터와 프레임, 프로세스 정의가 저장되는 시스템 메모리 영역입니다. 사용자가 프로그램 내에서 참조할 수 없습니다
- (*4) ETOS 동작 중에 발생한 중요 이벤트를 텍스트로 저장해두는 메모리입니다. [온라인] -> [메모리 저장]에서 시스템 메모리를 체크하면 파일로 저장할 수 있습니다.
- (*5) 로그 메모리 전체가 배터리에 의해 백업됩니다
- (*6) ETOS-200/1000 시리얼 통신 카드는 로그 메모리 위치가 0x1700000 번지입니다
- (*7) ETOS-100A/150 시리즈는 기본 2 포트에 옵션으로 2 포트를 추가하여, 최대 4 포트까지 가능합니다
- (*8) ETOS 에서 사용하는 프로그램 파일은 ETOS 가 지원하는 통신 기능에 따라 몇 가지의 드라이버로 구분됩니다.
- (*9) ETOS-200 시리즈의 SECS 기능 지원은 ETOS-200-HSMS, ETOS-200-Sgate 제품만 가능합니다

1.3.4. S/W 사양

제품 별 S/W 사양입니다. 다음 표는 제품 별로 지원 가능한 최대 크기, 또는 수를 표시합니다.

항 목	ETOS-50A	ETOS-100A/150	ETOS-200	ETOS-1000
프로그램 파일 최대 크기 ^(*1)	1.0MBytes	1.0MBytes	2.0MBytes	2.0Mbytes
등록가능 프로세스 ^(*2)	1024	1024	1024	1024
프로세스 당 등록가능 아이템 ^(*3)	4096	4096	4096	4096
타이머 프로세스 ^(*4)	16	32	64	128
동시실행가능 프로세스 ^(*5)	32	64	128	256
이더넷 커넥션 ^(*6)	4	16	24	32

-
- (*1) 프로그램 파일의 최대 크기는 프로그램 메모리(FLASH)의 크기와 동일합니다.
 - (*2) 사용자가 하나의 프로그램 파일안에서 정의할 수 있는 프로세스의 최대 수입니다. 프로그램 파일의 크기에 의해 제한을 받습니다. 따라서 실제로는 최대 수보다 적을 수 있습니다.
 - (*3) 하나의 프로세스안에 등록가능한 아이템의 최대 수입니다.
 - (*4) 하나의 프로그램 파일안에서 사용가능한 IF_TIMER 아이템의 개수입니다. IF_TIMER 로 시작하는 프로세스의 수는 제시된 수치보다 적거나 같아야 합니다.
 - (*5) ETOS 는 멀티 태스킹 기능을 내장하여 동시에 여러 개의 프로세스를 수행할 수 있습니다. 그러나, 내부 자원(메모리, CPU 부하)을 고려하여 여러 개의 프로세스를 등록하였을 경우 동시에 실행 가능한 프로세스의 개수를 제한 하고 있으며 ETOS 기종별로 동시에 실행 가능한 프로세스 개수는 표와 같이 제한됩니다. 동시 실행 가능 프로세스의 수는 시스템의 전체 수행 성능을 결정합니다.
 - (*6) ETOS 는 10/100 BaseT 포트를 통해 외부 기기와 이더넷 접속을 하는데 물리적인 통신포트는 하나 입니다.(이중화에서는 2 개) 위의 [표]에서 설명하는 이더넷 커넥션은 하드웨어적인 케이블 접속 포트를 의미하는 것이 아니고, 이더넷으로 접속 가능한 상대 통신기기의 커넥션 수를 의미합니다. 커넥션 수는 ETOS 의 내부자원(메모리, CPU 부하)을 고려하여 기종별로 표와 같이 제한되고 있습니다. 이에 대한 자세한 내용은 본 설명서 9 장 이더넷 통신을 참조하시기 바랍니다.

제품 규격

제 2 장

제품의 일반적인 성능 규격 전원, 케이블에 대한 규격 등을 설명합니다.

	내용	위치
2.1	일반 규격	10
2.2	전원 규격	11
2.3	케이블 규격	12
2.4	통신 방식 규격	14
2.5	LED 표시 내용	15

2. 제품 규격

제품의 일반 성능 규격 및 전원, 케이블에 대한 규격을입니다.

2.1. 일반 규격

ETOS 시리즈에 적용되는 사용 환경, 전기 및 기계적 규격에 대한 설명입니다.

항목	규격				관련 규격		
사용 온도	-10℃ ~ +60℃						
보관 온도	-25℃ ~ +70℃						
사용 습도	5~95%RH,이슬이 맺히지 않을 것						
보관 습도	5~95%RH,이슬이 맺히지 않을 것						
내 진동	단속적인 진동이 있는 경우						
	주파수	가속도	진폭	횟수	IEC 1131-2		
	10≤f< 57 Hz	-	0.075mm	X,Y,Z 각 방향 10 회			
	57≤f≤150 Hz	9.8 m/s²(1G)	-				
	연속적인 진동이 있는 경우						
	주파수	가속도	진폭				
	10≤f< 57 Hz	-	0.035mm				
57≤f≤150 Hz	4.8 m/s²(0.5G)	-					
내 충격	* 최대 충격 가속도:147 m/s²(15G) * 인가 시간 :11 ms * 펄스 파형 : 정현 반파 펄스(X,Y,Z 3 방향 각 3 회)				IEC 1131-2		
내 노이즈	방형파 임펄스 노이즈	±1,500V			자체 시험 규격 기준		
	정전기 방전	전압 : 4kV(접촉 방전)			IEC 1131-2, IEC 801-2		
	방사 전자계 노이즈	27 ~ 500 MHz, 10 V/m			IEC 1131-2, IEC 801-3		
	패스트 트랜지언트 /버스트 노이즈	구분	전원 모듈	디지털 입출력 (24V 이상)	디지털 입출력 (24V 이상) 아날로그 입출력 통신 인터페이스	IEC 1131-2, IEC 801-4	
전압	2 kV	1 kV	0.25 kV				
주위환경	부식성 가스, 먼지가 없을 것						
사용고도	2000m 이하						
오염도 (*1)	2 이하						
냉각 방식	자연 공랭식						

(*1) 오염도:장치의 절연 성능을 결정하는 사용환경의 오염 정도를 나타내는 지표이며 오염도 2 란 통상 비전도성 오염만 발생하는 상태입니다. 단,이슬 맺힘에 따라 일시적인 도전이 발생하는 상태를 말합니다

2.2. 전원 규격

제품별 전원에 대한 정격 입력 전압과 전원용량에 대한 규격입니다.

제품	입력 전원	전원 용량	소비 전력 (*24)
ETOS-50A	Free Voltage AC 85V~260V (50/60Hz)	5V/1A	2.5W
ETOS-100A		5V/2.5A	5W
ETOS-150		5V/2.5A	5W
ETOS-200		5V/7A	25W
ETOS-1000	AC 110V/220V 선택형	5V/15A +12V/1.5A, -12V/1.5A	65W

(*24) 소비전력은 제품의 구성과 장착되는 옵션 종류에 따라 다를 수 있습니다.
표에서 표시된 소비전력은 다음과 같은 구성시 예입니다.

- ETOS-50A : 이더넷 + RS232 2 포트
- ETOS-100A : 이더넷 + RS232 4 포트
- ETOS-150 : 이더넷 + RS232 4 포트
- ETOS-200 : CPU 1EA + RS232 16 포트
- ETOS-1000 : CPU 1EA + RS232 64 포트

2.3. 케이블 규격

ETOS 에서 통신용으로 사용되는 케이블에 대한 규격입니다. 추천 케이블을 사용하지 않을 경우(특히 장거리 통신 시) 통신이 불안정하거나 통신이 안될 수 있으므로 주의를 요합니다.

2.3.1. RS232C / RS422 / RS485 케이블

RS422 또는 RS485 를 이용하여 통신할 경우 통신거리 및 통신속도를 고려하여 RS422 용 트위스트 패어 실드 케이블을 사용하여야 합니다. 다음 표는 권장 케이블의 규격을 기재하고 있습니다. 권장 케이블 이외의 것을 사용할 경우에도 표의 특성에 맞는 케이블을 사용하여 주십시오.

품 명 : Low Capacitance LAN Interface Cable

형 명 : LIREV-AMESB

규 격 : 2P X 22AWG(D/0.254 TA)

제조원 : LG 전선

▶ 전기적 특성

항 목	단 위	특 성	조 건
도 체 저 항	Ohm/km	59 이하	상 온
내 전 압(DC)	V/1min	500V 에 1 분간 견딤	공 기 중
절 연 저 항	MEGA Ohm-km	1,000 이상	상 온
정 전 용 량	pF/m	45 이하	1 KHz
특성 임피던스	Ohm	120 (±12)	10 MHz

▶ 외관 특성

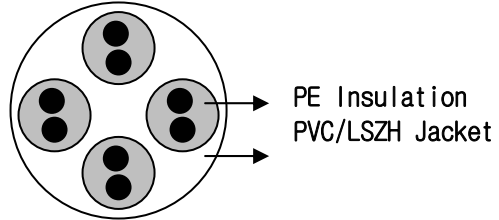
항 목		단 위	단 선	연 선
도 체	심선수	패어	2	2
	규 격	AWG	22	22
	구 성	NO./mm	1/0.64	7/0.254
	외 경	mm	0.64	0.76
절연체	두께	mm	0.55	0.55
	외 경	mm	1.64	1.76

2.3.2. 이더넷 케이블

ETOS 시리즈에서 제공되는 이더넷은 10/100 Base-T 로 RJ45 커넥터를 사용한 Category 5 Class 케이블을 사용하고, IEEE802.3 에서 제시하는 규격을 만족하는 케이블이어야 합니다.

케이블 사용 예

품 명 : UTP 케이블
 형 명 : Enhanced CAT.5 4P
 규 격 : 4P(Pair) X 24AWG
 제조원 : LG 전선



▶ 케이블 규격

항 목	단 위	수 치
심선수	패어	4
규격	AWG	24
외경	mm	5.0
중량	Kg/305m	9

2.4. 통신 방식 규격

ETOS 시리즈에 사용되는 각종 시리얼 / 이더넷 / 필드버스에 대한 규격입니다.

2.4.1. 이더넷 (10/100 Base-T)

항 목		규 격
전송 규격	데이터 전송속도	10Mbps /100Mbps
	전송방식	베이스 밴드
	최대 세그먼트 길이	100m(노드-허브)
	최대 프로토콜 크기	1500 바이트
	프로토콜	TCP/IP, UDP/IP
	Application 프로토콜	GLOFA Enet , MODBUS, MELSEC-Q
	TCP/UDP 포트 수	ETOS-50A:4 포트, ETOS-100A/150:16 포트 ETOS-200:24 포트, ETOS-1000:32 포트
	통신권 액세스 방식	CSMA/CD

2.4.2. 시리얼 (RS232C, RS422/485)

항 목	규 격
Data Bit	7 또는 8
Stop Bit	1 또는 2
Start Bit	1
Parity	Even / Odd / None / Space / Mark
채널 선택	Process 에서 Dynamic 하게 변경 가능
동기 방식	비동기 방식
전송 속도 (BPS)	300/600/1200/1800/2400/3600/4800/7200/9600 19200/38400/57600/64000/115200
전송 거리	RS232C : 최대 15m(모뎀 사용 가능)
	RS422 : 최대 500m
진단 기능	채널 당 Tx, Rx, Error LED 로 동작 상태 표시 ^(*25)

2.4.3. 필드버스

항 목	규 격	비 고
Profibus-DP	EN50170(유럽 규격)	ETOS-100A 옵션, Master/Slave 각각 지원
GLOFA Fnet	GLOFA PLC Fnet 시스템	ETOS-100A 옵션

(*25) Error LED 는 ETOS-100A/150 에서만 지원

2.5. LED 표시 내용

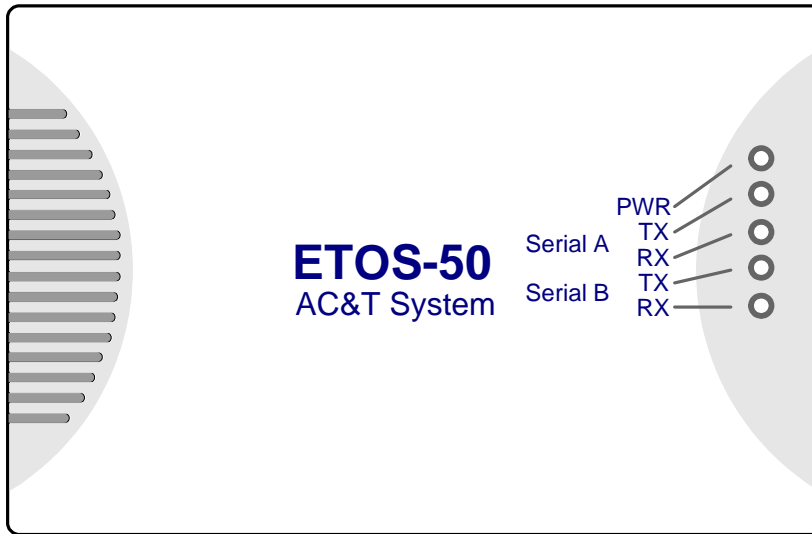
ETOS 는 시스템의 상태 정보와 송.수신 통신 상태를 LED 를 통해 표시합니다. 사용자는 해당 LED 상태를 통해 제품의 상태를 판단할 수 있습니다.

2.5.1. ETOS-50A

ETOS-50A 의 각 LED 의 역할 및 동작 모드에 따른 LED 의 동작 상태를 설명합니다.

2.5.1.1. ETOS-50A 시리얼

ETOS-50A 시리얼(모델명: ETOS-50A-SXX)은 이더넷 포트는 없으며, 시리얼 2 포트로 구성되어 있습니다.



▶ 정상 동작 시 LED 상태

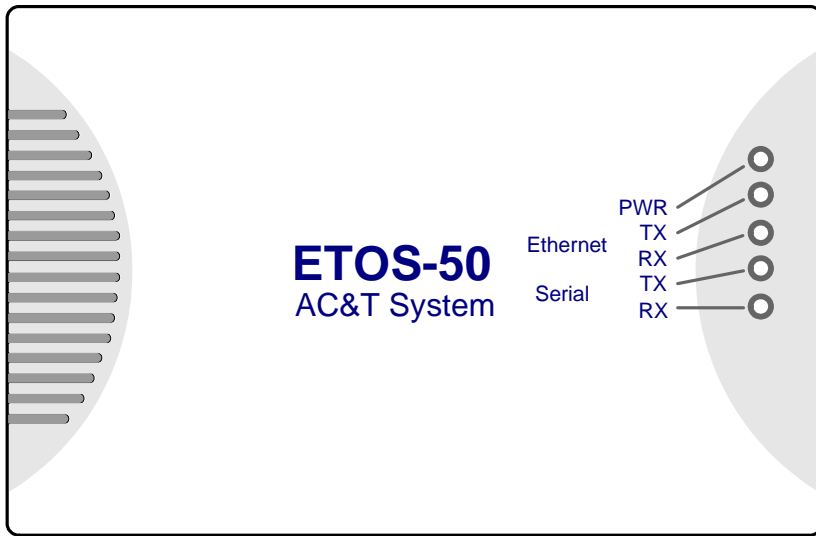
전면 표기		LED 색상	동작 상태
PWR		Red	Power On 시 점등
Serial A	TX	Green	Serial A 송신 시 점등
	RX	Green	Serial A 수신 시 점등
Serial B	TX	Green	Serial B 송신 시 점등
	RX	Green	Serial B 수신 시 점등

▶ OS Download 시 LED 상태

전면 표기		LED 색상	동작 상태
PWR		Red	Power On 시 점등
Serial A	TX	Green	LED 점멸
	RX	Green	LED 점멸
Serial B	TX	Green	-
	RX	Green	-

2.5.1.2. ETOS-50A 이더넷

ETOS-50A 이더넷(모델명: ETOS-50A-EXX)은 이더넷 1 포트와 시리얼 1 포트로 구성되어 있습니다.



▶ **정상 동작 시 LED 상태**

전면 표기		LED 색상	동작 상태
PWR		Red	Power On 시 점등
Ethernet	TX	Green	이더넷 송신 시 점등
	RX	Green	이더넷 수신 시 점등
Serial	TX	Green	Serial 송신 시 점등
	RX	Green	Serial 수신 시 점등

▶ **OS Download 시 LED 상태**

전면 표기		LED 색상	동작 상태
PWR		Red	Power On 시 점등
Ethernet	TX	Green	-
	RX	Green	-
Serial	TX	Green	LED 점멸
	RX	Green	LED 점멸

OS Download 수행 동안 LED 가 점멸 됩니다.이 과정은 정상 상태로 복귀될 때 까지 반복됩니다.

2.5.2. ETOS-100A

ETOS-100A의 각 LED의 역할 및 동작 모드에 따른 LED의 동작 상태를 설명합니다. 다음 그림은 ETOS-100A의 전면 패널^(*26) 모습입니다.

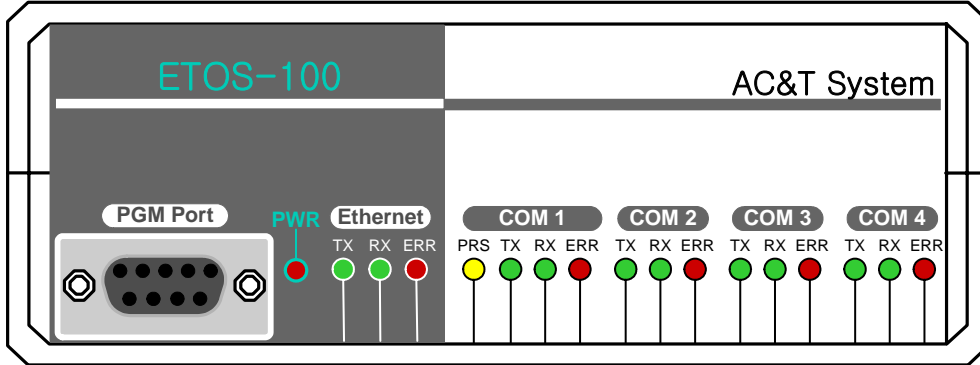


그림 2.5-1 ETOS-100A 전면 패널

▶ 정상 동작 시 LED 상태

전면 표기		LED 색상	동작 상태
PWR		Red	Power On 시 점등
Ethernet ^(*27)	TX	Green	이더넷 송신 시 점등
	RX	Green	이더넷 수신 시 점등
	ERR	Red	이더넷에서 Error가 발생하면 점등
COM1 ^(*28)	PRS	Yellow	파라미터 및 프로세스 이상 시 점등
	TX	Green	COM1 송신 시 점등
	RX	Green	COM1 수신 시 점등
	ERR	Red	COM1에서 Error가 발생하면 점등
COM2	TX	Green	COM2 송신 시 점등
	RX	Green	COM2 수신 시 점등
	ERR	Red	COM2에서 Error가 발생하면 점등
COM3	TX	Green	COM3 송신 시 점등
	RX	Green	COM3 수신 시 점등
	ERR	Red	COM3에서 Error가 발생하면 점등
COM4	TX	Green	COM4 송신 시 점등
	RX	Green	COM4 수신 시 점등
	ERR	Red	COM4에서 Error가 발생하면 점등

(*26) 전면 명판엔 ETOS100과 ETOS100A 구분을 하지 않습니다. 제품 하단의 시리얼 번호가 적힌 스티커에서 구분됩니다

(*27) 이더넷이 지원되는 제품에 한함.

(*28) ETOS-100A에서는 파라미터 및 프로세스 이상 시 점등하는 LED로 사용

▶ 자기 진단 및 OS Download 시 LED 상태

전면 표기		LED 색상	동작 상태
PWR		Red	Power On 시 점등
Ethernet	TX	Green	의미 없음
	RX	Green	의미 없음
	ERR	Red	이더넷에서 Error 가 발생하면 점등
COM1	RS422	Yellow	-
	TX	Green	OS 가 없으면 점등
	RX	Green	OS 에서 OS Upgrade 를 요청하면 점등
	ERR	Red	Off
COM2	TX	Green	Off
	RX	Green	Off
	ERR	Red	Off
COM3	TX	Green	COM2 Port Loop back 이 되면 점등
	RX	Green	COM1 Port Loop back 이 되면 점등
	ERR	Red	-
COM4	TX	Green	이더넷 Controller 가 정상이면 점등
	RX	Green	RAM Test 정상이면 점등
	ERR	Red	LED 점멸 후 점등

자기 진단 및 OS Download 모드에서는 PWR, 이더넷 TX, RX 를 제외한 모든 LED 가 점멸한 후 자기 진단을 하고 그 결과를 LED 에 표기합니다. 이 과정은 정상 동작 상태로 전환 될 때까지 계속 반복됩니다.

2.5.3. ETOS-150

ETOS-150의 각 LED의 역할 및 동작 모드에 따른 LED의 동작 상태를 설명합니다. 다음 그림은 ETOS-150의 전면 패널 모습입니다.

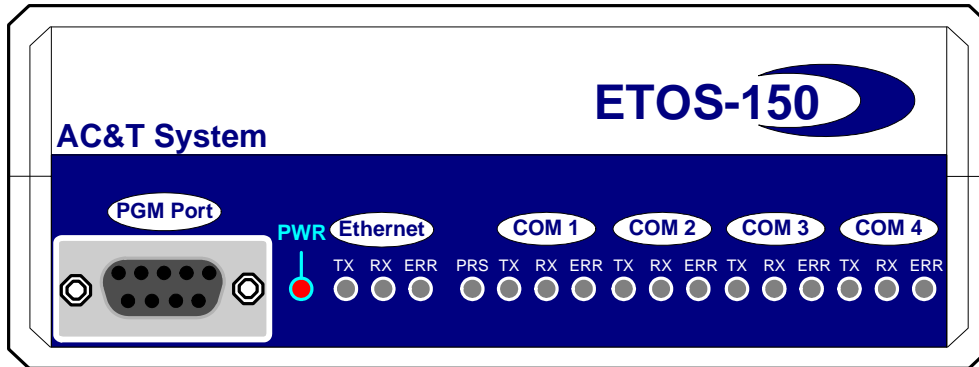


그림 2.5-2 ETOS-150 전면 패널

▶ 정상 동작 시 LED 상태

전면 표기		LED 색상	동작 상태
PWR		Red	Power On 시 점등
Ethernet	TX	Green	이더넷 송신 시 점등
	RX	Green	이더넷 수신 시 점등
	ERR	Red	이더넷에서 Error가 발생하면 점등
PRS	PRS	Yellow	파라미터 및 프로세스 이상 시 점등
COM1	TX	Green	COM1 송신 시 점등
	RX	Green	COM1 수신 시 점등
	ERR	Red	COM1에서 Error가 발생하면 점등
COM2	TX	Green	COM2 송신 시 점등
	RX	Green	COM2 수신 시 점등
	ERR	Red	COM2에서 Error가 발생하면 점등
COM3	TX	Green	COM3 송신 시 점등
	RX	Green	COM3 수신 시 점등
	ERR	Red	COM3에서 Error가 발생하면 점등
COM4	TX	Green	COM4 송신 시 점등
	RX	Green	COM4 수신 시 점등
	ERR	Red	COM4에서 Error가 발생하면 점등

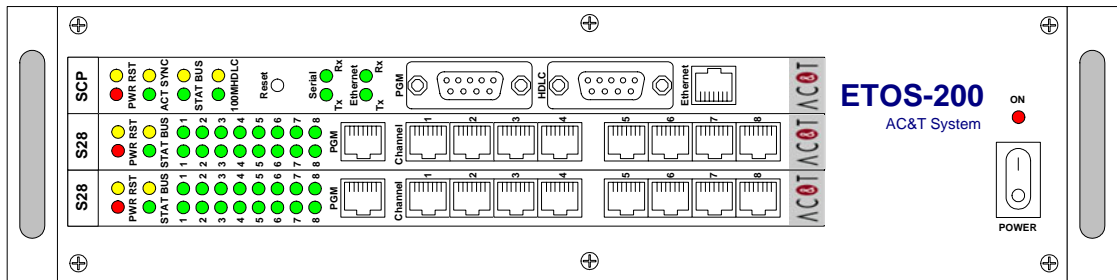
▶ 자기 진단 및 OS Download 시 LED 상태

전면 표기		LED 색상	동작 상태
PWR		Red	Power On 시 점등
Ethernet	TX	Green	의미 없음
	RX	Green	의미 없음
	ERR	Red	이더넷에서 Error 가 발생하면 점등
PRS	PRS	Yellow	파라미터 및 프로세스 이상 시 점등
COM1	TX	Green	OS 가 없으면 점등
	RX	Green	OS 에서 OS Upgrade 를 요청하면 점등
	ERR	Red	Off
COM2	TX	Green	Off
	RX	Green	Off
	ERR	Red	Off
COM3	TX	Green	COM2 Port Loop back 이 되면 점등
	RX	Green	COM1 Port Loop back 이 되면 점등
	ERR	Red	-
COM4	TX	Green	이더넷 Controller 가 정상이면 점등
	RX	Green	RAM Test 정상이면 점등
	ERR	Red	LED 점멸 후 점등

자기 진단 및 OS Download 모드에서는 PWR, 이더넷 TX, RX 를 제외한 모든 LED 가 점멸한 후 자기 진단을 하고 그 결과를 LED 에 표기합니다. 이 과정은 정상 동작 상태로 전환 될 때까지 계속 반복됩니다.

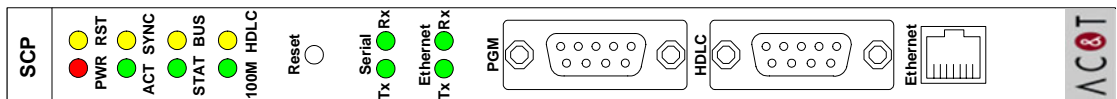
2.5.4. ETOS-200

ETOS-200의 각 LED의 역할 및 동작 방식을 설명합니다. 다음 그림은 CPU 모듈 1장과 시리얼 통신 모듈 2장으로 구성된 ETOS-200의 전면 패널 모습입니다.



2.5.4.1. CPU 모듈

ETOS-200 CPU 모듈(SCP로 표기)입니다. 정상 동작시의 LED 상태를 설명합니다.



▶ System Status 표시용 LED

전면 표기	LED 색상	동작 상태
PWR	Red	Power On 시 점등
RST	Yellow	Reset 시 점등
ACT (*29)	Green	CPU 모듈 정상 동작일 때 점멸
SYNC (*30)	Yellow	-
STAT	Green	Error 발생시 점등
BUS	Yellow	사용자가 작성한 Program 에서 시리얼 통신카드를 Access 할 경우 점등됨.
100M	Green	100Mbps 이더넷 통신 시 점등
HDLC (*2)	Yellow	-

(*29) CPU가 정상 동작 중이면 주기적으로 점멸합니다.

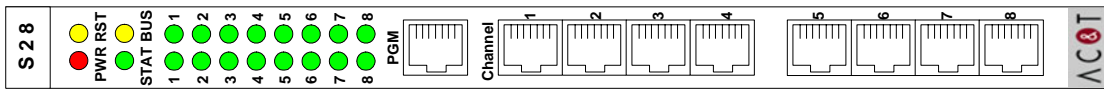
(*30) 이중화 모델에서만 동작합니다

▶ Port Status 표시용 LED

전면 표기	LED 색상	동작 상태
Serial	Tx	Green Monitor Port 송신 시 점등
	Rx	Green Monitor Port 수신 시 점등
Ethernet	Tx	Green 이더넷 송신 시 점등
	Rx	Green 이더넷 수신 시 점등

2.5.4.2. 시리얼 통신 모듈

다음 그림은 시리얼 통신 모듈(S28 로 표기)입니다. 정상 동작시의 LED 상태를 설명합니다.



▶ System Status 표시용 LED

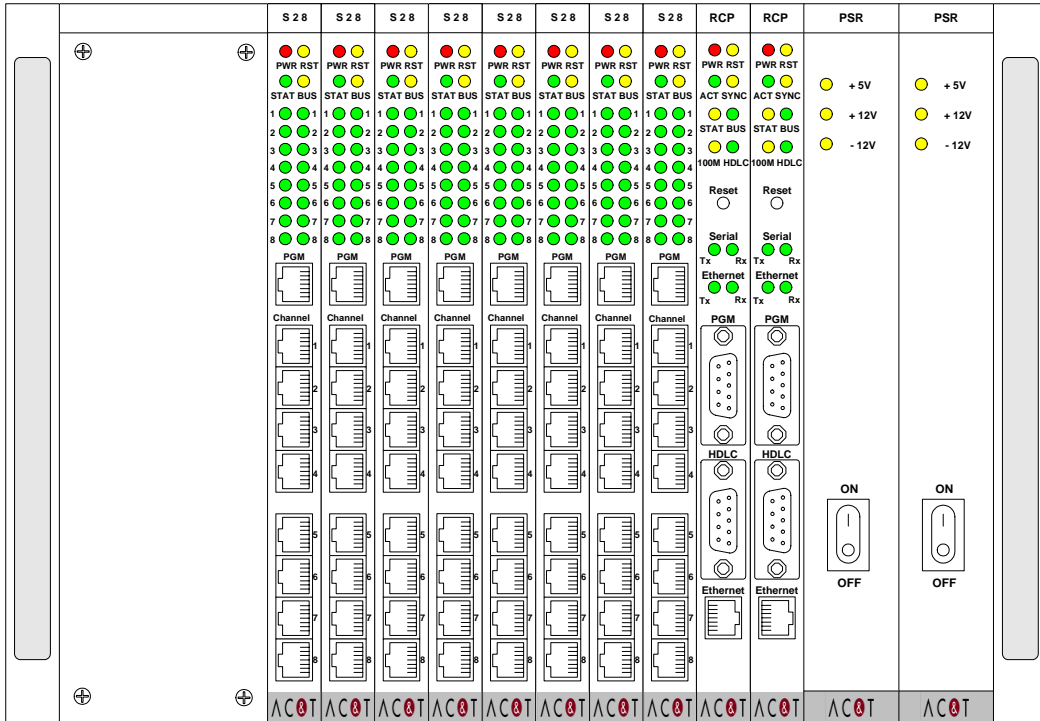
전면 표기	LED 색상	동작 상태
PWR	Red	Power On 시 점등
RST	Yellow	Board Reset 시 점등
BUS	Yellow	사용자가 작성한 Program 이 시리얼 통신카드를 Access 할 경우 점등됨.
STAT	Green	정상 동작 여부 표시 (정상 동작일 때 점멸)

▶ Port Status 표시용 LED

전면 표기	LED 색상	동작 상태
Tx	Green	Data 송신 시 점등
Rx	Green	Data 수신 시 점등

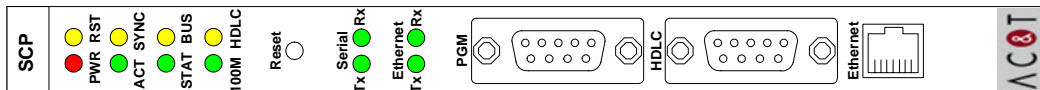
2.5.5. ETOS-1000

ETOS-1000 의 각 LED 의 역할 및 동작 방식을 설명합니다. 다음 그림은 CPU 모듈 2 장과 시리얼 통신 모듈 8 장으로 구성된 ETOS-1000 이중화의 전면 패널 모습입니다.



2.5.5.1. Master CPU 모듈

▶ 단중화 모델(SCP) System Status 표시용 LED

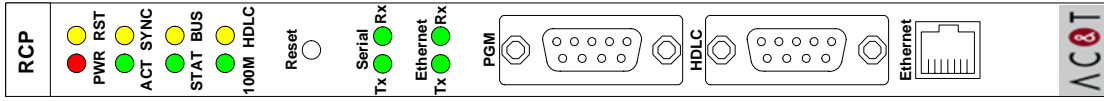


전면 표기	LED 색상	동작 상태
PWR	Red	Power On 시 점등
RST	Yellow	Reset 시 점등
ACT ^(*31)	Green	CPU 모듈 정상 동작일 때 점멸
SYNC ^(*32)	Yellow	-
STAT	Green	Error 발생시 점등
BUS	Yellow	사용자가 작성한 Program 이 시리얼 통신카드를 Access 할 경우 점등됨.
100M	Green	100Mbps 이더넷 통신 시 점등
HDLC ^(*2)	Yellow	-

(*31) CPU 가 정상 동작 중이면 주기적으로 점멸합니다

(*32) 이중화 모델에서만 동작합니다

▶ 이중화 모델(RCP) System Status 표시용 LED



전면 표기	LED 색상	동작 상태
PWR	Red	Power On 시 점등
RST	Yellow	Reset 시 점등
ACT	Green	Active 동작 시 점멸
SYNC	Yellow	사용자 메모리 Sync 시 점등
STAT ^(*33)	Red	Error 발생시 점등, Standby 상태로 동작 시 점멸
BUS ^(*34)	Green	사용자가 작성한 Program 이 시리얼 통신카드를 Access 할 경우 점등됨.
100M	Red	100Mbps 이더넷 통신 시 점등
HDLC	Green	Active CPU 모듈과 Standby CPU 모듈간의 Alive Check

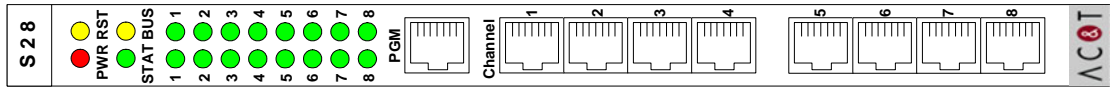
(*33) Active CPU 에서는 Error 상태 표시로 사용되며, Standby CPU 에서는 정상 동작 상태 표시로 사용됩니다.

(*34) BUS LED 는 Active CPU 에서만 점등됩니다

▶ Port Status 표시용 LED (단중화 및 이중화 모델)

전면 표기	LED 색상	동작 상태
Serial	Tx	Green Monitor Port 송신 시 점등
	Rx	Green Monitor Port 수신 시 점등
이더넷	Tx	Green 이더넷 송신 시 점등
	Rx	Green 이더넷 수신 시 점등

2.5.5.2. 시리얼 통신 모듈



▶ System Status 표시용 LED

전면 표기	LED 색상	동작 상태
PWR	Red	Power On 시 점등
RST	Yellow	Board Reset 시 점등
BUS	Yellow	사용자가 작성한 Program 이 시리얼 통신카드를 Access 할 경우 점등됨
STAT	Green	Active 동작 시 점멸, Error 발생 시 점등 (*35)

(*35) 정상 동작 중이면 주기적으로 점멸합니다.

▶ Port Status 표시용 LED

전면 표기	LED 색상	동작 상태
Tx	Green	Data 송신 시 점등
Rx	Green	Data 수신 시 점등

설치 및 배선

제 3 장

ETOS 시리즈 설치 및 배선에 필요한 각종 규격에 대해서 설명합니다

	내용	위치
3.1	ETOS-50A 전원 및 커넥터 규격	28
3.2	ETOS-100A/150 전원 및 커넥터 규격	30
3.3	ETOS-50A/100A/150 배선 방법	34
3.4	ETOS-200/1000 설치 및 배선	41

3. 설치 및 배선

ETOS 시리즈 설치 및 배선에 필요한 커넥터 핀 규격과 전원 규격에 대해 설명합니다.

3.1. ETOS-50A 전원 및 커넥터 규격

ETOS-50A의 전원 규격 및 커넥터 핀 규격에 대해 설명합니다.

3.1.1. 전원 규격

- * 입력전원 : AC 85V~260V (Free Voltage), 50/60 Hz
- * 출력 : DC 5V 1A

3.1.2. ETOS-50A 시리얼 모델 커넥터 규격

다음 그림은 ETOS-50A 시리얼 모델(모델명: ETOS-50A-SXX)의 후면 패널 모습입니다. 전원 입력 단자 옆에 두개의 시리얼 포트를 제공하는데 Serial A/B 는 외부 기기 통신용 시리얼 통신포트입니다. PGM 포트는 ETOS-PD 와의 온라인 연결용 포트이며 외부기기 통신에는 사용할 수 없으므로 주의하여야 합니다.

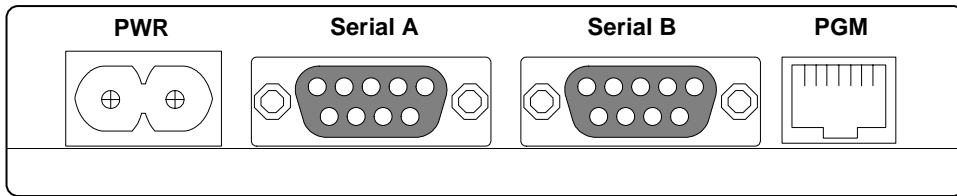


그림 3.1-1 ETOS-50A 시리얼 모델 후면 패널

외부기기 통신용 포트 Serial A 는 프로그램에서 COM1 포트에 배정되고, Serial B 는 COM2 에 각각 배정되어 있습니다. ETOS-50A 시리얼 모델은 두개의 시리얼 통신포트를 제공하며 이더넷 포트는 제공하지 않습니다. 시리얼 포트는 RS232C, RS422/485 두가지를 제공하는데 하나의 포트로는 RS232C/RS422 포트를 동시에 지원하지 않습니다. 통신 용도에 맞게 선택하여 사용하여야 합니다. ETOS-50A 시리얼 모델에는 시리얼 포트의 구성에 따라 다음과 같이 세 종류의 모델이 있습니다.

모델명	제품 구성	Serial A	Serial B
ETOS-50A-S20	RS232C 용 2 포트	RS232C	RS232C
ETOS-50A-S02	RS422/485 용 2 포트	RS422/485	RS422/485
ETOS-50A-S11	RS232C 1 포트/ RS422/485 1 포트	RS422/485	RS232C

3.1.3. ETOS-50A 이더넷 모델 커넥터 규격

다음 그림은 ETOS-50A 이더넷 모델(모델명: ETOS-50A-EXX)의 후면 패널 모습입니다.

이더넷 1 포트와 시리얼 1 포트가 지원됩니다. 전원 입력단자 옆 PGM 포트는 ETOS-PD 를 이용한 온라인 연결용 포트입니다. ETOS-50A 시리얼 모델과 달리 PGM 포트는 D-SUB9 (Female) 커넥터를 이용하여 PC 와 연결되며 외부기기 통신에는 사용할 수 없으므로 주의하여야 합니다. PGM 포트 옆의 Serial 포트는 외부기기와 통신용 시리얼 통신포트입니다. RS232C 또는 RS422 통신방식을 용도에 맞게 선택하여 사용하여야 합니다. Serial 포트 옆의 Ethernet 포트는 이더넷 10/100 Base-T 통신용 RJ-45 단자이며 외부기기와 이더넷 통신시 사용합니다.

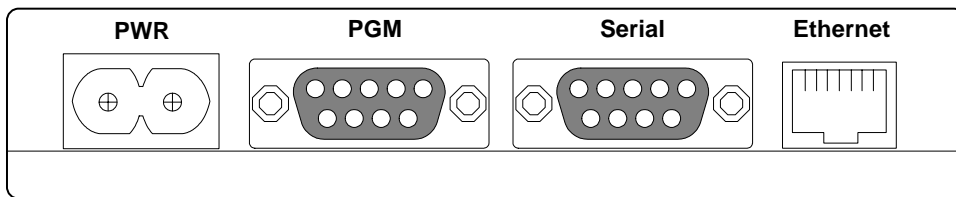


그림 3.1-2 ETOS-50A 이더넷 모델 후면 패널

ETOS-50A 이더넷 모델은 시리얼 통신 방식에 따라 다음의 두가지로 구분됩니다.

모델명	통신포트 구성	Ethernet	Serial
ETOS-50A-E10	이더넷 1 포트, RS232 1 포트	10/100Base-T	RS232C
ETOS-50A-E01	이더넷 1 포트, RS422 1 포트	10/100Base-T	RS422/485

3.2. ETOS-100A/150 전원 및 커넥터 규격

ETOS-100A/150의 전원 규격 및 커넥터 핀 규격에 대해 설명합니다.

3.2.1. 전원 규격

- * 입력전원 : AC 85V~260V (Free Voltage), 50/60 Hz
- * 출력 : DC 5V 2.5A

전원은 반드시 접지(F.G)가 연결되어야 하고 제 3종 접지 규격이어야 합니다.

3.2.2. 온라인 연결 포트

ETOS-PD를 이용하여 ETOS로 프로그램 읽기/쓰기 및 프레임 모니터 등의 기능을 사용하기 위해서는 전면의 PGM 포트를 이용하여야 합니다. PGM 포트는 외부기기와의 통신용으로 사용할 수 없는 온라인 연결용 전용 포트입니다.

그림은 ETOS-100A PGM 포트 및 전면 배치도입니다.

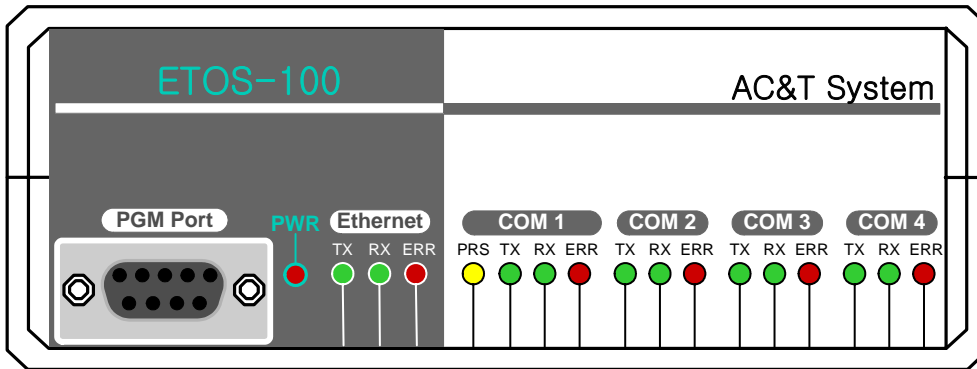


그림 3.2-4 ETOS-100A PGM 포트 배정

그림은 ETOS-150 PGM 포트 및 전면 배치도입니다.

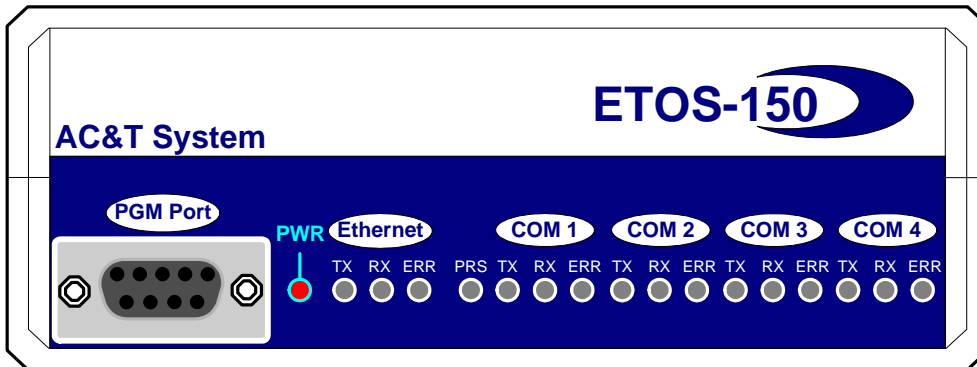


그림 3.2-5 ETOS-150 PGM 포트 배정

3.2.3. 외부 통신용 포트

다음 그림은 ETOS-100A/150 의 후면 패널입니다. 후면 패널은 ETOS-100A 와 ETOS-150 이 동일한 구조를 가지고 있습니다. ETOS-100A/150 은 4 개의 RS232C 또는 RS422/485 통신용 포트와 1 개의 이더넷 포트(10/100Base-T)를 제공합니다. COM1 ~ COM4 는 시리얼 통신용 포트이며 COM1, COM2 는 기본사양으로 장착되고 COM3, COM4 는 옵션 사양으로 시리얼 통신, 필드버스 통신, 또는 I/O 입출력 용도로 사용할 수 있습니다. 10/100 Base-T 포트는 이더넷 통신 포트로서 RJ-45 커넥터를 사용합니다.

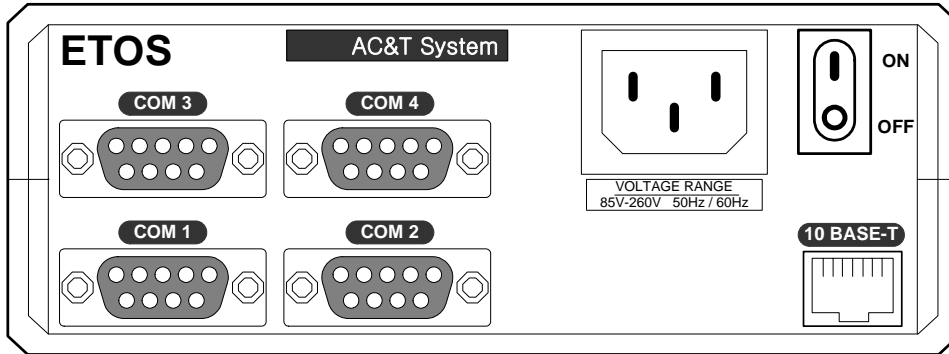


그림 3.2-1 ETOS-100A/150 후면 패널

3.2.3.1. 지원되는 통신 포트

ETOS-100A/150 은 COM1~COM4 의 최대 4 포트를 사용할 수 있습니다. 하나의 포트는 RS232C 와 RS422/485 중 한 가지로만 동작 가능합니다. 또한 COM3 과 COM4 에는 필드버스 옵션 모듈을 사용할 수 있습니다. 다음 표는 각 포트별로 지원되는 통신 포트의 종류입니다.

표 3.2-1 포트별 지원 가능 모듈

포트	COM1	COM2	COM3	COM4
RS232C	가능	가능	가능	가능
RS422/485	가능	가능	가능	가능
필드버스	불가	불가	가능	가능
Digital I/O	불가	불가	가능	가능

3.2.4. ETOS-100A/150 옵션카드 설치 및 배선

ETOS-100A/150 은 COM3, COM4 포트에 옵션모듈을 장착할 수 있습니다. 디지털 입출력(I/O) 옵션 카드에 대해 설명합니다.

3.2.4.1. ETOS-100A/150 디지털 입출력 옵션 카드

ETOS 100A/150 I/O 옵션 카드는 입력 8 점, 릴레이 출력 6 점을 제공합니다. COM3 은 입력용, COM4 는 출력용 포트고 고정되어 있습니다.

입력 포트(COM3) 규격

항목	규격
입력 점수	8
절연 방식	포토 커플러
정격 입력 전압	DC 24V
정격 입력 전류	9mA
동작 전압	24 VDC ± 20% (리플 5% 이하)
On시 전압 / 전류	DC 9.5V 이상 / 3.5mA 이상
입력 Common	8점 / 1Common
커넥터	D-SUB 9 Female

출력 포트(COM4) 규격

항목	규격	
출력 점수	6	
출력 소자	릴레이	
절연 방식	릴레이 절연	
정격 부하 전압, 전류	DC24V, 1A/1점, 6A/1Common	
최소 부하	DC 5V / 1mA	
서지 킬러	없음	
릴레이 수명(전기적)	10만회 이상	
출력 Common	6점 / 1Common	
외부 공급 전원	전압	24 VDC ± 10% (리플 4Vp-p 이하)
	전류	150mA(24VDC 모든 점 ON시)
커넥터	D-SUB 9 Female	

D-SUB 9 핀 커넥터(Mail)의 핀 배열은 다음과 같습니다.

입력 포트 (COM3) 핀 배열

핀 번호	신호 명	기능 설명
1	IN 0	입력 0
2	IN 1	입력 1
3	IN 2	입력 2
4	IN 3	입력 3
5	IN 4	입력 4
6	IN 5	입력 5
7	COMMON	8 점 공통 접지선
8	IN 6	입력 6
9	IN 7	입력 7

출력 포트 (COM4) 핀 배열

핀 번호	신호 명	기능 설명
1	OUT 0	출력 0
2	OUT 1	출력 1
3	OUT 2	출력 2
4	OUT 3	출력 3
5	OUT 4	출력 4
6	OUT 5	출력 5
7	COMMON	6 점 공통 접지선
8	24V (-)	DC 24V - 입력
9	24V (+)	DC 24V + 입력

3.3. ETOS-50A/100A/150 배선방법

3.3.1. ETOS-50A/100A/150 PGM 포트

PGM 포트는 컴퓨터의 COM(RS-232C)포트와 연결하여 프로그램 읽기/쓰기, 프레임 모니터 등의 용도로 사용됩니다. ETOS 기종에 따라 D-SUB9 (Female) 커넥터 또는 RJ-45 커넥터를 이용하며, RS-232C 신호선 중 TX, RX, GND 3 개의 신호선만을 사용합니다.

(ETOS-50A 시리얼 모델은 RJ-45 커넥터를 PGM 포트에 사용하며 핀 배열은 3.3.2 절의 RJ-45 커넥터 핀번호와 동일합니다.)

표 3.3-1 ETOS-50A/100A/150 PGM 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
2	RxD	←	RS232C 수신 데이터 신호
3	TxD	→	RS232C 송신 데이터 신호
5	SG	-	신호용 공통 접지선
1,4,6,7,8,9	-		사용 않음

3.3.2. ETOS-50A/100A/150 이더넷 포트

ETOS-50A/100A/150 이더넷 포트는 RJ-45 커넥터를 사용합니다. 핀 배열은 다음과 같습니다.

RJ-45 커넥터(Plug End)의 핀 번호는 배면(접촉용 금속이 노출되어 있는 쪽)을 기준으로 왼쪽이 1 번이고, 오른쪽이 8 번입니다.



그림 3.3-1 RJ-45 커넥터(Plug End) 핀 번호

표 3.3-2 ETOS-50A/100A/150 이더넷 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
1	TD +	→	이더넷 송신 (+)극성
2	TD -	→	이더넷 송신 (-)극성
3	RD +	←	이더넷 수신 (+)극성
4	-		사용 않음
5	-		사용 않음
6	RD -	←	이더넷 수신 (-)극성
7	-		사용 않음
8	-		사용 않음

3.3.3. ETOS-50A/100A/150 시리얼 포트

ETOS-50A/100A/150 는 시리얼 통신 포트로서 D-SUB9 (Female) 커넥터를 사용하며 핀 설정은 모든 기종이 같습니다. 시리얼 통신방식은 RS232C 와 RS422/485 의 두가지 방식이 있습니다. RS232C 와 RS422 은 통신방식이 전혀 다른 통신사양으로, ETOS 에서는 각각의 포트가 두가지 사양중 하나만을 지원하도록 되어 있으므로, 필요 사양에 맞게 RS232C 또는 RS422 방식으로 선택하여 구입을 하셔야 합니다. 두 가지 모드로 동시 사용은 불가능하며 반드시 RS232C 또는 RS422/485 중 한가지로만 동작 합니다. RS422 통신은 RS485 통신과 방식이 유사하므로 결선을 변경하여 RS485 통신포트로 사용할 수 있습니다.

3.3.3.1. 시리얼 포트 규격 및 배선

다음 표는 ETOS-50A/100A/150 시리얼 포트 각 핀의 명칭 및 기능과 데이터 방향을 나타냅니다.

표 3.3-3 ETOS-50A/100A/150 시리얼 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
1	CD	←	DCE 가 캐리어 검출을 DTE 에 알림
2	RXD	←	수신 데이터 신호
3	TXD	→	송신 데이터 신호
4	DTR	→	DTE 가 통신 준비되었음을 DCE 에게 알림
5	SG	-	신호용 접지선
6	DSR	←	DCE 가 통신 준비되었음을 DTE 에게 알림
7	RTS	→	DTE 가 DCE 에게 데이터 송신을 요구
8	CTS	←	DCE 가 DTE 에게 데이터 송신가능을 알림
9	N/A	-	-

▶ 전용모뎀 접속시의 RS232C 커넥터 접속 방법.

ETOS RS232C 포트는 모뎀을 접속하여 장거리 통신이 가능하며, 전용모뎀과 ETOS 의 RS232C 포트는 다음 그림과 같이 연결합니다. 모뎀이 2선식인지 4선식인지에 따라 결선이 달라지므로 주의하시기 바랍니다.

전용모뎀(2선식, Full Connect) ^(*36)

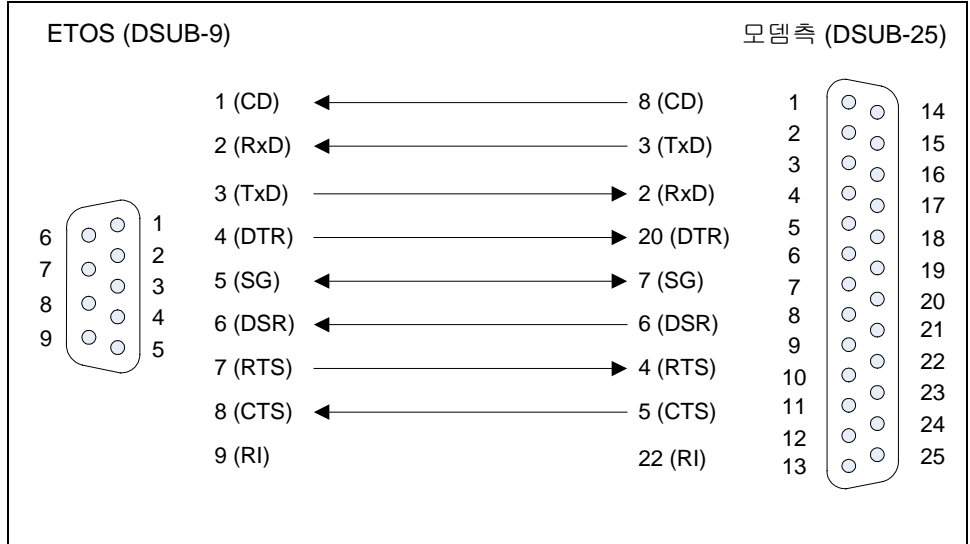


그림 3.3-1 RS232C 전용모뎀 결선도 (2선식)

^(*36) ETOS 측은 전용모뎀으로 설정

전용모뎀(4선식, 널모뎀 접속) ^{(*37) (*38)}

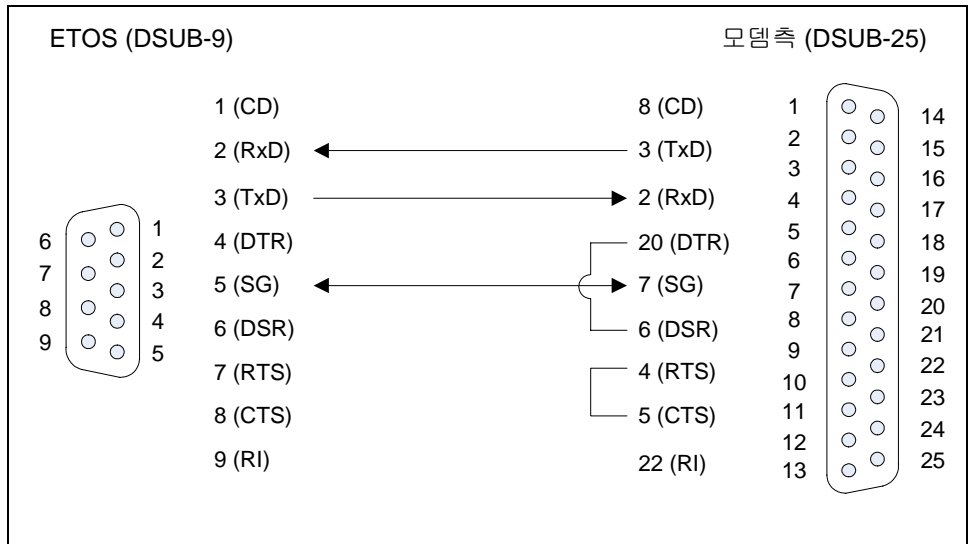


그림 3.3-2 RS232C 전용모뎀 결선도 (4선식)

^(*37) ETOS 측은 널모뎀으로 설정

^(*38) 모뎀측은 DTR, RTS 신호를 강제 ON 해야 합니다.

▶ 널(Null)모뎀 접속시의 RS232C 커넥터 접속 방법.

ETOS RS232C 포트를 외부기기와 모뎀을 거치지 않고 직접 연결하는 방식을 널 모뎀 통신이라 하며, 외부 기기와 ETOS의 RS232C 포트는 다음 그림과 같이 연결합니다.

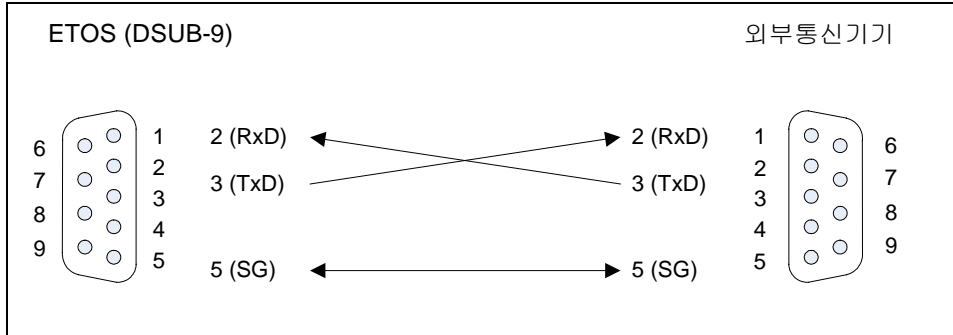


그림 3.3-3 RS232C 널모뎀 결선도

- * ETOS에서는 9번 RI 신호를 사용하지 않습니다.
- * 외부 기기에서 하드웨어적인 Handshake를 사용할 경우 외부기기의 핀을 강제로 접속해야 하는 경우가 있으며 외부 통신기기의 설명서를 참조하십시오.

3.3.3.2. RS422 포트 규격 및 배선

다음 표는 ETOS-50A/100A/150 RS422 통신 포트의 핀 명칭 및 기능과 데이터 방향을 나타냅니다.

표 3.3-4 ETOS-50A/100A/150 의 RS422 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
2	RXD+	←	RS422 수신 데이터(+)
3	TXD+	→	RS422 송신 데이터(+)
5	SG ^(*39)	-	신호용 접지선
7	TXD-	→	RS422 송신 데이터(-)
8	RXD-	←	RS422 수신 데이터(-)
1,4,6,9			사용 없음

다음 그림은 ETOS-50A/100A/150 RS422 통신 포트와 외부기기와의 결선 방법을 나타냅니다.

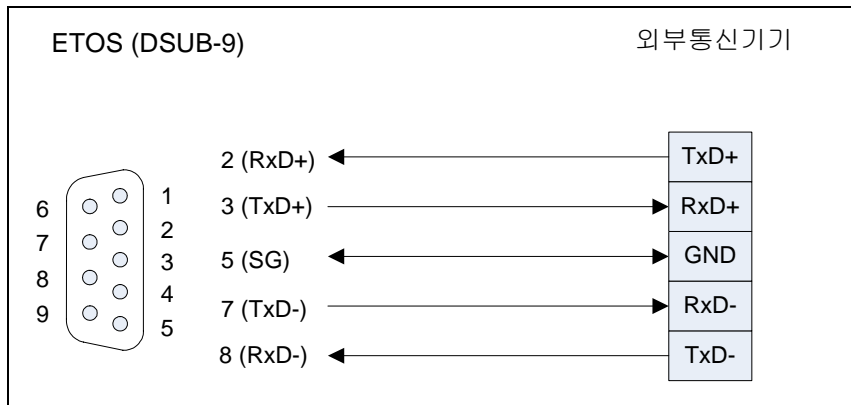


그림 3.3-4 ETOS-50A/100A/150 RS422 결선도

(*39) 5번 신호선은 Signal Ground 신호로 상대 기기에 GND 단자가 있을 경우 연결합니다. GND 신호는 상대기기와의 전위차를 맞추기 위한 단자입니다.

3.3.3.3. RS485 포트 규격 및 배선

다음 표는 ETOS-50A/100A/150 RS485 통신 포트의 핀 명칭 및 기능과 데이터 방향을 나타냅니다.

표 3.3-5 ETOS-50A/100A/150 RS485 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
2-3 ^(*40)	TRX+	↔	RS485 송수신 데이터(+)
5 ^(*41)	SG	-	신호용 접지선
7-8 ^(*1)	TRX-	↔	RS485 송수신 데이터(-)
1,4,6,9			사용 없음

다음 그림은 ETOS-50A/100A/150 RS485 통신 포트와 외부기기와의 결선 방법을 나타냅니다.

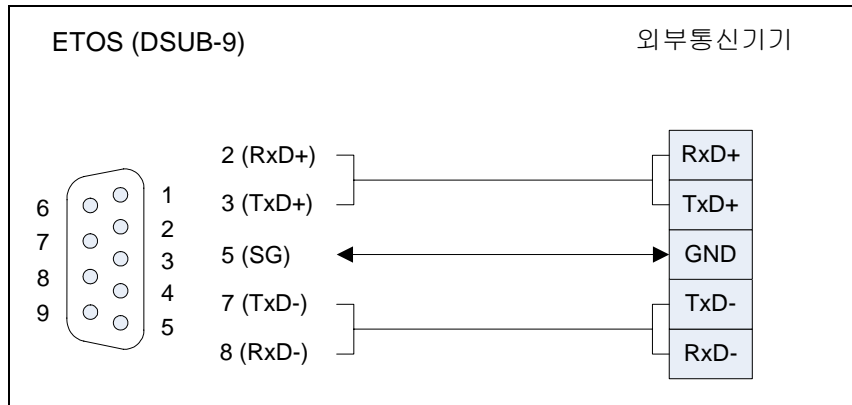


표 3.3-6 ETOS-50A/100A/150의 RS485 결선

(*40) 2,3 번 핀을 Short 시켜 상대국의 송수신 +신호선에 연결하고 7,8 번 핀은 short 시켜 상대국의 - 신호선에 연결합니다.

(*41) 5 번 신호선은 Signal Ground 신호로 상대 기기에 GND 단자가 있을 경우 연결합니다. GND 신호는 상대기기와의 전위차를 맞추기 위한 단자입니다.

3.3.3.4. 종단저항

RS422 포트를 통하여 통신할 경우 반드시 외부에 종단저항을 연결해 주어야 합니다. 종단저항은 장거리 통신을 할 때, 케이블의 반사파에 의한 신호 왜곡을 방지하는 역할을 하는 것으로 케이블의 특성 임피던스 값과 동일값의 저항(1/2W)을 네트워크의 종단에 연결하여 주어야 합니다. 권장 케이블을 사용할 경우에는 120Ω 종단저항을 선로양단에 연결하여 주십시오. 권장 케이블 이외의 케이블을 사용시에도 사용 케이블의 특성 임피던스 값과 동일한 값의 1/4W 저항을 선로양단에 연결하여 주십시오. 다음 그림은 각각 RS422와 RS485 연결시의 종단저항 연결방법입니다.

▶ RS422 종단 저항

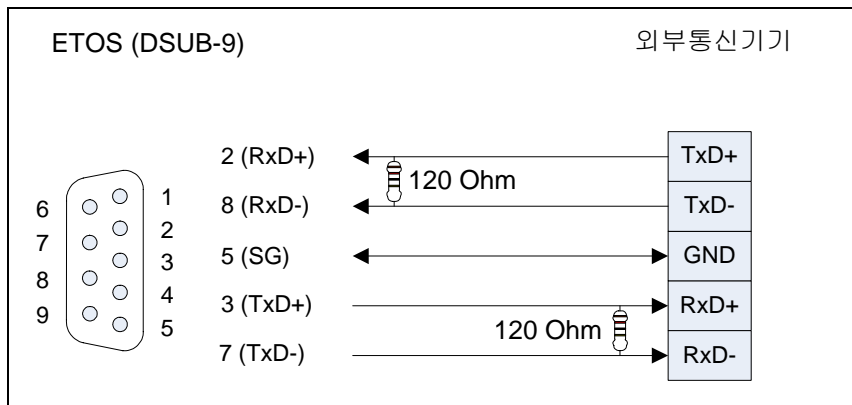


그림 3.3-5 RS422 종단저항 연결 방법

▶ RS485 종단 저항

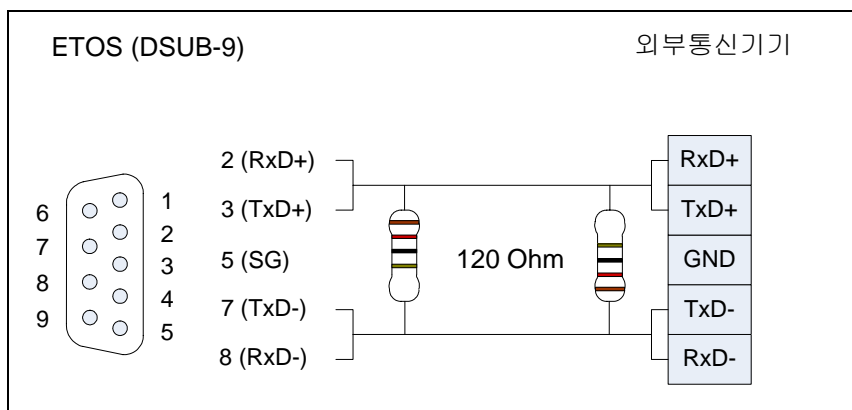


그림 3.3-6 RS485 종단저항 연결 방법

3.4. ETOS-200/1000 설치 및 배선

ETOS-200/1000 의 전원 규격 및 통신 포트 규격을 설명합니다.

3.4.1. ETOS-200 전원 규격

입력전원 : AC 85V~260V (Free Voltage), 50/60 Hz

출력 : DC 5V/7A

전원은 반드시 접지(F.G)가 연결되어야 하고, 제 3 종 접지 규격이어야 합니다.

3.4.2. ETOS-1000 전원 규격

입력전원 : AC 110V/220V (전압 선택형, 출하 시 220V), 50/60 Hz

출력 : DC 5V/15A, +12V/1.5A , -12V/1.5A

전원은 반드시 접지(F.G)가 연결되어야 하고, 제 3 종 접지 규격이어야 합니다.

3.4.3. ETOS-200/1000 CPU 모듈 커넥터 규격

다음 그림은 ETOS-200/1000 CPU 모듈의 전면 패널 모습입니다.

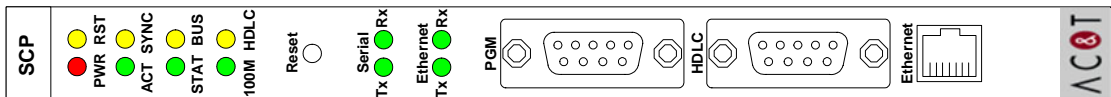


그림 3.4-1 ETOS-200/1000 CPU 전면 패널

3.4.3.1. ETOS-200/1000 CPU 모듈 PGM 포트

ETOS-200/1000 CPU 모듈의 PGM 포트는 컴퓨터의 COM(RS232C)포트와 연결합니다. 프로그램 읽기/쓰기, 프레임 모니터 등의 온라인 연결 전용 포트입니다. D-SUB 9 (Female) 커넥터를 사용하며, RS232C 신호선 중 TX, RX, GND 3 개의 신호선만을 사용합니다.

표 3.4-1 ETOS-200/1000 CPU 모듈 PGM 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
2	RxD	←	RS232C 수신 데이터 신호
3	TxD	→	RS232C 송신 데이터 신호
5	SG	-	신호용 공통 접지선
1,4,6,7,8,9	-		사용 없음

3.4.3.2. ETOS-200/1000 CPU 모듈 HDLC 포트

Shelf 이중화 용의 포트입니다. 현재 지원되지 않습니다. (추후 지원 예정)

3.4.3.3. ETOS-200/1000 CPU 모듈 이더넷 포트

ETOS-200/1000 이더넷 포트는 RJ-45 커넥터를 사용합니다. 핀 배열은 다음과 같습니다.

RJ-45 커넥터(Plug End)의 핀 번호는 배면(접촉용 금속이 노출되어 있는 쪽)을 기준으로 왼쪽이 1 번이고, 오른쪽이 8 번입니다.



그림 3.4-1 RJ-45 커넥터(Plug End) 핀 번호

표 3.4-2 ETOS-200/1000 CPU 모듈 이더넷 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
1	TD +	→	이더넷 송신 (+)극성
2	TD -	→	이더넷 송신 (-)극성
3	RD +	←	이더넷 수신 (+)극성
4	-		사용 없음
5	-		사용 없음
6	RD -	←	이더넷 수신 (-)극성
7	-		사용 없음
8	-		사용 없음

3.4.4. ETOS-200/1000 시리얼 통신 모듈 커넥터 규격

다음 그림은 ETOS-200/1000 시리얼 통신 모듈의 전면 패널 모습입니다.

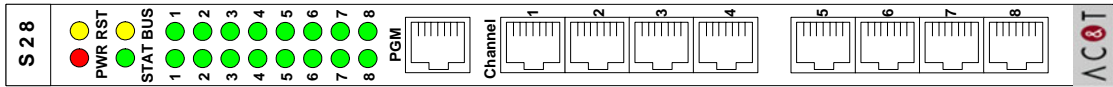


그림 3.4-2 ETOS-200/1000 시리얼 통신 모듈 전면 패널

3.4.4.1. ETOS-200/1000 시리얼 통신 모듈 PGM 포트

ETOS-200/1000 시리얼 모듈 전면 패널에 위치한 PGM 포트는 시리얼 모듈의 OS 다운로드나 시스템 정보 읽기, 프레임 모니터 등의 온라인 연결 전용 포트입니다. 그러나, CPU 모듈이 있는 경우, 위에 나열한 온라인 기능들은 모두 CPU 모듈의 PGM 포트를 통해 할 수 있습니다. CPU 모듈이 고장 등으로 기능을 하지 못하는 경우 이외에는 시리얼 통신 모듈의 PGM 포트는 사용하지 않습니다.

ETOS-200/1000 시리얼 통신 모듈의 PGM 포트는 이더넷 커넥터와 동일한 RJ-45 커넥터를 사용합니다. 핀 배열은 다음과 같습니다.

표 3.4-3 ETOS-200/1000 시리얼 통신 모듈 PGM 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
2	RxD	←	RS232C 수신 데이터 신호
3	TxD	→	RS232C 송신 데이터 신호
5	SG	-	신호용 공통 접지선
1,4,6,7,8	-		사용 않음

3.4.4.2. ETOS-200/1000 시리얼 통신 모듈 시리얼 포트

시리얼 통신 모듈은 RS232C 와 RS422/485 통신용 포트를 제공합니다. 각각의 포트는 RS232C 또는 RS422 중 한가지로만 동작합니다. (*42)

▶ 전면 RJ-45 커넥터

시리얼 통신 모듈내의 도터(dauther) 보드의 위치를 바꿈으로써, 전면의 RJ-45 커넥터 대신 후면의 P2 Back Plane 커넥터로 시리얼 통신 입, 출력을 전환할 수 있습니다. (*43) (P2 Back Plane 커넥터의 A 열 및 C 열). 그러나 이경우 전면 커넥터로는 입, 출력이 불가능합니다. (두 개 커넥터 중 한 가지만 사용 가능)



그림 3.4-1 RJ-45 커넥터(Plug End) 핀 번호

▶ 시리얼 포트 규격

다음 표는 ETOS-200/1000 시리얼 포트 각 핀의 명칭 및 기능과 데이터 방향을 나타냅니다. (RJ-45 커넥터의 핀 배열입니다.)

표 3.4-4 ETOS-200/1000 시리얼 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
1	CD	←	DCE 가 캐리어 검출을 DTE 에 알림
2	RXD	←	수신 데이터 신호
3	TXD	→	송신 데이터 신호
4	DTR	→	DTE 가 통신 준비되었음을 DCE 에게 알림
5	SG	-	신호용 접지선
6	DSR	←	DCE 가 통신 준비되었음을 DTE 에게 알림
7	RTS	→	DTE 가 DCE 에게 데이터 송신을 요구
8	CTS	←	DCE 가 DTE 에게 데이터 송신가능을 알림

(*42) 포트 구성은 구입시에 결정합니다. 선택시에는 2 포트 단위로 포트의 종류를 변경할 수 있습니다. 예를 들어, 1,2 번 포트는 RS232C, 3,4 번 포트는 RS422/485 와 같이 지정할 수 있습니다.

(*43) 후면 커넥터로 입, 출력이 필요한 경우 제조사로 문의 바랍니다.

▶ ETOS-200/1000 P2 Back Plane 커넥터

P2 커넥터의 핀 배열은 다음과 같습니다

번호	A 열	C 열	Channel
1	RTS0	DTR0	COM1
2	TXD0	DCD0	
3	RXD0	CTS0	
4	DSR0	GND0	
5	RTS1	DTR1	COM2
6	TXD1	DCD1	
7	RXD1	CTS1	
8	DSR1	GND1	
9	RTS2	DTR2	COM3
10	TXD2	DCD2	
11	RXD2	CTS2	
12	DSR2	GND2	
13	RTS3	DTR3	COM4
14	TXD3	DCD3	
15	RXD3	CTS3	
16	DSR3	GND3	
17	RTS4	DTR4	COM5
18	TXD4	DCD4	
19	RXD4	CTS4	
20	DSR4	GND4	
21	RTS5	DTR5	COM6
22	TXD5	DCD5	
23	RXD5	CTS5	
24	DSR5	GND5	
25	RTS6	DTR6	COM7
26	TXD6	DCD6	
27	RXD6	CTS6	
28	DSR6	GND6	
29	RTS7	DTR7	COM8
30	TXD7	DCD7	
31	RXD7	CTS7	
32	DSR7	GND7	

▶ 전용모뎀 접속시의 RS232C 커넥터 접속 방법.

ETOS RS232C 포트는 모뎀을 접속하여 장거리 통신이 가능하며, 전용모뎀과 ETOS 의 RS232C 포트는 다음 그림과 같이 연결합니다. 모뎀이 2선식인지 4선식인지에 따라 결선이 달라지므로 주의하시기 바랍니다.

전용모뎀(2선식, Full Connect) (*44)

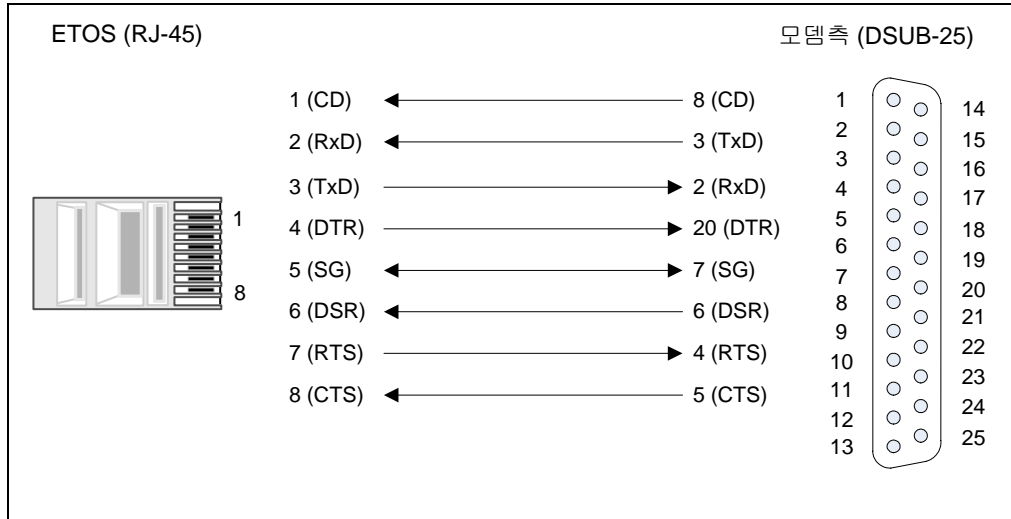


그림 3.4-3 RS232C 전용모뎀 결선도 (2선식)

(*44) ETOS 측은 전용모뎀으로 설정

전용모뎀(4선식, 널모뎀 접속) (*45) (*46)

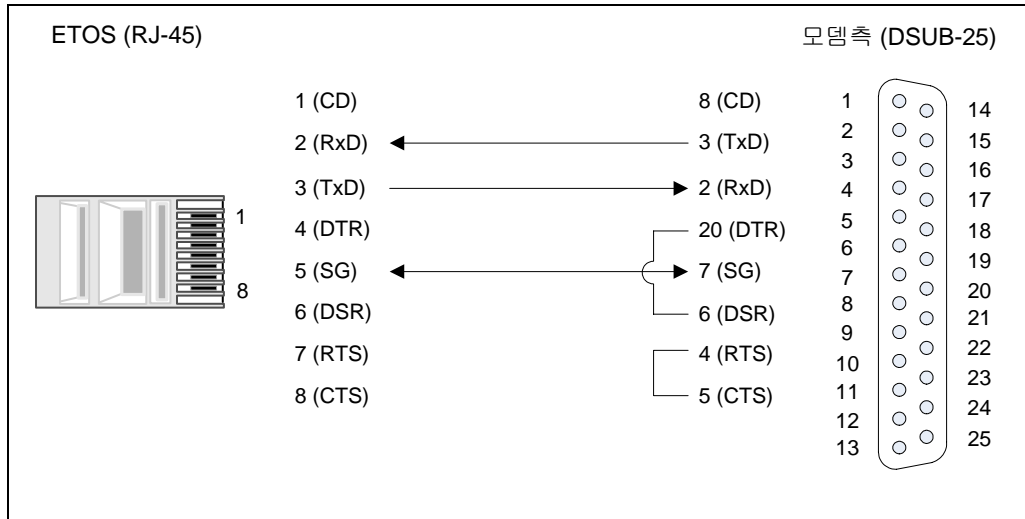


그림 3.4-4 RS232C 전용모뎀 결선도 (4선식)

(*45) ETOS 측은 널모뎀으로 설정

(*46) 모뎀측은 DTR, RTS 신호를 강제 ON 해야 합니다.

▶ 널(Null)모뎀 접속시의 RS232C 커넥터 접속 방법.

ETOS RS232C 포트를 외부기기와 모뎀을 거치지 않고 직접 연결하는 방식을 널 모뎀 통신이라 하며, 외부 기기와 ETOS의 RS232C 포트는 다음 그림과 같이 연결하여야 합니다.

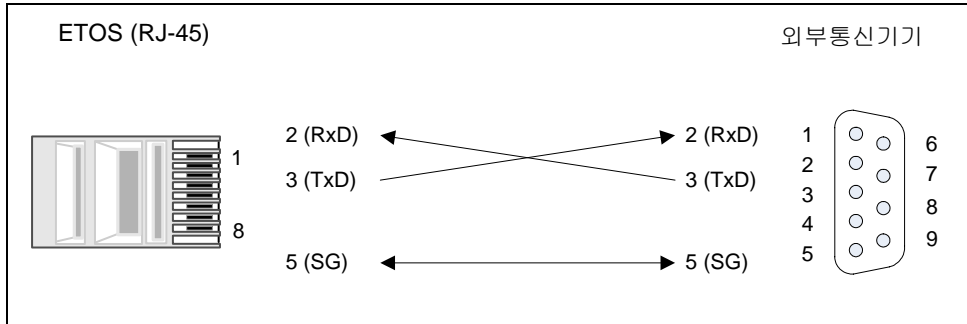


그림 3.4-5 RS232C 널모뎀 결선도

* 외부 기기에서 하드웨어적인 Handshake 를 사용할 경우 외부기기의 핀을 강제로 접속해야 하는 경우가 있으며 외부 통신기기의 설명서를 참조하십시오.

3.4.4.3. RS422 포트 규격 및 배선

다음 표는 ETOS-200/1000 RS422 포트의 핀 명칭 및 기능과 데이터 방향을 나타냅니다.

표 3.4-5 ETOS-200/1000 의 RS422 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
2	RXD+	←	RS422 수신 데이터(+)
3	TXD+	→	RS422 송신 데이터(+)
5	SG ^(*47)	-	신호용 접지선
7	TXD-	→	RS422 송신 데이터(-)
8	RXD-	←	RS422 수신 데이터(-)
1,4,6			사용 없음

다음 그림은 ETOS-200/1000 RS422 통신 포트와 외부기기와의 결선 방법입니다.

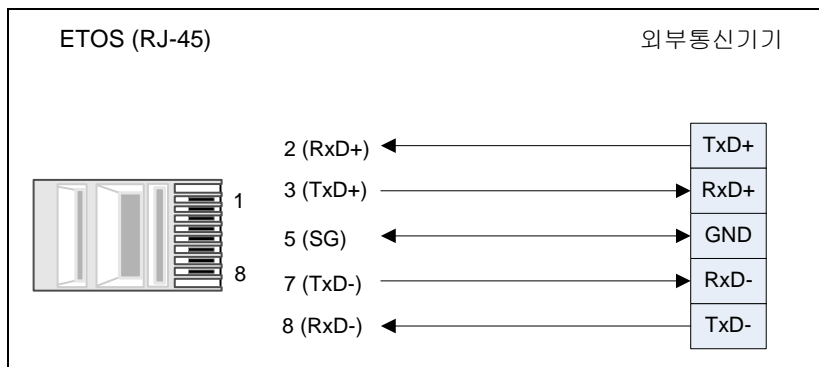


그림 3.4-6 ETOS-200/1000 RS422 결선도

(*47) 5 번 신호선은 Signal Ground 신호로 상대 기기에 GND 단자가 있을 경우 연결합니다. GND 신호는 상대기기와의 전위차를 맞추기 위한 단자입니다.

3.4.4.4. RS485 포트 규격 및 배선

다음 표는 ETOS-200/1000 RS485 통신 포트의 핀 명칭 및 기능과 데이터 방향을 나타냅니다.

표 3.4-6 ETOS-200/1000 RS485 포트 사양

핀 번호	이름	신호 방향 ETOS - 외부기기	기능 설명
2-3 (*48)	TRX+	↔	RS485 송수신 데이터(+)
5 (*49)	SG	-	신호용 접지선
7-8 (*1)	TRX-	↔	RS485 송수신 데이터(-)
1,4,6			사용 않음

다음 그림은 ETOS-200/1000 RS485 통신 포트와 외부기기와의 결선 방법을 나타냅니다.

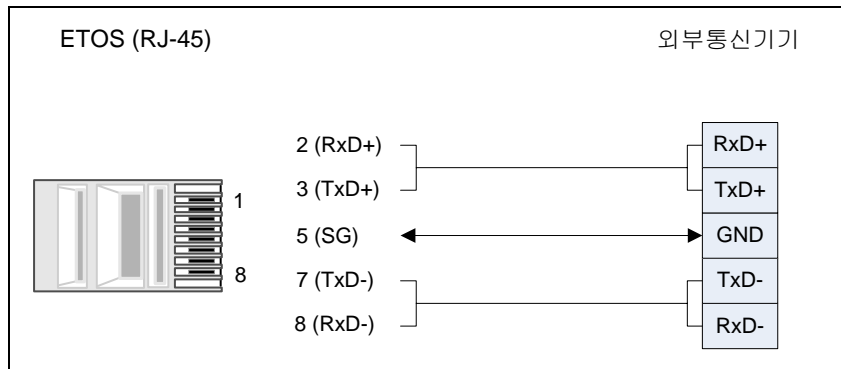


그림 3.4-7 ETOS-200/1000 RS485 결선도

(*48) 2,3 번 핀을 Short 시켜 상대국의 송수신 +신호선에 연결하고 7,8 번 핀은 상대국의 - 신호선에 연결합니다.

(*49) 5 번 신호선은 Signal Ground 신호로 상대 기기에 GND 단자가 있을 경우 연결합니다. GND 신호는 상대기기와의 전위차를 맞추기 위한 단자입니다.

ETOS-PD 사용법

제 4 장

기본적인 ETOS-PD 사용법에 대해서 설명합니다.

	내용	위치
4.1	개요	52
4.2	시스템 환경 및 설치	54
4.3	화면 설명	54

4. ETOS-PD 사용법

4.1. 개요

ETOS-PD (Protocol Designer) 는 ETOS 를 운전하기 위해 꼭 필요한 시스템 설정, 프레임, 프로세스를 편집, 수정할 수 있는 Windows Software 입니다. 이 장은 ETOS-PD 를 컴퓨터에 설치하는 방법과 기본적인 ETOS-PD 사용법을 설명합니다. 이 장의 내용은 ETOS-PD V2.3 을 기준으로 작성한 것으로 버전에 따라 다소 차이가 있을 수 있습니다. (최신 버전의 ETOS-PD 를 에이씨앤티시스템 홈페이지 <http://www.acnt-sys.co.kr> 에서 다운로드 받으실 수 있습니다.)

4.1.1. ETOS-PD 의 기능 및 특징

- * ETOS 시스템 설정기능(시각/이더넷 IP 주소/통신 포트 종류/사용 프로토콜 종류 등)
- * 프레임 편집기능
- * ETOS 의 동작을 정의하는 프로세스 편집기능
- * 데이터 백업이 용이한 단일 파일형태의 프로그램 파일 관리
- * 데이터 확인을 위한 사용자 메모리 모니터
- * 통신 상태를 확인할 수 있는 프레임 모니터기능
- * SECS 메시지 편집기능

4.1.2. 파라미터, 프레임 및 프로세스

ETOS 가 동작하기 위해서는 파라미터, 프레임, 그리고 프로세스의 세가지가 필요합니다. 이 중 파라미터는 반드시 있어야 하며, 프레임과 프로세스는 경우에 따라 필요합니다.

4.1.2.1. 파라미터

시리얼, 이더넷, 필드버스 등의 통신포트 설정, 이중화 시스템 설정 등 시스템의 전반적인 구성입니다.

4.1.2.2. 통신 프레임

이더넷 및 시리얼 포트에서 사용할 프로토콜을 구성하는 단위 메시지입니다.

4.1.2.3. 사용자 프로세스

파라미터 설정을 통해 상대 장비와의 통신 프로토콜을 결정했으면, 이제 상대 장비로부터 실제로 데이터를 가져오거나 보내야 합니다. 프로세스는 데이터를 보내거나 가져오는 구체적인 동작을 정의하며, 통신 동작 뿐 아니라 송,수신 데이터의 가공, 편집 등도 정의합니다. 사용자는 프로세스의 이러한 기능을 통해 두 개의 서로 다른 프로토콜을 사용하는 장비간의 데이터 교환을 정의할 수 있습니다.

다음은 ETOS-PD 를 이용하여 ETOS 의 동작에 필요한 파라미터, 프레임 및 프로세스를 설정하는 일반적인 순서를 나타낸 것입니다.

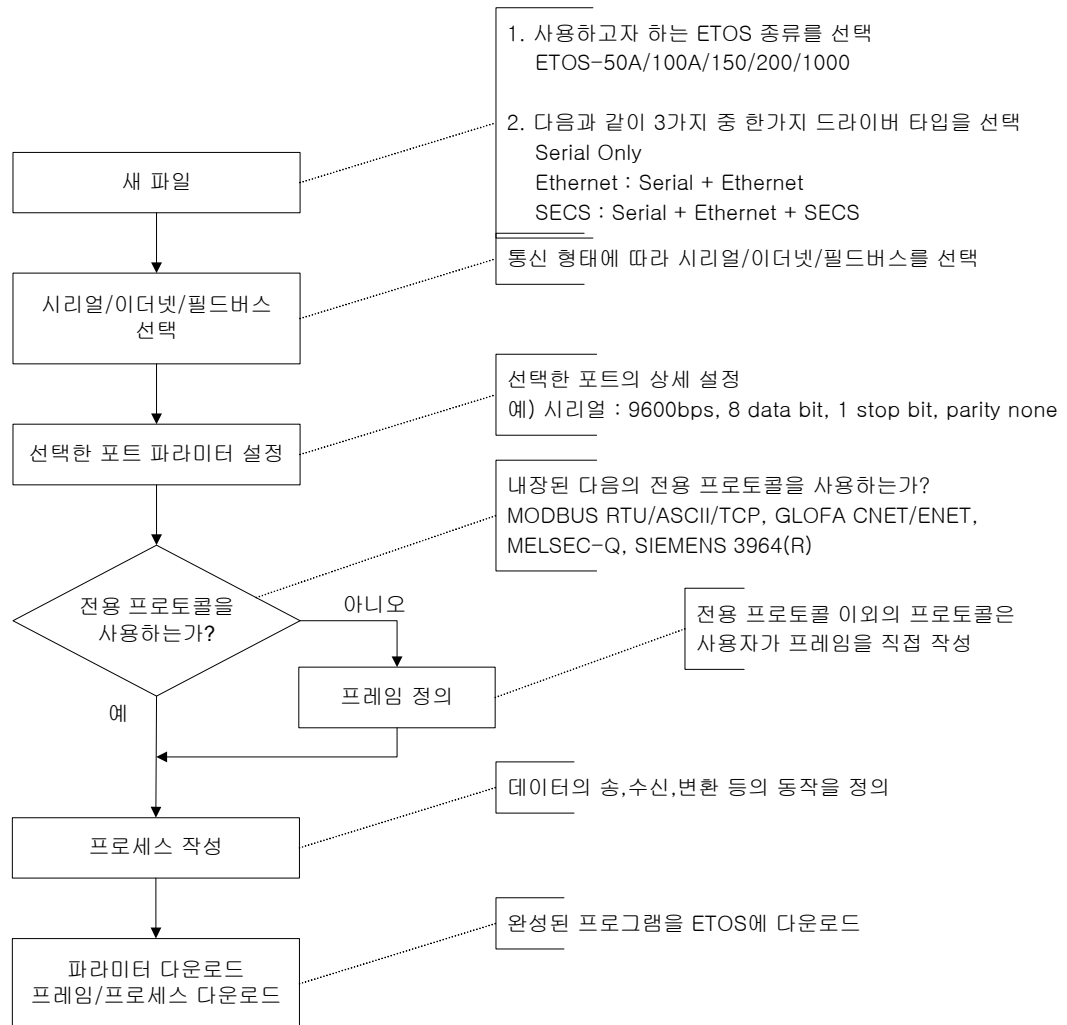


그림 4.1-1 ETOS-PD 작업 순서도

4.2. 시스템 환경 및 설치

ETOS-PD 를 컴퓨터에 설치하기 위한 컴퓨터 환경과 설치 방법에 대해 설명합니다.

4.2.1. 시스템 환경

CPU: Pentium 이상
 MEMORY: 32MBytes
 OS: MS Windows 95/98/ME/NT/2000/XP

4.2.2. 설치 방법

ETOS-PD 는 별도 인스톨버전으로 제공되지 않으며, 압축파일 형태로 제공되는 파일을 폴더에 복사하여 사용하면 됩니다. (ETOS-PD 는 에이씨앤티시스템 홈페이지 <http://www.acnt-sys.co.kr> 에서 다운로드 받으실 수 있습니다.)

4.3. 화면 설명

ETOS-PD 화면 내용에 대한 설명입니다.

4.3.1. ETOS-PD 기본화면

다음은 ETOS-PD 를 실행한 직후의 화면입니다. ETOS-PD V2.3^(*50)부터는 프로그램 시작시에 새파일을 기본적으로 열도록 변경되었습니다. 새파일 열기에서 ETOS의 종류(ETOS-50/100/150/200/1000)와 드라이버의 종류(Serial Only/Ethernet/SECS)를 동시에 설정합니다.

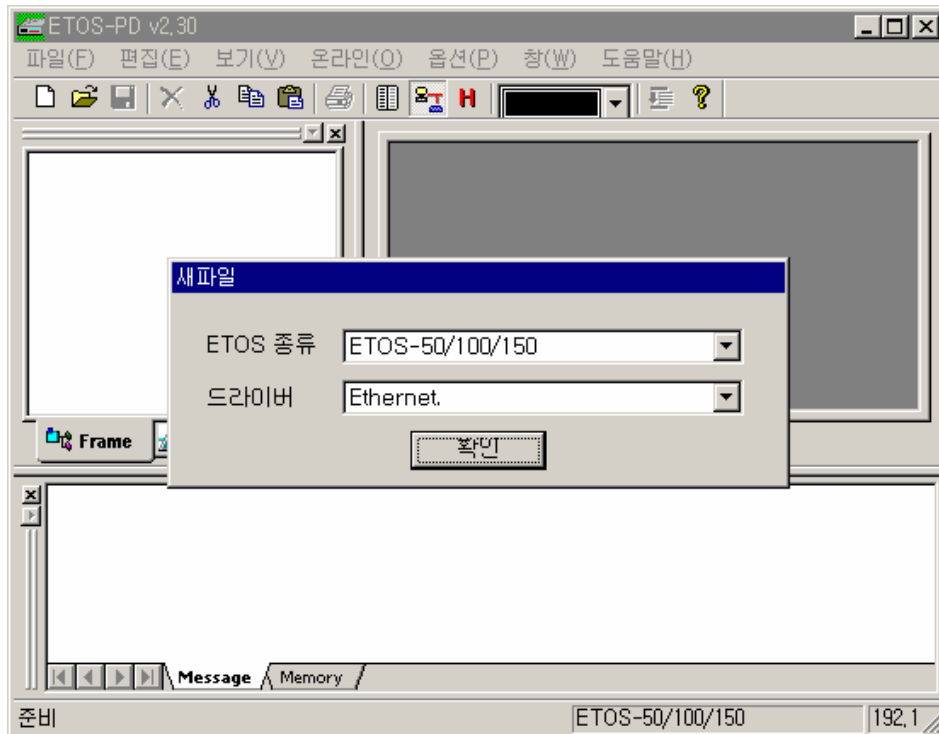


그림 4.3-1 ETOS-PD 의 기본 화면

(*50) ETOS-PD V2.3 을 기준으로 작성되었습니다. 이전 버전과 다소의 차이가 있으므로, 최신 버전을 다운로드하시기 바랍니다.

4.3.2. 화면 구성

파일을 연 상태에서의 화면 구성입니다. 좌측의 {프레임 등록}/{프로세스 등록} 창, 우측의 {프레임 편집}/{프로세스 편집} 창, 하단의 {메시지}/{메모리} 출력 창으로 이루어져 있습니다.
(화면의 실제 데이터는 열려있는 파일에 따라 달라집니다.)

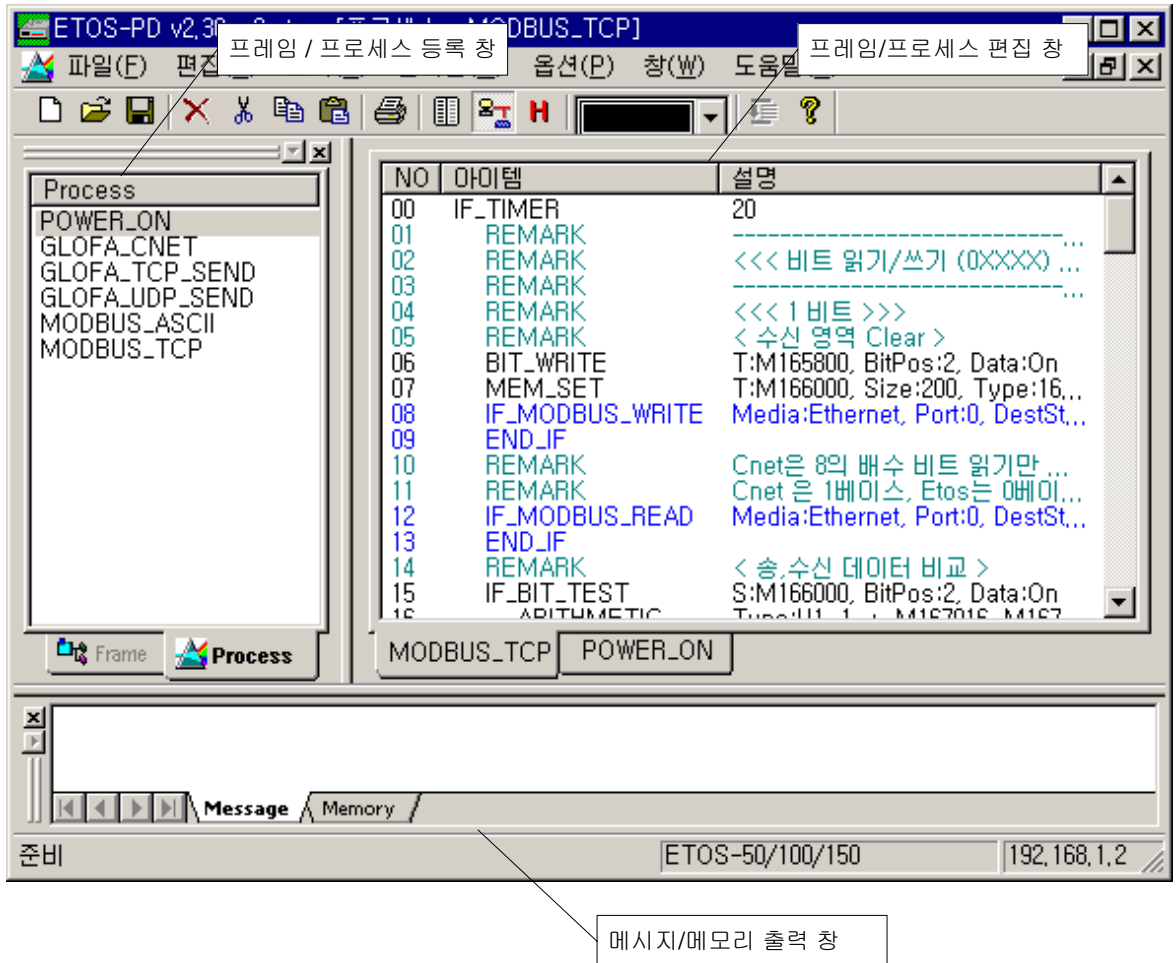


그림 4.3-2 ETOS-PD 화면 구성

4.3.3. 메뉴 구성

ETOS-PD 메뉴에 대한 간략한 설명입니다. 보다 상세한 설명은 각 메뉴의 상세 설명을 참조하시기 바랍니다.

4.3.3.1. 파일(F)

하위 메뉴	설 명	비 고
새 파일 (N)	새 문서를 작성합니다.	Ctrl-N (단축키)
열기 (O)	기존 문서를 엽니다.	Ctrl-O (단축키)
저장 (S)	현재 열린 문서를 저장합니다.	Ctrl-S (단축키)
다른 이름으로 저장 (A)	현재 열린 문서를 다른 이름으로 저장합니다.	
인쇄 (P)	현재 문서를 프린터로 인쇄하거나 텍스트 파일 형태로 봅니다.	
종료 (X)	현재 열린 문서를 닫고, 프로그램을 종료합니다.	

4.3.3.2. 편집(E)

하위 메뉴	설 명
파라미터 설정	시리얼, 이더넷, 이중화 파라미터를 설정합니다.
복사(C)	선택 부분을 복사하여 클립보드에 넣습니다.
붙여넣기(T)	클립보드의 내용을 붙여 넣습니다.
잘라내기(P)	선택 부분을 잘라내어 클립보드에 넣습니다.
삭제(D)	선택 부분을 삭제합니다.

4.3.3.3. 보기(V)

하위 메뉴	설 명
도구 모음 (I)	도구 모음 보이거나 감춥니다.
상태 표시줄 (S)	상태 표시줄을 보이거나 감춥니다.
작업 창	작업 창을 보이거나 감춥니다.
출력 창	출력 창을 보이거나 감춥니다.
ASCII Table	16 진수, 10 진수, ASCII 변환 표를 보여줍니다.

4.3.3.4. 온라인(O)

하위 메뉴	설 명
파라미터 쓰기	시스템 기본정보 쓰기
프로그램 쓰기	프레임과 프로세스 쓰기(SECS 인 경우 메시지)
파라미터 읽기	ETOS 로부터 시스템 기본정보 읽기
프로그램 읽기	ETOS 로부터 프로그램 읽기
시스템 정보 읽기	ETOS 의 버전, MAC ID 등의 정보 읽기
메모리 저장	메모리를 텍스트 파일 포맷으로 저장
OS 업데이트	ETOS OS 업그레이드
파라미터, 프로그램 지우기	ETOS 에 저장된 파라미터, 프로그램 지우기
편집 내용 검사	편집하고 있는 파라미터, 프로그램 문법 체크
TCP / IP 설정	ETOS 의 TCP/IP 설정 읽기/ 쓰기
시간 설정	ETOS 의 현재 시각 읽기 / 쓰기
카드 리셋 (*51)	ETOS 시리얼 통신 모듈 리셋
암호 설정	타사용자의 접근을 막도록 암호 설정
암호 지우기	설정된 암호를 지움
온라인 연결 설정	온라인 연결용 포트 설정(PGM 포트)
프레임 모니터	시리얼 송,수신 프레임 모니터링
IFOR Shared Memory 설정 (*52)	네트워크 안의 모듈들이 서로 같은 공유메모리를 사용하도록 설정
IFOR Direct Pass 설정 (*2)	모듈의 시리얼/이더넷 포트를 원거리의 다른 모듈의 포트와 연결시 설정
IFOR 정보읽기 (*2)	모듈과 링의 상태를 모니터링
HSMS Diagnostic Status 읽기 (*53)	HSMS 의 진단 및 상태 모니터링

(*51) ETOS 종류가 ETOS-200/1000 인 경우만 메뉴가 활성화됩니다.

(*52) [옵션] -> [편집 옵션] 에서 IFOR 편집을 체크한 경우만 메뉴에 나타납니다.

(*53) [옵션] -> [편집 옵션] 에서 HSMS Diagnostic 편집을 체크한 경우만 메뉴에 나타납니다.

4.3.3.5. 옵션(P)

하위 메뉴	설 명
편집 옵션	특수 모듈 등의 편집 옵션 On/Off 합니다.
작업 폴더	임시파일이나 작업에 필요한 폴더 지정합니다.
16 진수 보기	편집 내용 중 "주소" 부분을 16 진수로 보여줍니다.

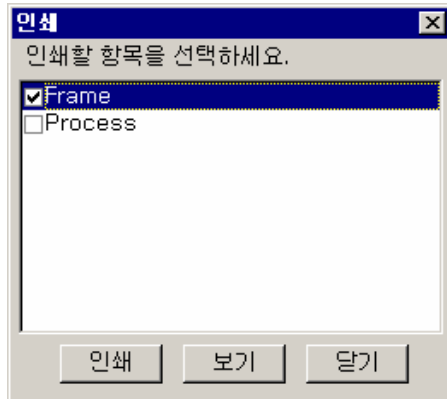
4.3.3.6. 도움말(H)

하위 메뉴	설 명
ETOS-PD 정보	ETOS-PD 버전 정보 표시합니다.

4.3.4. 메뉴 상세 설명

주요 메뉴에 대한 상세 설명입니다.

4.3.4.1. [파일(F)] -> [인쇄(P)]



ETOS-PD 에서 편집하고 있는 현재 파일을 프린터로 인쇄하거나 텍스트 파일 형태로 볼 수 있습니다. 프레임, 프로세스 중에서 보기를 원하는 항목을 선택한 후, [보기]를 클릭합니다. 다음 그림은 현재 편집 중인 파일의 프레임의 보기 예입니다.

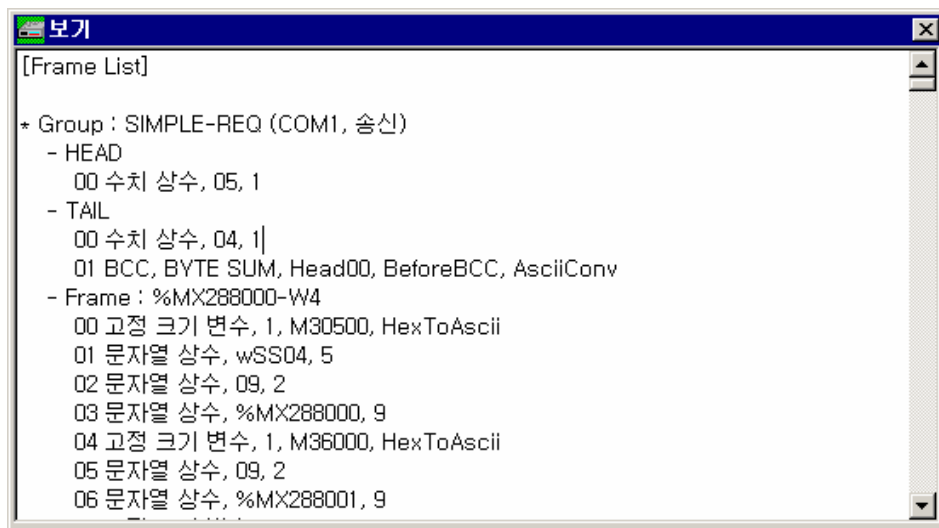


그림 4.3-3 프레임 보기

4.3.4.2. [옵션(P)] -> [편집 옵션]

ETOS-PD V2.3 부터 사용자들의 혼동을 막기 위해 특수 모듈 전용의 메뉴 및 아이템은 PD 화면에 나타나지 않습니다. 대신 특수 모듈을 사용하는 사용자는 {편집 옵션} 대화상자에서 편집하고자 하는 모듈을 체크하면, 관련된 메뉴 및 아이템을 편집할 수 있습니다. 각 모듈에 대한 자세한 정보는 제조사로 문의하시기 바랍니다.

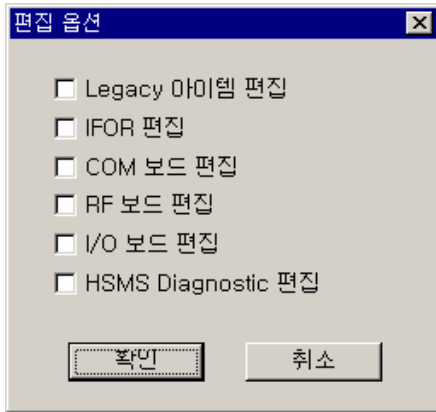


그림 4.3-4 편집 옵션 대화상자

표 4.3-1 편집 옵션 설정 항목

항 목	설 명
Legacy 아이템 편집	ETOS-50/100 단종 모델 및 ETOS-50A/100A/150/200/1000 초기버전에 사용
IFOR 편집	ETOS-200/1000 용 IFOR 보드 장착시
COM 보드 편집	
RF 보드 편집	
I/O 보드 편집	ETOS-200/1000 용 I/O 보드 장착시
HSMS Diagnostic 편집	모델명이 ETOS-150/200-DIAG인 경우 사용

4.3.4.3. [온라인(Q)] -> [온라인 연결 설정]

ETOS-PD 에서 편집한 파라미터나 프로그램을 ETOS 에 쓰거나, ETOS 로부터 필요한 정보를 읽기 위해서는 ETOS-PD 가 사용할 컴퓨터의 통신포트를 설정해야 합니다. 사용자는 시리얼 또는 이더넷 중 한가지 통신 방식을 선택해야 합니다.

다음 그림은 통신 방식으로 시리얼(RS232C)을 선택했을 경우의 예입니다. 시리얼의 경우 컴퓨터의 COM 포트(RS232C)는 ETOS 의 PGM 포트와 연결되어야 합니다. 그림의 COM 포트는 ETOS 가 아니라 컴퓨터의 COM 포트를 의미합니다. 사용하고자 하는 COM 포트를 선택하시기 바랍니다. 통신 속도 등의 값은 리모트 접속의 경우를 위해 변경할 수 있도록 되어 있으나, 로컬 접속인 경우 57600/8/1/None 으로 고정시켜야 합니다. 값이 다른 경우엔 ETOS 와의 통신이 불가능하니 주의하시기 바랍니다. [기본값]버튼을 눌러 기본값으로 설정할 수 있습니다.

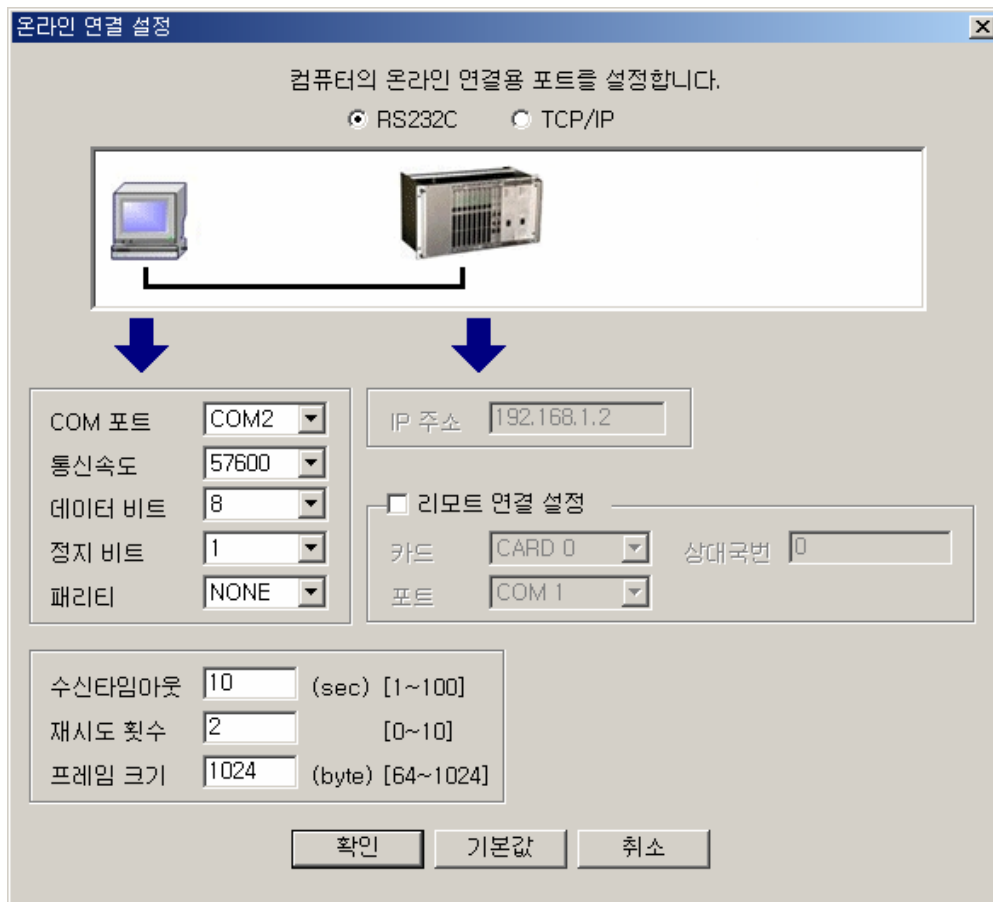


그림 4.3-5 온라인 연결 설정 화면

표 4.3-2 온라인 연결 설정 항목

항 목	설 명	
RS232C / TCP/IP	ETOS-PD가 실행되는 컴퓨터와 ETOS간의 통신 방식 선택	
COM 포트 설정 (*54)	COM포트	ETOS-PD가 실행되는 컴퓨터의 COM포트 선택
	통신 속도	57600bps 로 선택
	데이터 비트	8로 선택
	정지 비트	1로 선택
패리티	NONE 패리티 선택	
IP 주소	ETOS의 IP주소를 입력합니다. 통신 방식을 TCP/IP로 선택한 경우에만 활성화됩니다.	
수신타임아웃	ETOS로부터 응답을 기다리는 최대 시간을 초 단위로 설정	
재시도 횟수	ETOS로부터 응답이 없는 경우 재시도하는 횟수.	
프레임 크기	ETOS와 통신하는 프레임의 크기를 설정.	

▶ 리모트 접속

로컬에 있는 ETOS 를 거쳐 원격지 ETOS 에 다운로드나 메모리 읽기 등의 작업을 가능하게 하는 기능입니다. ETOS-PD 가 다운로드 하는 프레임은 사용자가 설정한 카드의 포트를 통해 원격지 ETOS 로 전달됩니다. (그림 참조)

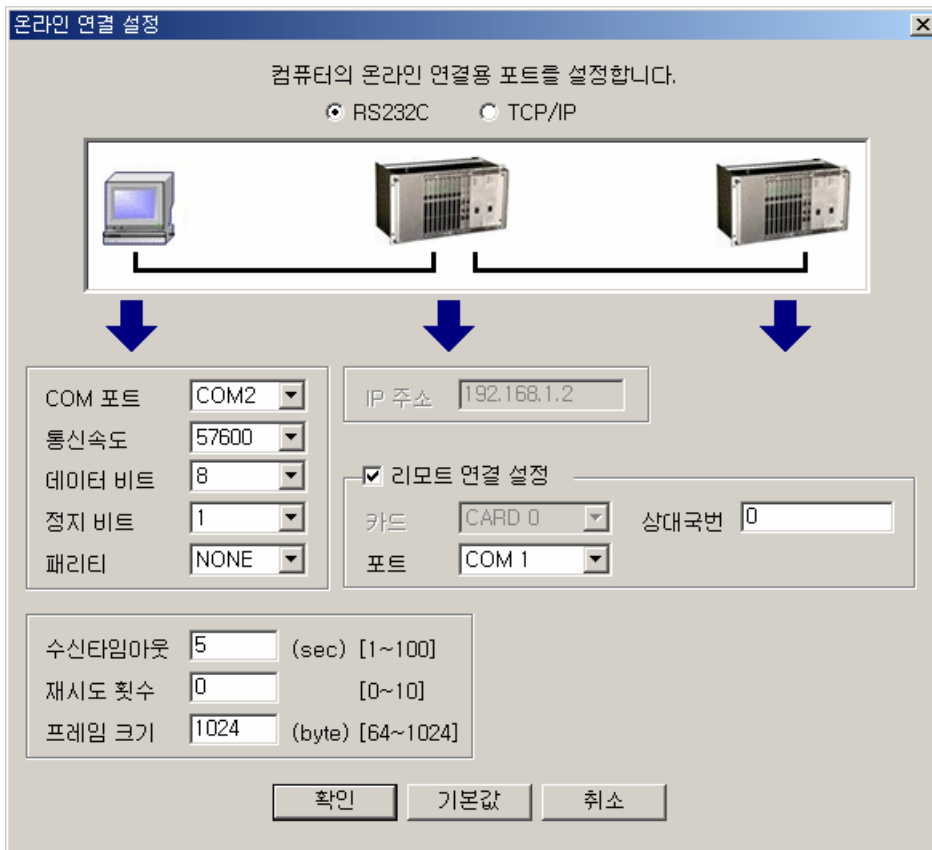


그림 4.3-6 리모트 연결 설정

(*54) 통신방식으로 RS232C 를 선택한 경우에만 활성화됩니다.

표 4.3-3 온라인 연결 설정 항목(리모트 연결)

항 목	설 명
리모트 연결 설정	원격지 ETOS에 접속하는 경우 선택
카드 (*55)	리모트 접속용 로컬 ETOS의 포트가 위치한 카드 선택
포트	리모트 접속용 로컬 ETOS의 포트 번호를 선택
상대국번	리모트 접속할 원격지의 ETOS 국번을 설정

일반적으로 수신타임아웃, 재시도 횟수 등의 설정치를 로컬 접속보다 크게 하여야 하며, 경우에 따라서는 프레임 크기를 1024 보다 작게 해야 하는 경우도 있습니다.

상대국번은 원격지 ETOS의 리모트 접속용 포트 국번을 의미합니다. ETOS는 포트 별로 서로 다른 국번을 설정할 수 있습니다. 포트 별로 서로 다른 응용을 가능하게 하기 위해서 입니다. 단, 1:N 통신(RF 모뎀이나 RS422/485)의 경우를 대비한 옵션으로 1:1 통신 상(RS232)에서는 실질적인 의미는 없습니다. 따라서 RS232C 통신인 경우엔 양쪽 모두를 0번 국 등으로 동일하게만 설정하면 됩니다.

▶ 원격지 ETOS의 리모트 연결 설정

리모트 연결을 위해서, 원격지 ETOS의 해당 포트는 다음과 같이 설정되어야 합니다.

리모트 연결용 COM 포트를 선택하고, {COM 포트 설정 (Serial)} 대화상자의 [고급] 버튼을 누릅니다. <리모트 접속>을 체크하고 국번을 할당합니다. 온라인 연결 설정에서 동일한 국번을 입력해야 합니다.

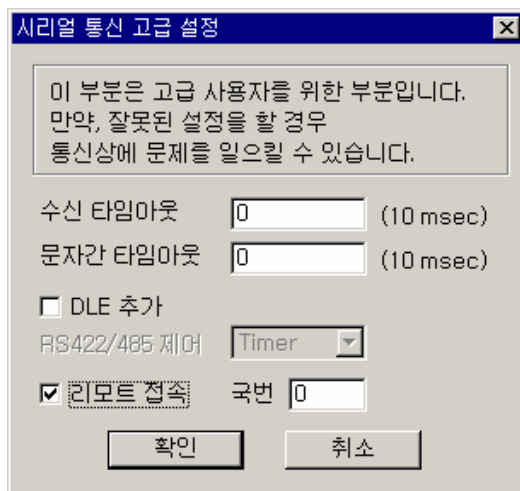


그림 4.3-7 원격지 ETOS의 리모트 연결 설정

(*55) ETOS-200/1000 인 경우만 활성화됩니다.

4.3.4.4. [온라인(O)] -> [편집 내용 검사]

파라미터 설정과 프로그램(프레임, 프로세스) 편집에 오류가 있는지 확인합니다. 파라미터 쓰거나 프로그램 쓰기 시에 자동으로 실행되어, 편집 내용의 오류를 검사합니다. 편집 내용이 정상인 경우에만 파라미터 또는 프로그램 쓰기가 수행됩니다.

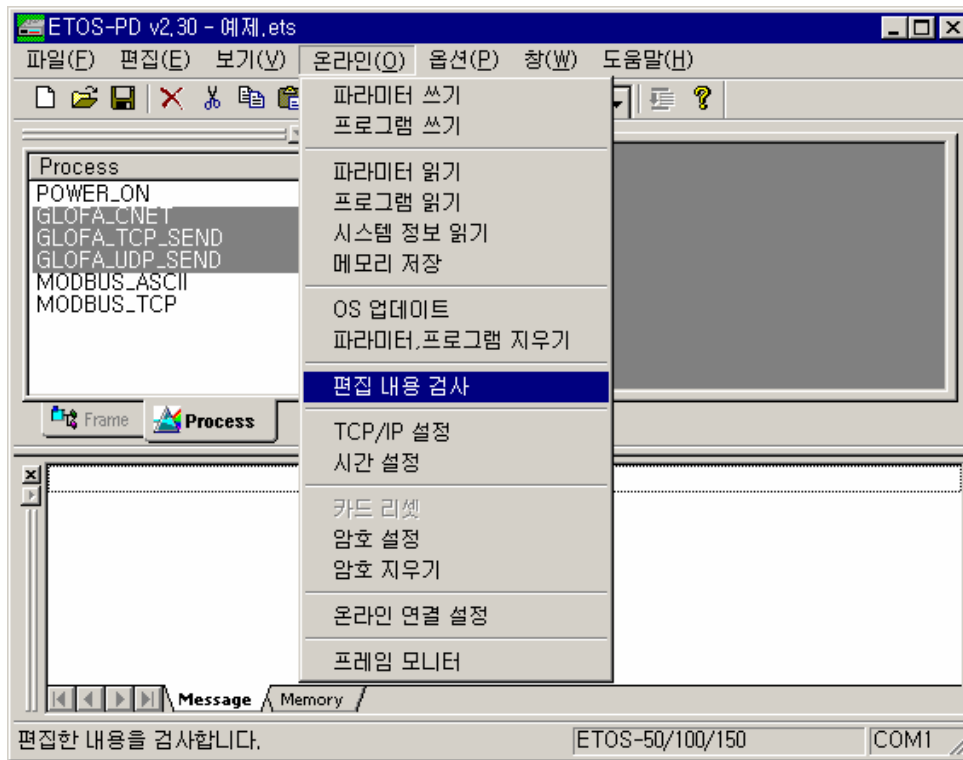


그림 4.3-8 편집내용 검사

편집내용에 이상이 없는 경우에는 {메시지} 창에 '[편집내용 검사]을[를] 완료하였습니다.'라는 메시지가 출력됩니다. 그러나 편집내용에 오류가 있으면 그에 해당하는 설명이 {메시지} 창에 표시됩니다.

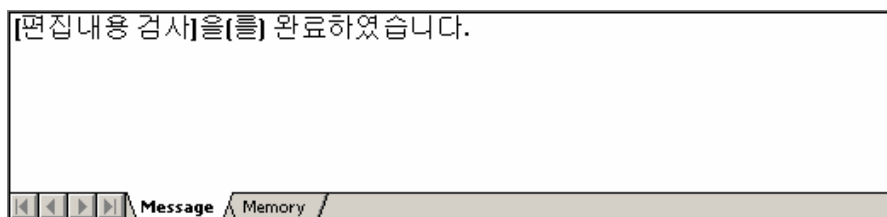


그림 4.3-9 편집내용 검사 결과 화면

4.3.4.5. [온라인(O)] -> [파라미터 쓰기]

편집한 파라미터를 ETOS 에 다운로드 합니다. 파라미터에는 시리얼 파라미터, 이더넷 파라미터, 이중화 파라미터^(*56)가 포함됩니다. 파라미터에 대한 보다 자세한 설명은 [5. 파라미터](#) 를 참조하시기 바랍니다.

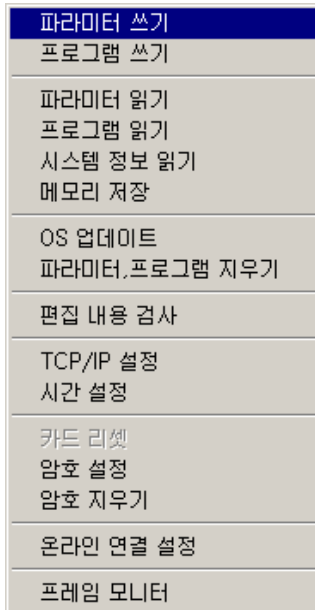


그림 4.3-10 파라미터 쓰기 메뉴

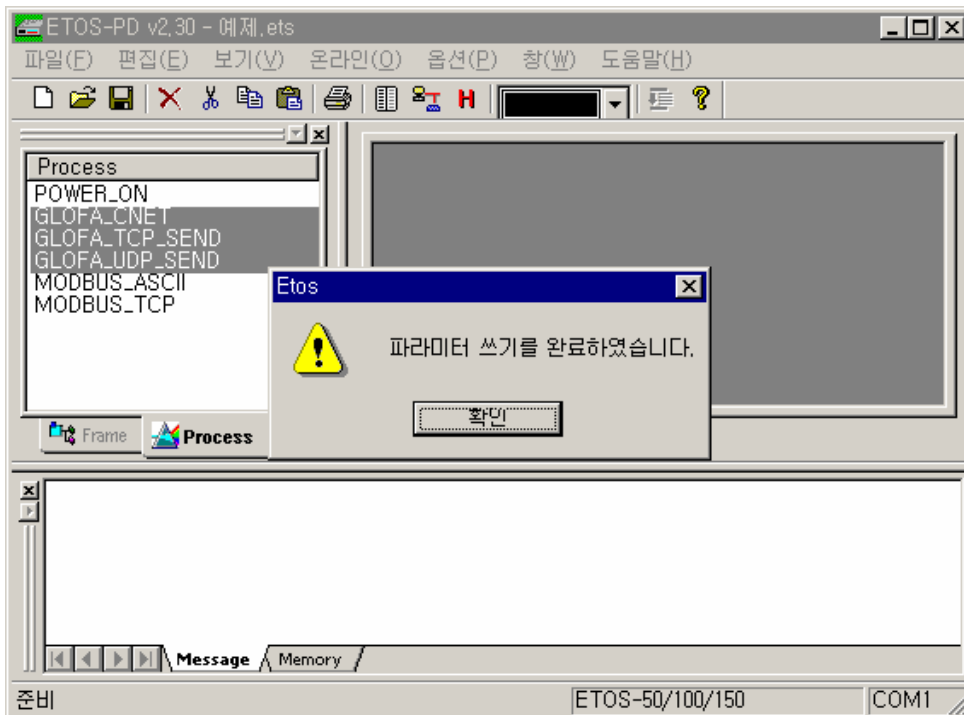


그림 4.3-11 파라미터 쓰기 결과 화면

(*56) ETOS-1000 이중화 모델만 해당됩니다.

4.3.4.6. [온라인(O)] -> [프로그램 쓰기]

편집한 프레임과 프로세스를 ETOS 에 다운로드 합니다. 프레임과 프로세스에 대한 보다 자세한 설명은 6. 프레임 및 7. 프로세스 및 아이템 편을 참조하시기 바랍니다

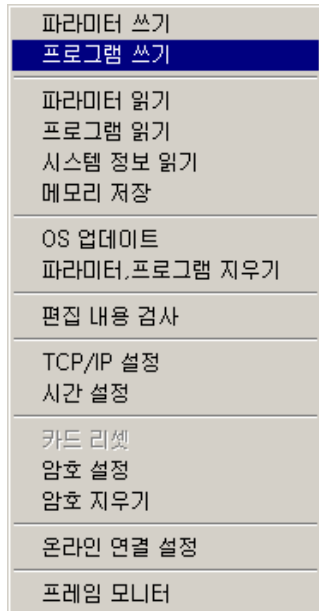


그림 4.3-12 프로그램 쓰기 메뉴

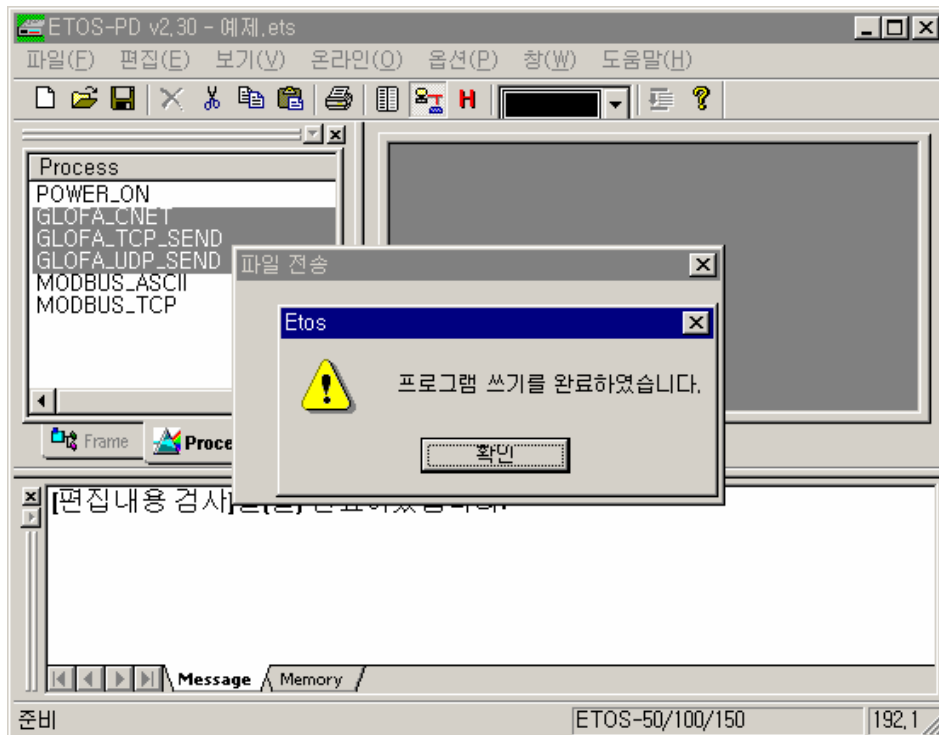


그림 4.3-13 프로그램 쓰기 결과 화면

4.3.4.7. [온라인(O)] -> [파라미터 읽기]

ETOS 에 다운로드되어 있는 파라미터를 읽어들이니다. 프로그램 파일을 별도로 가지고 있지 않은 경우 유용한 기능입니다.

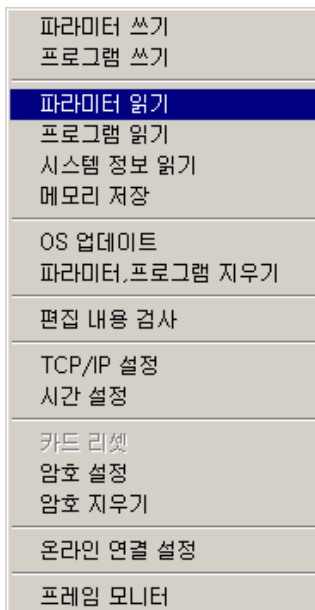


그림 4.3-14 파라미터 읽기 메뉴

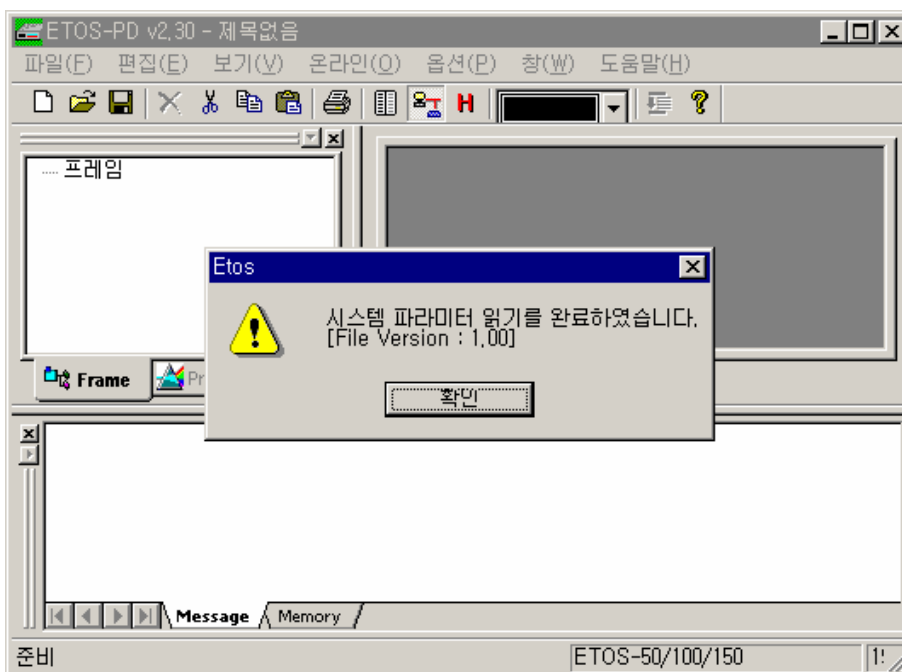


그림 4.3-15 파라미터 읽기 결과 화면

4.3.4.8. [온라인(O)] -> [프로그램 읽기]

ETOS 에 다운로드되어 있는 프레임, 프로세스를 읽어들이니다. 프로그램 파일을 별도로 가지고 있지 않은 경우, 파라미터 읽기, 프로그램 읽기를 차례로 실행하여 파일로 저장할 수 있습니다.

동작중인 ETOS 에 기능을 추가하는 경우, 백업해 놓은 파일로부터 작업을 시작하는 것보다, ETOS 로부터 파라미터, 프로그램 읽기를 실행해서 업로드한 파일에 추가 작업을 하는 것이 바람직합니다.

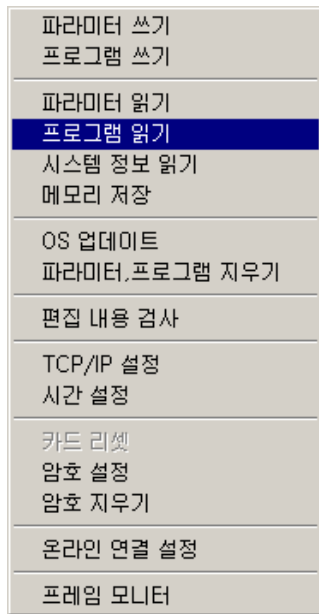


그림 4.3-16 프로그램 읽기 화면

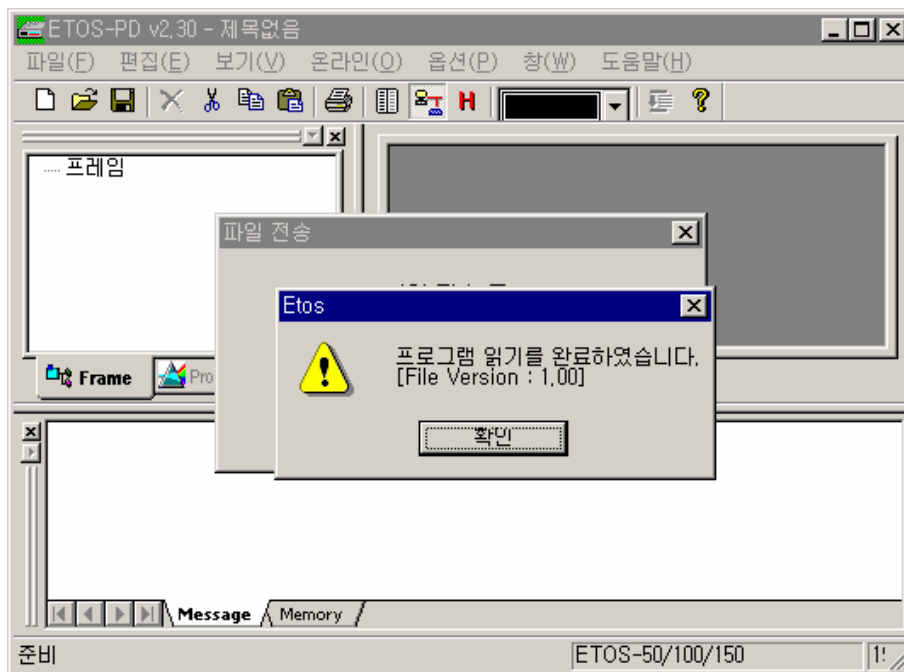


그림 4.3-17 프로그램 읽기 결과 화면

4.3.4.9. [온라인(O)] -> [시스템 정보 읽기]

ETOS의 S/W 타입(PLUS, SECS), 버전, MAC ID 등의 정보를 확인할 수 있습니다.

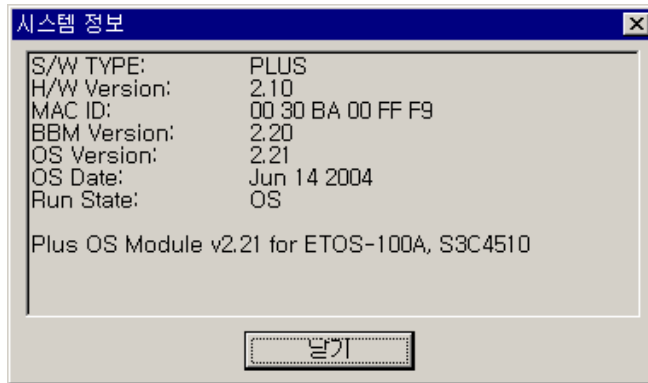


그림 4.3-18 시스템 정보 화면

4.3.4.10. [온라인(O)] -> [메모리 저장]

사용자(데이터) 메모리 또는 시스템(로그) 메모리를 텍스트 파일로 컴퓨터에 저장합니다. ETOS는 시스템 메모리 상에 운전 중의 이벤트를 로그로 남기고 있습니다. ETOS의 동작에 이상이 있다고 판단되는 경우, 로그 내용을 파일로 저장해서 여러 상황을 분석하는데 사용합니다.

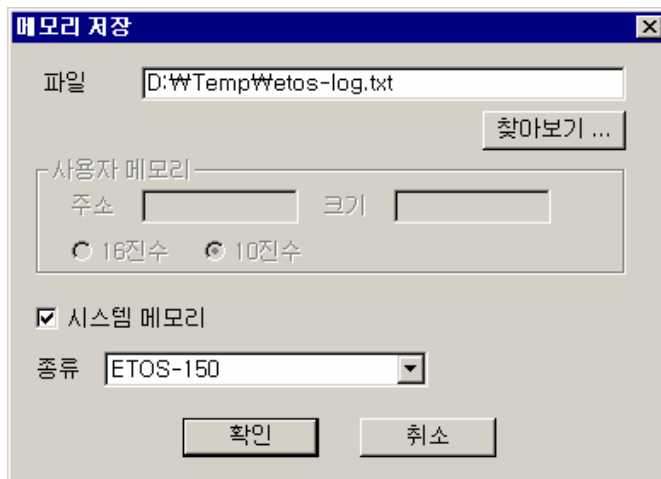


그림 4.3-19 메모리 저장 화면

표 4.3-4 메모리 저장 설정 항목

항 목		내 용
파일		저장하고자 하는 파일 이름 입력합니다. 찾아보기를 통해 폴더 위치 선택
사용자 메모리 (*57)	주소	사용자 메모리 시작 위치를 입력
	크기	저장하고자 하는 메모리 양을 바이트 단위로 입력
	16진수/10진수	메모리 시작 위치와 크기를 16/10진수 형태로 인식
시스템 메모리 종류		로그 메모리를 파일로 저장합니다. ETOS-150/200/1000/150-Dual 중에서 선택

(*57) 시스템 메모리 선택을 해제한 경우만 활성화됩니다.

4.3.4.11. [온라인(O)] -> [OS 업데이트]

ETOS 의 OS 를 업그레이드 할 때 사용합니다. ETOS 의 OS 는 기능 향상을 위해 [AC&T 홈페이지](#)를 통해 주기적으로 업데이트 되고 있습니다. 이때, 사용중인 ETOS 의 S/W 타입과 다운로드 하고자 하는 OS 의 타입이 일치해야 합니다.^(*58) ETOS 의 S/W 타입은 [온라인(O)] -> [시스템 정보 읽기]에서 확인할 수 있습니다.

- * S/W 타입에는 PLUS 와 SECS 가 있습니다. SECS 는 반도체 장비가 표준으로 사용하는 SECS 프로토콜을 구현한 ETOS 의 타입을 지칭합니다. (ETOS-150, ETOS-200-HSMS 등). 반도체 장비가 아닌 범용 장비를 위한 ETOS 의 타입은 PLUS 로 표현합니다.

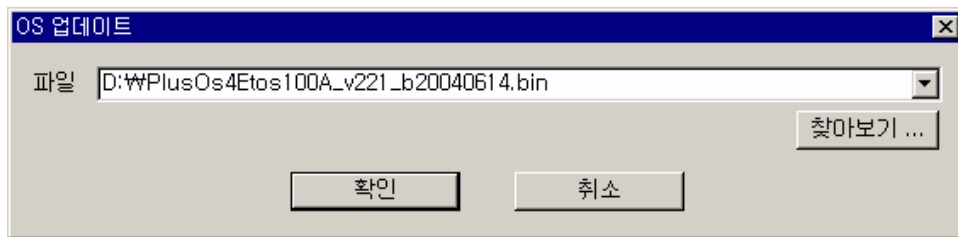


그림 4.3-20 OS 업데이트 화면

업그레이드 하고자 하는 OS 를 [찾아보기]로 선택한 후 [확인]을 누르면 쓰기가 시작됩니다. 시리얼로 ETOS 에 다운로드하는 경우, 수분 이상이 걸릴 수 있습니다. 쓰기 중엔, ETOS 의 송,수신 LED 가 일정한 패턴으로 점멸합니다. 쓰기가 모두 끝나면 ETOS 는 자동으로 리부팅됩니다. 이때, 전원을 끄면 ETOS 가 정상 동작하지 않을 수 있으니, 반드시 자동으로 리부팅될 때까지 기다려야 합니다.

^(*58) S/W 타입이 PLUS 인 경우, OS 는 PlusOS4Etosxxx_v...bin 으로 업데이트 해야 하며, S/W 타입이 SECS 인 경우, OS 는 SecsOS4Etosxxx_v...bin 으로 업데이트 해야 합니다. 이때, xxx 는 ETOS 종류를 지칭합니다. (50A/100A/150/200/1000)

4.3.4.12. [온라인(O)] -> [파라미터, 프로그램 지우기]

ETOS 에 저장되어 있는 파라미터, 프로그램을 모두 지웁니다.

다운로드한 프로그램에 문제가 있어서 ETOS 가 정상 동작하지 않는 경우에는 다운로드한 프로그램을 ETOS 프로그램 메모리(FLASH)에서 지워야 합니다. (*59)

또한, [파라미터, 프로그램 지우기] 후에 OS 를 업그레이드 할것을 권장합니다.

동작중인 ETOS 의 프로그램을 읽어서 파일로 저장해 둔 후, [파라미터, 프로그램 지우기]로 ETOS 프로그램을 ETOS 프로그램 메모리(FLASH)에서 지워 주시기 바랍니다. OS 업그레이드가 모두 끝나면 [파라미터 쓰기], [프로그램 쓰기]를 통해 백업해 둔 프로그램을 다시 ETOS 에 다운로드 합니다.

[파라미터, 프로그램 지우기]는 ETOS-PD 에서 편집 중인 내용에는 영향을 미치지 않습니다.

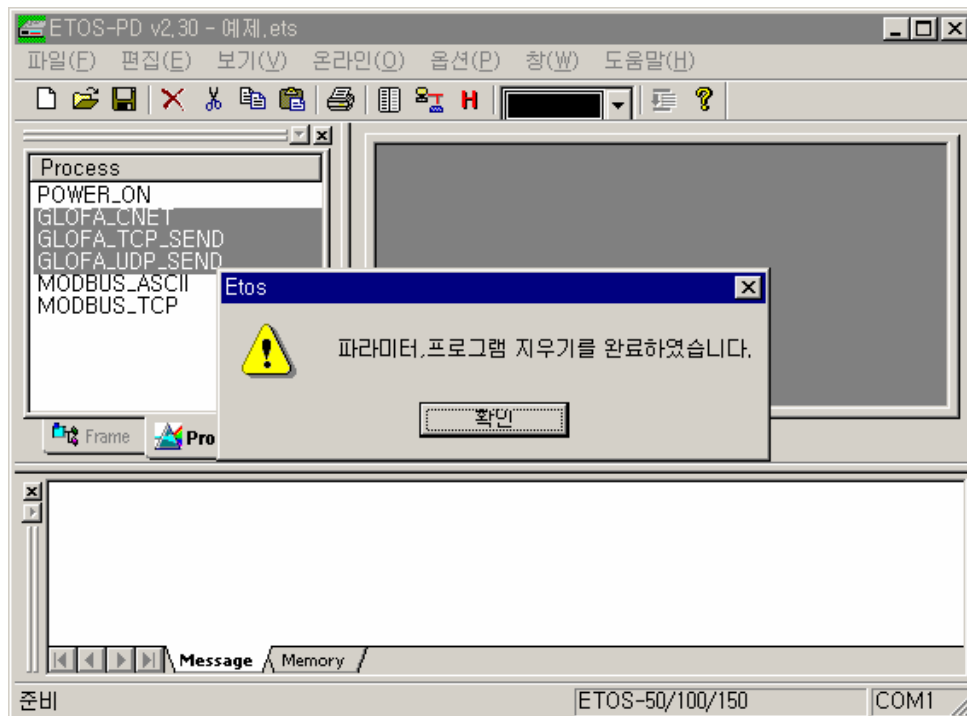


그림 4.3-21 파라미터, 프로그램 지우기 결과 화면

(*59) ETOS 가 정상 동작하지 않는 경우 [파라미터, 프로그램 지우기]도 실행되지 않는 경우가 대부분입니다. 이 경우, ETOS 의 PGM 포트 2,3 번을 Short 시킨 후 전원을 켜면, LED 가 주기적으로 점멸하는 상태가 됩니다. 이 상태에서 2,3 번 short 를 제거하고, 컴퓨터와 PGM 포트를 다시 연결하고 [파라미터, 프로그램 지우기]를 실행하면 됩니다.

4.3.4.13. [온라인(O)] -> [TCP/IP 설정]

ETOS의 이더넷 관련 옵션을 설정합니다.

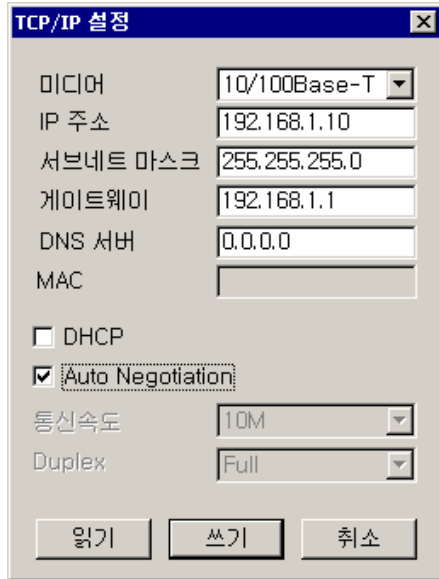


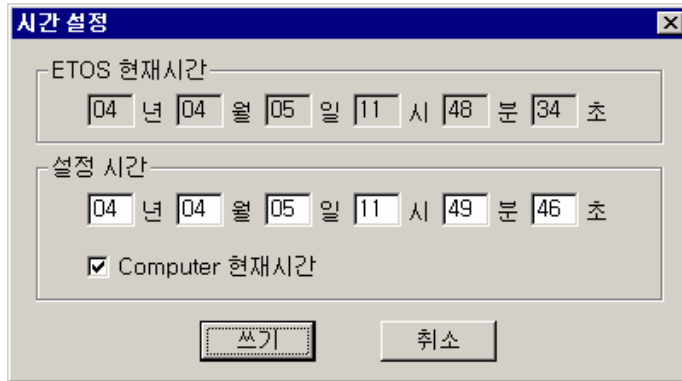
표 4.3-5 TCP/IP 설정 항목

항목	내 용	비 고
미디어	10/100Base-T 선택	
IP 주소	ETOS의 IP 주소를 설정	고정 IP인 경우
서브네트 마스크	ETOS의 IP가 속해있는 네트워크 그룹	
게이트웨이	ETOS가 속해있는 그룹의 게이트웨이 IP주소	
DNS 서버	Domain Name Service Server IP주소	
DHCP	유동 IP를 사용하는 경우 선택	
Auto Negotiation	통신속도와 duplex를 자동으로 설정	
통신속도	10Mbps / 100Mbps 중에서 선택	Auto Negotiation을 끈 경우 만 설정
Duplex	Full / Half duplex 중에서 선택	

각 항목에 대한 보다 구체적인 사항은 네트워크 관리자에게 문의하시기 바랍니다.

4.3.4.14. [온라인(O)] -> [시간 설정]

ETOS의 RTC(Real Time Clock)를 읽거나 설정합니다. 여러 가지 원인에 의해 ETOS의 RTC 값은 조금씩 오차가 누적됩니다. 따라서 사용자가 주기적으로 ETOS의 RTC 값을 보정할 필요가 있습니다.



<ETOS 현재시간>

ETOS의 RTC 현재값입니다.

<설정시간>

사용자 PC의 시각으로 동기화 시킬 수 있습니다. <Computer 현재시간>을 체크한 상태에서 [쓰기]를 누르면 ETOS의 시각을 컴퓨터의 현재 시각으로 동기화 시킵니다.

4.3.4.15. [온라인(O)] -> [암호 설정]

관리자 이외의 사용자가 임의로 ETOS 의 정보를 읽거나 쓰지 못하도록 암호를 걸어 타 사용자의 접근을 제한합니다. 반드시 암호를 통해 인증을 받아야만 ETOS 의 설정을 변경할 수 있습니다.

암호 변경 시에는 기존암호를 입력하고, 새 암호를 입력, 확인 하여 변경합니다.

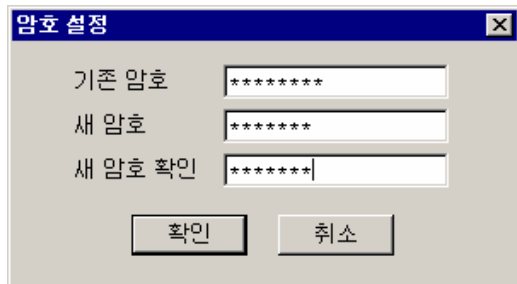


그림 4.3-22 암호설정 및 변경 화면

4.3.4.16. [온라인(O)] -> [암호 지우기]

설정된 암호를 잊어버린 경우, 암호를 지우기 위해 사용합니다. 이때, 입력하는 내용은 제품별로 모두 다릅니다. 암호를 지우고자 하는 ETOS 의 MAC ID(시스템 정보 읽기로 확인)를 확인하여 제조사로 문의하시면 입력할 내용을 알려드립니다.

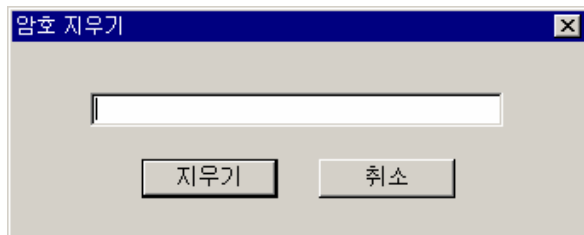


그림 4.3-23 암호 지우기 화면

4.3.4.17. [온라인(O)] -> [IFOR Shared Memory 설정]

▶ [옵션] -> [편집 옵션] 에서 <IFOR 편집>을 체크해야 나타납니다.

Optic Network Card(IFOR) 모듈이 꽂혀있을 때 사용합니다.

Optic Network 로 구성된 네트워크 안의 모듈들이 같은 메모리를 공유하는 기능을 설정할 때 사용합니다.

[온라인]->[IFOR Shared Memory 설정]을 선택하면 아래의 창이 열립니다. Shared Memory 설정은 여러 국이 동일한 메모리 맵을 가지고, 데이터를 공유하는 방식입니다. 따라서 한 국에서 설정한 정보는 다른 국에도 동일하게 다운로드 되어야 합니다. 아래쪽의 [추가]버튼을 이용해서 메모리 위치와 시간을 설정합니다. 파일로 저장하고, Card ID 를 적고, 다운로드 합니다. [전국에 쓰기]는 IFOR Card 의 국번이 0 번인 국에서만 가능합니다. 전국에 쓰기를 하는 동안에는 Shared Memory Service 가 잠시 정지됩니다. 각 국 별로 개별 쓰기는 해당국만 잠시 정지됩니다.

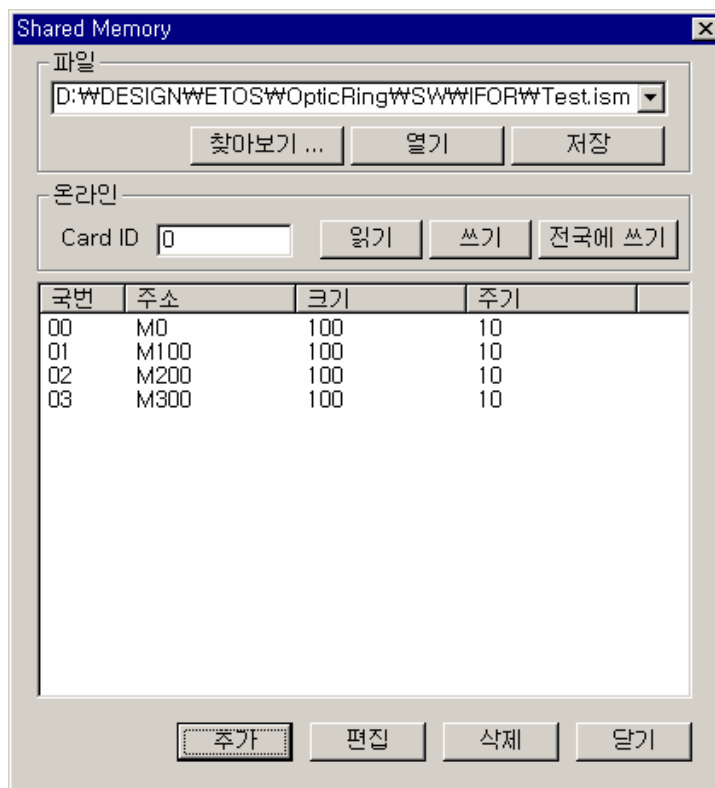


그림 4.3-24 Shared Memory 설정

Shared Memory 는 최대 56540Bytes 까지 사용 가능합니다. 한 국이 전부 다 사용할 수도 있고, 모든 국이 나눠 사용할 수도 있습니다. 사용자의 필요에 따라 설정하면 됩니다. 같은 국에서 여러 블록을 설정 할 수도 있습니다. 다만 총 블록은 64 개로 제한하였습니다.

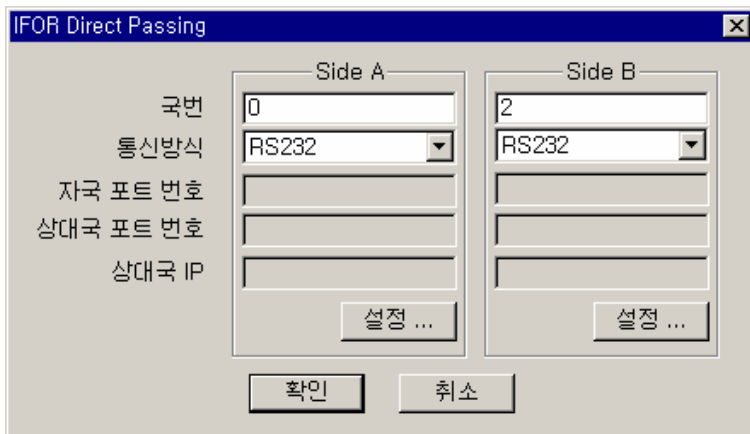
4.3.4.18. [온라인(O)] -> [IFOR Direct Pass 설정]

▶ [옵션] -> [편집 옵션] 에서 <IFOR 편집>을 체크해야 나타납니다.

Optic Network Card(IFOR) 모듈이 꽂혀있을 때 사용합니다.

Optic Network 로 구성된 한 모듈의 시리얼/이더넷 포트를 원거리의 다른 모듈의 포트와 연결해 주는 기능을 설정할 때 사용합니다.

Direct Passing 은 한 IFOR 의 이더넷이나 시리얼포트를 다른 국의 것과 가상의 연결을 맺어 거리와 공간의 제약을 극복하기 위한 것입니다. 예를 들면 PC 의 시리얼 포트는 25 미터안에 다른 장비가 있어야 통신이 가능합니다. IFOR 를 도입해서 그런 거리의 제약을 없앨 수 있습니다. 또 물리적인 연결의 변경 없이 설정만으로 다른 장비를 Access 할 수 있습니다. ETOS-PD 의 [온라인]->[IFOR Direct Pass 설정]을 선택후 <Card ID>를 설정하고 [추가]버튼을 클릭합니다. 아래 창에서 Side A 에는 자국의 정보를, Side B 에는 Remote 국의 정보를 설정합니다.



설정후 다음과 같이 추가됩니다.

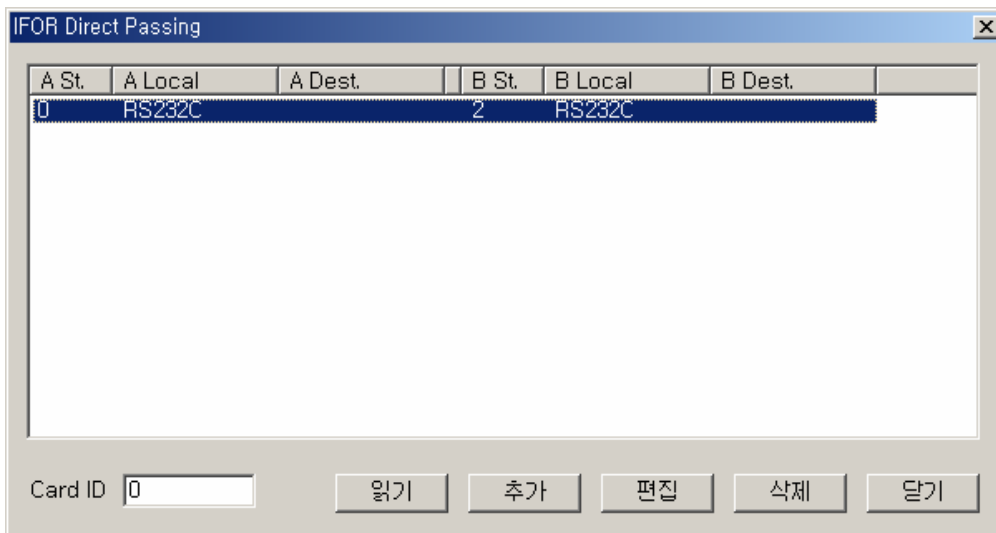


그림 4.3-25 IFOR Direct Pass 설정

이런 Direct Passing 설정은 전원을 Off 하면 모두 사라지게 됩니다.

4.3.4.19. [온라인(O)] -> [IFOR 정보읽기]

▶ [옵션] -> [편집 옵션] 에서 <IFOR 편집>을 체크해야 나타납니다.

Optic Network Card(IFOR) 모듈이 꽂혀있을 때 사용합니다.

Optic Network 로 구성된 모듈과 링의 상태를 모니터 하는 기능입니다.

[온라인]->[IFOR 정보읽기]를 선택하면 아래의 창이 열립니다. [읽기] 또는 [연속읽기]를 클릭하면 현재 링의 상태가 보여집니다. 물리적으로 연결되어 있는 순서 그대로 표시됩니다.

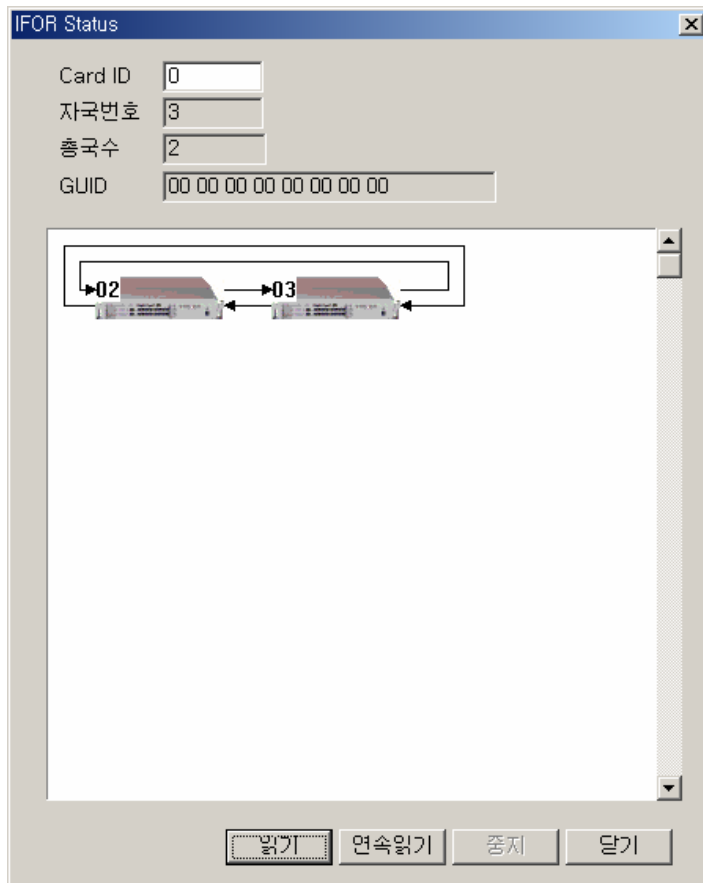


그림 4.3-26 IFOR Staus 창

그림에서 특정국을 다시 클릭하시면 해당국의 상세 정보가 나옵니다. Power On Time, 총 송수신량, 초당 송수신량, 에러 패킷 수 등을 알 수 있습니다.

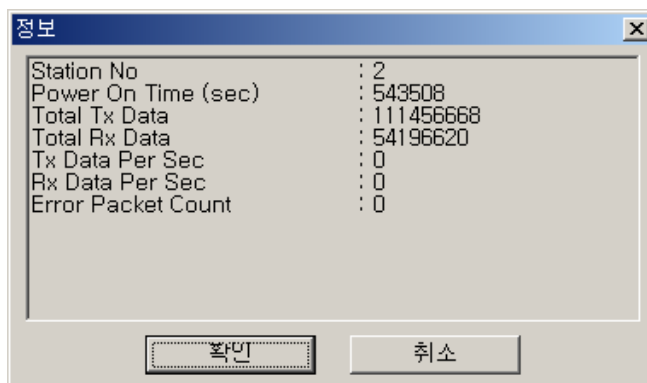


그림 4.3-27 IFOR Staus 세부정보

4.3.4.20. [온라인(O)] -> [HSMS Diagnostic Status 읽기]

▶ [옵션] -> [편집 옵션] 에서 <HSMS Diagnostic 편집>을 체크해야 나타납니다.

ETOS-150/200-DIAG 모델일 경우만 사용합니다.

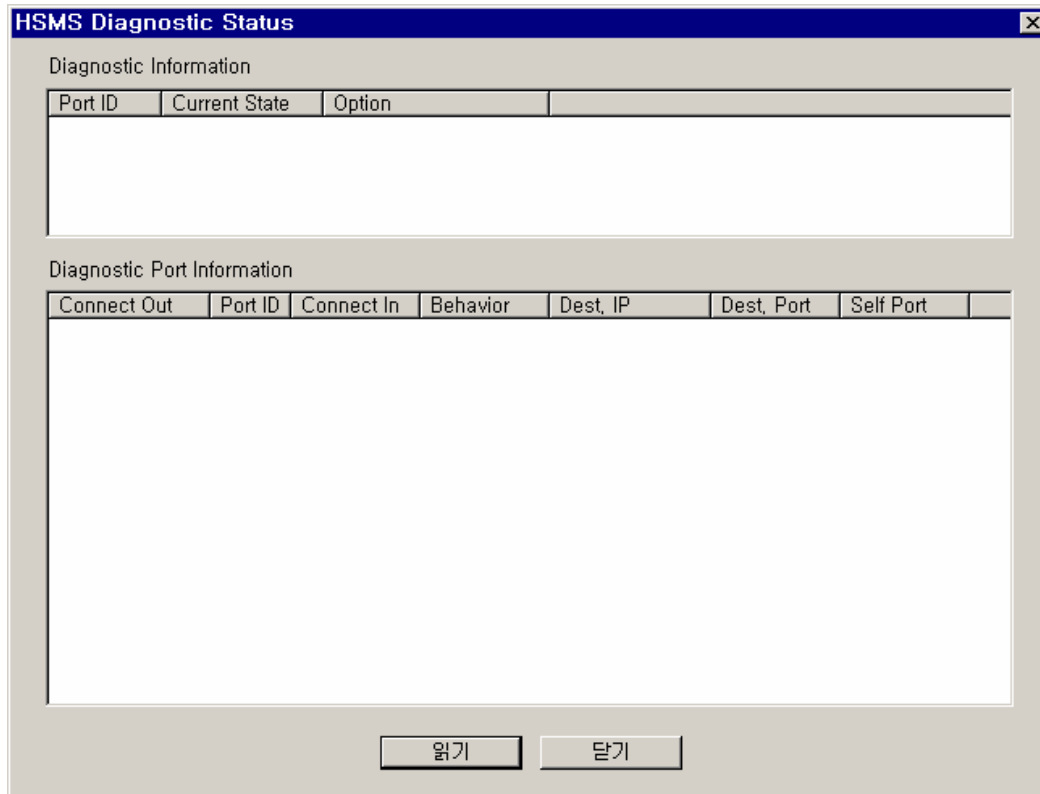


그림 4.3-28 HSMS Diagnostic Status 읽기 화면

4.3.5. 통신 모니터링

편집한 프로그램을 ETOS 에 다운로드 했으면, 프로그램이 의도한대로 정상동작 하는지 확인해야 합니다. ETOS 의 TX, RX LED 의 점멸로 데이터의 송,수신 여부를 알 수 있으나, 실제 내용은 확인할 수 없습니다. 이때 프레임 모니터를 통해 해당 시리얼 포트의 송,수신 프레임을 확인합니다. 또한, {메모리} 창에서 프레임이 원하는 메모리 위치에 저장되는지도 확인할 수 있습니다.

4.3.5.1. [온라인(O)] -> [프레임 모니터]

프레임 모니터링은 특정 시리얼 포트의 송,수신 프레임을 모니터링하는 기능입니다.

[온라인] -> [프레임 모니터]를 선택합니다.

모니터링 하고자 하는 ETOS 의 카드^(*60)와 포트를 선택한 후, [시작]버튼을 누르면 모니터가 시작됩니다. 모니터를 멈추려면 [정지]버튼을 누릅니다.

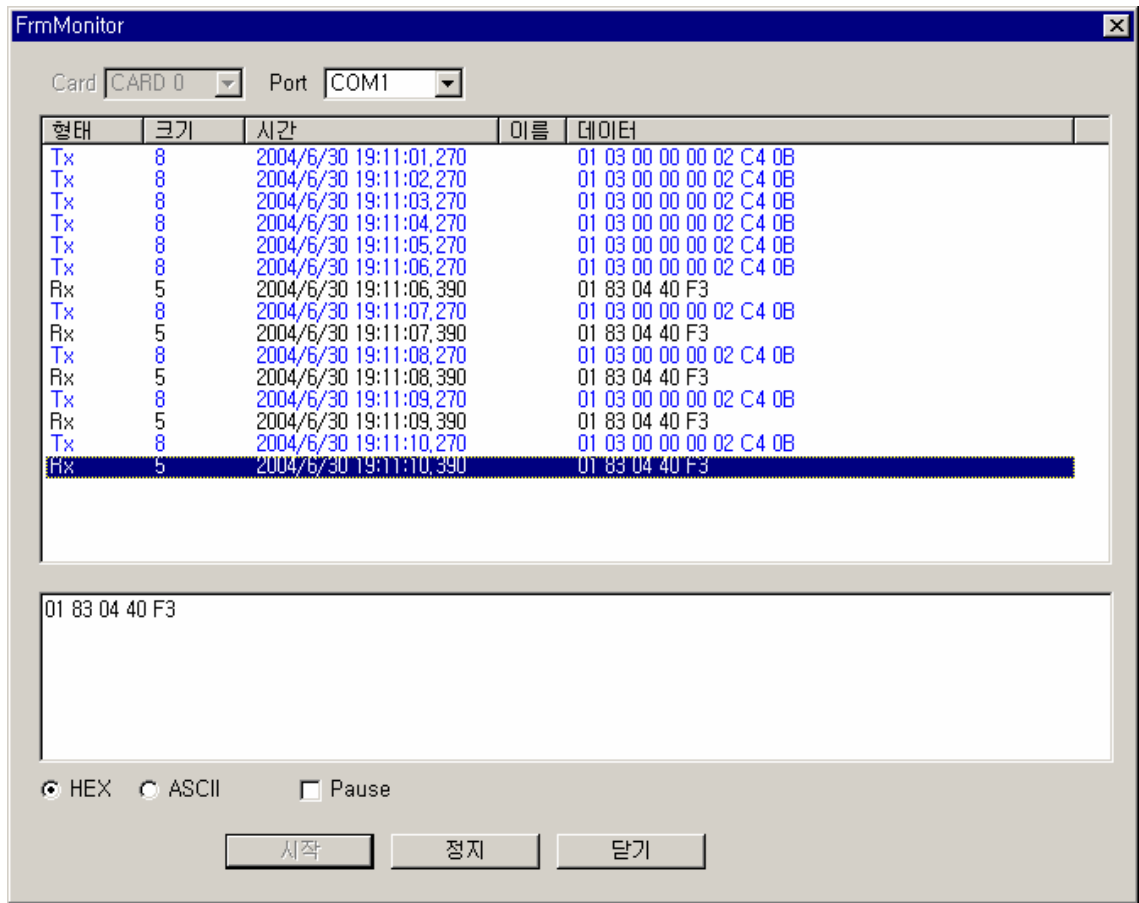


그림 4.3-29 프레임 모니터링

형태에 Tx 로 표시되는 프레임은 송신 프레임이며, Rx 로 표시되는 프레임은 수신 프레임입니다. 또, 시간은 해당 프레임이 송,수신 된 ETOS 의 시각을 의미합니다.

[정지]버튼으로 모니터링을 중단한 후에 HEX 와 ASCII 를 선택하면 이미 모니터링한 프레임이 해당 데이터 형식으로 표현됩니다.

(*60) ETOS 종류가 ETOS-200/1000 일 경우만 활성화됩니다.

{프레임 모니터} 창에서 설정하는 각 항목의 의미는 다음과 같습니다.

표 4.3-6 프레임 모니터 설정 항목

항 목	내 용
카드	ETOS-200 과 ETOS-1000 을 사용하는 경우에만 적용되는 옵션. 통신 프레임을 모니터링 할 시리얼 포트가 속해있는 ETOS 의 시리얼 통신 모듈의 번호를 입력합니다. (ETOS-200/1000 인 경우 시리얼 통신 모듈의 PGM 포트로 직접 모니터링할 때는 의미가 없고 CPU 모듈의 PGM 포트로 모니터링할 때만 적용)
포트	모니터링 할 ETOS 의 시리얼 포트를 선택.
HEX	통신프레임을 16 진수 데이터 형식으로 볼 때 사용.
ASCII	통신프레임을 ASCII 데이터 형태로 볼 때 사용.
Pause	모니터 스크롤을 중지합니다. 이때, 모니터는 백그라운드로 수행되고 있습니다. 체크를 삭제하면, 중지하고 있는 동안 송,수신 된 프레임들이 한번에 표시됩니다.
시작	모니터 보기를 실행.
정지	모니터 보기를 중단.
닫기	프레임 모니터 창을 닫습니다.

- * ETOS-1000 이중화 모델인 경우, 두 CPU 모듈 중에서 Active CPU 를 통해서만 모니터링이 가능합니다. Active CPU 는 ETOS Scan 프로그램을 이용하거나, Active LED(ACT)의 점멸로 확인할 수 있습니다.

4.3.6. 메모리 모니터

ETOS의 사용자 메모리 영역을 모니터하는 기능입니다.

{메시지}/{메모리} 출력 창의 메모리 탭을 클릭한 후, <주소>에 읽고자 하는 영역의 선두 번지를 입력하고, [읽기] 버튼을 누르면 ETOS 메모리를 읽어 표시하게 됩니다. 이때, <16진수>/<10진수> 선택에 따라 입력하는 주소도 16진수 또는 10진수로 입력해야 하며, 출력창의 주소 또한 16진수 또는 10진수로 표현됩니다. 그러나, 진수 선택에 관계없이 메모리값 자체는 항상 16진수로 표현됩니다. 주의하시기 바랍니다.

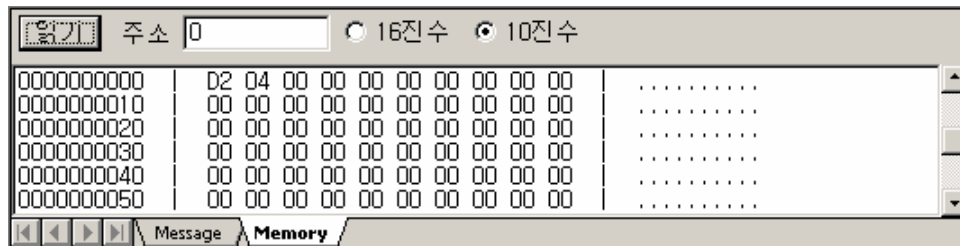


그림 4.3-30 메모리 출력 창

4.3.6.1. ETOS 사용자 메모리

ETOS 사용자 메모리는 송, 수신 프레임 데이터의 저장, 연산 대상 및 결과값의 저장 등 프로그램 내에서 참조하는 메모리 영역입니다. ETOS 사용자 메모리는 바이트 주소만을 지원하며, 비트 또는 워드 주소는 지원하지 않습니다. 비트는 나누기 8, 워드 주소는 곱하기 2를 해서, 즉 바이트 주소로 변환해서 읽을 수 있습니다.

ETOS 사용자 메모리는 하위 바이트가 메모리의 낮은 번지에 저장되는 little-endian^(*61) 방식의 저장 방법을 사용합니다.

10진수 1234가 ETOS 메모리에서 어떻게 표현되는지 예를 들어보겠습니다.

▶ 1234 (10진수) = 4D2_H (16진수)

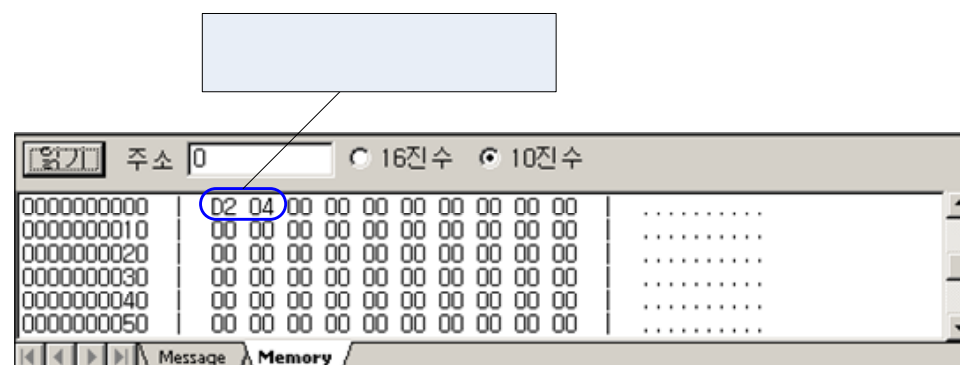


그림 4.3-31 ETOS 메모리 체계

(*61) 여러 바이트로 이루어진 하나의 데이터를 메모리에 저장하는 순서를 나타내는 용어로, 인텔 x86 계열 프로세서(펜티엄 4 등)가 해당합니다. 반면에 PowerPC 같은 프로세서는 하위 바이트가 메모리의 상위 바이트에 저장되는 big-endian 방식을 사용합니다.

파라미터

제 5 장

이 장은 파라미터의 정의 및 설정방법 등에 대한 전반적인 내용을 다루고 있습니다.

	내용	위치
5.1	개요	84
5.2	파라미터 설정 순서	85
5.3	시리얼 파라미터	89
5.4	이더넷 파라미터	94

5. 파라미터

ETOS 가 동작하기 위해서는 파라미터, 프레임, 그리고 프로세스의 세가지 요소가 필요합니다. 이 장에서는 파라미터의 의미와 사용자가 요구하는 다양한 통신 시스템 구성을 위해 파라미터가 어떻게 설정되어야 하는지 설명합니다.

- * 이 장의 내용은 ETOS-PD V2.3 을 기준으로 작성한 것으로 버전에 따라 다소 차이가 있을 수 있습니다. (최신 버전의 ETOS-PD 를 에이씨앤티시스템 홈페이지 <http://www.acnt-sys.co.kr> 에서 다운로드 받으실 수 있습니다.)

5.1. 개요

파라미터는 ETOS 의 시스템 전반에 대한 설정을 말하는 것으로, 구체적으로는 ETOS-PD 의 [편집 (E)] -> [파라미터 설정] 메뉴에서 설정하는 모든 내용입니다.

파라미터는 크게 시리얼, 이더넷 그리고 이중화 파라미터로 구성됩니다.

▶ 시리얼 파라미터

범용 시리얼 통신 파라미터와 필드버스 파라미터 등을 포함하고 있습니다.

▶ 이더넷 파라미터

이더넷 드라이버, 채널, 고속링크 설정과 SECS 파라미터(HSMS 설정, HSMS Diagnostic 설정) 설정으로 크게 나누어 집니다. [HSMS 설정]은 파일을 SECS 로 설정했을 경우만 활성화됩니다.

▶ 이중화 파라미터

ETOS 종류가 ETOS-1000 일 경우에만 활성화 되며, ETOS-1000 이중화 시스템인 경우만 (CPU 가 ETOS-1000RCP) 의미를 갖습니다.

제품에 따른 하드웨어 구성은 1.3 시리즈 구성에 정리되어 있습니다. 모델명과 제품 구성을 확인하여, 구입 제품의 H/W 구성에 맞게 파라미터를 설정해야 합니다.

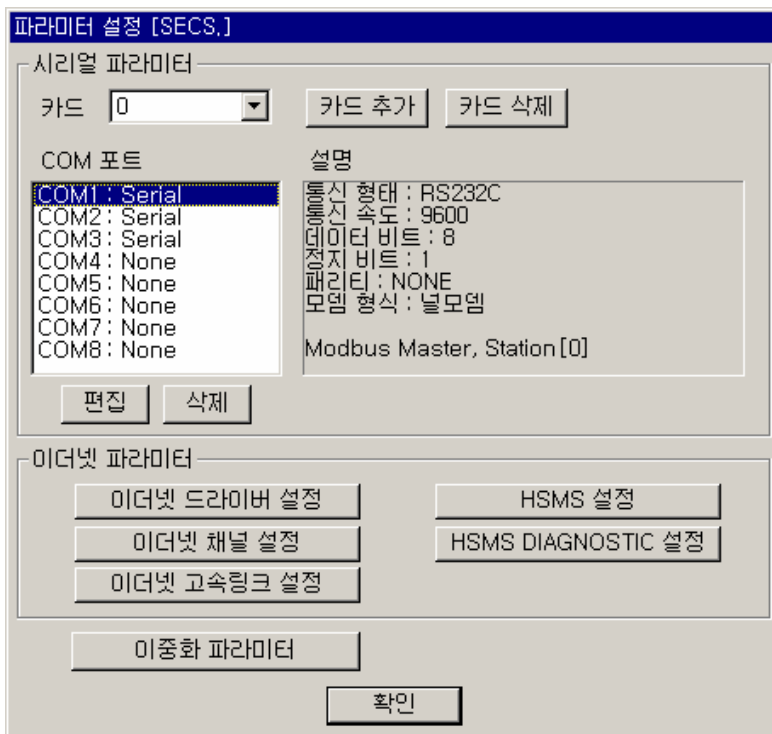


그림 5.1-1 파라미터 설정 화면

5.2. 파라미터 설정 순서

연결할 상대 기기의 종류와 수, 연결 방법 등에 따라 파라미터를 설정합니다.
 다음 그림은 파라미터를 설정하는 일반적인 순서입니다.

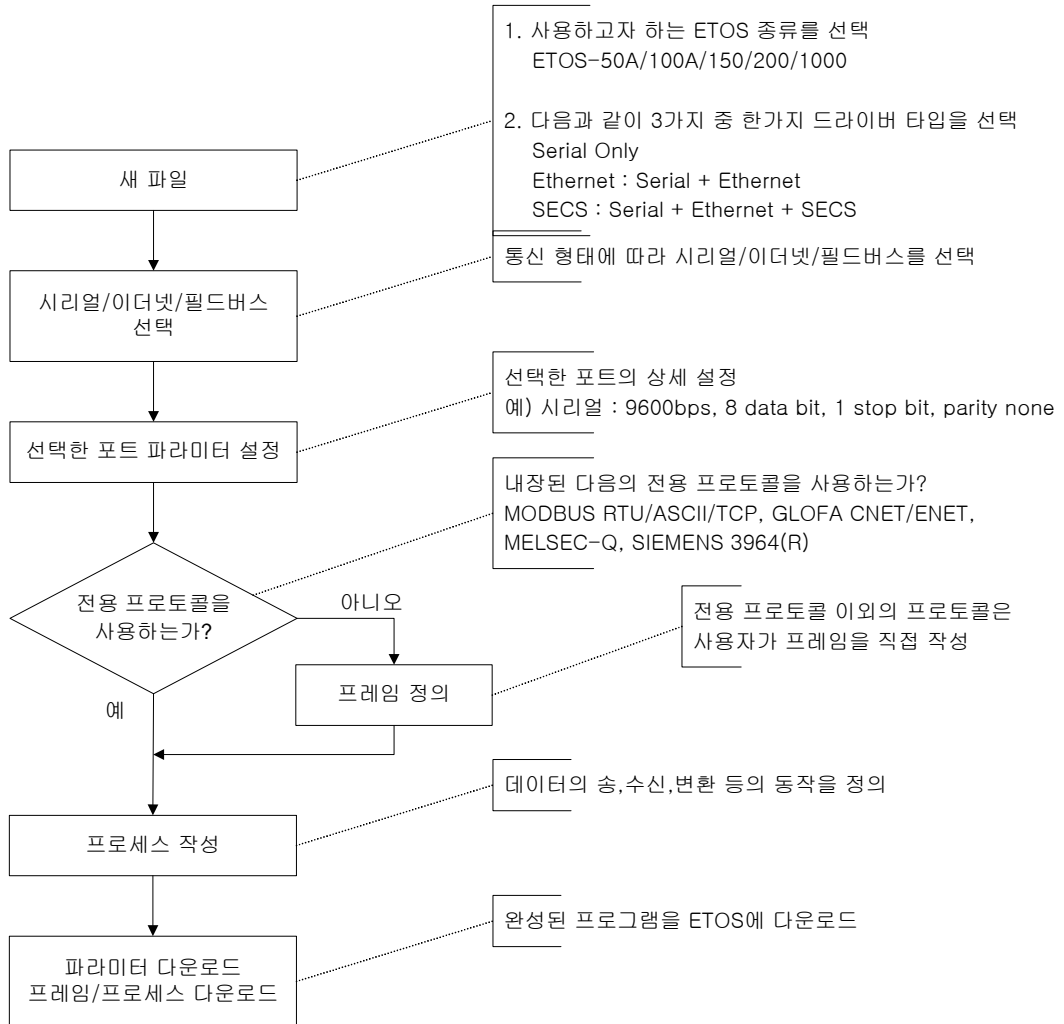


그림 5.2-1 파라미터 설정 순서

5.2.1. 새 파일 시작

ETOS-PD V2.3 부터는 프로그램 시작시에 자동으로 새파일을 열도록 되어 있습니다. (이전 버전사용자는 [파일] -> [ETOS TYPE 변경], [파일] -> [새 파일] 메뉴를 차례로 실행해야 합니다.)

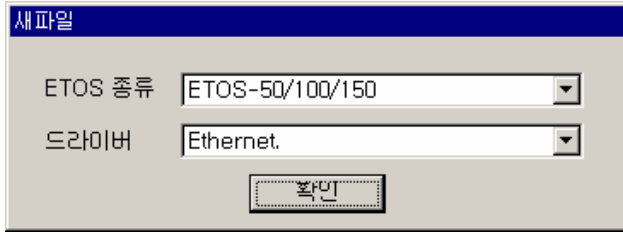


그림 5.2-2 새파일 대화상자

5.2.1.1. ETOS 종류

{새파일} 대화상자에서 ETOS의 종류를 선택해야 합니다. 구입하신 ETOS와 동일하게 맞추어 주시기 바랍니다. 설정 이후에는 종류 변경이 되지 않으므로 ETOS 종류를 잘못 선택한 경우엔 파일을 다시 만들어야 합니다. ETOS 종류를 아래 표를 참고하여 선택합니다.

표 5.2-1 제품별 ETOS 종류 선택

ETOS 종류	해당 제품
ETOS-50/100/150	ETOS-50A/100A/150
ETOS-200	ETOS-200-Plus/HSMS/SGate
ETOS-1000	ETOS-1000
ETOS-100/150 Dual	ETOS-100/150 Dual

5.2.1.2. 드라이버 종류

드라이버는 사용하려는 용도에 맞추어 선택합니다. 드라이버의 종류는 다음과 같습니다. 사용자는 반드시 다음 중 한가지의 드라이버를 선택해야 합니다.

표 5.2-2 통신 시스템 구성에 따른 드라이버 선택

드라이버 종류	통신 시스템 구성
Serial Only	Serial (범용 시리얼, 필드버스, I/O 등)
Ethernet	Serial + Ethernet
SECS	Serial + Ethernet + SECS-I + HSMS

▶ Serial Only

이더넷을 사용하지 않을 경우에 선택합니다. 시리얼 통신이나 필드버스 통신 또는 시리얼과 필드버스 통신을 동시에 사용하는 경우 선택합니다. (PLUS, SECS 모델 모두 지원)

▶ Ethernet : Serial + Ethernet

시리얼이나 필드버스 통신외에 이더넷 통신을 사용하는 경우 선택합니다. ETOS를 슬레이브나 마스터로 사용할 수 있습니다. 이더넷 내장 프로토콜 및 사용자정의 프레임 등록 모두를 지원합니다. (PLUS, SECS 모델 모두 지원)

▶ **SECS : Serial + Ethernet + SECS-I + HSMS**

SECS-I, HSMS 통신을 사용하는 경우에 선택합니다. SECS 는 일반 ETOS(ETOS-Plus)가 아닌 ETOS-SECS 모델에서만 지원합니다. (SECS 모델에서만 지원)

5.2.1.3. 모델 별 선택가능 드라이버

다음 표는 모델 별로 선택가능한 드라이버의 목록입니다. 구입하신 ETOS 의 모델명을 확인하시어, 적절한 드라이버를 선택하시기 바랍니다.

표 5.2-3 모델 별 선택가능 드라이버

모델명	선택가능 드라이버
ETOS-50A-SXX	Serial Only
ETOS-50A-EXX	Serial Only/Ethernet
ETOS-50A-SGate	SECS
ETOS-100A-SXX	Serial Only
ETOS-100A-EXX	Serial Only/Ethernet
ETOS-150-SGate	SECS
ETOS-150-HSMS	Serial Only/Ethernet/SECS
ETOS-200-EXX	Serial Only/Ethernet
ETOS-200-SGate	SECS
ETOS-200-HSMS	Serial Only/Ethernet/SECS
ETOS-1000	Serial Only/Ethernet
ETOS-150 Dual	Serial Only/Ethernet/SECS

* 한 번 선택된 드라이버 종류는 그 파일(*.ets) 내에서는 변경할 수 없습니다. 드라이버를 해당 기준에 맞게 선택하지 않고, 다른 드라이버를 선택해서 ETOS 에 다운로드하면 제품이 정상 동작하지 않습니다. 드라이버 선택 시 주의하시기 바랍니다.

5.2.1.4. ETOS 종류 확인

ETOS 종류와 드라이버를 선택한 후 [확인] 버튼을 누릅니다. 이미 열려 있는 파일의 ETOS 종류는 상태 창에서 확인할 수 있습니다.



5.2.1.5. 단종 드라이버

제품 성능 향상을 위해서 드라이버의 종류가 추가되거나 통합될 수 있습니다. 다음 표는 ETOS-PD V2.2 이하에서만 지원하는 드라이버 목록입니다. 더 이상 사용하지 않는 경우입니다. 참고하시기 바랍니다.

표 5.2-4 사용하지 않는 드라이버 종류

드라이버 종류	적용 모델	비고
SECS	ETOS-100-SECS	단종
SECS Multi Port	ETOS-150 SECS, SECS Enet 으로 통합	기능 통합
Ethernet(TCP/IP)	ETOS-100-Plus	단종
Ethernet Any(TCP/IP)	ETOS-100-Plus	단종
Ethernet Multi	ETOS-100A/200/1000, Ethernet Plus 로 통합	기능 통합
POSCON PCS	ETOS-100-Plus	단종
M2I Corp.	ETOS-100-Plus	단종
Ethernet None	ETOS-50A/100A/150/200/1000	Serial Only.로 명칭 변경
Ethernet Plus	ETOS-50A/100A/150/200/1000	Ethernet. 으로 명칭 변경
SECS Enet	ETOS-50A/100A/150/200/1000	SECS.로 명칭 변경

5.3. 시리얼 파라미터

시리얼 파라미터의 설정 방법을 시스템 구성이 Serial Only 인 경우를 가정하여 설명합니다. Serial Only 는 이더넷 통신을 사용하지 않는 경우에 선택합니다. 이더넷을 지원하지 않기 때문에 {파라미터 설정} 대화상자에서 시리얼 파라미터만 활성화되고, 이더넷 파라미터는 나타나지 않습니다. (설정 버튼이 활성화 되지 않은 상태로 나타납니다.)

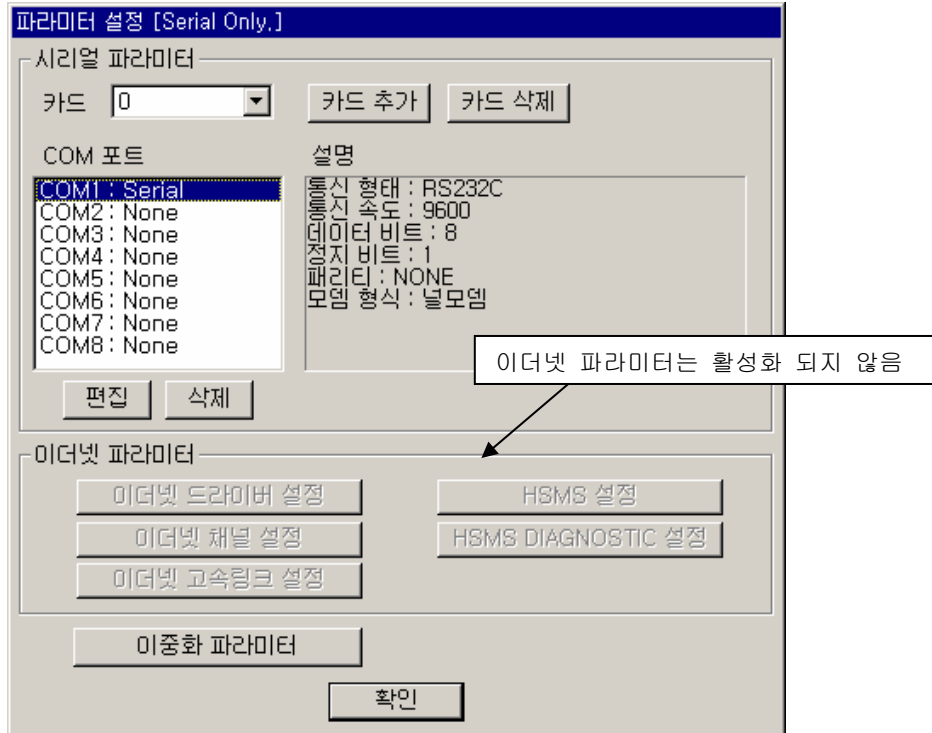


그림 5.3-1 Serial Only 시스템 설정

시리얼 포트를 추가하기 위해서 시스템 설정화면에서 COM 포트를 선택하고 [편집]버튼을 누릅니다. {포트 종류 선택} 대화상자에서 원하는 포트 종류를 선택합니다. 필요없는 포트는 [삭제]버튼을 눌러 삭제할 수 있습니다.

- * [카드 추가] 및 [카드 삭제]버튼은 ETOS-200/1000 인 경우만 활성화됩니다.
- * [이중화 파라미터] 버튼은 ETOS-1000 인 경우만 활성화됩니다.

5.3.1. 포트 종류 선택

{포트 종류 선택} 대화상자에서 포트의 종류를 선택할 수 있습니다. 그러나, 실제 장착된 하드웨어에 해당하는 포트를 설정해야만 정상 동작합니다. 따라서 구입하신 제품명을 확인하시고 [1.3 시리즈 구성](#)을 참고하시어 장착된 포트의 종류를 확인하시기 바랍니다.

포트를 선택하고 [확인]버튼을 누르면 해당 포트의 세부 옵션을 설정하는 화면이 나타납니다. 옵션은 선택한 포트의 종류에 따라 달라집니다.

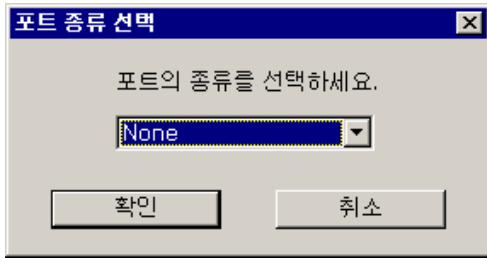


그림 5.3-2 포트 종류 선택 화면

표 5.3-1 COM 포트 종류

종 류	내 용	해당 제품
None	사용하지 않는 포트	ETOS-50A/100A/150/200/1000
Serial	범용시리얼 포트 (RS232C/422/485 통신)	ETOS-50A/100A/150/200/1000
I/O ^(*62)	입력8점, 출력6점 디지털 입출력	ETOS-100A/150
Profibus ^(*63)	Profibus-DP Master 포트	ETOS-100A/150
SECS-I ^(*64)	반도체 전용 통신 포트	ETOS-150/ETOS-200-HSMS
IFOR ^(*65)	ETOS 전용 광 링 포트(IFOR 보드 장착시)	ETOS-200/1000
Enet ^(*4)	이더넷 포트 (IFOR 보드 장착시)	ETOS-200/1000

(*62) 디지털 입,출력 옵션 카드(형명:OPT-HYB-14)을 추가로 장착한 경우만 사용할 수 있습니다.

(*63) Profibus-DPMaster 옵션 카드(형명:OPT-Pnet)을 추가로 장착한 경우만 사용할 수 있습니다.

(*64) 드라이버를 SECS로 선택하여 파일을 연 경우만 나타납니다. ETOS-XXX-HSMS/SGate 모델에서만 사용할 수 있습니다. (XXX는 50A/150/200)

(*65) [옵션]-> [편집 옵션]에서 IFOR 편집을 체크한 경우만 나타납니다. IFOR 보드를 장착한 경우만 사용할 수 있습니다.

5.3.1.1. Serial 포트

범용 시리얼 장비(RS232C, RS422, RS485)를 사용하는 경우에 선택합니다. 통신 파라미터 설정 이외의 시리얼 통신에 대한 보다 자세한 사항은 [8. 시리얼 통신](#)을 참고하시기 바랍니다.

{COM 포트 설정 (Serial)} 대화상자에서 통신하려고 하는 범용 시리얼 통신 장비와 통신 형태, 속도 등의 통신 파라미터를 일치시켜야 합니다.

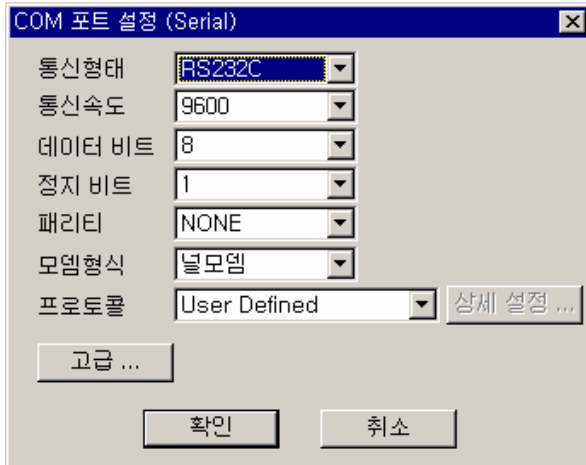


그림 5.3-3 COM 포트 설정 예

표 5.3-2 COM 포트 설정 항목

항 목	내 용
통신형태	RS232C/RS422/RS485/LOG 중에서 선택합니다.
통신속도	통신 속도(baud rate, bps)를 설정합니다.
데이터비트	7 / 8 비트
정지비트	1 / 2 비트
패리티	NONE(없음)/EVEN(짝수)/ODD(홀수)/SPACE(1)/MARK(0)
모뎀 형식	널모뎀/전용모뎀 에서 선택 (*66)
프로토콜	내장 프로토콜을 사용하는 경우 선택합니다. (*67)
상세 설정	내장 프로토콜 별 상세 설정 (*68)
고급	시리얼 통신 고급 설정 참조

(*66) 다이얼업 모뎀은 전용모뎀 선택

(*67) MODBUS, GLOFA CNET 등을 지원합니다. (OS V1.2 이상에서만 지원)

(*68) 프로토콜 별로 다릅니다. 해당 프로토콜 편을 참조하시기 바랍니다.

5.3.1.2. 디지털 입출력(I/O) 포트

입력 8 점 출력 6 점인 디지털 입출력(I/O) 옵션카드를 장착하는 경우입니다. COM3 은 입력 포트로 COM4 는 출력 포트로 고정됩니다. 그러나 포트 설정은 COM3 은 “I/O”로, COM4 는 “None”으로 설정해야 합니다.

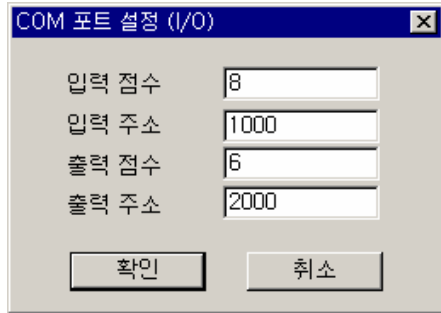


그림 5.3-4 I/O 파라미터 설정 예

표 5.3-3 I/O 파라미터 설정 사항

항 목	내 용
입력점수	장착한 옵션카드의 입력 점수를 입력합니다. 8 점
입력주소	입력 값이 저장되는 메모리 상의 주소를 설정합니다.
출력점수	장착된 옵션카드의 출력 점수를 입력합니다. 6 점
출력주소	출력 값에 대응되는 메모리 상의 주소를 설정합니다.

5.3.1.3. Profibus, IFOR, Enet 포트

해당 사용설명서를 참조하시기 바랍니다. (별도 제공)

5.3.1.4. SECS-I 포트

반도체 장비용 프로토콜인 SECS-I 을 사용하는 경우에 선택합니다. SECS-I 파라미터 설정 이외의 SECS 통신에 대한 보다 자세한 사항은 [10. SECS 통신](#)을 참고하시기 바랍니다.

{COM 포트 설정 (SECS-I)} 창에서 통신하려고 하는 SECS-I 장비와 통신 형태, 속도 등의 통신 파라미터를 일치시켜야 합니다.

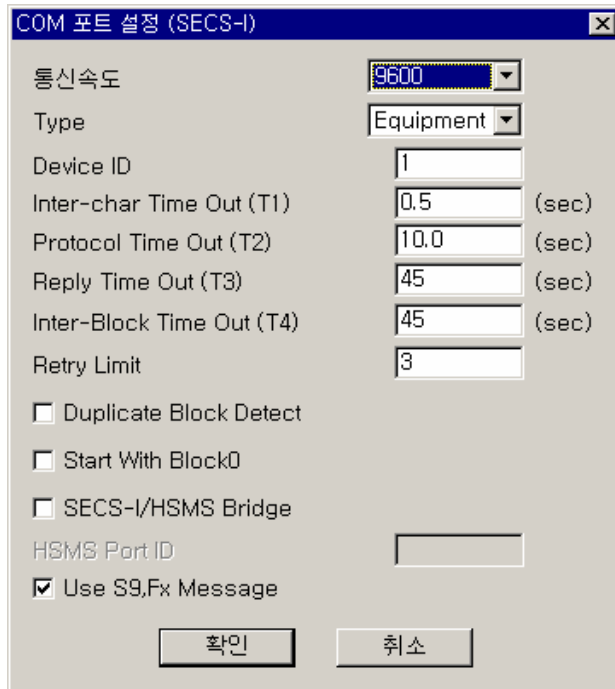


그림 5.3-5 SECS-I 설정

표 5.3-4 SECS-I 설정 항목

항 목	내 용
통신속도	통신 속도(baud rate, bps)를 설정.
Type	ETOS 의 동작 Type(Host, Equipment)을 선택합니다.
Device ID	SECS 통신에 사용할 Device ID 를 설정합니다.
Inter-char Time Out(T1)	Inter-char Time Out(T1)을 설정합니다.
Protocol Time Out(T2)	Protocol Time Out(T2)을 설정합니다.
Reply Time Out(T3)	Reply Time Out(T3)을 설정합니다.
Inter-Block Time Out(T4)	Inter-Block Time Out(T4)을 설정합니다.
Retry Limit	재전송 횟수를 설정합니다.
Duplicate Block Detect	중복된 블록을 자동으로 제거합니다.
Start With Block0	Multiple Block 의 경우, Block No.를 0 번부터 시작합니다. Check 하지 않으면 1 번부터 시작합니다.
SECS-I/HSMS Bridge	프레임과 프로세스를 정의하지 않고, SECS-I 과 HSMS 사이에 Bridge 만으로 사용할 경우에 체크합니다.
HSMS Port ID	SECS-I/HSMS Bridge 로 사용할 경우, 본 SECS-I 포트와 연결할 HSMS Port ID 를 지정합니다.

5.4. 이더넷 파라미터

- ▶ 모델명이 ETOS-1000, ETOS-XXX^(*69)-EXX, ETOS-XXX^(*1)-HSMS/SGate 이고,
- ▶ 드라이버를 Ethernet 또는 SECS 로 설정한 경우만 이더넷 파라미터를 설정할 수 있습니다.

이더넷 통신을 하기 위해서는 먼저 ETOS 의 IP 주소 등이 설정되어야 합니다.

[4.3.4.13 \[온라인\(O\)\] -> \[TCP/IP 설정\]](#) 을 참고하여 설정하시기 바랍니다.

시리얼, 필드버스 통신과 이더넷 통신 또는 이더넷 통신만을 사용하는 경우 드라이버 타입을 Ethernet 으로 선택합니다. 제품별로 선택가능한 드라이버의 목록은 [5.2.1.3. 모델 별 선택가능 드라이버](#) 를 참조하시기 바랍니다.

▶ 이더넷 드라이버

내장 프로토콜 슬레이브

▶ 이더넷 채널

내장 프로토콜 마스터, 사용자정의 프로토콜 마스터/슬레이브를 포함

▶ 이더넷 고속링크

ETOS 간, 또는 GLOFA PLC 이더넷 통신 모듈과의 고속링크 통신

▶ HSMS 설정

모델명이 ETOS-XXX^(*70)-HSMS/SGate 인 경우, HSMS 통신 파라미터 설정

▶ HSMS DIAGNOSTIC 설정

모델명이 ETOS-XXX-DIAG 인 경우 사용

이더넷 드라이버는 슬레이브로 동작하며, 파라미터 설정만으로 통신할 수 있습니다. 이더넷 채널은 마스터 또는 슬레이브로 동작하며 내장 프로토콜을 사용하거나(전용 아이템 사용) 시리얼과 같은 방법으로 프레임을 등록하여 사용합니다.

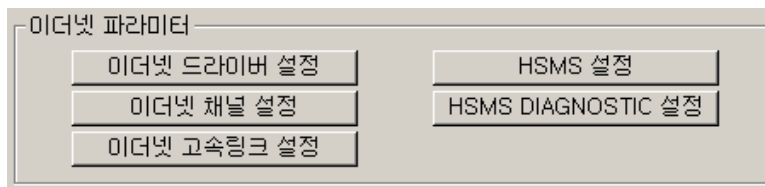


그림 5.4-1 이더넷 파라미터

(*69) 50A/100A/150/200 중 하나

(*70) 50A/150/200 중 하나

5.4.1. 이더넷 드라이버 설정

이더넷 드라이버란 ETOS 내장 프로토콜 슬레이브를 말합니다. 내장 프로토콜에는 GLOFA Enet, MODBUS/TCP 등이 있습니다. 슬레이브란 통신 상대국의 요청에 대해 수동적으로 응답하는 동작 방식을 말하며, 통신 상대국은 마스터라고 합니다.

이더넷 드라이버는 상대국이 MODBUS 나 GLOFA 프로토콜을 사용하여 ETOS 의 데이터를 읽거나, 상대국의 데이터를 ETOS 메모리에 쓰는 경우 사용합니다. 통신 상대국은 주로 MMI^(*71)(또는 HMI) 인 경우가 많습니다. 이더넷 드라이버는 사용자가 별도의 통신 프로그램을 작성할 필요 없이 파라미터 설정만으로 상대 기기와 통신이 가능합니다. 다음 그림은 이더넷 드라이버를 사용하는 전형적인 예인 MMI PC 와 통신하는 경우입니다. MMI PC 에서 데이터를 요청하면, ETOS 가 응답하는 형식으로 동작합니다.

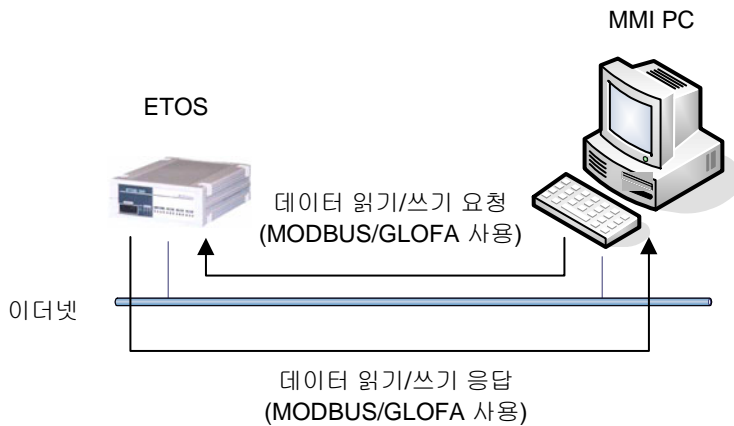


그림 5.4-2 이더넷 드라이버 사용 예

다음 예는 이더넷 드라이버로 GLOFA Enet 과 MODBUS TCP 의 두 가지를 설정한 경우입니다. 그림에서와 같이 서로 다른 프로토콜을 사용하는 두 개 이상의 마스터 기기와 각각 통신이 가능합니다.

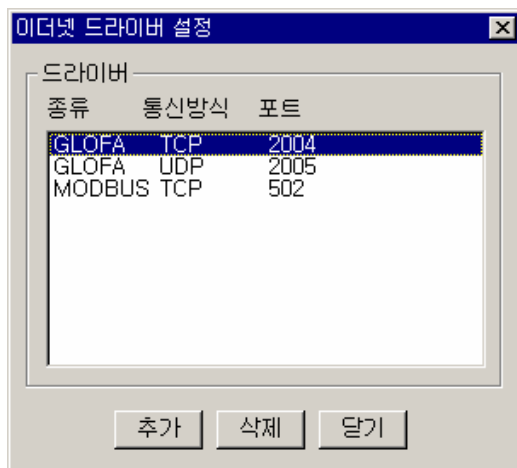


그림 5.4-3 이더넷 드라이버 설정 창

(*71) Man Machine Interface

5.4.1.1. 이더넷 드라이버 종류

지원하는 드라이버 종류는 다음과 같습니다. (OS 버전 1.20 기준)

표 5.4-1 이더넷 드라이버 종류

종 류	설 명
MODBUS/TCP	Modicon 사의 오픈 프로토콜
GLOFA Enet	LG 산전의 이더넷 전용 프로토콜
Etos	ETOS 전용 프로토콜 ^(*72)

{이더넷 드라이버 설정}에서 서로 다른 복수의 전용 드라이버를 포트별로 설정할 수 있습니다. 그러나 제품별로 설정할 수 있는 드라이버의 수에 제한이 있습니다. 다음 표를 참고하시기 바랍니다.

표 5.4-2 ETOS 제품별 설정 가능한 드라이버 수

종 류	설정 가능한 드라이버 수
ETOS-50A	4
ETOS-100A/150	16
ETOS-200	24
ETOS-1000	32

- * 설정 가능한 드라이버의 수는 설정된 이더넷 채널이 없는 경우를 가정한 것입니다. 이더넷 채널을 설정했다면, 설정한 수만큼 사용가능한 드라이버의 수가 줄어들게 됩니다. 주의하시기 바랍니다.
- * 이더넷 드라이버는 1:N 통신이 가능합니다. 따라서 여러대의 마스터 기기가 설정한 한 개 포트 로 연결하여 데이터를 가져갈 수 있습니다.

(*72) 프로토콜 자료는 문의바랍니다.

5.4.1.2. 이더넷 드라이버 포트 설정

이제 {이더넷 드라이버 설정} 창에서 [추가]버튼을 눌러 해당 포트를 설정합니다.

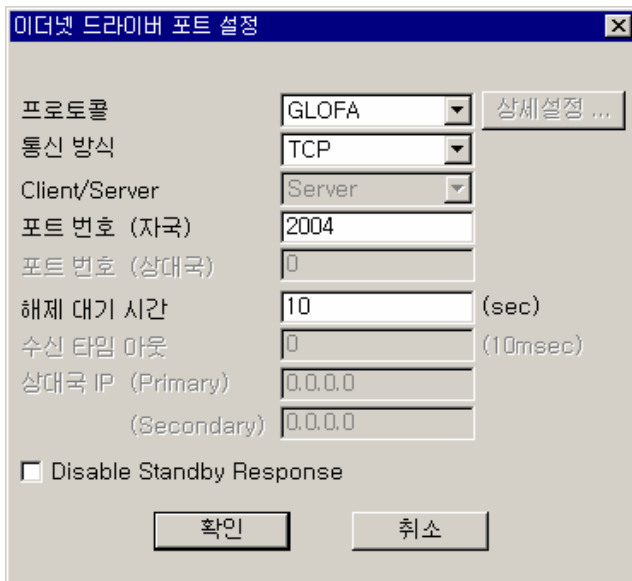


그림 5.4-4 드라이버 포트 설정 예

표 5.4-3 드라이버 포트 설정 항목

항 목	내 용	
프로토콜	상대기기와의 통신에 사용할 프로토콜을 선택합니다. MODBUS/GLOFA Enet/ETOS 전용 중에서 선택	
통신방식	TCP/UDP 중에서 선택합니다.	
Client/Server	Server 로 고정됩니다.	
포트번호(자국)	서버로 동작할 소켓 포트의 번호를 설정합니다.(서버만 적용)	
해제대기시간	이 시간 동안 통신을 수행하지 않으면 통신 연결을 끊습니다.	
수신타임아웃	해당 사항 없음	
상대국	IP (Primary)	해당 없음
	(Secondary)	해당 없음
Disable Standby Response	ETOS-1000 이중화 모델의 사용자정의 시 사용	

프로토콜을 MODBUS 로 선택하면 [상세 설정] 버튼이 활성화 되며, 상세 설정 창이 나타납니다.
상세 설정 등 프로토콜에 따른 보다 자세한 설명은 [9.2 이더넷 드라이버](#)를 참조하시기 바랍니다.

5.4.2. 이더넷 채널 설정

이더넷 채널은 내장 프로토콜 마스터로 동작해야 하는 경우, 또는 내장되어 있지 않은 프로토콜 (사용자 정의 프로토콜)을 가지고 통신을 해야 하는 경우(마스터/슬레이브 모두 가능)에 사용됩니다. 이더넷 드라이버가 슬레이브 포트를 설정하는 것이라고 한다면, 이더넷 채널은 마스터 포트를 설정하는 것입니다. 다음 그림은 이더넷 채널을 사용하는 전형적인 예인 PLC 와 통신하는 경우입니다. ETOS 가 마스터가 되어 PLC 에 데이터를 요청하면, PLC 가 응답하는 형식으로 동작합니다.

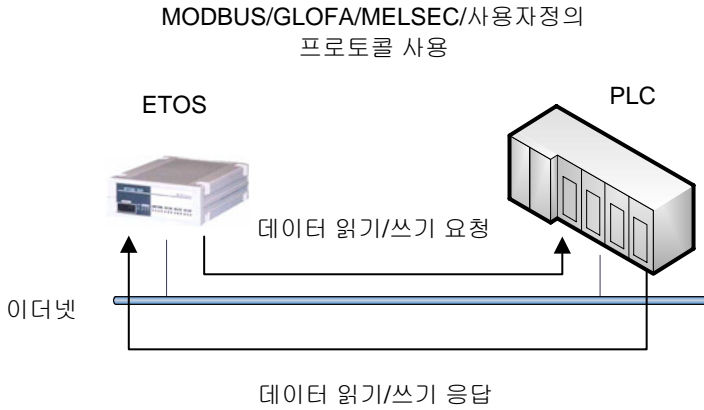


그림 5.4-5 이더넷 채널 사용 예

{이더넷 채널 설정} 창에서 편집하고자 하는 채널 번호에 마우스를 클릭하고 [편집]버튼을 누릅니다. 초기 화면에는 설정된 채널이 없습니다.

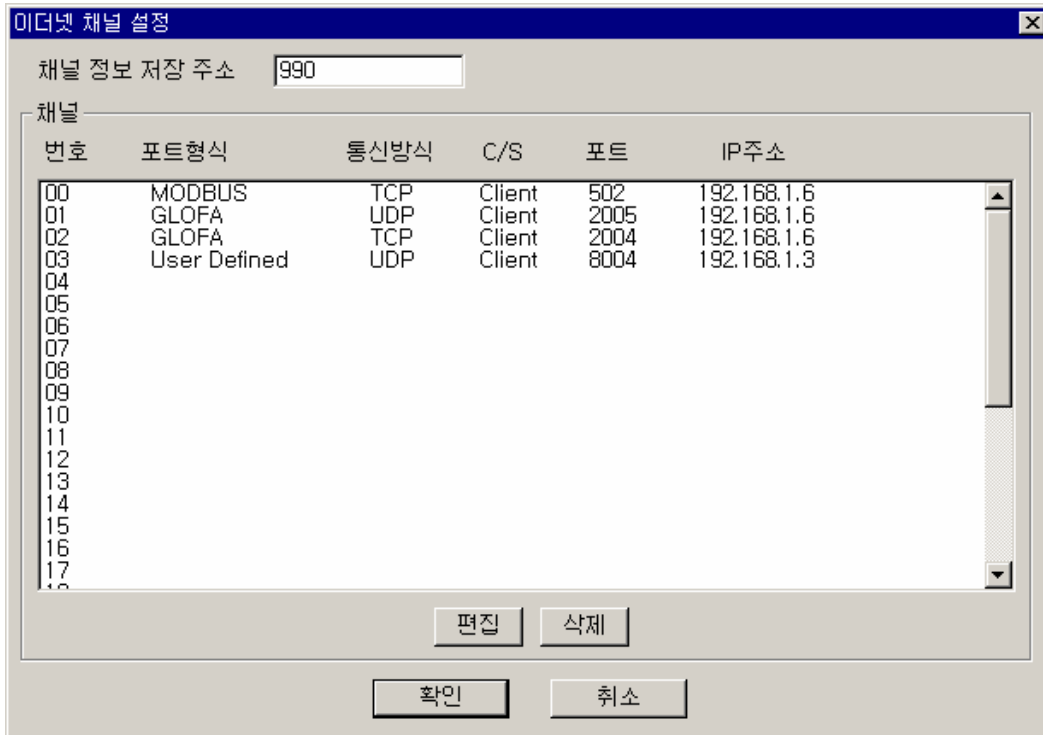


그림 5.4-6 이더넷 채널 설정 화면

<채널정보 저장주소>는 각 채널의 접속 상태를 저장하는 번지입니다. 사용자는 프로그램 작성 시 이 번지를 적절히 이용하여 채널 접속 정보를 알 수 있습니다. 예를 들어 0 번지로 되어 있으면 0 번지 0 비트는 채널 번호 00의 연결 상태, 1번 비트는 채널 번호 01의 연결 상태를 나타냅니다. 해당 데이터가 1 이면 연결, 0 이면 연결되지 않았다는 의미입니다.

<채널정보 저장주소>는 지정한 위치부터 1~4 바이트를 사용하며, 제품에 따라 다릅니다.^(*73)

표 5.4-4 제품 별 채널정보 저장에 사용되는 바이트 수

제품	사용 바이트 수
ETOS-50A	1
ETOS-100A/150	2
ETOS-200	3
ETOS-1000	4

5.4.2.1. 포트 종류

지정할 수 있는 포트의 형식은 다음과 같습니다.

표 5.4-5 이더넷 채널 프로토콜 종류

포트 형식	설 명	특 징
MODBUS	Modicon사의 MODBUS/TCP 마스터	프레임 정의 필요없음
GLOFA	LG산전의 GLOFA Enet 마스터	프레임 정의 필요없음
MELSEC	Mitsubishi 사의 MELSEC 마스터	프레임 정의 필요없음
User Defined	위의 두가지 이외의 모든 프로토콜 마스터/슬레이브	프레임 정의 필요함

▶ MODBUS (마스터 포트)

MODBUS 마스터로 동작해야 하는 경우, 이더넷 채널의 포트 형식을 MODBUS 로 지정합니다. 통신 방식은 TCP 로 고정되며, Client/Server 역시 Client 로 고정됩니다. 통상 포트 번호(상대국)에는 502 를 입력합니다. 상대국 IP 에는 MODBUS 서버로 동작할 상대 기기의 IP 주소를 입력합니다.

▶ GLOFA (마스터 포트)

상대기기가 GLOFA Enet 슬레이브이고, 상대기기로부터 데이터를 읽어오거나, 데이터를 상대 기기에 쓰고자 하는 경우에 사용합니다. 보통 TCP 인 경우 2004 번 포트를, UDP 인 경우 2005 번 포트를 사용합니다.

▶ MELSEC (마스터 포트)

상대기기가 MELSEC PLC 일 경우, 상대기기로부터 데이터를 읽거나, ETOS 데이터를 PLC 에 쓸 수 있습니다.

^(*73) 시리즈별 설정 가능 채널 수와 관련이 있습니다. 채널 수/8 로 나눈 값이 사용하는 메모리 크기입니다.

▶ User Defined (마스터/슬레이브 포트)

ETOS 가 제공하지 않는 프로토콜을 사용하는 기기와 통신하려고 하는 경우, 사용자는 해당 프로토콜의 프레임을 정의하여 상대기기와 통신 할 수 있습니다
이더넷 드라이버와 마찬가지로 복수의 이더넷 채널을 설정할 수 있습니다.

표 5.4-6 제품 별 설정가능한 채널 수

제품	설정 가능한 이더넷 채널 수
ETOS-50A	4
ETOS-100A/150	16
ETOS-200	24
ETOS-1000	32

- * 위 표의 수치는 설정된 이더넷 드라이버가 없는 경우를 가정한 것입니다. 이더넷 드라이버를 설정했다면, 설정한 수만큼 사용가능한 채널의 수가 줄어들게 됩니다. 주의하시기 바랍니다.
- * 이더넷 채널은 상대와 1:1 통신을 합니다. 따라서, 2대 이상의 상대기기와 통신하고자 할 때는 복수개의 채널을 설정해야 합니다.

5.4.2.2. 이더넷 채널 포트 설정

이제 {이더넷 채널 설정} 창에서 [편집]버튼을 눌러 포트 별 세부 설정을 합니다.
{이더넷 채널 포트 설정} 대화상자에서 해당 포트의 파라미터를 설정합니다.

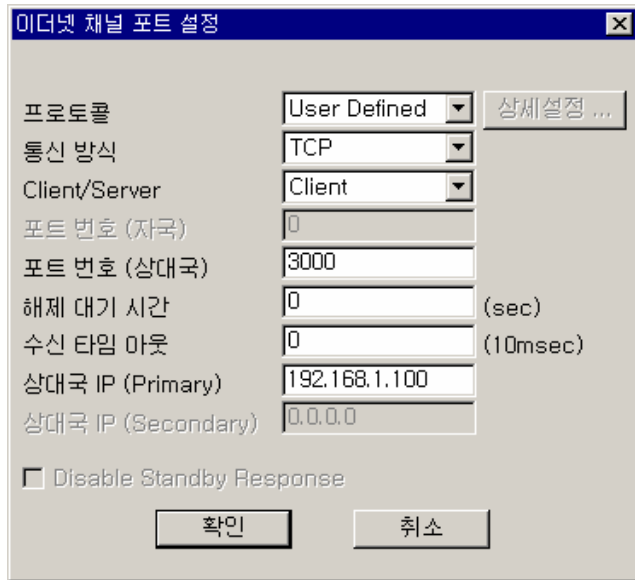


그림 5.4-7 이더넷 채널 포트 설정 화면

표 5.4-7 이더넷 채널 포트 설정 항목

항 목	내 용
프로토콜	상대기기와의 통신에 사용하는 프로토콜을 선택합니다. 사용자정의 프로토콜(프레임을 정의함)은 User Defined를 선택합니다. User Defined / MODBUS/ GLOFA / MELSEC
통신 방식	TCP/UDP 중에서 선택, MODBUS를 선택하면 TCP로 고정됩니다.
Client/Server	Client/Server 중에서 선택, 프로토콜을 MODBUS/GLOFA/MELSEC중에서 선택하면 Client로 고정됩니다. (MODBUS, GLOFA, MELSEC은 마스터로 동작합니다.)
포트 번호(자국)	자국의 포트 번호를 입력합니다. 통신 방식이 TCP인 경우는 입력하지 않습니다. (User Defined-TCP-Server인 경우는 입력)
포트 번호(상대국)	상대국의 포트 번호를 입력합니다. (User Defined-TCP-Server인 경우는 입력하지 않습니다.)
해제 대기 시간	Server인 경우만 설정합니다. 초 단위
수신 타임 아웃	IF_REQ_RESP 아이템 사용시 프레임 송신 후 응답을 기다리는 최대 시간 10ms 단위로 설정
상대국 IP	지정된 채널로 ETOS와 통신할 상대 장비의 IP주소를 입력합니다.

▶ 상대국 IP

ETOS 가 클라이언트인 경우 반드시 서버 장비의 IP 주소를 설정합니다. 만약, 서버가 DHCP 를 사용하여 IP 를 동적으로 할당받는 경우라면, IP 주소가 계속 바뀌므로 설정한 IP 와 달라져 통신이 불가능하게 됩니다. 따라서, 서버는 반드시 고정 IP 주소를 할당받아야 하며, DHCP 를 사용해서는 통신이 불가능합니다.

ETOS 가 서버로 사용될 경우는 클라이언트의 IP 주소를 설정합니다. 상대 장비는 반드시 지정한 IP 주소로 설정되어 있어야 합니다. 만약, 상대 장비가 DHCP 를 사용하여 동적으로 IP 를 할당받는 경우라면 특정 IP 주소를 입력할 수 없게 됩니다. 이 경우, 상대국 IP 에 0.0.0.0 을 입력하면, 상대 장비의 IP 주소가 변경되더라도 접속을 허용합니다.

* 이더넷 채널 설정에 대한 보다 자세한 사항은 [9.3 이더넷 채널 설정](#)을 참조하시기 바랍니다.

5.4.3. HSMS 설정

▶ 모델명이 ETOS-XXX^(*74)-HSMS/SGate 인 경우만 해당합니다.

HSMS 설정은 ETOS 가 HSMS 로 통신을 해야 하는 경우 설정합니다.

{HSMS 설정} 창에서 [추가]버튼을 눌러 포트를 추가하거나, 편집하고자 하는 채널 번호에 마우스를 클릭하여 선택한 후, [편집]버튼을 누릅니다. 초기 화면에는 설정된 채널이 없습니다.

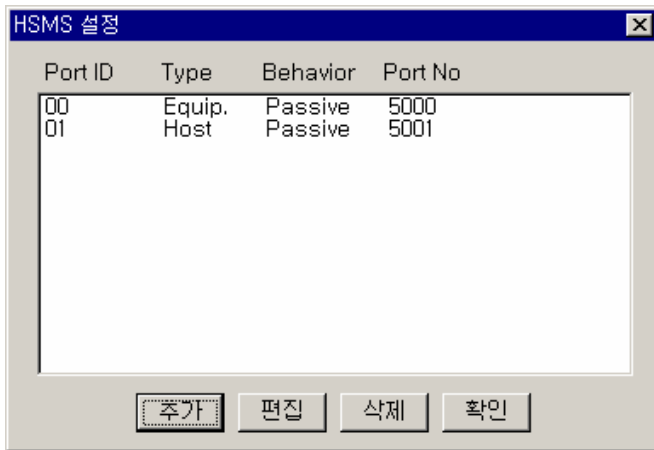


그림 5.4-8 HSMS 설정 화면

(*74) 50A/150/200 중 하나

5.4.3.1. HSMS port 설정

[추가] 또는 [편집]버튼을 눌러 포트 별 세부 설정을 합니다.

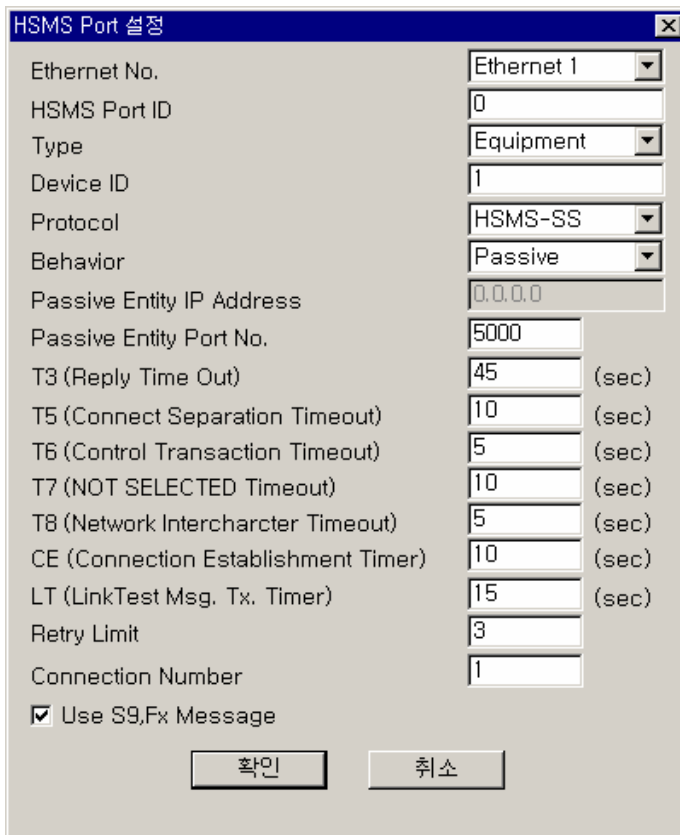


그림 5.4-9 HSMS Port 설정 화면

표 5.4-8 HSMS Port 설정 항목

항 목	내 용
Ethernet No.	ETOS-150-Dual의 경우, Ethernet 1/2 중 선택
HSMS Port ID	IF_SECS_MC_REQ_RESP, IF_SECS_MC_FRAME_RCV, SECS_MC_FRAME_SND등의 아이템에서 Port를 구별하기 위한 ID입니다.
Device ID	SECS통신에 사용할 Device ID를 설정합니다.
Protocol	현재 HSMS-SS만이 가능합니다.
Behavior	Passive/Active를 선택합니다.
Passive Entity IP Address	Behavior를 Active로 한 경우, Passive의 IP Address를 설정합니다.
Passive Entity Port No.	Behavior를 Active로 한 경우, Passive의 Port No.를 설정합니다.
T3(Reply Time Out)	Reply Time Out(T3)을 설정합니다.
T5(Connect Separation Time Out)	Connect Separation Time Out(T5)을 설정합니다.
T6(Control Transaction Time Out)	Control Transaction Time Out(T6)을 설정합니다.

표 5.4-8 HSMS Port 설정 항목 (계속)

항 목	내 용
T7(NOT SELECTED Time Out)	NOT SELECTED Time Out(T7)을 설정합니다.
T8(Network Intercharacter Time Out)	Network Intercharacter Time Out(T8)을 설정합니다.
CE(Connection Establishment Timer)	Connection Establishment Timer 을 설정합니다.
LT(Link Test Msg. Tx. Timer)	Link Test Msg. Tx. Timer을 설정합니다.
Retry Limit	재전송 시도 횟수를 설정합니다.
Connection Number	Passive로 설정한 경우, 허용해 줄 Connection의 수를 설정합니다.
Use S9,Fx Message	Error발생시 ETOS가 자동으로 S9Fx 프레임을 전송할 지를 선택합니다.

* HSMS 에 대한 보다 자세한 사항은 [10. SECS 통신](#)을 참조하시기 바랍니다.

프레임

제 6 장

이 장은 프레임의 정의와 설정방법에 대한 전반적인 내용을 다루고 있습니다.

	내용	위치
6.1	개요	106
6.2	프레임의 구성	107
6.3	프레임 등록	107
6.4	프레임 등록 예제	123
6.5	SECS 메시지 등록	134
6.6	SECS 메시지 등록 예제	139

6. 프레임

ETOS가 동작하기 위해서는 파라미터, 프레임, 그리고 프로세스의 세가지가 필요합니다. 그 중 두 번째인 프레임을 설명합니다. 프레임은 이더넷 및 시리얼 통신에서 사용할 프로토콜을 구성하는 단위 메시지입니다. MODBUS, GLOFA CNET, GLOFA ENET, MELSEC 등을 제외한 프로토콜을 사용하는 장비와 통신하기 위해서는 사용자가 해당 시스템의 통신 프레임을 정의해야 합니다. 통신 프레임 정의시 사용자는 송, 수신 프레임의 저장 위치 및 저장 형식 등을 지정할 수 있습니다

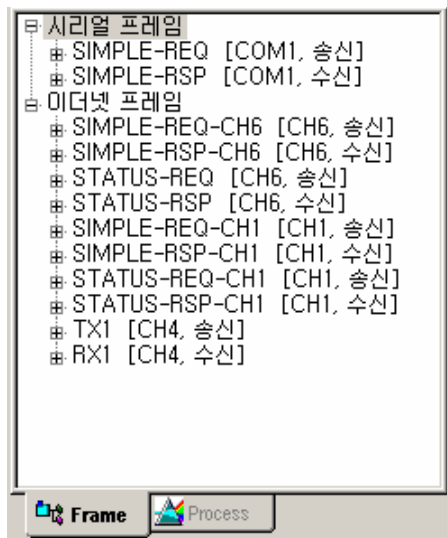
- * MODBUS, GLOFA CNET, GLOFA ENET, MELSEC을 사용하고자 한다면 프레임 등록은 필요없습니다. [7. 프로세스 및 아이템](#) 으로 건너 뛸 수 있습니다.

6.1. 개요

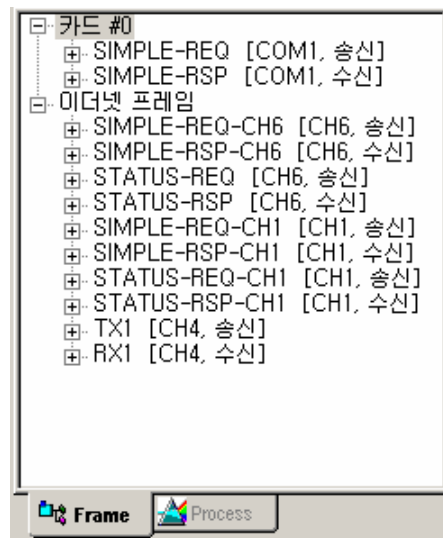
이더넷 및 시리얼 통신에서 사용할 프로토콜을 구성하는 단위 메시지입니다. MODBUS, GLOFA CNET, GLOFA ENET, MELSEC 등을 제외한 통신 프로토콜을 사용하는 장비와 통신하기 위해서는 사용자가 해당 시스템이 사용하는 프레임을 정의해야 합니다. 통신 프레임 정의시 사용자는 송, 수신 프레임의 저장 위치 및 저장 형식 등을 지정할 수 있습니다.

다음은 {프레임 등록} 창의 화면입니다. 시리얼 송, 수신 프레임과 이더넷 송, 수신 프레임을 등록하였습니다. 시리얼 프레임은 포트별로 정의하며, 이더넷 프레임도 파라미터에서 등록한 채널별로 정의합니다. 따라서 프레임을 정의하기 위해서는 해당 포트 또는 채널을 파라미터에서 미리 설정해 주어야 합니다.

프레임 등록 창(ETOS-50A/100A/150)



프레임 등록 창(ETOS-200/1000)



- * ETOS-50A/100A/150 인 경우에는 '시리얼 프레임'이라는 제목아래 시리얼 프레임이 등록되며, ETOS-200/1000 인 경우에는 '카드 #0, 카드 #1,...' 제목아래에 시리얼 프레임이 등록됩니다.

6.2. 프레임의 구성

프레임은 통신 프로토콜을 구성하는 개별 메시지입니다. 데이터를 송, 수신한다는 것은 프로토콜별로 정해진 형식을 갖는 각각의 프레임들을 송, 수신 한다는 의미입니다. 각각의 프레임은 논리적으로 헤더(HEAD), 메시지(BODY), 테일(TAIL)로 구분할 수 있습니다. 시리얼 통신의 경우 헤더, 메시지, 테일로 구성된 경우가 일반적입니다. 또, 이더넷 통신의 경우는 보통 헤더와 메시지만을 갖는 것이 일반적입니다.

일부 시스템은 BODY 만으로 프레임을 정의하기도 하는데, 통신의 신뢰성을 위해서 HEAD, BODY, TAIL 로 하나의 프레임을 정의하는 것이 바람직합니다. 프레임의 구체적인 구성은 통신 장비마다 다르므로 해당 장비의 프로토콜 자료를 참조해서 작성해야 합니다.

다음 그림은 특정 프레임을 논리적으로 HEAD, BODY, TAIL 로 구분한 예입니다. (실제 프레임은 05H 01H 57H ... 데이터의 연속적인 흐름일 뿐입니다.)

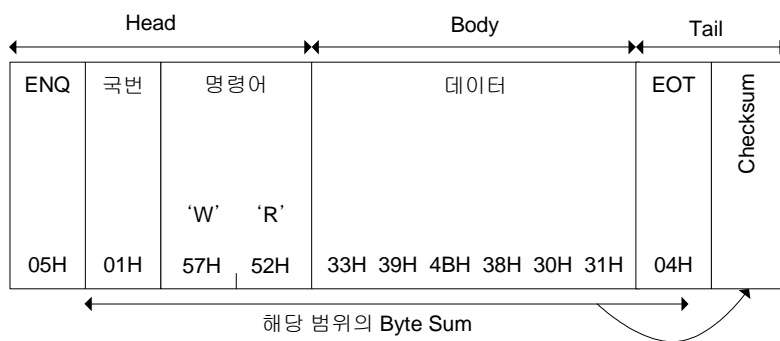


그림 6.2-1 프레임 구성 예

6.3. 프레임 등록

통신하고자 하는 상대 장비가 MODBUS RTU/ASCII/TCP, GLOFA Cnet/Enet, MELSEC 프로토콜을 사용하는 장비가 아니라면 프레임을 정의해야 합니다.^(*75) 프레임 정의는 왼쪽 상단의 [프레임]탭을 눌러 {프레임 등록} 창을 활성화시킨 상태에서 합니다.

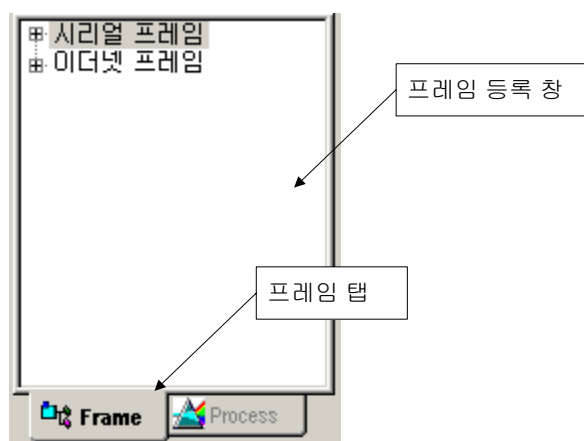


그림 6.3-1 프레임 창

(*75) MODBUS, GLOFA, MELSEC 프로토콜을 사용하는 경우에는 프로토콜별 전용 아이템을 사용하면 프레임을 정의하지 않아도 통신이 가능합니다.

6.3.1. 그룹 추가

ETOS 는 HEAD, TAIL 이 동일한 프레임들을 하나의 그룹으로 묶어서 관리합니다. 따라서, 한 개의 그룹안에는 HEAD, TAIL 을 한 개만 정의할 수 있으며, BODY 는 여러 개를 정의할 수 있습니다. HEAD 나 TAIL 이 다른 프레임은 별도의 그룹으로 등록해야 합니다. 그러나, HEAD, TAIL 이 같은 프레임들이 반드시 하나의 그룹안에 모두 등록되어야 하는 것은 아닙니다. 즉, HEAD, TAIL 이 같은 프레임일지라도 서로 다른 그룹으로 등록하는 것이 가능합니다.

6.3.1.1. ETOS-200/1000

{프레임 등록} 창의 빈 공간에 마우스를 위치하고 오른쪽 버튼을 클릭하면, [카드 추가]와 [이더넷 프레임 추가] 팝업 메뉴가 나타납니다. CPU 보드(SCP 또는 RCP) 다음 위치부터 차례로 0,1,2,...7 번 카드입니다. CPU 보드만 있는 경우에는 카드 추가가 필요 없습니다.

시리얼 프레임을 등록하기 위해서 [카드 #N]에 마우스를 위치하고 오른쪽 버튼을 클릭하면 [그룹 추가] 팝업 메뉴가 활성화 됩니다. [그룹 추가]를 메뉴를 선택합니다.

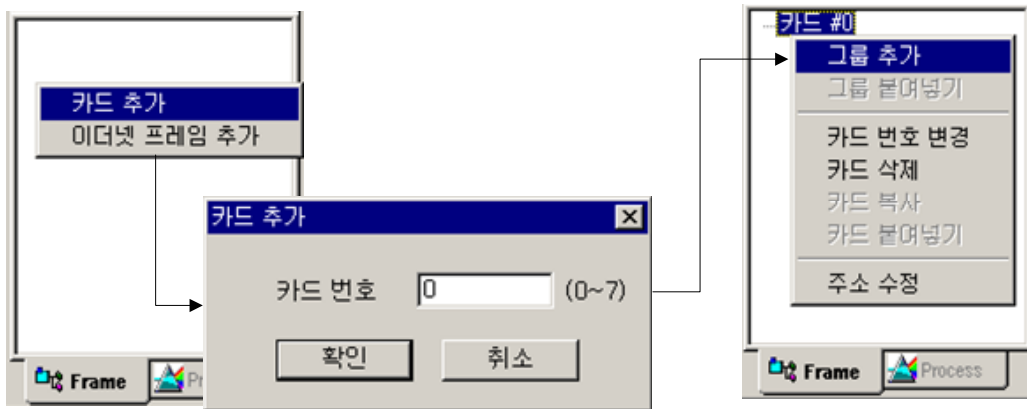


그림 6.3-2 그룹 추가 (ETOS-200/1000)

6.3.1.2. ETOS-50A/100A/150

ETOS-50A/100A/150 의 경우는 카드 추가가 필요없습니다. <시리얼 프레임> 에 마우스를 위치하고 오른쪽 버튼을 클릭하면, [그룹 추가] 팝업 메뉴가 나타납니다. [그룹 추가]를 메뉴를 선택합니다.

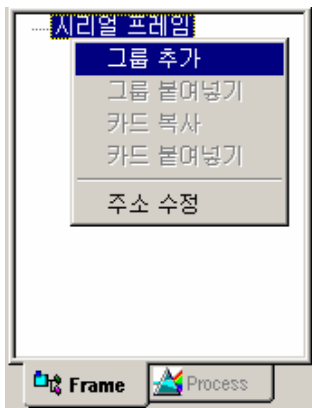


그림 6.3-3 그룹 추가 (ETOS-50A/100A/150)

이제 {그룹 편집} 대화상자에서 <그룹명>과 <포트번호>(시리얼) 또는 <채널번호>(이더넷), <프레임 종류>를 선택합니다. (*76)

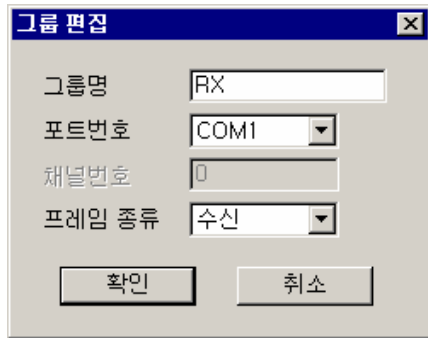


그림 6.3-4 그룹 편집

표 6.3-1 그룹 편집 설정 항목

항 목	내 용
그룹 명	그룹이름을 지정하며, 최대 영문 20 자까지 입력 가능합니다. 그룹 명은 중복하여 사용할 수 없습니다. 한글입력 및 특수문자의 사용이 가능합니다.
사용 포트	ETOS 에서 사용할 시리얼 포트를 지정합니다.
채널 번호	<이더넷 프레임>의 경우에 활성화됩니다. 이더넷 채널 설정에서 설정한 채널 번호를 입력합니다.
프레임 종류	송신, 수신 프레임을 지정합니다.

다음 그림은 “RX”란 그룹 이름으로 COM1 수신 프레임 그룹을 등록한 화면입니다.

수신 그룹 등록(ETOS-50A/100A/150)



수신 그룹 등록(ETOS-200/1000)

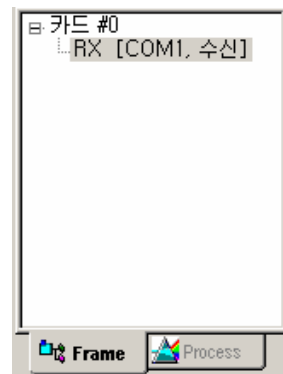


그림 6.3-5 수신 그룹 등록 예

(*76) 프레임 종류는 등록 후에 변경할 수 없습니다. 프레임 종류를 변경하기 위해서는 그룹을 삭제하고 다시 등록해야 합니다.

6.3.2. 프레임 추가

그룹은 HEAD, TAIL 이 동일한 여러 개의 프레임을 가상으로 묶어놓은 것입니다. 따라서 HEAD, TAIL 은 그룹 안에 한 개씩만 등록가능하며, BODY 부분은 여러 개를 등록할 수 있습니다. 이제, 그룹을 구성하는 HEAD, TAIL, BODY 각 부분을 추가합니다. 그룹을 선택하고 오른쪽 마우스 버튼을 클릭해서 [프레임 추가] 메뉴를 선택합니다. [프레임 추가]를 통해 프레임을 구성하는 HEAD, BODY, TAIL 을 추가합니다.

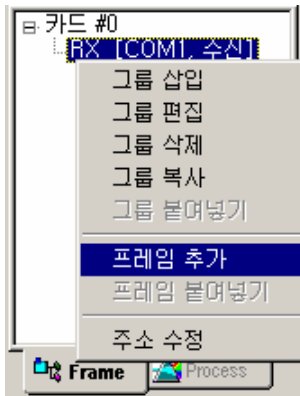


그림 6.3-6 프레임 추가

프레임 편집화면에서 프레임의 종류와 이름을 선택하여 지정합니다. HEAD, TAIL 은 그룹안에서 유일하므로 이름을 입력하지 않습니다. 또, 경우에 따라, HEAD, TAIL 은 생략이 가능합니다. 그러나 신뢰성 있는 통신을 위해서는 HEAD, TAIL 을 설정하는 것이 바람직합니다.

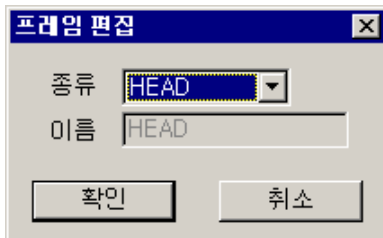


그림 6.3-7 프레임 편집 창

6.3.2.1. 프레임 종류

[프레임 추가] 메뉴에 의해 사용자는 프레임을 구성하는 HEAD, BODY, TAIL 을 추가할 수 있습니다. 다음 표는 HEAD, BODY, TAIL 각 프레임의 특징입니다.

표 6.3-2 프레임 종류

항 목	이 름	설정개수	내 용
HEAD	설정불가	0 ~ 1	* HEAD, TAIL 은 그룹안에서 1 개만 설정 가능합니다. * 프로토콜에 따라 생략이 가능합니다. * 프로세스에서 참조가 불가능 합니다.
TAIL	설정불가	0 ~ 1	
BODY	설정가능	1 ~ 64	* BODY 는 최대 64 개까지 설정 가능합니다. * 국문,영문,숫자,특수문자를 조합하여 설정 가능합니다. * 그룹 내에서 중복된 이름을 사용할 수 없습니다. * 1 개 이상을 반드시 설정해야 합니다. * 프로세스에서 [그룹 이름.BODY 이름] 형식으로 사용됩니다.

* 프레임 최대 길이는 4096 바이트 입니다. HEAD, BODY, TAIL, BCC 를 모두 합한 길이가 4096 바이트 이내여야 합니다.

다음 예는 HEAD, TAIL, BODY 프레임을 모두 등록한 경우입니다. “RX” 그룹에 “EVENT1”, “EVENT2” 라는 이름으로 두 개의 BODY 프레임이 등록되어 있습니다.



그림 6.3-8 프레임 등록 예

이때 COM1 포트로 수신되는 프레임의 실제 모습은 다음과 같습니다.

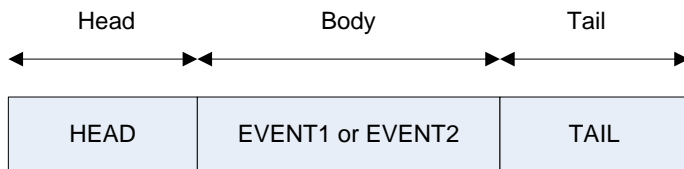


그림 6.3-9 실제 수신 프레임

6.3.3. 세그먼트 추가

통신 프레임은 논리적으로 HEAD, BODY, TAIL 로 구분할 수 있습니다. 그러나 실제 통신에서는 연속적인 데이터의 흐름일 뿐입니다. 마찬가지로, 프레임을 용이하게 다루기 위해서 프레임을 구성하는 HEAD, BODY, TAIL 의 각 부분을 논리적으로 더 여러 개로 구분 지을 수 있습니다. ETOS 에선 이를 세그먼트라고 합니다. 따라서 하나의 완전한 통신 프레임은 HEAD, BODY, TAIL 로 이루어져 있고, HEAD, BODY, TAIL 은 각각 한 개 또는 그 이상의 세그먼트로 이루어져 있습니다. 동일한 프레임을 나타내는 방법은 사용자에게 따라 여러 가지 일 수 있습니다.

{프레임 편집} 창에 마우스를 위치한 상태에서 오른쪽 버튼을 누르고 [세그먼트 추가] 메뉴를 선택합니다.

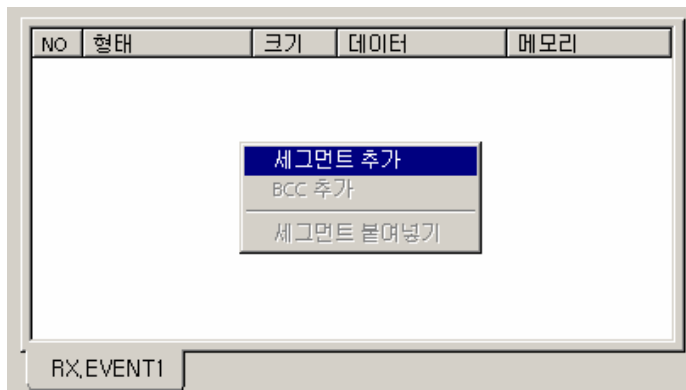


그림 6.3-10 세그먼트 추가

6.3.3.1. 세그먼트 종류

다양한 방법으로 프레임을 정의할 수 있도록, ETOS 는 여러가지의 세그먼트 정의 방식을 지원합니다. 세그먼트는 크게 상수 세그먼트와 변수 세그먼트로 나뉘어 집니다. 상수 세그먼트는 그 값이 고정된 경우로 프레임 정의시에 이미 값을 알고 있는 경우에 사용합니다. 주로 HEAD 나 TAIL 에 고정값으로 정의된 값을 지정할 때 사용합니다. 반면에 변수 세그먼트는 그 값이 고정되지 않고, 프레임 송, 수신 시마다 변할 수 있는 데이터를 지정할 때 사용합니다. 주로, 송,수신 데이터를 변수 세그먼트로 지정합니다.

표 6.3-3 세그먼트 종류

형 태		적용 위치	최대 입력크기
상수	수치 상수	HEAD, BODY, TAIL	30 바이트
	문자열 상수	HEAD, BODY, TAIL	영문 30 자, 한글 15 자
변수	고정 크기 변수	BODY	4096 바이트
	가변 크기 변수	BODY	N/A
BCC ^(*77)		TAIL	N/A

변수 세그먼트는 BODY 에만 지정할 수 있습니다. 반면에 상수 세그먼트는 프레임의 종류에 관계없이 HEAD, BODY, TAIL 어느 곳에도 지정할 수 있습니다.

(*77) Block Check Character

▶ 수치 상수

프레임 중 크기와 내용이 고정되어 있고, 문자열로 표현되지 않는 부분은 수치 상수 세그먼트로 정의합니다. HEAD, TAIL, BODY 어느 곳에도 설정 가능합니다.

수신 프레임의 수치 상수 세그먼트

<메모리 지정> 옵션이 있으나 수신 프레임의 수치 상수는 이미 알고 있는 부분이므로 저장하지 않는 것이 일반적입니다. 입력 데이터는 16 진수로 취급합니다. 따라서 다음 예는 16 진수 1234₁₆를 나타냅니다.

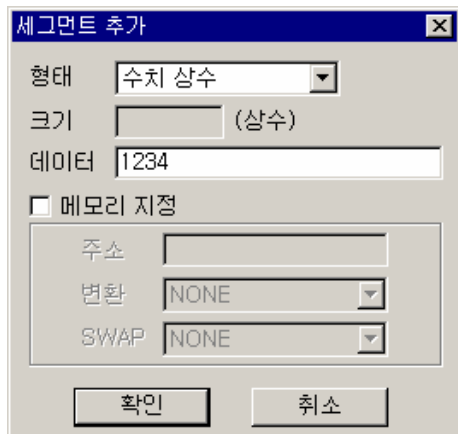


그림 6.3-11 수신 프레임의 수치 상수 세그먼트 추가

송신 프레임의 수치 상수 세그먼트

<메모리 지정> 옵션이 없습니다. 송신의 경우 이미 알고 있는 값이므로 메모리에서 불러올 이유가 없기 때문입니다. 역시 입력 데이터는 16 진수로 취급합니다.

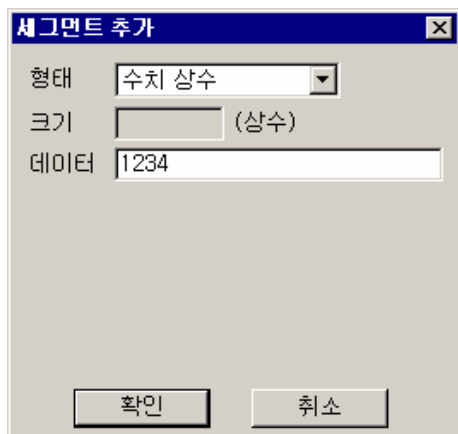


그림 6.3-12 송신 프레임의 수치 상수 세그먼트 추가

* 수치 상수 세그먼트의 최대 크기는 30 바이트 입니다. 30 바이트 이상의 데이터는 여러 개의 세그먼트로 등록합니다.

▶ 문자열 상수

프레임 중 크기와 내용이 고정되어 있는 문자열은 문자열 상수 세그먼트로 정의합니다. HEAD, TAIL, BODY 어느 곳에도 설정 가능합니다.

수신 프레임의 문자열 상수 세그먼트

입력 데이터는 문자열로 취급합니다. 따라서 다음 예는 문자열 “1234”를 나타냅니다. (16 진수로 31H 32H 33H 34H 의 4 바이트 데이터입니다.) 수치 상수와 마찬가지로 수신 프레임에서는 보통 메모리 지정을 하지 않습니다.

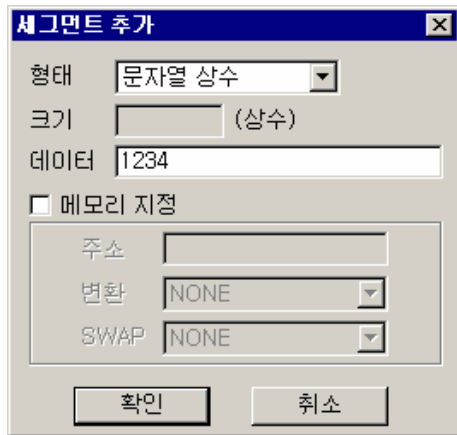


그림 6.3-13 수신 프레임의 문자열 상수 세그먼트 추가

송신 프레임의 문자열 상수 세그먼트

송신 프레임은 <메모리 지정> 옵션이 없습니다. 다음 예는 문자열 “ABCD”를 나타냅니다.

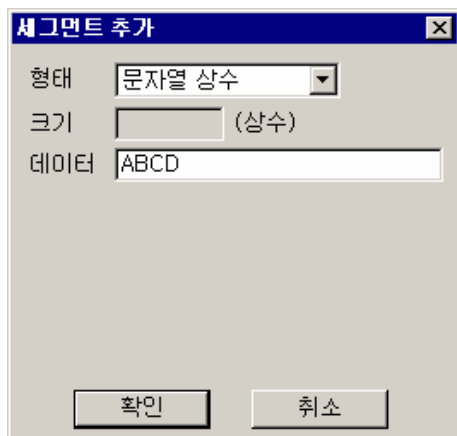


그림 6.3-14 송신 프레임의 문자열 상수 세그먼트 추가

- * 문자열 상수 세그먼트의 최대 크기는 30 자 입니다. 30 자 이상의 데이터는 두 개 이상의 세그먼트로 분리해서 등록합니다.

▶ 고정 크기 변수

프레임 중 크기는 고정되어 있으나 데이터의 내용이 송,수신 시 변경되는 부분은 고정 크기 변수 세그먼트로 정의합니다. BODY 에만 설정 가능합니다.

수신 프레임의 고정 크기 변수 세그먼트

수신 프레임의 경우 일반적으로 메모리에 저장합니다. 그러나 필요 없는 부분은 저장을 하지 않는 경우도 있습니다.

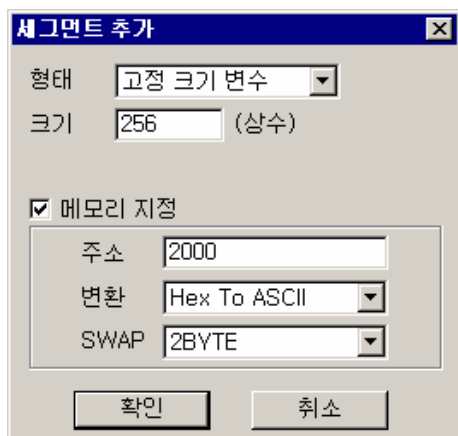


그림 6.3-15 수신 프레임의 고정 크기 변수 세그먼트 추가

송신 프레임의 고정 크기 변수 세그먼트

변수이므로 송신 전에 메모리에서 반드시 읽어와야 합니다.(<메모리 지정> 옵션이 항상 체크되어 있습니다.)

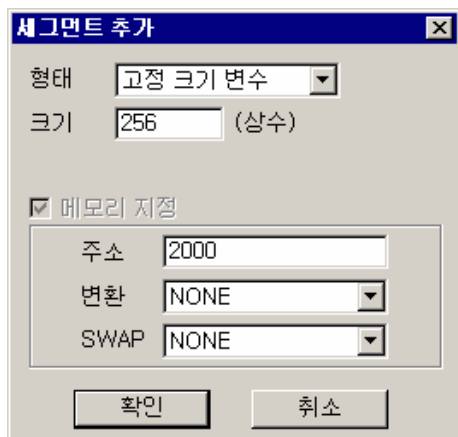


그림 6.3-16 송신 프레임의 고정 크기 변수 세그먼트 추가

* 고정 크기 변수 세그먼트의 최대 크기는 4096 바이트 입니다. 그러나 이 경우에도 프레임의 전체 크기는 4096 바이트를 초과할 수 없습니다.

▶ 가변 크기 변수

프레임 중 크기와 내용이 모두 변경되는 부분은 가변 크기 변수 세그먼트로 정의합니다. BODY에만 설정 가능합니다. 또, 다른 세그먼트와 달리 가변 크기 변수 세그먼트 이후에는 세그먼트를 추가로 설정할 수 없습니다. 가변 크기 세그먼트를 사용하는 경우에는 반드시, HEAD, TAIL 중 하나가 반드시 있어야 합니다. HEAD, TAIL 없이 가변 크기 변수만으로 프레임을 등록하면, 통신 상황에 따라 프레임 수신시에 오류가 발생할 가능성이 있습니다. 신뢰성 있는 통신을 위해서는 반드시 HEAD, TAIL 중 하나 이상을 설정해야 합니다.

수신 프레임의 가변 크기 변수 세그먼트

수신 시에 수신 크기와 데이터를 메모리에 저장합니다. 그러나 고정 크기 변수와 마찬가지로 메모리 저장을 생략할 수 있습니다. (수신은 하지만 데이터를 이용하지 않을 경우)
 메모리 주소 10000 번지에는 가변 크기 변수의 크기(바이트 수)가 2 바이트로 저장됩니다. (크기가 10000 바이트가 아니라 크기가 저장되어 있는 주소가 10000 번지입니다.)
 메모리 주소 10002 번지부터는 실제 데이터가 저장됩니다.

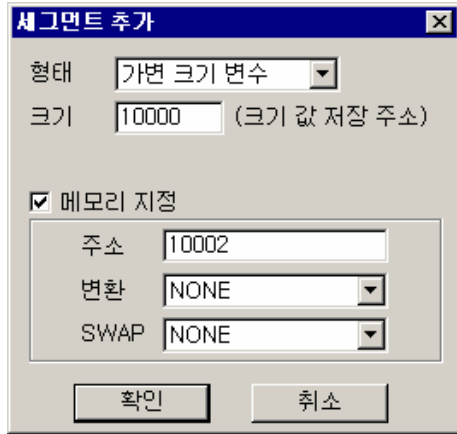


그림 6.3-17 수신 프레임에 가변 크기 변수 세그먼트 추가

송신 프레임의 가변 크기 변수 세그먼트

송신 전에 메모리에서 읽어 옵니다. (메모리 지정 옵션이 항상 체크되어 있습니다.)
 메모리 10000 번지에는 가변 크기 변수의 크기가 송신 전에 기록되어 있어야 합니다.
 메모리 10002 번지부터는 지정된 크기만큼의 데이터가 송신 전에 기록되어 있어야 합니다.

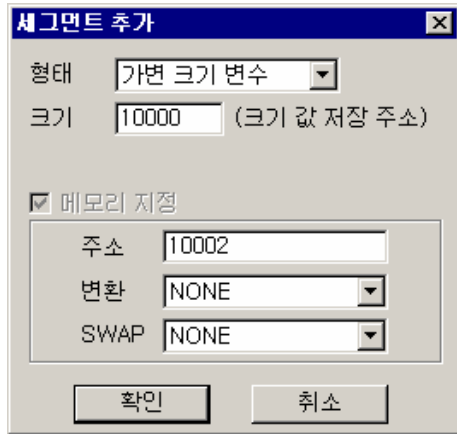


그림 6.3-18 송신 프레임에 가변 크기 변수 세그먼트 추가

- * 가변 크기 세그먼트는 동작 상 최대 크기는 4096 바이트로 제한되어 있으나 프로그램 시 크기를 직접 지정하지 않습니다. 따라서 사용자는 동작 시에 세그먼트의 크기가 4096 바이트를 초과하지 않도록 해야 합니다. 이를 위해 송신 전 또는 수신 후에 크기를 확인하는 코드를 반드시 추가해야 합니다.
- * 송신 시 프로세스에서 세그먼트의 크기를 잘 못 지정하여 4096 바이트를 초과하는 경우 ETOS가 오동작 할 수 있습니다.
- * 가변크기 변수의 크기 값은 2 바이트로 저장됩니다. 앞의 그림 예에서는 10000 ~ 10001 번지에 크기가 저장됩니다.

▶ 메모리 지정

수신 프레임의 특정 세그먼트를 메모리에 저장하고자 하는 경우, 또는 송신 프레임의 특정 세그먼트를 메모리에서 불러오자 하는 경우에 사용합니다. <메모리 지정> 옵션을 선택하면, <주소>, <변환>, <SWAP>의 세가지 옵션을 지정할 수 있습니다.

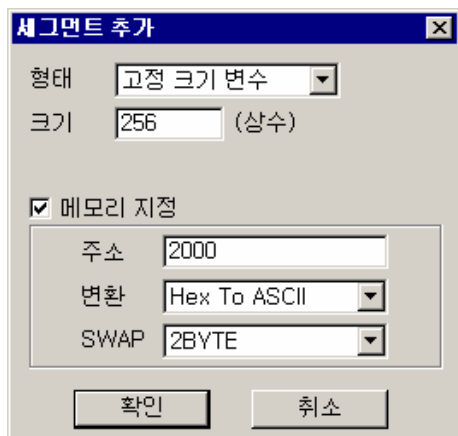


그림 6.3-19 메모리 지정 옵션

<주소>

수신 프레임 : 수신 데이터가 이 주소에 저장됩니다
 송신 프레임 : 송신 데이터를 이 주소에서 읽어옵니다.

<변환>

송, 수신 데이터를 지정하는 방식으로 변환합니다.

수신 프레임 : 수신 데이터를 지정된 형태로 변환하여 주소 위치에 저장합니다.

송신 프레임 : 주소 위치에서 읽어온 데이터를 지정된 형태로 변환하여 송신합니다.

표 6.3-4 변환 방식

변 환	내 용
NONE	변환 없이 송,수신 합니다.
HEX_TO_ASCII	16 진수 데이터를 ASCII 문자열로 변환. 주로 송신 시 사용하고 데이터 크기가 2 배로 늘어납니다.
ASCII_TO_HEX	ASCII 문자열을 16 진수 데이터로 변환. 주로 수신 시 사용하고 데이터 길이가 반으로 줄어듭니다.

<SWAP>

데이터의 상, 하위 바이트 위치를 바꿔서 저장합니다. 통신 장비에 따라 메모리의 상, 하위 바이트 위치가 바뀌어 있는 경우 사용합니다.

표 6.3-5 SWAP 설정에 따른 데이터 자리 이동 예

메모리 번지	0	1	2	3	4	5	6	7	비고
NONE	00	10	20	30	40	50	60	70	변환 없음
2BYTE	10	00	30	20	50	40	70	60	WORD 변환
4BYTE	30	20	10	00	70	60	50	40	DWORD 변환
8BYTE	70	60	50	40	30	20	10	00	LWORD 변환

6.3.4. BCC 세그먼트

BCC는 송, 수신 프레임의 오류를 검사하기 위해 사용되는 세그먼트이며, TAIL에 오직 1개만 설정할 수 있습니다. BCC를 설정한 경우 설정 방식에 따라 송수신 프레임에 대한 BCC 연산을 하고 연산결과가 다를 경우 해당 프레임을 무시하므로 통신의 신뢰성을 높이는 수단이 됩니다. 시리얼 프레임에서만 사용하며, 이더넷 프레임의 경우 사용할 필요가 없습니다.^(*78)

장비와 프로토콜에 따라 BCC 계산 방식이 달라지므로, ETOS-PD는 다양한 BCC 설정 옵션을 제공합니다.

6.3.4.1. BCC 추가

TAIL 프레임 편집 창에서 마우스 오른쪽 버튼을 누르고 [BCC 추가] 메뉴를 선택합니다.

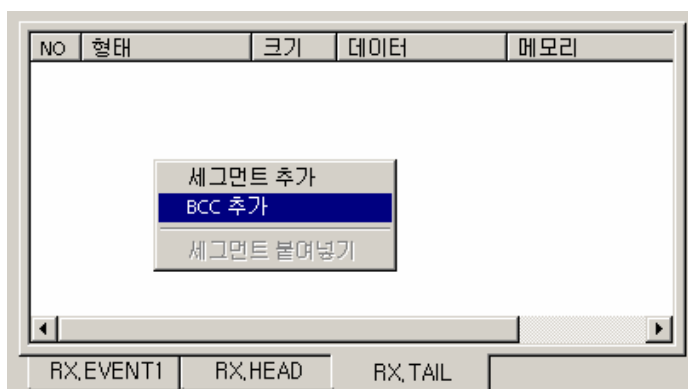


그림 6.3-20 BCC 추가 메뉴

[BCC 추가]메뉴를 선택하면 {BCC 설정} 대화상자가 나타납니다.

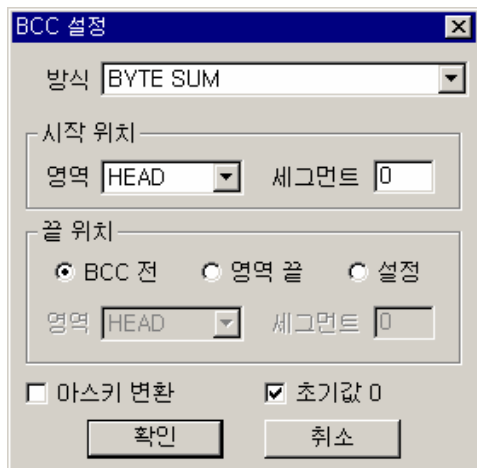


그림 6.3-21 BCC 설정 창

(*78) 이더넷 프레임은 CRC가 TCP 또는 UDP 헤더에 이미 포함되어 있어, 사용자가 별도로 BCC를 사용할 필요가 없습니다.

6.3.4.2. BCC 종류

BCC 방식은 크게 범용통신 오류 검출방식과 전용통신 오류 검출방식으로 나눌 수 있습니다.

표 6.3-6 BCC 종류

분 류	BCC 방식	내용 설명
범용통신 오류검출 방식	BYTE SUM	지정 영역의 데이터를 1 바이트 단위로 더한 결과의 하위 바이트값을 사용
	WORD SUM	지정 영역의 데이터를 1 워드 단위로 더한 결과의 하위 워드값을 사용
	BYTE XOR	지정 영역의 데이터를 1 바이트 단위로 Exclusive OR 결과의 하위 바이트를 사용
	7BIT SUM	BYTE SUM 결과값의 최상위 비트를 제외한 값을 사용
	7BIT XOR	BYTE XOR 결과값의 최상위 비트를 제외한 값을 사용
	7BIT SUM#1	7BIT SUM 결과값이 20 _H 보다 작으면 20 _H 를 더해서 사용
	BYTE SUM 2'S COMP	BYTE SUM 결과를 2의 보수(Complement)를 취함
전용통신 오류검출 방식	LGIS CRC	LG 산전 PLC 에서 사용하는 오류검출 방식
	SIEMENS TI	Texas Instrument 의 SIEMENS-TI Protocol 의 오류검출 방식
	MODBUS LRC	MODBUS 의 ASCII 통신에 사용되는 오류검출 방식
	MODBUS CRC	MODBUS 의 RTU 통신에 사용되는 오류검출 방식
	DLE AB	Allen Bradley 의 DF1 Protocol 의 오류검출 방식
	DLE SIEMENS	Siemens 3964R 통신에 사용되는 오류검출 방식
	DLE XOR	-
	LANDIS CRC	LANDIS 의 시리얼 통신에 사용되는 오류검출 방식
	ROBOSTAR	ROBOSTAR 의 시리얼 통신에 사용되는 오류검출 방식
	PUB XOR	-

* BCC 계산방식에서 전용통신 방식으로 분류된 경우는 BCC 설정범위 및 표시방식을 설정할 필요가 없으며, 범용통신으로 분류된 경우는 BCC 설정범위 및 표시방식을 설정하여야 합니다.

표 6.3-7 BCC 설정 항목

항목		내 용
시작 위치	영역	HEAD/BODY/TAIL 어디부터 BCC 계산을 시작할 지 지정
	세그먼트	HEAD/BODY/TAIL 내에서 BCC 계산을 시작할 세그먼트 위치를 지정. 0은 프레임 처음부터 BCC 계산에 포함
끝 위치	BCC 전	시작 위치부터 BCC 전까지 계산에 포함
	영역 끝	시작 위치부터 지정하는 영역 끝까지 계산에 포함
	설정	시작 위치부터 지정하는 영역 세그먼트 위치까지 계산에 포함
아스키 변환		결과값을 아스키 변환, 크기가 2배 증가합니다.
초기값 0		BCC 계산 초기값을 0으로 지정 가능. 지정하지 않을 경우 FF _H 가 초기값이 됩니다.

▶ 시작위치

프레임 중 BCC 계산의 시작위치를 지정합니다. 영역은 HEAD 나 BODY 등의 프레임 시작영역을 지정하며 세그먼트 번호를 지정하여 HEAD, BODY 등의 프레임 내에서 시작위치를 지정합니다. 세그먼트 내의 숫자는 지정 영역 내에서 BCC 연산을 시작할 세그먼트 영역을 설정합니다.

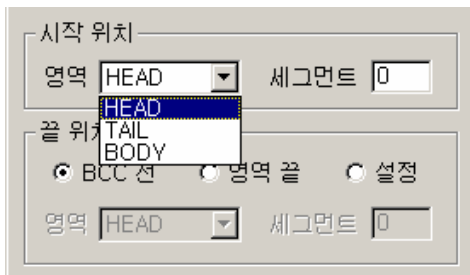


그림 6.3-22 BCC 영역 설정

▶ 끝 위치

프레임 중 BCC 계산의 끝 위치를 설정하는데 다음의 세가지 방법이 있습니다.

<BCC 전> : BCC의 바로 앞 세그먼트를 끝 위치로 지정한 경우입니다. BCC 전을 선택한 경우 영역을 설정할 수 없고 시작영역부터 BCC 설정영역 바로 전까지 계산합니다.

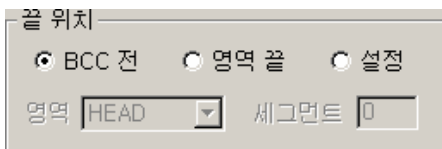


그림 6.3-23 BCC 끝 위치 설정

<영역 끝> : 영역 끝을 지정한 경우 영역을 설정할 수 있으며 BODY/TAIL에 대해 영역을 설정할 수 있습니다. 다음 그림은 BODY를 설정한 경우인데 시작영역부터 BODY의 가장 끝까지 계산합니다. 이 경우 세그먼트 번호를 지정할 수 없습니다.

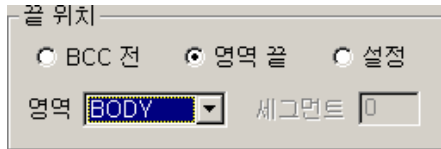


그림 6.3-24 영역 끝을 끝 위치로 설정

<설정> : 영역과 세그먼트를 모두 지정하여 BODY/TAIL 등의 영역과 영역내의 세그먼트 번호를 모두 지정할 수 있습니다.

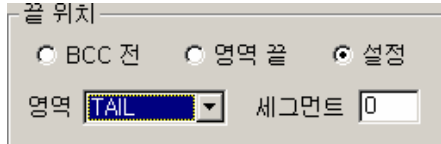


그림 6.3-25 세그먼트 번호로 끝 위치를 설정

NO	형태	크기	데이터	메모리
00	문자열 상수	10	1234567890	
01	가변 크기 변수	M220000		M225000

세그먼트 번호
↖

수신-1,B

그림 6.3-26 세그먼트 번호

▶ 아스키 변환

생성된 BCC 데이터의 최종결과를 아스키 변환할 것인지를 설정합니다. 아스키 변환을 설정한 경우 최종결과 값의 아스키 변환 값을 BCC 결과로 사용하므로 최종 BCC 값은 연산결과와 2 배 길이가 됩니다. 아스키 변환을 지정하지 않은 경우는 BCC 연산결과를 그대로 BCC 값으로 사용합니다.

6.4. 프레임 등록 예제

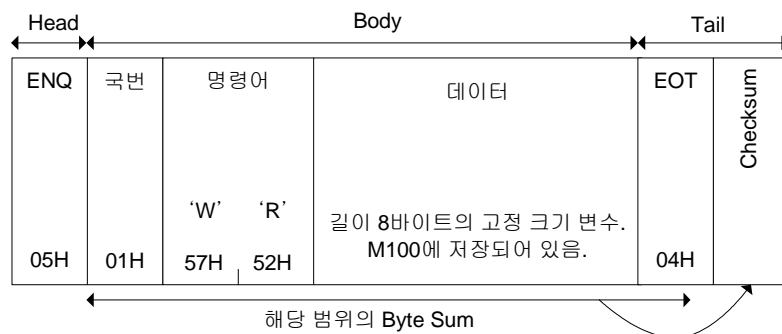
송수신 프레임의 등록방법을 예를 들어 설명합니다.

6.4.1. 예제 프레임

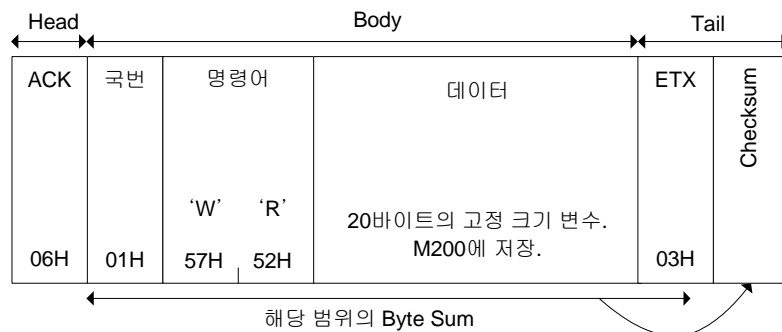
다음의 프레임으로 통신하는 경우의 프레임 등록 과정에 대해 설명합니다.

프레임의 Head 는 상수 부분만을 포함합니다. 따라서, ENQ 부터 명령어까지를 Head 에 포함할 수 있습니다. 그러나, ETOS 가 국번이 다른 여러대의 장비와 멀티드롭으로 묶여있다면(RS422/485), 국번은 변수가 되어야 합니다.^(*79) 이 예제에서는 ENQ 만을 Head 에 포함하기로 합니다.

송신 프레임 (ETOS → 상대기기)



수신 프레임 (상대기기 → ETOS)



(*79) 국번을 상수로 해서 헤더(Head)에 포함하면, 같은 형식의 프레임을 각 국에 대해 모두 정의해야 합니다.

6.4.2. 송신 프레임 등록

송신 프레임(ETOS → 상대기기)을 먼저 등록하겠습니다.

6.4.2.1. 그룹등록

다음 그림과 같이 {프레임 등록} 창의 <시리얼 프레임> 위치에 마우스를 위치하고 우측버튼을 누르고 [그룹 추가] 메뉴를 선택합니다. ETOS-200/1000의 경우에는 카드 추가 후에 <카드 #N>에 마우스를 위치하고 우측버튼을 누릅니다.

{그룹 편집} 대화상자에서 <그룹명>과 <포트번호>, <프레임 종류>를 다음 그림과 같이 설정합니다.

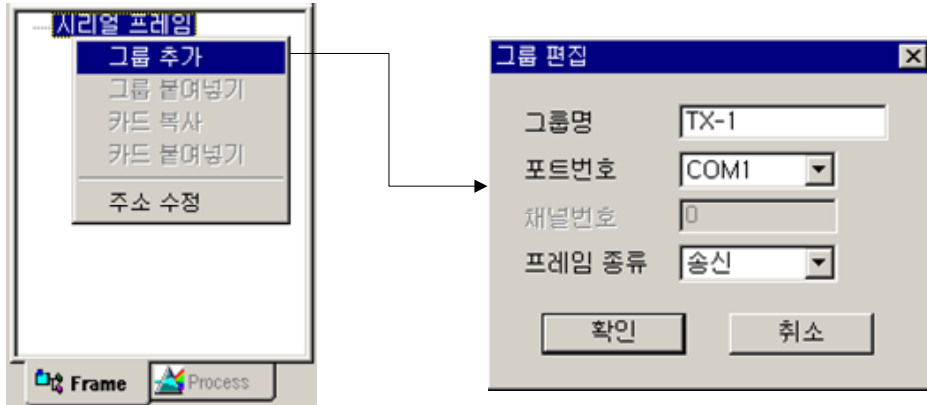


그림 6.4-1 송신 그룹 추가 및 편집

6.4.2.2. HEAD 프레임 추가

다음 그림과 같은 {프레임 등록} 창의 송신그룹에 마우스를 놓고 우측버튼을 눌러 [프레임 추가] 메뉴를 선택합니다. {프레임 편집} 대화상자에서 <프레임 종류>를 HEAD로 선택하고 [확인]을 누릅니다.

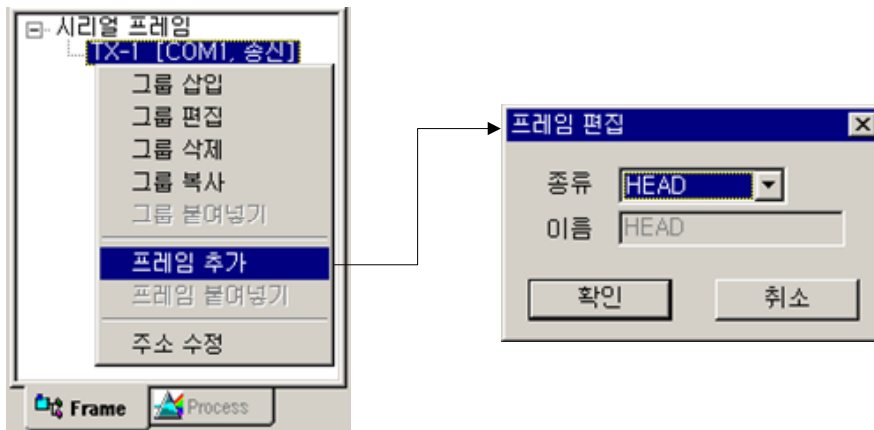


그림 6.4-2 송신 프레임 HEAD 입력

▶ 세그먼트 추가

프레임 등록이 끝나면 우측 {프레임 편집} 창의 HEAD 프레임에 마우스를 위치하고, 우측 버튼을 눌러 [세그먼트 추가]메뉴를 선택합니다. {세그먼트 추가} 대화상자에 다음 그림과 같이 입력합니다.

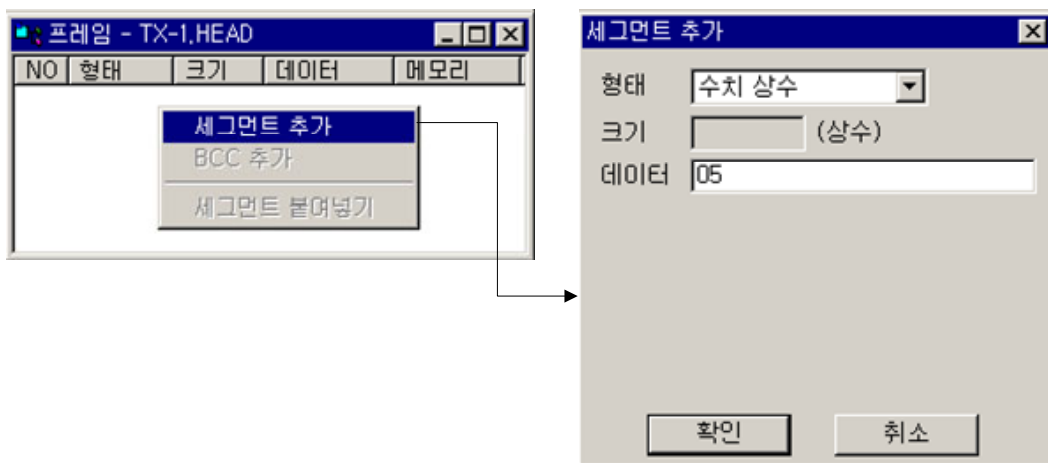


그림 6.4-3 송신 세그먼트 추가

다음 그림은 등록된 HEAD 프레임입니다. 이제 HEAD 프레임 편집이 모두 끝났습니다.

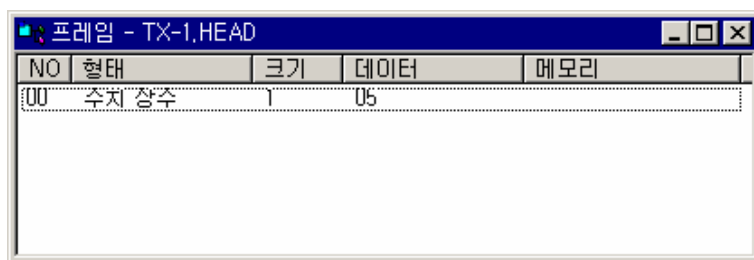


그림 6.4-4 송신 프레임 HEAD

6.4.2.3. TAIL 프레임 추가

HEAD 프레임과 동일하게 {프레임 등록} 창에서 TAIL 프레임을 등록합니다.

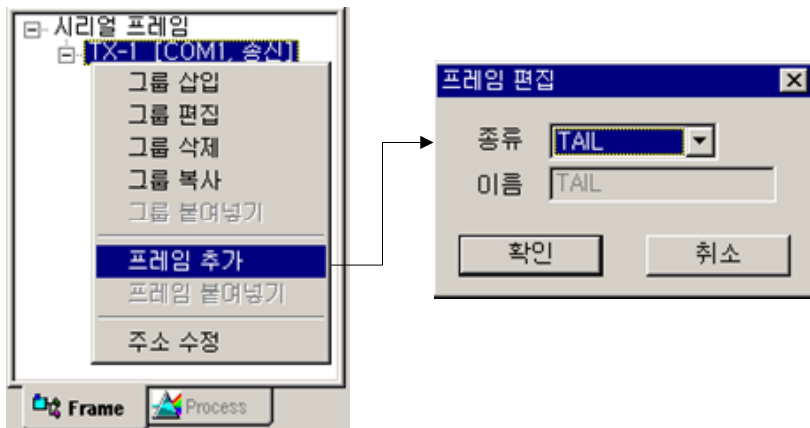


그림 6.4-5 송신 TAIL 프레임 등록

{프레임 편집} 창에서 TAIL 프레임을 열고, 우측 버튼을 눌러 [세그먼트 추가] 메뉴를 선택합니다. TAIL 프레임의 세그먼트를 다음 그림과 같이 입력합니다. 데이터에는 EOT의 ASCII CODE 값인 "04"를 입력합니다.

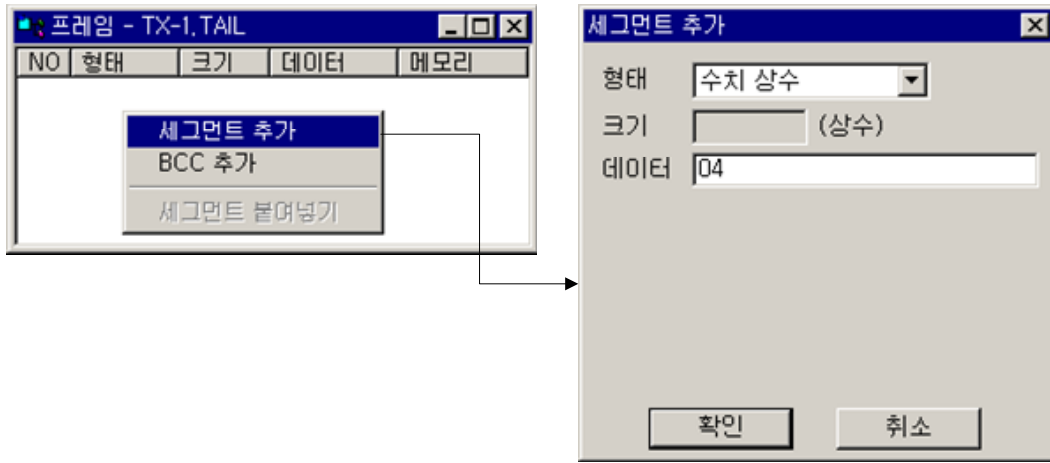


그림 6.4-6 송신 TAIL 세그먼트 추가

6.4.2.4. BCC 세그먼트 추가

TAIL의 마지막은 BCC로 끝납니다. BCC는 일반 세그먼트와 달리 TAIL에만 올 수 있습니다. 예제에서는 BODY 부분부터 BCC 전까지 바이트 단위로 더한 값을 사용합니다. 다음 그림과 같이 TAIL 프레임 편집 창에서 마우스의 우측버튼을 누르고, [BCC 추가]를 누릅니다. 시작 위치는 BODY 처음부터이므로 BODY의 세그먼트 0로 설정합니다. 끝나는 위치는 BCC 전까지 입니다.

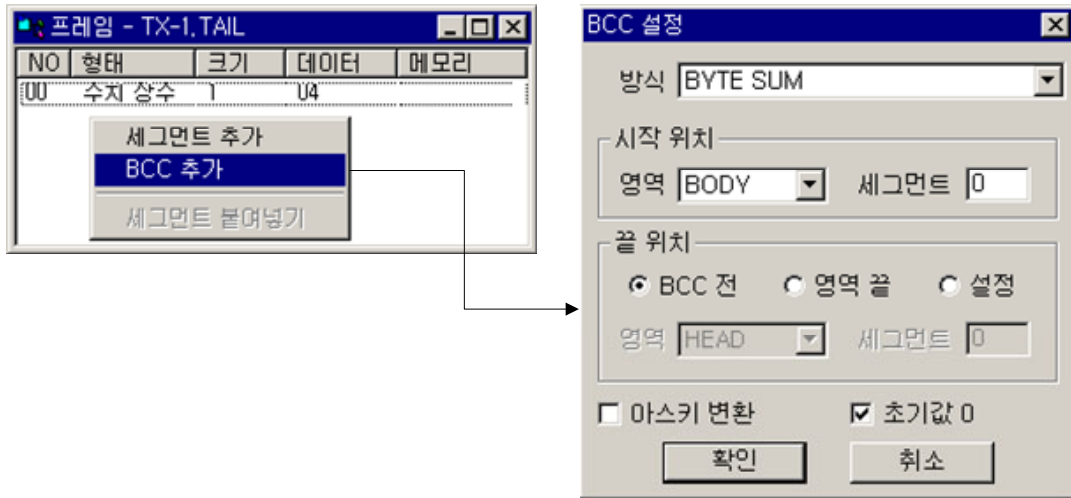


그림 6.4-7 송신 프레임 BCC 설정

다음 그림은 테일 프레임의 수치 상수 세그먼트와 BCC 세그먼트를 모두 설정한 화면입니다.

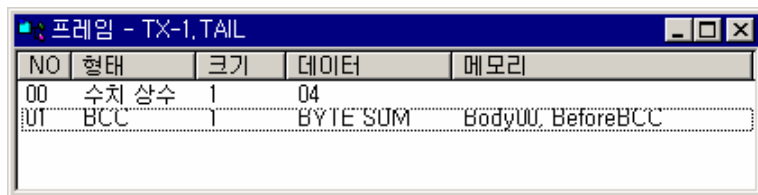


그림 6.4-8 송신 프레임 TAIL

6.4.2.5. BODY 프레임 추가

예제의 송신 프레임은 고정된 크기의 데이터를 송신합니다. 국번은 변수로 등록해서 프로그램내에서 변경이 가능하도록 하고, 명령어는 문자열 상수로, 데이터는 고정 크기 변수로 할당합니다.

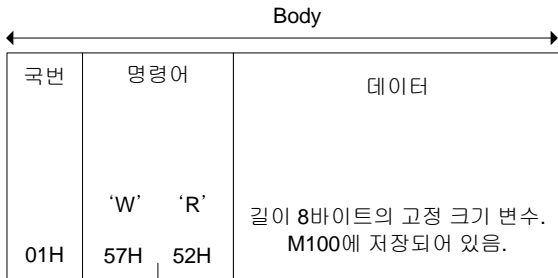


그림 6.4-9 송신 프레임 BODY

BODY 프레임을 추가합니다. 이름은 그룹 안에서 유일해야 합니다.

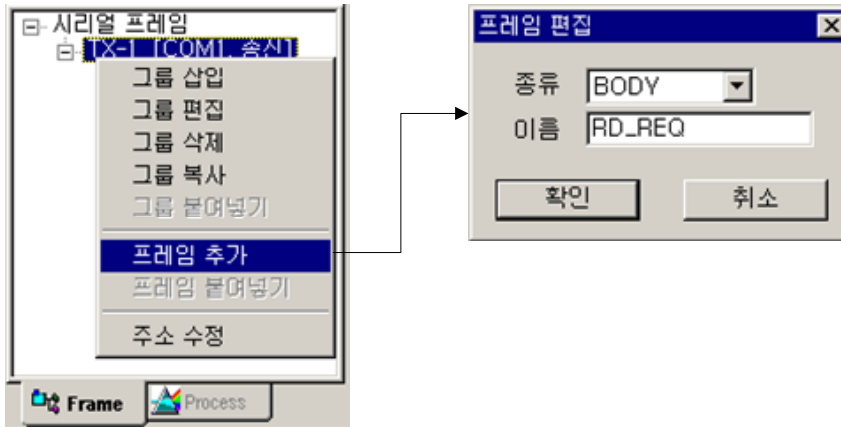


그림 6.4-10 송신 BODY 프레임 등록

BODY 프레임의 이름은 프로세스 아이템에서 TX-1.RD_REQ 와 같은 형식으로 참조할 수 있습니다. 국번, 명령어, 데이터를 세개의 세그먼트로 하여 등록합니다.

▶ 국번

고정 크기 변수 세그먼트로 등록합니다. 크기는 1 바이트고, 실제 데이터는 메모리 98 번지를 참조합니다.

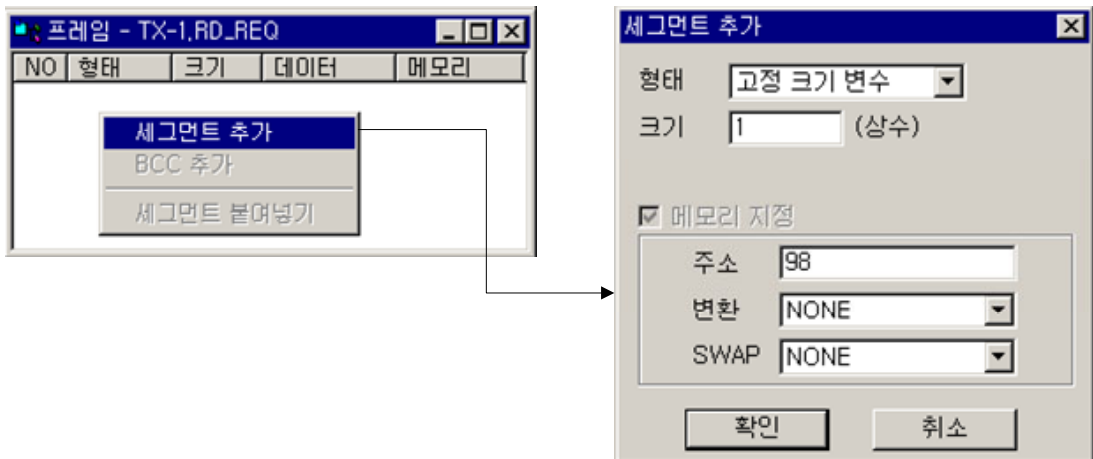


그림 6.4-11 세그먼트 추가 (국번, BODY)

▶ 명령어

문자열 상수 세그먼트로 등록합니다. 입력 데이터는 문자열로 취급됩니다.

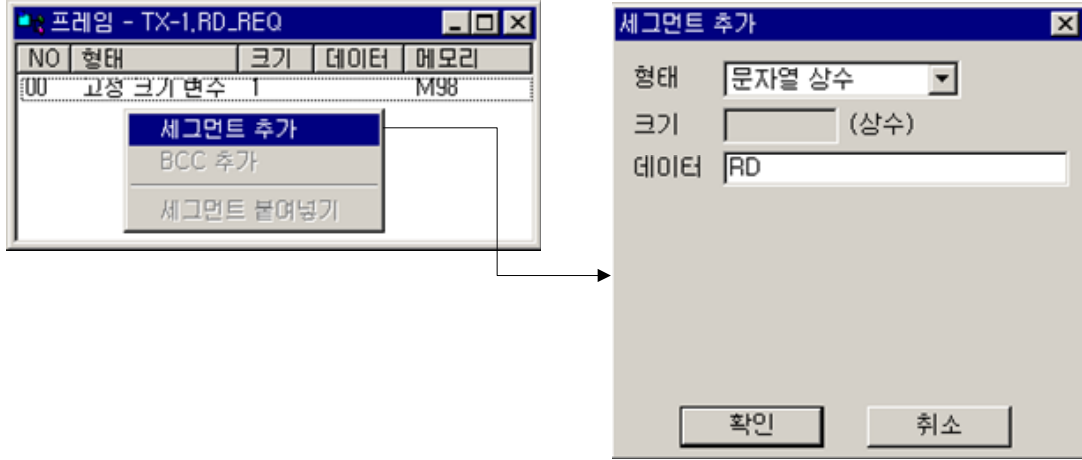


그림 6.4-12 세그먼트 추가 (명령어, BODY)

▶ 데이터

8 바이트의 고정 크기 변수 세그먼트로 등록합니다. 실제 데이터는 메모리 100 번지부터 8 바이트를 참조합니다.

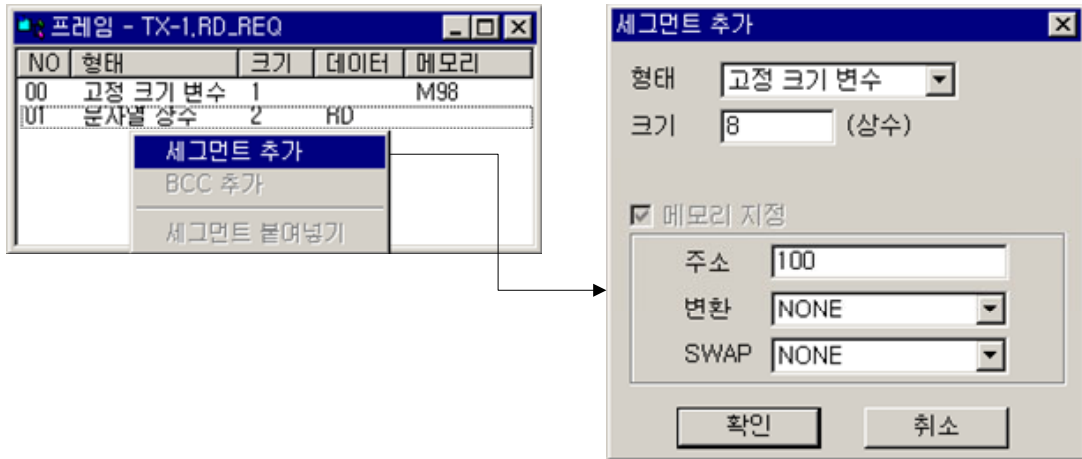


그림 6.4-13 세그먼트 추가 (데이터, BODY)

다음은 BODY 프레임 편집을 마친 상태입니다. 송신 전에 국번 자리인 98 번지와 데이터 자리인 100 번지부터 8 바이트를 원하는 값으로 설정해야 합니다. 특정값으로 설정하기 위해서는 프로세스에서 아이템을 사용합니다. (MEM_SET, ARITHMETIC 등 사용) 프로세스에 대한 설명은 다음장 [7. 프로세스 및 아이템](#) 을 참조하시기 바랍니다.

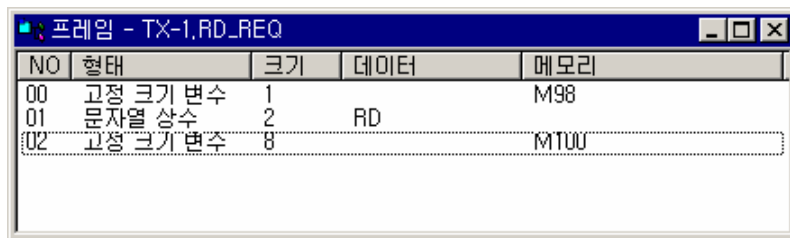


그림 6.4-14 송신 프레임 BODY

6.4.3. 수신 프레임 등록

수신 프레임은 ACK 만을 HEAD 로 하고, BODY 와 TAIL 은 송신 프레임과 동일하게 정의합니다.

6.4.4. 그룹 추가

다음 그림과 같이 {프레임 등록} 창의 <시리얼 프레임> 위치에 마우스를 위치하고 우측버튼을 누르고 [그룹 추가] 메뉴를 선택합니다. ETOS-200/1000 의 경우에는 카드 추가 후에 <카드 #N>에 마우스를 위치하고 우측버튼을 누릅니다.

{그룹 편집} 창에서 <그룹명>과 <포트번호>, <프레임 종류>를 다음 그림과 같이 설정합니다.

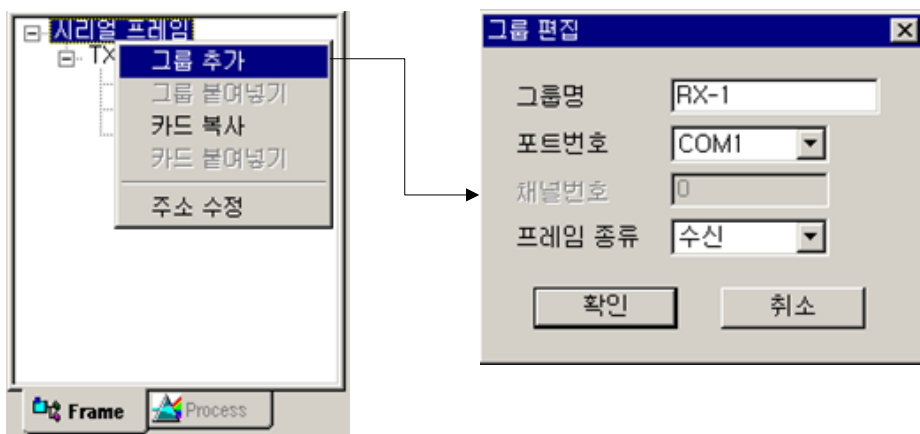


그림 6.4-15 수신 그룹 추가 및 편집

6.4.4.1. HEAD 프레임 추가

다음 그림과 같은 {프레임 등록} 창의 송신그룹에 마우스를 놓고 우측버튼을 눌러 [프레임 추가] 메뉴를 선택합니다. {프레임 편집} 대화상자에서 <프레임 종류>를 HEAD 로 선택하고 [확인]을 누릅니다.

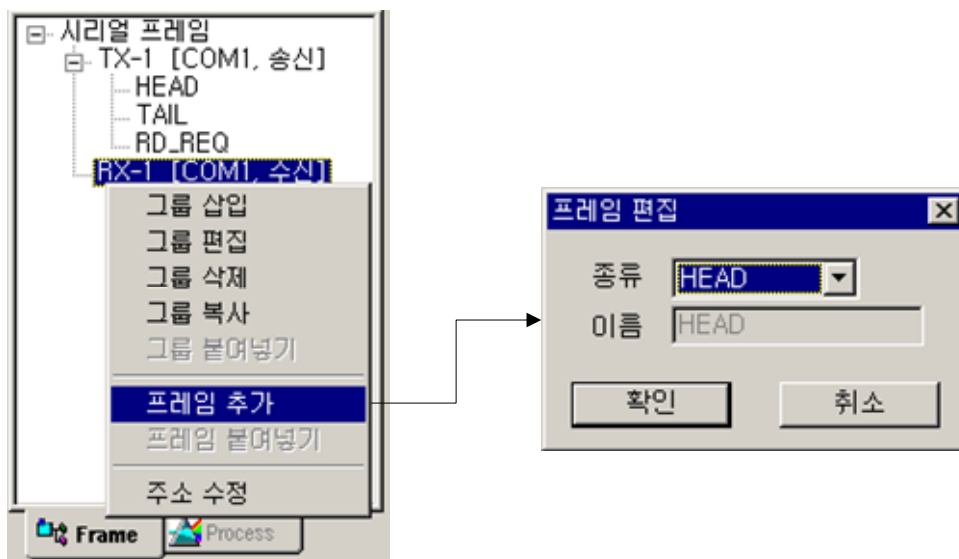


그림 6.4-16 수신 프레임 HEAD 입력

프레임 등록이 끝나면 우측 {프레임 편집} 창의 HEAD 프레임에 마우스를 위치하고, 우측 버튼을 눌러 [세그먼트 추가]메뉴를 선택합니다. {세그먼트 추가} 대화상자에 다음과 같이 입력합니다. 데이터는 ACK의 ASCII Code 값인 "06"을 입력합니다. 메모리에 저장할 필요가 없으므로 [메모리 지정] 옵션은 체크하지 않습니다.

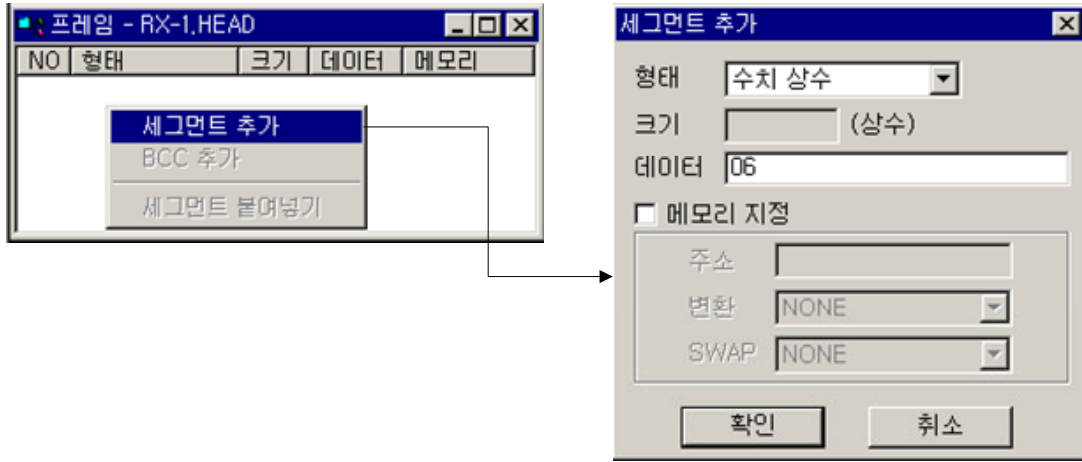


그림 6.4-17 수신 세그먼트 추가

6.4.4.2. TAIL 프레임 추가

HEAD 프레임과 동일하게 {프레임 등록} 창에서 TAIL 프레임을 등록합니다. {프레임 편집} 창에서 TAIL 프레임을 열고, 우측 버튼을 눌러 [세그먼트 추가] 메뉴를 선택합니다. TAIL 프레임의 세그먼트를 다음과 같이 입력합니다. ETX의 ASCII Code 값인 "03"을 입력합니다. HEAD와 마찬가지로 [메모리 지정] 옵션은 체크하지 않습니다.

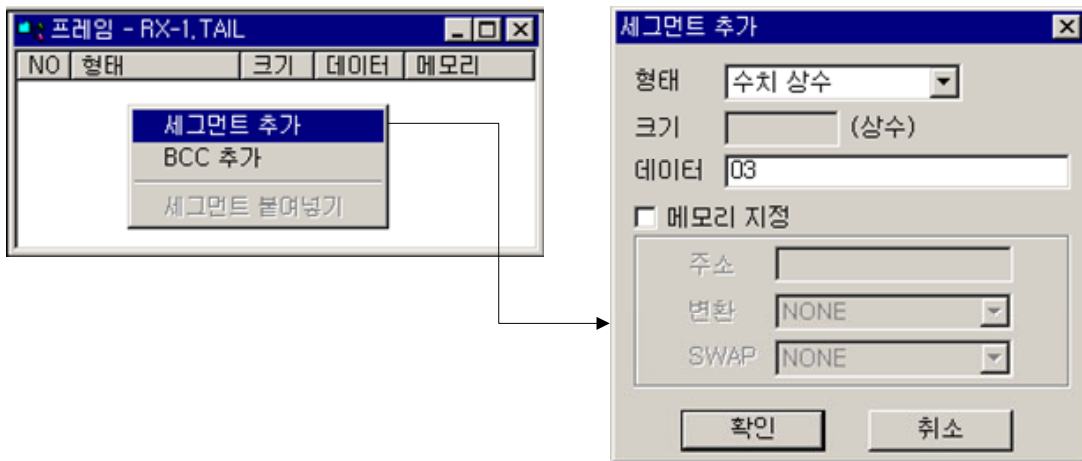


그림 6.4-18 수신 TAIL 세그먼트 추가

6.4.4.3. BCC 세그먼트 추가

BCC 세그먼트도 송신 프레임과 동일하게 설정합니다.

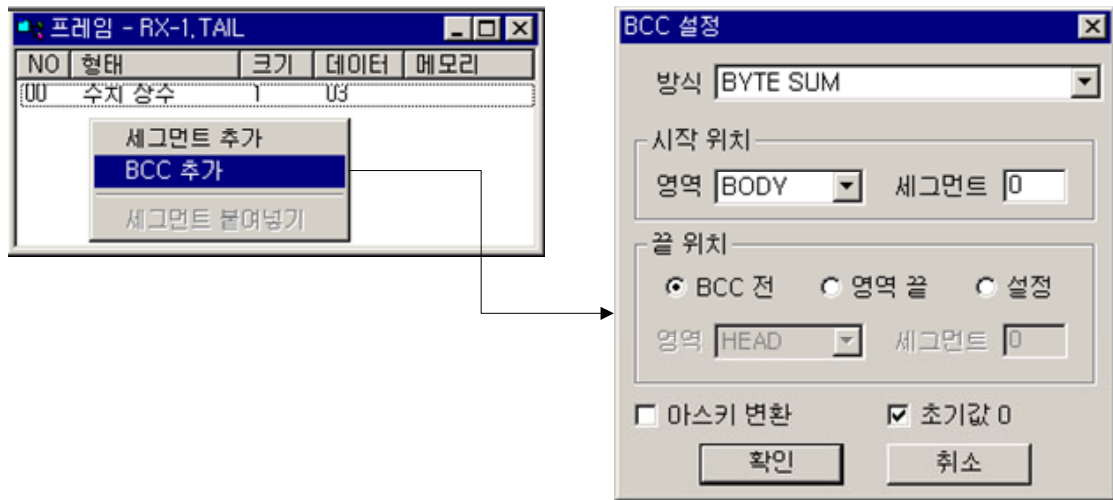


그림 6.4-19 수신 프레임 BCC 설정

다음 그림은 편집을 마친 TAIL 프레임입니다.

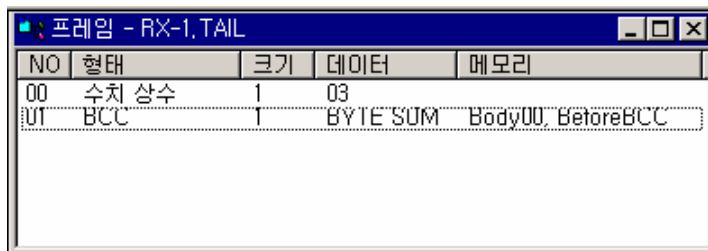


그림 6.4-20 수신 프레임 TAIL

6.4.4.4. BODY 프레임 추가

수신 프레임의 Body 프레임입니다. 송신 프레임과 동일하게 국번, 명령어, 데이터를 별도의 세그먼트로 등록합니다.

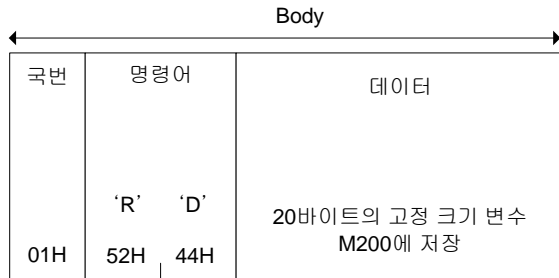


그림 6.4-21 수신 프레임 BODY

▶ 국번

고정 크기 변수로 등록합니다. 크기는 1 바이트며, 198 번지에 저장됩니다. 프로세스에서 해당 메모리를 참조하여 국번을 판단할 수 있습니다.

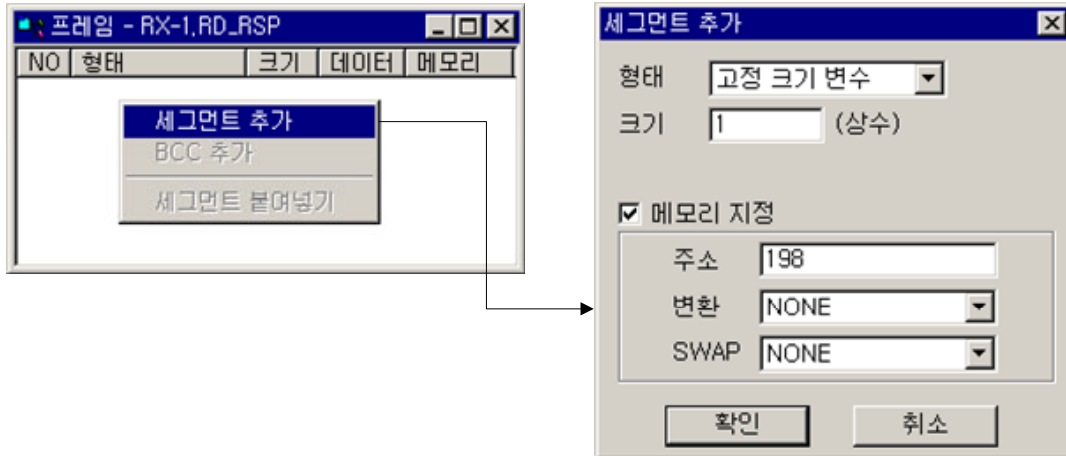


그림 6.4-22 세그먼트 추가 (국번, BODY)

▶ 명령어

문자열 상수로 등록합니다. 데이터에 입력한 RD 는 문자열로 취급됩니다.

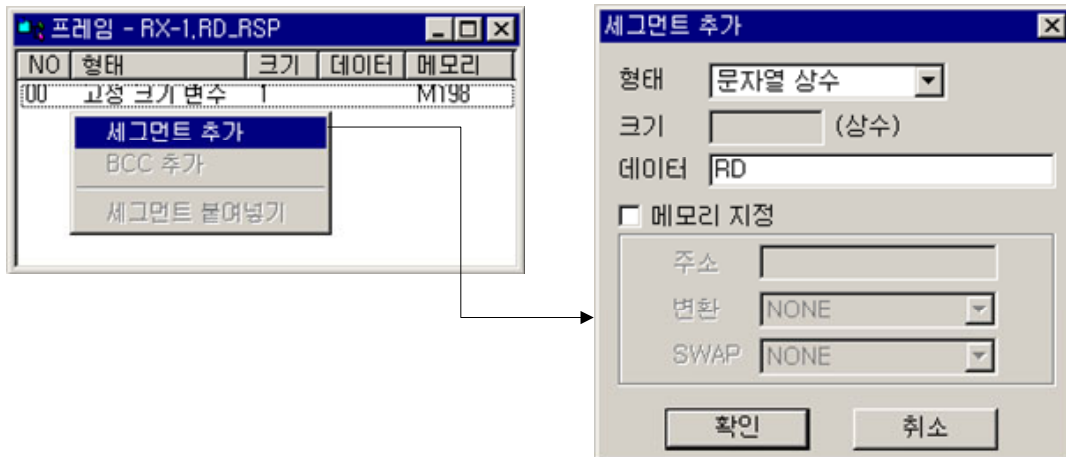


그림 6.4-23 세그먼트 추가 (명령어, BODY)

▶ 데이터

크기 20 바이트의 고정 크기 변수로 등록합니다. 수신 데이터는 200 번지부터 20 바이트에 걸쳐 메모리에 저장됩니다.

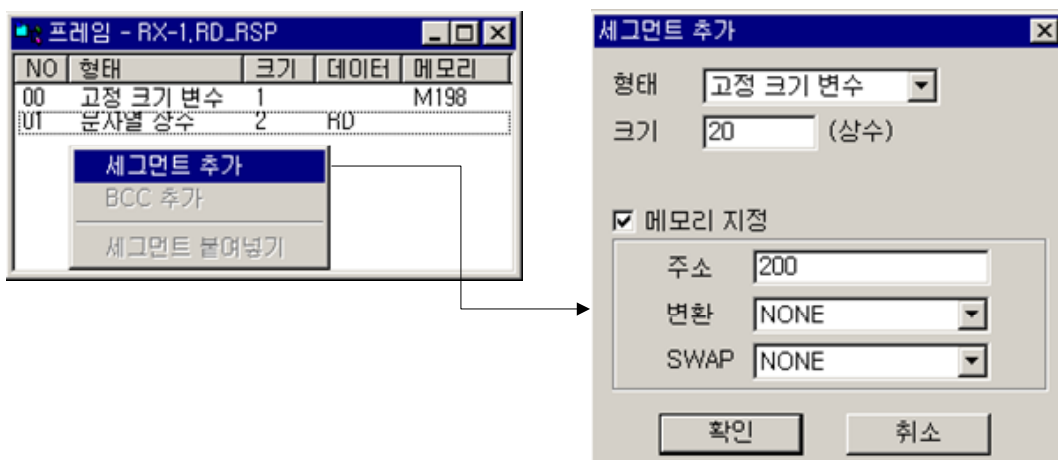


그림 6.4-24 세그먼트 추가 (데이터, BODY)

다음 그림은 편집을 끝낸 BODY 프레임입니다.

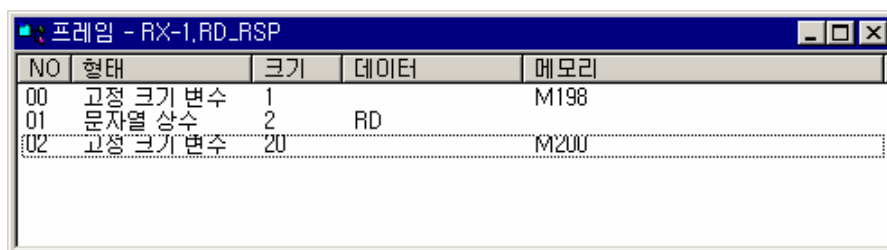


그림 6.4-25 수신 프레임 BODY

이상으로 송,수신 예제 프레임의 등록을 모두 마쳤습니다. 등록된 프레임을 이용한 데이터 송,수신은 프로세스에서 정의합니다. 등록된 프레임을 이용한 송,수신 프로세스 작성은 [7.3 프로세스 등록 예제](#) 에서 다루겠습니다.

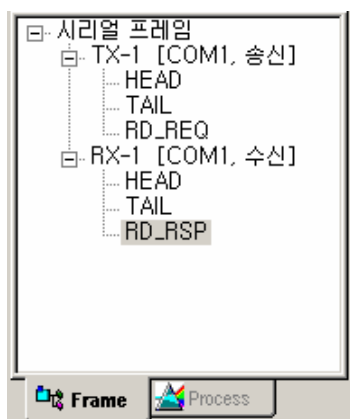


그림 6.4-26 프레임 등록 결과

6.5. SECS 메시지 등록

▶ ETOS-XXX-HSMS 인 경우만 해당되는 내용입니다.

ETOS 가 SECS-I/HSMS Bridge 로 사용되는 경우를 제외하곤, 통신하고자 하는 SECS 메시지를 정의해야 합니다. SECS 메시지 정의는 왼쪽 상단의 [SECS]탭을 눌러 {메시지 등록} 창을 활성화시킨 상태에서 시작합니다.



그림 6.5-1 SECS 메시지 편집 창

6.5.1. 메시지 추가

{SECS 메시지 등록} 창의 빈 공간에 마우스를 위치하고 오른쪽 버튼을 클릭하면, [메시지 추가]와 [메시지 붙여 넣기] 팝업 메뉴가 나타납니다. [메시지 추가]를 클릭합니다.



그림 6.5-2 SECS 메시지 추가

아래처럼 {SECS 메시지} 창에서 필요한 부분을 설정합니다.

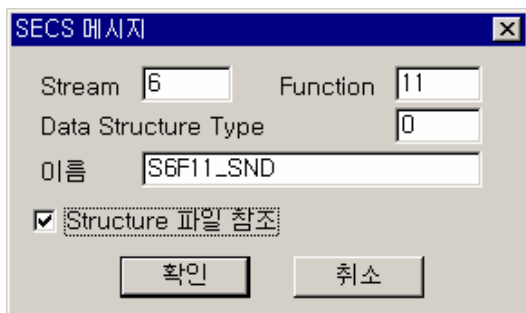


그림 6.5-3 SECS 메시지

표 6.5-1 SECS 메시지 설정 항목

항목	내용
Stream	정의하고자 하는 메시지의 Stream No를 설정합니다.
Function	정의하고자 하는 메시지의 Function No를 설정합니다.
Data Structure Type	이미 정의된 “Structure 파일을 참조”할 경우, 한 메시지에 대해서 여러 개의 정의가 있으면 그 중 어느 것을 사용할지를 결정합니다.
이름	메시지의 이름을 사용자가 구별하기 쉽게 설정합니다.
Structure 파일 참조	표준 Structure 파일을 참조 할지를 결정합니다. 표준 Structure와 동일하거나, 유사하면 표준 Structure 파일을 참조해서 값만 바꿔주는 것이 편리합니다. 많이 다르다면 처음부터 정의 할 수도 있습니다.

표준 Structure 파일을 참조하는 경우, 그림 6.5-4 와 같은 기본적인 메시지 구조가 나옵니다. 값을 설정하거나 구조를 변경할 수 있습니다.

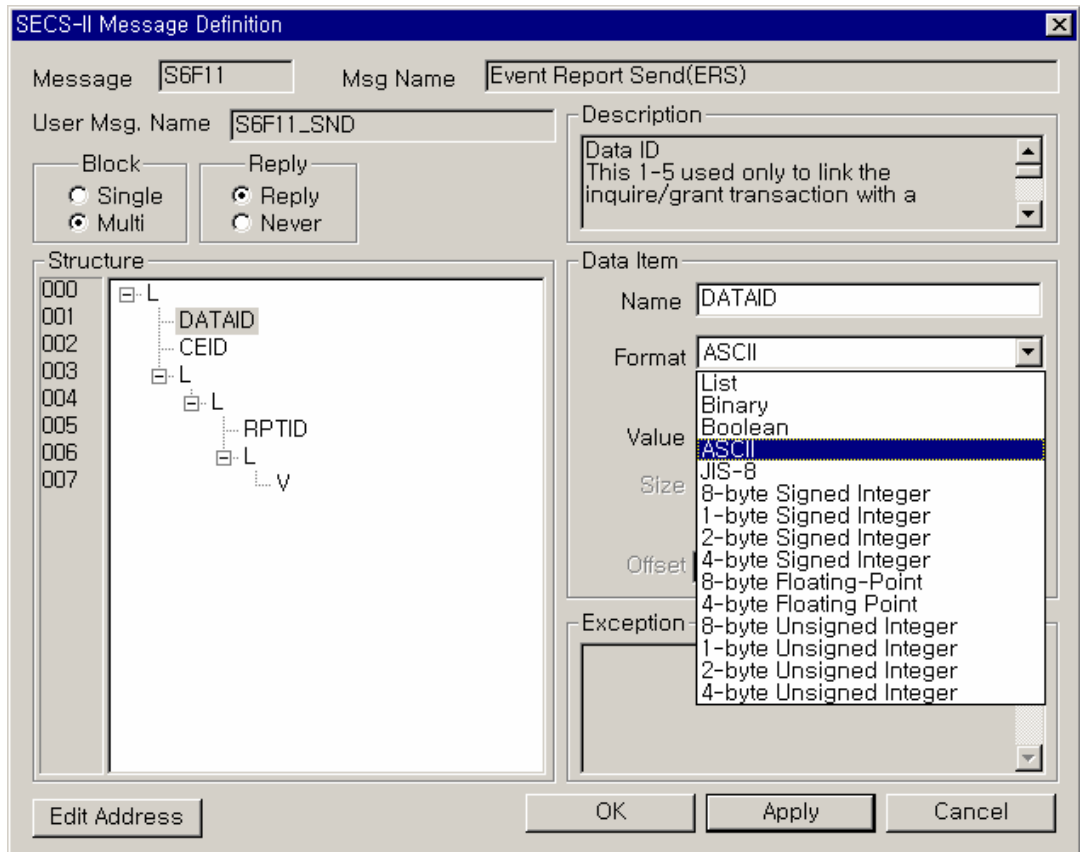


그림 6.5-4 SECS-II Message Definition

표 6.5-2 SECS-II Message Definition 설정 항목

항목	내용
Message	정의하는 메시지의 Stream/Function No를 보여 줍니다.
Msg. Name	정의하는 메시지의 이름을 나타냅니다.
User Msg. Name	사용자가 정의한 이름을 나타냅니다.
Description	메시지에 대한 설명이 나옵니다.
Block	Multi Block이 가능한지 설정합니다.
Reply	응답(Secondary Msg.)이 있는지를 설정합니다.
Structure	메시지의 구조를 Tree로 보여줍니다. 각 항목을 선택하고 오른쪽 <Data Item>에서 내용을 수정할 수 있습니다. 구조의 편집은 아래에서 다시 설명합니다.
Data Item/Name	Data Item의 이름입니다.
Data Item/Format	List/Binary/Boolean/ASCII/JIS-8/Signed Integer/Unsigned Integer/Float 등을 선택할 수 있습니다.
Data Item/Address	아래의 <Value>가 주소임을 설정합니다.
Data Item/Value	Data Item의 값을 직접 입력하거나, 값이 저장된 또는 값을 저장할 ETOS의 메모리 주소를 입력합니다.
Data Item/Size	<Address>를 선택한 경우, <Value>에 설정한 메모리의 크기를 지정합니다. 예를 들면 2Byte Signed Integer처럼 Data Item/Format이 크기를 포함하고 있는 경우는 초기값으로 Format의 크기(여기선 2byte)를 기본값으로 보여줍니다. 배열(Array)를 표현하고자 한다면 이 값을 Item Format의 크기의 정수 배로 설정할 수 있습니다.
Data Item/Variable	Data Item/Format 이 ASCII인 경우 가변 크기의 문자열을 송수신할 수 있습니다. 가변 크기가 필요한 경우 선택합니다.
Data Item/Address of Size	Data Item/Format 이 ASCII이고, Variable을 선택한 경우, a문자열의 크기를 저장하거나, 지정된 ETOS 메모리 주소를 설정합니다.
Data Item/Offset	동일한 구조의 Item이 반복되는 경우, Offset을 설정해서 구조를 반복하도록 할 수 있습니다.
Edit Address	메시지의 주소를 일괄 수정합니다. 메시지에 연관된 주소를 이동하는 경우 편리하게 사용할 수 있습니다.

자세한 사용 예는 [6.6. SECS 메시지 등록 예제](#)에서 상황 별로 설명하겠습니다.

다음은 메시지의 구조를 편집하기 위한, 팝업 창입니다.

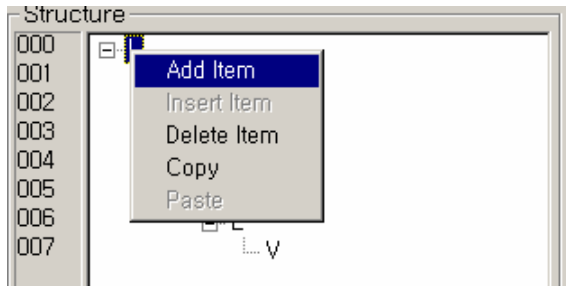


그림 6.5-5 SECS 메시지 편집 팝업 창

표 6.5-3 SECS 메시지 편집

항목	내용
Add Item	Structure의 마지막에 Item을 추가합니다.
Insert Item	Structure의 현재 위치에 Item을 추가합니다.
Delete Item	현재 아이템을 삭제합니다.
Copy	현재 아이템을 복사합니다.
Paste	현재 아이템을 붙여 넣습니다.

[Edit Address]를 클릭하면 아래의 {주소 수정} 창이 나옵니다.

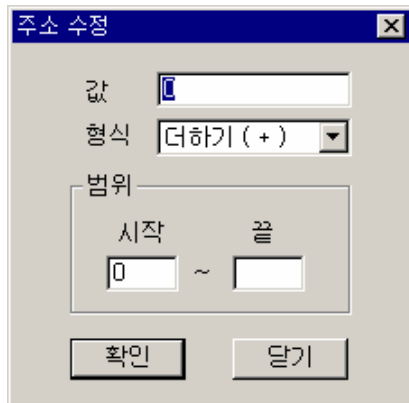


그림 6.5-6 주소 수정

표 6.5-4 주소 수정

항목	내용
값	변경할 값을 지정합니다.
형식	설정: 현재의 값으로 모든 주소 값을 변경합니다. 더하기: 모든 주소 값에 현재 값을 더 합니다. 빼기: 모든 주소 값에서 현재 값을 뺍니다.
범위	주소를 변경할 범위를 설정합니다. <Structure>의 좌측 일렬 번호를 입력합니다.

6.6. SECS 메시지 등록 예제

6.6.1. 헤더만 있는 S1F1

{SECS 메시지} 창에서 다음 그림과 같이 <Structure 파일 참조>를 체크하고, <이름>과 <Stream>, <Function>을 정합니다.

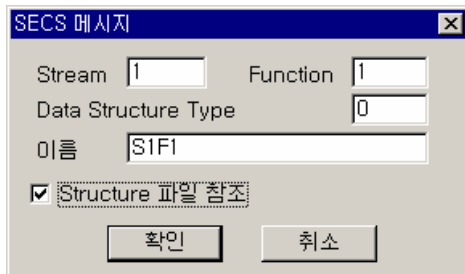


그림 6.6-1 SECS 메시지 추가

[확인]을 누르면 아래처럼 메시지가 만들어 집니다. <Block>과 <Reply>만 설정하면 메시지의 정의가 끝납니다.

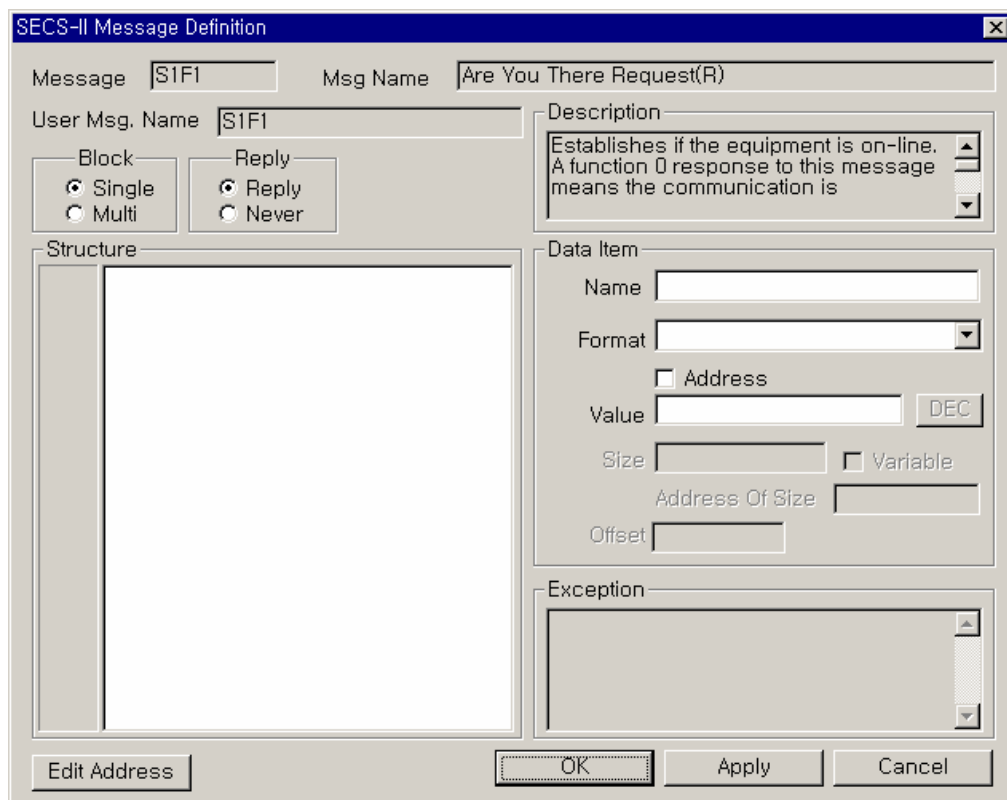


그림 6.6-2 SECS-II Message Definition

메시지가 정의되었습니다.



그림 6.6-3 정의된 SECS 메시지

6.6.2. 고정 값을 갖는 S1F2

{SECS 메시지} 창에서 다음 그림과 같이 <Structure 파일 참조>를 체크하고, <이름>과 <Stream>, <Function>을 정합니다.

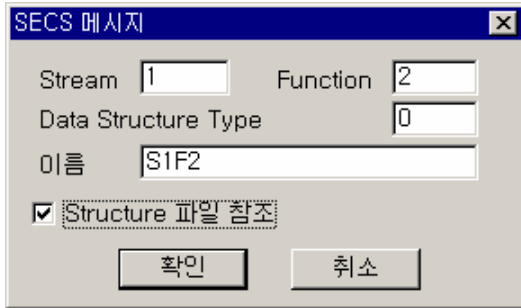


그림 6.6-4 SECS 메시지 추가

[확인]을 누르면 아래처럼 메시지가 만들어 집니다. <Block>과 <Reply>를 확인하고, 아이템을 선택하고 우측의 <Data Item>에 값을 직접 넣어 줍니다. 여기서 <MDLN>아이템을 ASCII 로 "ETOS"라고 설정하였습니다. <SOFTREV>아이템도 비슷하게 설정합니다. ASCII 가 아니라 다른 FORMAT 을 원한다면 <Format>에서 변경이 가능합니다.

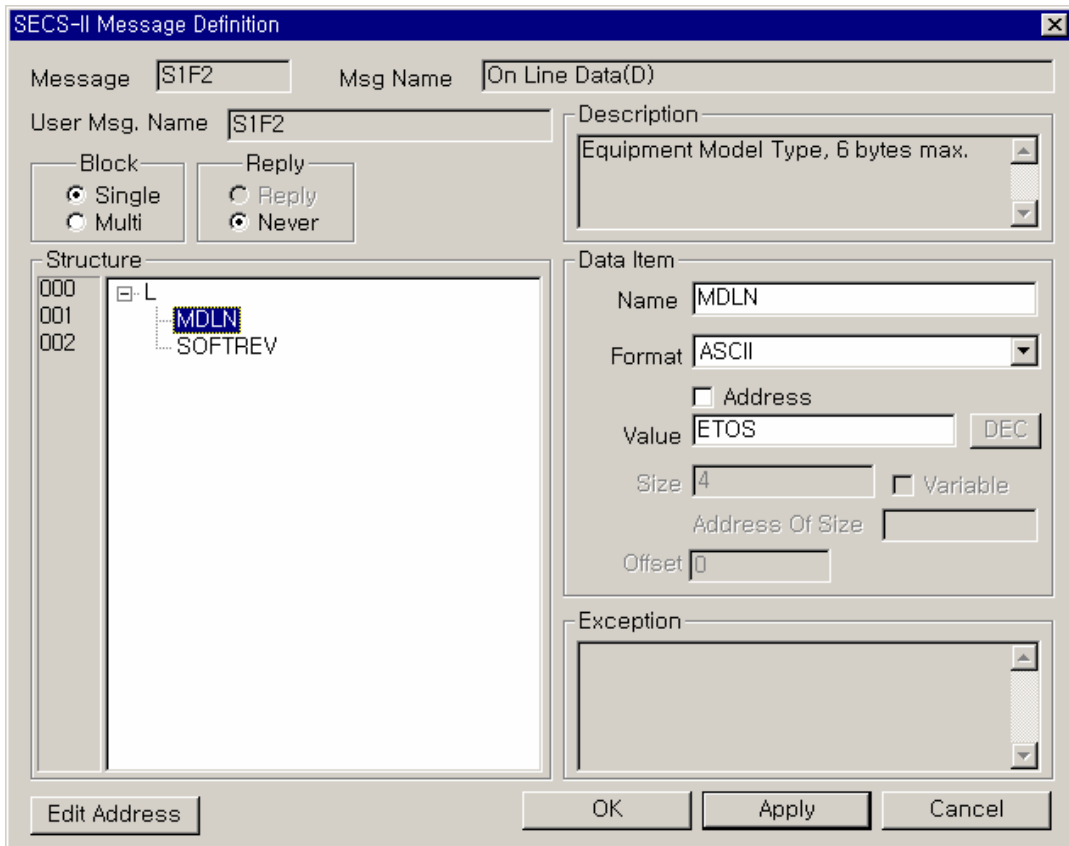


그림 6.6-5 SECS-II Message Definition

6.6.3. 가변 값을 갖는 S1F2

앞의 예에 이어서 <SOFTREV>아이템을 2-byte Signed Integer 로, 송신 시 ETOS 의 메모리 100 번지 값을 보내거나, 수신 시 100 번지에 수신한 <SOFTREV> 값을 저장하도록 해 보겠습니다. 아래처럼 <Structure>에서 <SOFTREV> 아이템을 선택하고, 우측의 <Data Item>에서 <Format>은 2-byte Signed Integer 로, <Address>를 체크하고, <Value>에 100 번지를 지정합니다. <Size>는 2-byte Signed Integer 를 선택하면 자동으로 2 로 변경됩니다.

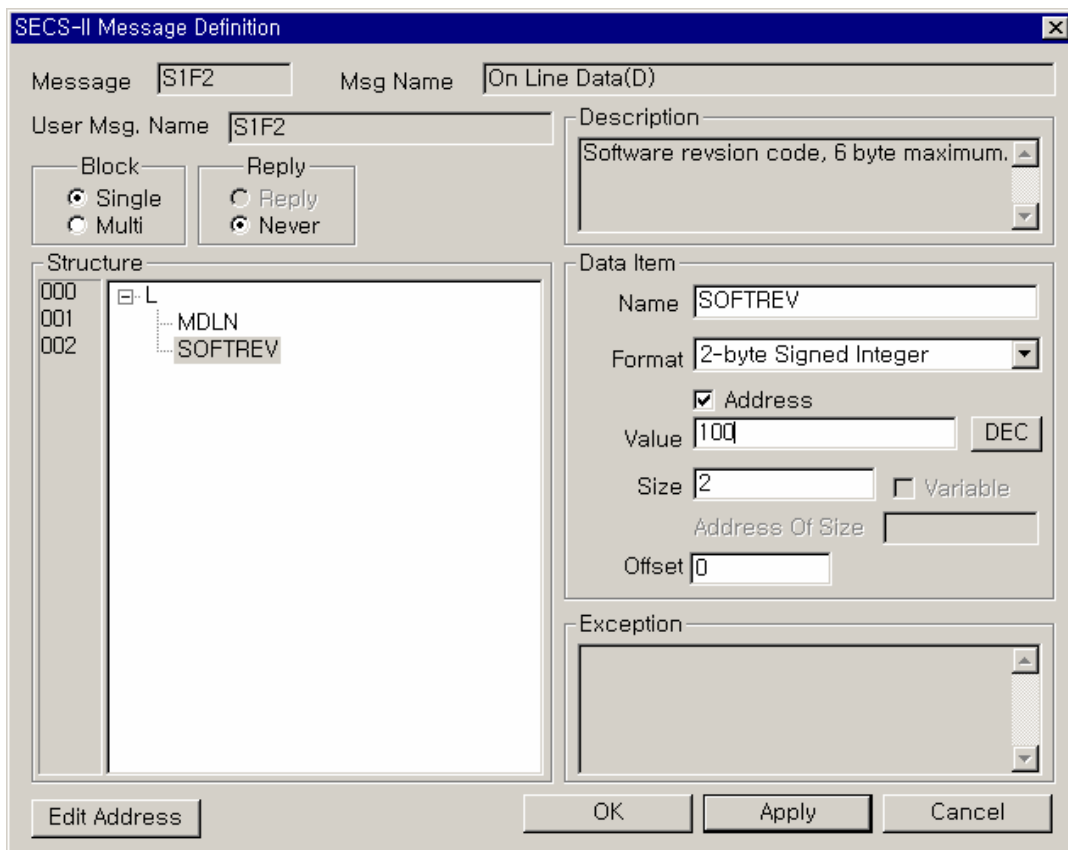


그림 6.6-6 SECS-II Message Definition

6.6.4. 가변 String 을 갖는 S6F11

S6F11 의 <DATAID>아이템을 가변 String 으로, 송신 시 ETOS 의 메모리 16 번지 값을 20 바이트 를 보내거나, 수신 시 16 번지에 수신한 <DATAID> 값 20 바이트를 저장하도록 해 보겠습니다. 아래처럼 <Structure>에서 <DATAID> 아이템을 선택하고, 우측의 <Data Item>에서 <Format>은 ASCII 로, <Address>를 체크하고, <Value>에 16 번지를 지정합니다. <Size>는 20 바이트로 설정 합니다. 이때 <Size>는 최대값을 의미합니다.

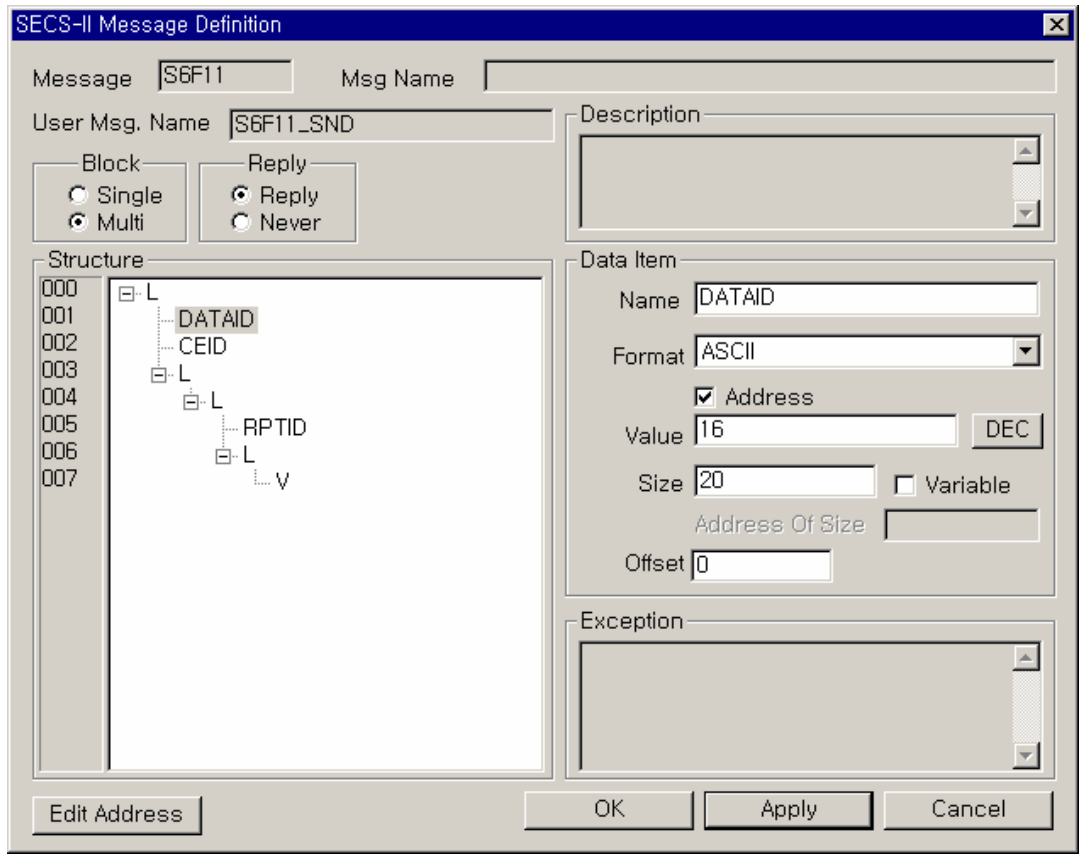


그림 6.6-7 SECS-II Message Definition

- * 송신 시 16 번지의 String 이 20 바이트가 안 된다면 나머지는 Space(20H)으로 채워집니다.
- * 수신 시는 <DATAID>의 크기가 20 바이트가 안되어도 예러 없이 16 번지에 저장하고, 처리 합니다. <DATAID>d 의 크기가 20 바이트가 넘는다면, 20 바이트만 저장합니다.

6.6.5. 크기 지정, 가변 String 을 갖는 S6F11

앞의 예제와 비슷하나 String 의 크기를 지정하겠습니다. S6F11 의 <DATAID>아이템을 가변 String 으로, 송신 시 ETOS 의 메모리 16 번째 값을 최대 20 바이트로 32 번째 크기로 보내거나, 수신 시 16 번째에 수신한 <DATAID> 값 최대 20 바이트를, 실제 길이는 32 번째에 저장하도록 해 보겠습니다. 아래처럼 <Structure>에서 <DATAID> 아이템을 선택하고, 우측의 <Data Item>에서 <Format>은 ASCII 로, <Address>를 체크하고, <Value>에 16 번째를 지정합니다. <Size>는 20 바이트로 설정합니다. 역시 <Size>는 최대값을 의미합니다. <Variable>을 체크하고, <Address of Size>에 32 를 지정합니다.

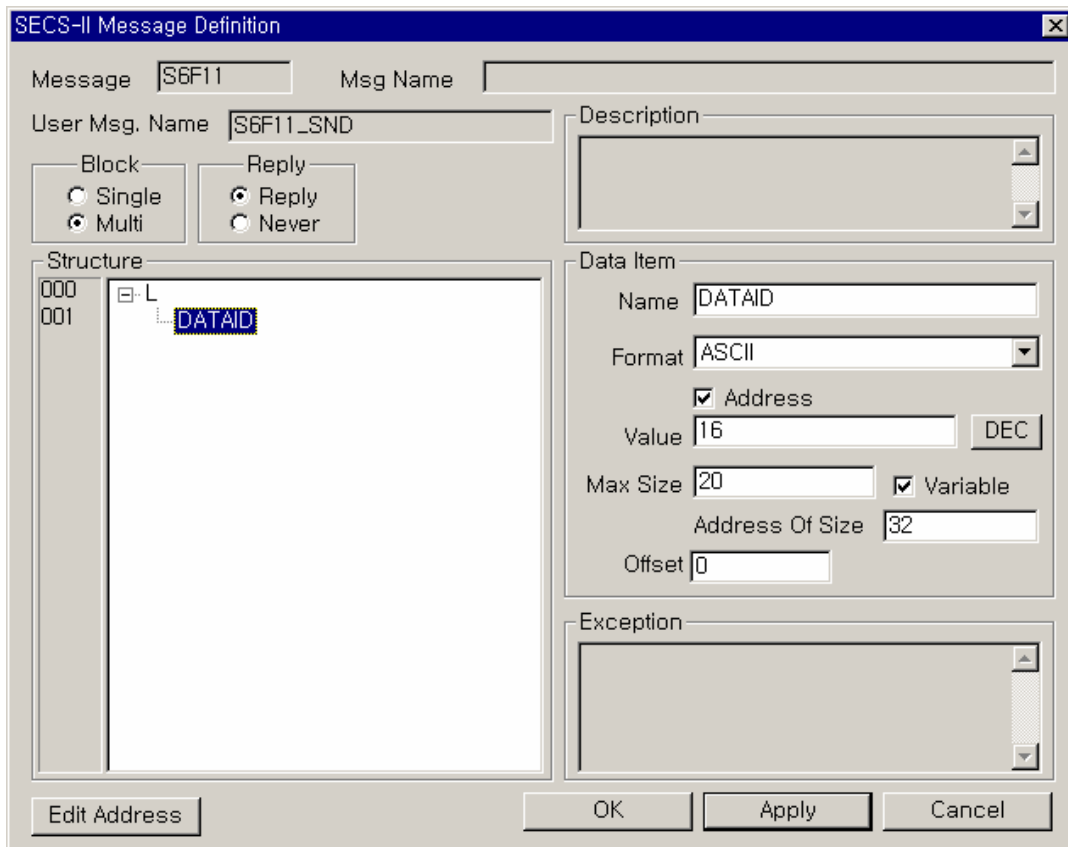


그림 6.6-8 SECS-II Message Definition

예) 송신 시, 16 번째에는 “ABCD”, 32 번째에는 0x0004(2-byte unsigned integer)가 쓰여져 있다면, <DATAID>는 “ABCD” 4 바이트가 전송됩니다. 32 번째에는 0x0002 라면 “AB” 2 바이트만 나갑니다. 32 번째에는 0x0008 라도, 이 경우엔 “ABCD” 4 바이트만 나갑니다. 빈 공간을 Space(0x20)로 채우지 않습니다.

수신 시는 <DATAID>의 크기가 20 바이트가 넘는다면, 20 바이트만 저장합니다. 32 번째에는 실제로 처리된 크기가 저장됩니다. <DATAID>의 크기가 30 바이트라면 16 번째에는 20 바이트만 복사되고, 32 번째에는 20 이 기록됩니다.

6.6.6. 동일 format 이 반복되는 S6F11

0x0000, 0x0001, 0x0002, 2-byte integer 3 개가 배열로 들어가는 경우를 보겠습니다.

<L [1]

<DATAID, 2-byte integer, 0x0000, 0x0001, 0x0002>

>

아래처럼 <Structure>에서 <DATAID> 아이템을 선택하고, 우측의 <Data Item>에서 <Format>은 2-byte Signed Integer 로, 직접 값을 넣는 경우에는 <Address>를 선택하지 않고, <Value>에 값을 쉼표(,)로 구분해서 0,1,2 를 넣습니다.

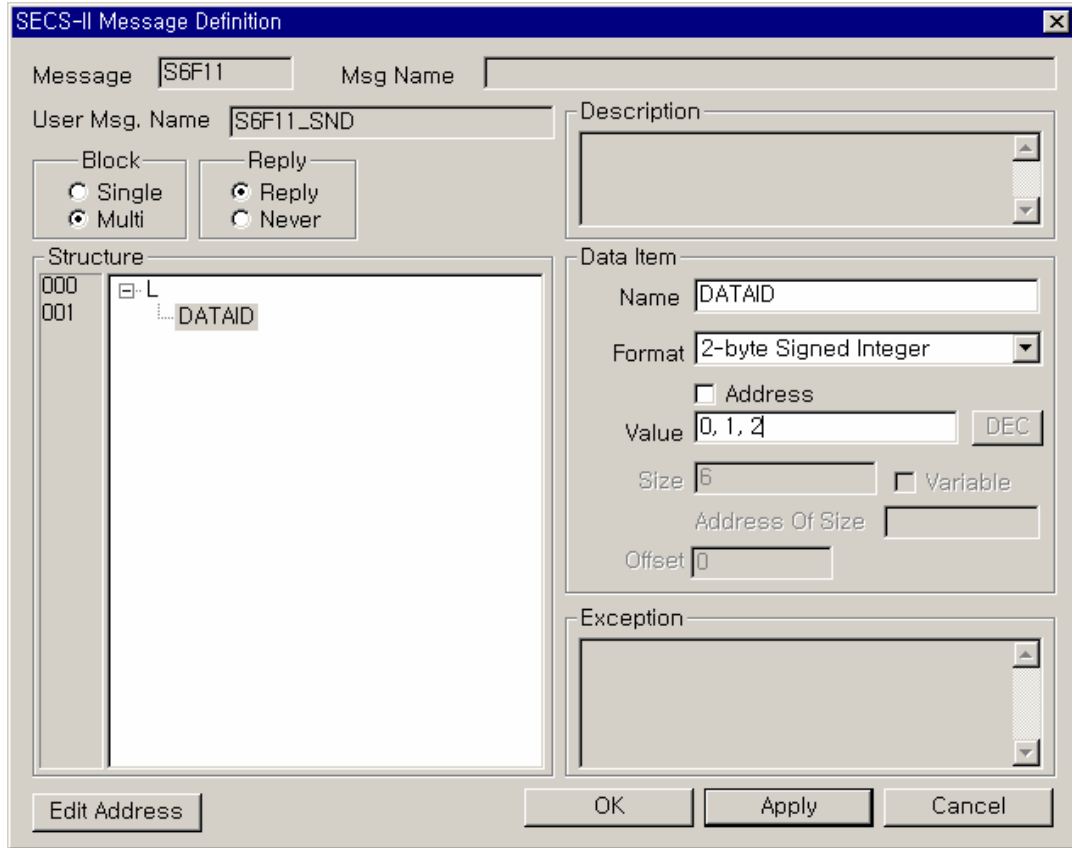


그림 6.6-9 SECS-II Message Definition

6.6.7. 동일 format 이 반복되는 S6F11, 값이 가변인 경우

2-byte integer, 3 개가 100 번지를 참조하는 경우를 보겠습니다.

<L [1]

<DATAID, 2-byte integer, 100 번지의 값, 102 번지의 값, 104 번지의 값 >

>

아래처럼 <Structure>에서 <DATAID> 아이템을 선택하고, 우측의 <Data Item>에서 <Format>은 2-byte Signed Integer 로, ETOS 의 메모리를 참조하므로 <Address>를 선택하고, <Value>에 ETOS 의 메모리 주소 100 을, <Size>엔 2 가 아니라 6(2byte * 3 개)을 넣습니다.

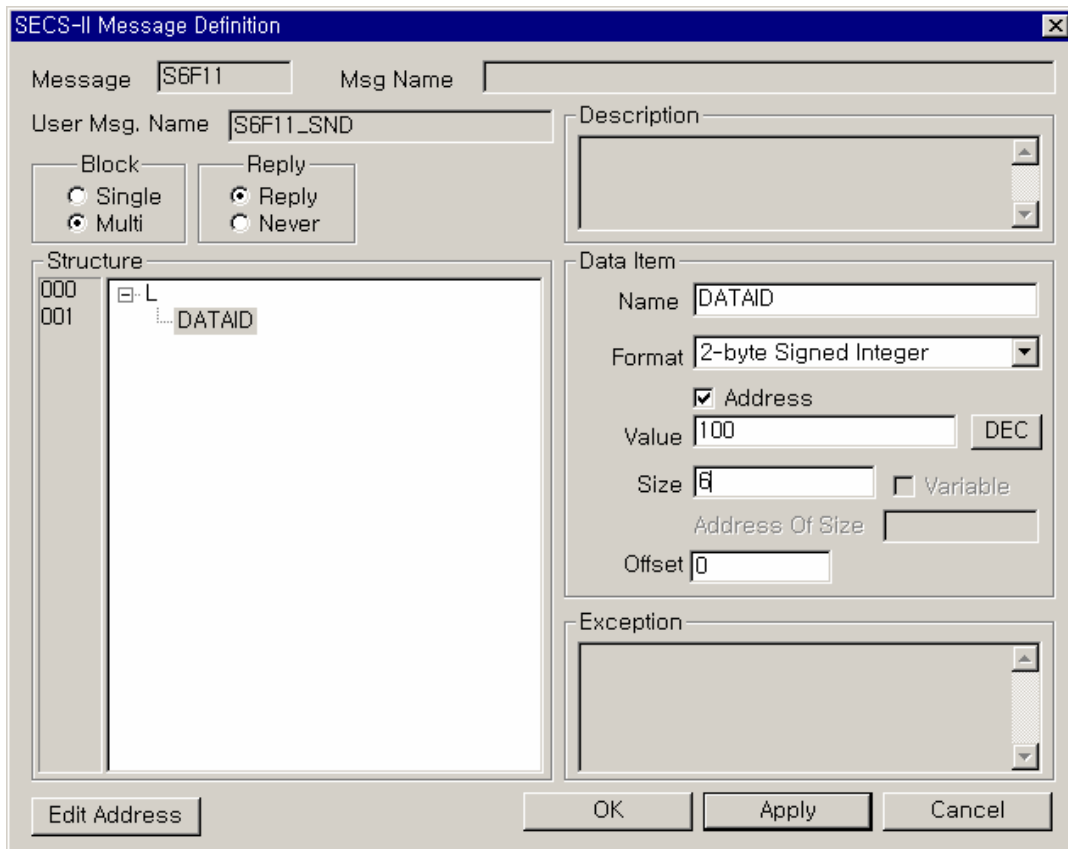


그림 6.6-10 SECS-II Message Definition

예) 송신 시, 100 번지에 0x0001, 102 번지에 0x0002, 104 번지에 0x0003 이 들어있다면, <DATAID>는 <DATAID, 2-byte integer, 0x0001, 0x0002, 0x0003> 6 바이트가 전송됩니다. <DATAID, 2-byte integer, 0x0001, 0x0002, 0x0003>가 수신 되었다면, 100 번지에 0x0001, 102 번지에 0x0002, 104 번지에 0x0003 이 차례대로 저장됩니다.

6.6.8. List 구조의 반복

아래처럼 <DATAID>를 감싸고 있는 LIST 가 n 개 만큼 반복되는 구조를 만들어 보겠습니다.

<L [n : ETOS 의 0 번째 값]

<L [2]

<DATAID, 4-byte integer, ETOS 의 16 번째 값>

<CEID, 2-byte integer, ETOS 의 32 번째 값>

<L [2]

<DATAID, 4-byte integer, ETOS 의 20 번째 값>

<CEID, 2-byte integer, ETOS 의 34 번째 값>

<L [2]

<DATAID, 4-byte integer, ETOS 의 24 번째 값>

<CEID, 2-byte integer, ETOS 의 36 번째 값>

...

>

먼저 <Structure>에서 첫 번째 <LIST> 아이템을 선택하고, 우측의 <Data Item>에서 <Format>은 LIST 로, ETOS 의 메모리를 참조하므로 <Address>를 선택하고, <Value>에 ETOS 의 메모리 주소 0 을, 다음에 올 LIST 가 최대 10 개까지 반복될 수 있게 <Max Size>엔 10 을 넣습니다.

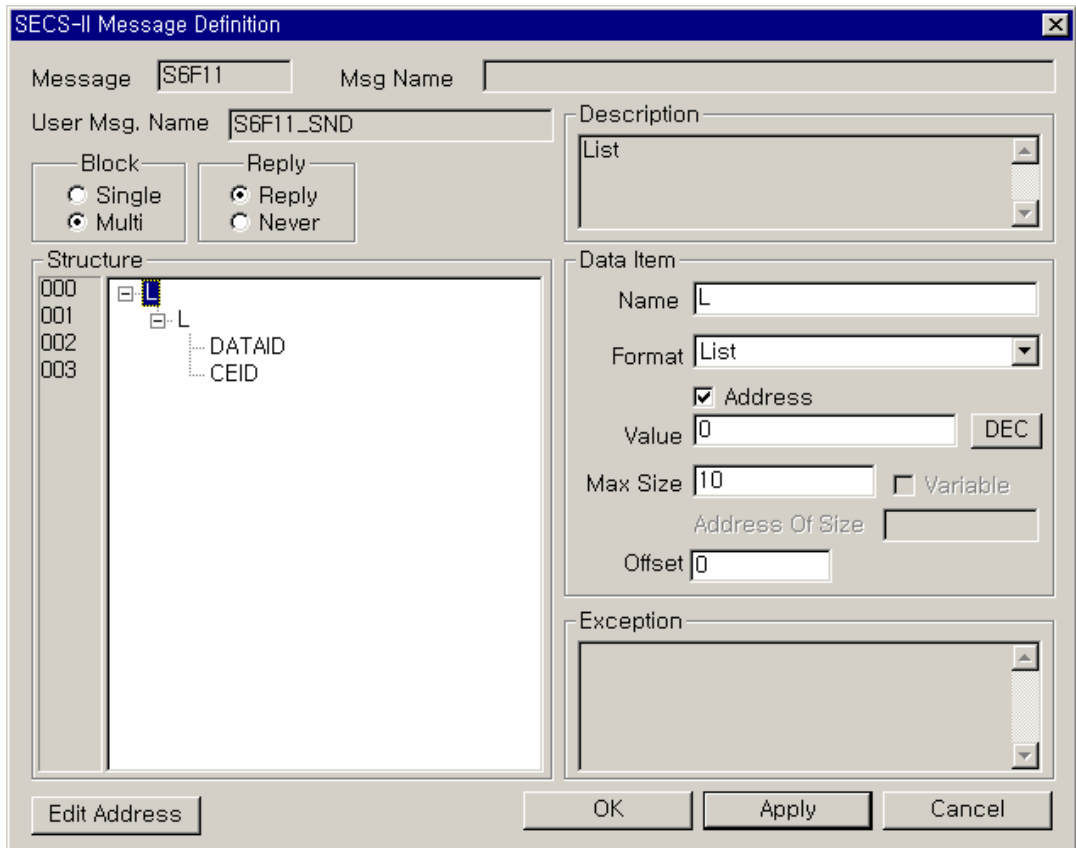


그림 6.6-11 SECS-II Message Definition

두 번째 <LIST> 아이템을 선택하고, 우측의 <Data Item>에서 <Format>은 LIST 로, 이후의 Structure 는 <DATAID>와 <CEID>로 고정이므로, <Value>에 “2”를 넣습니다.

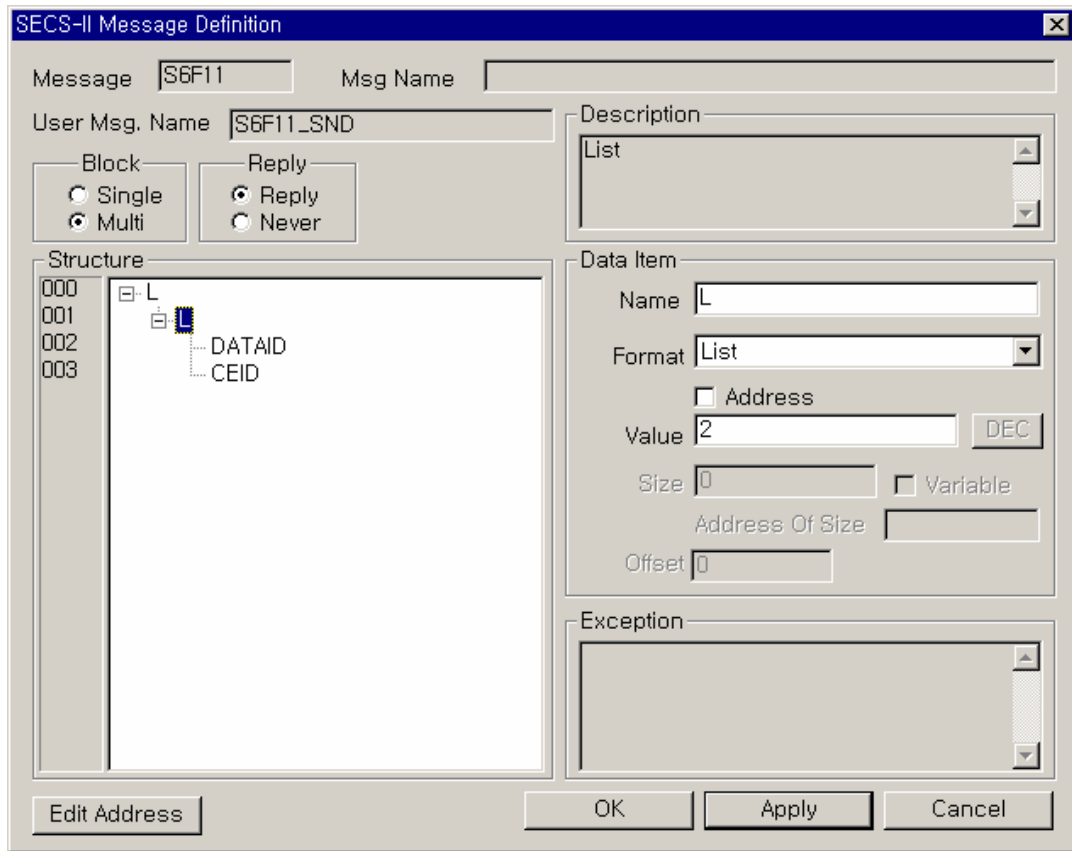


그림 6.6-12 SECS-II Message Definition

세 번째 <DATAID> 아이템을 선택하고, 우측의 <Data Item>에서 <Format>은 '4-byte Signed Integer'로, ETOS의 메모리를 참조하므로 <Address>를 체크하고, <Value>에 ETOS의 메모리 주소 "16"을, <Size>는 4-byte Signed Integer 이니까 "4"이고, <Offset>은 16,20,24...번지에 차례로 값이 있으므로 "4"입니다.

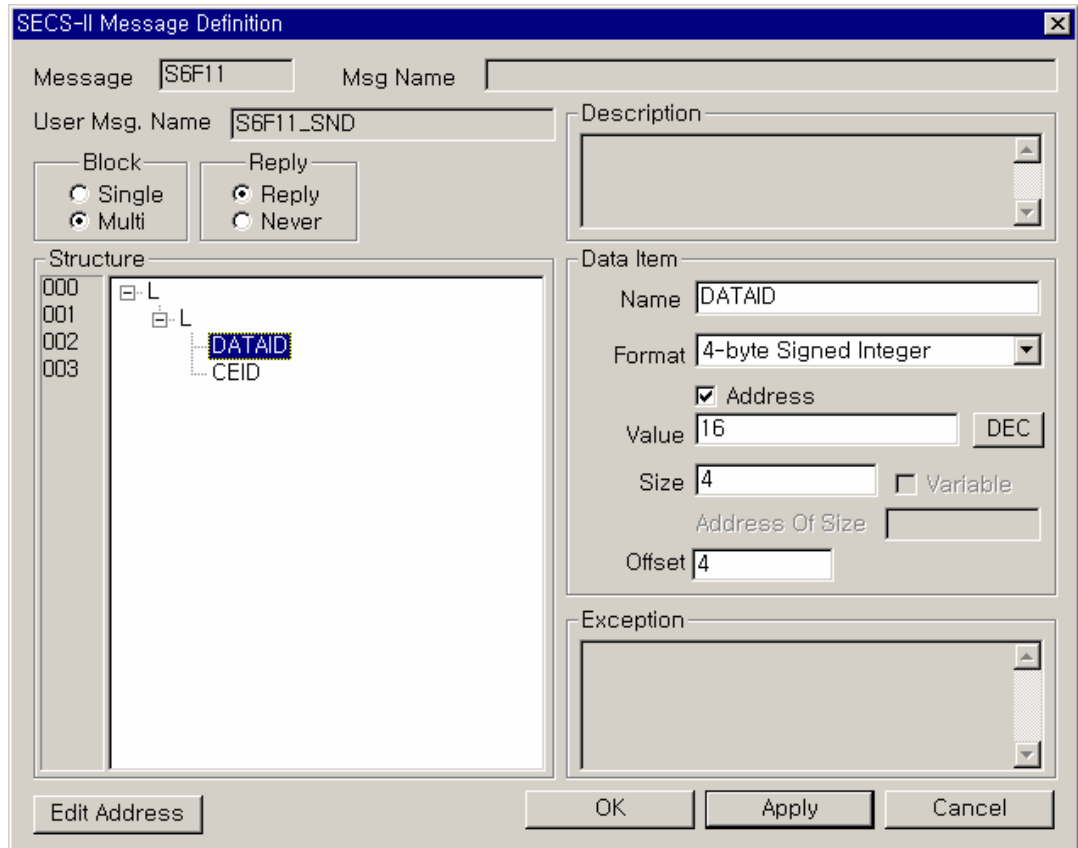


그림 6.6-13 SECS-II Message Definition

네 번째 <CEID> 아이템을 선택하고, 우측의 <Data Item>에서 <Format>은 '2-byte Signed Integer'로, ETOS의 메모리를 참조하므로 <Address>를 체크하고, <Value>에 ETOS의 메모리 주소 "32"를, <Size>는 2-byte Signed Integer 이니까 "2"이고, <Offset>은 32,34,36...번지에 차례로 값이 있으므로 "2"입니다.

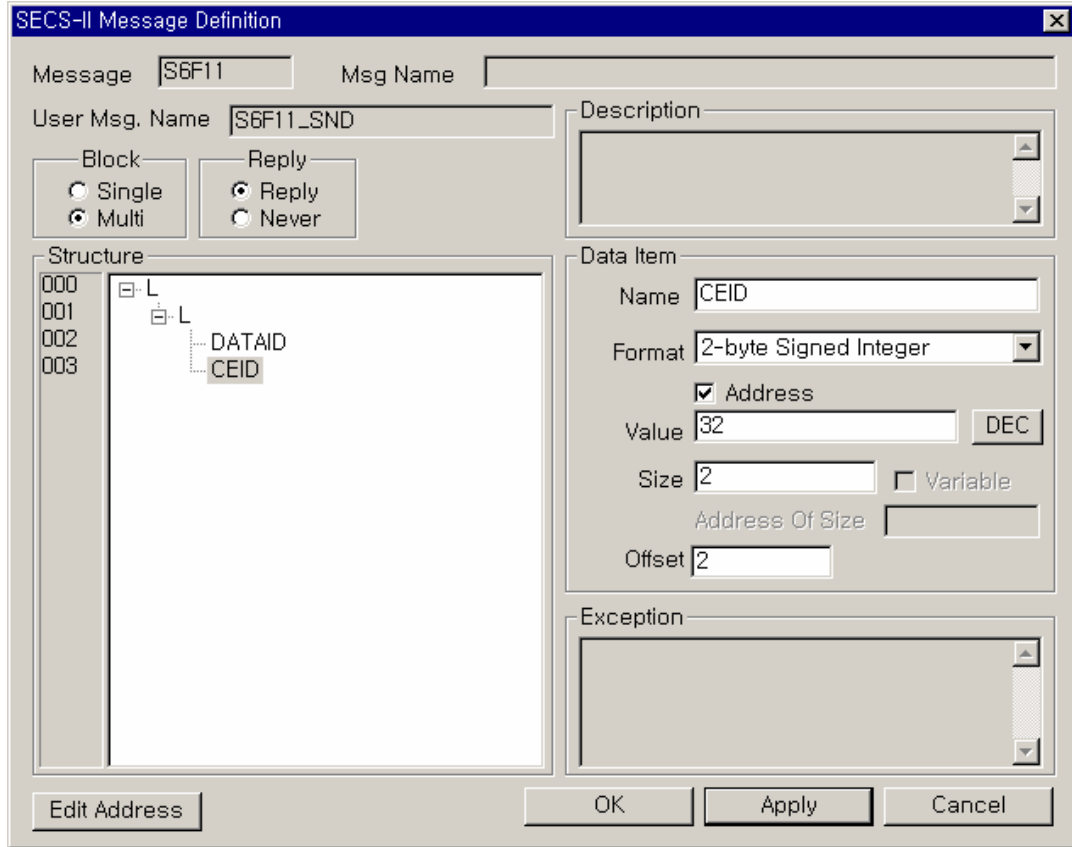


그림 6.6-14 SECS-II Message Definition

예) 송신 시

ETOS 의 메모리에 아래처럼 0 번지는 3, 16 번지는 0x00000001, 0x00000002, 0x00000003 이,
32 번지는 0x0004, 0x0005, 0x0006 이 들어있다면

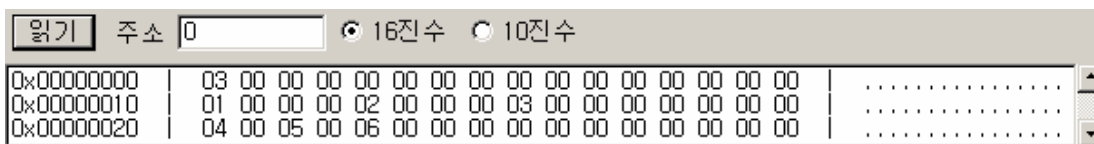


그림 6.6-15 ETOS 메모리

아래와 같은 메시지가 전송됩니다.

```
<L [3 : ETOS 의 0 번지 값]
  <L [2]
    <DATAID, 0x00000001, ETOS 의 16 번지 값>
    <CEID, 0x0004, ETOS 의 32 번지 값>
  <L [2]
    <DATAID, 0x00000002, ETOS 의 20 번지 값>
    <CEID, 0x0005, ETOS 의 34 번지 값>
  <L [2]
    <DATAID, 0x00000003, ETOS 의 24 번지 값>
    <CEID, 0x0006, ETOS 의 36 번지 값>
```

>
수신 시는 역으로 메시지의 값이 ETOS 의 해당 메모리에 저장됩니다.

주의) 가변 List 의 반복에는 몇 가지 제약 조건이 있는데, 아래처럼 LIST 가 없이 직접 아이템을 반복하는 것은 Item 별로 Offset(x, y)이 다를 수 있기 때문에 안됩니다.

```
<L [n]
  <DATAID, 4-byte integer, Offset x*0>
  <CEID, 2-byte integer, Offset y*0>
  <DATAID, 4-byte integer, Offset x*1>
  <CEID, 2-byte integer, Offset y*1>
  ...
  <DATAID, 4-byte integer, Offset x*(n-1)>
  <CEID, 2-byte integer, Offset y*(n-1)>
>
```


프로세스 및 아이템

제 7 장

이 장은 프로세스의 정의 및 설정 방법과 프로세스 구성의 기본 단위인 아이템에 대하여 설명합니다.

	내용	위치
7.1	개요	154
7.2	프로세스 추가/편집	155
7.3	프로세스 등록 예제	157
7.4	아이템 분류	164
7.5	아이템 일람	169
7.6	아이템 설명	174
7.7	단종 아이템	308

7. 프로세스 및 아이템

7.1. 개요

파라미터와 프레임 설정이 완료되면, 이제 상대 장비와의 통신을 통해 데이터를 송,수신해야 합니다. 또한 송,수신 데이터의 가공, 편집 등도 필요합니다. 이러한 일련의 동작은 프로세스를 통해 정의됩니다.

사용자는 프로세스의 이러한 기능을 통해 서로 다른 프로토콜을 사용하는 장비간의 데이터 교환을 정의할 수 있습니다.

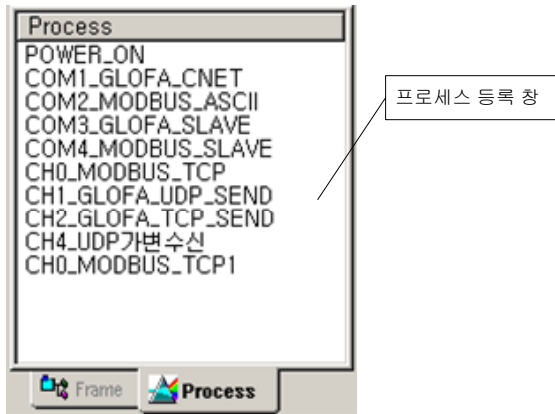


그림 7.1-1 프로세스 등록 창

ETOS의 동작은 프로세스로 정의하며, 프레임 송,수신, 데이터 변환, 데이터 복사/이동, 조건분기 등의 단위 명령어들의 구성에 의해 구체적인 동작이 결정됩니다. 이와 같이 프로세스를 구성하는 기본 단위인 명령어들을 아이템이라고 합니다. 반대로 프로세스는 명령어에 해당하는 아이템의 집합이라고 할 수 있습니다.

각각의 아이템은 기능에 따라 기본, 조건 분기, 통신, 시작, 응용 등의 아이템으로 분류합니다. 대부분의 아이템은 프로세스내에서 단독으로 동작하나, IF_XXX 아이템은 END_IF 아이템과 쌍(Pair)을 이루어야 합니다. 또한, 프로세스의 시작은 반드시 시작 아이템이어야 하며, 프로세스의 종료도 반드시 종료 아이템이어야 합니다.

7.1.1. 특징

프로세스 및 아이템은 다음과 같은 특징을 가지고 있습니다.

- ▶ 프로세스는 ETOS 동작을 정의하는 단위 프로그램으로 정의합니다.
- ▶ 프로세스는 아이템이라는 명령어의 조합으로 이루어집니다.
- ▶ 아이템은 ETOS 동작 정의의 기본 단위가 됩니다
- ▶ 프로세스는 최대 1024 개까지 등록할 수 있습니다.
- ▶ 아이템은 프로세스당 최대 4096 개까지 등록할 수 있습니다.
- ▶ 각각의 프로세스는 고유의 이름을 통해 구분하며 최대 20 자까지 입력 가능합니다.
- ▶ 프로세스에 등록된 아이템들은 사용자의 프로그램에 따라 순차적인 동작을 합니다.
- ▶ 여러 개의 프로세스들은 각기 독립적으로 수행됩니다.
- ▶ 프로세스는 선두에 등록할 수 있는 특정 아이템을 통해서만 시작됩니다. (시작 아이템)
- ▶ 프로세스는 후미에 등록할 수 있는 특정 아이템을 통해서만 종료됩니다. (종료 아이템)

- ▶ 프로세스는 선두 아이템의 조건이 충족될 때 기동되며 순차적 수행을 통해 종료 아이템에서 종료됩니다.
- ▶ 여러 개의 프로세스를 등록한 경우 등록순서는 프로세스의 동작순서와는 무관하며 프로세스 선두에 입력한 아이템 실행조건이 만족할 경우에 해당 프로세스가 실행됩니다.
- ▶ 각각의 프로세스들은 MULTI TASKING(동시수행)이 가능하도록 설계되어 있습니다.
- ▶ 프로세스별로 ENABLE/DISABLE 기능을 통해 프로세스 단위의 기동/정지가 가능합니다.
- ▶ 프로세스내에는 분기명령을 통해 프로그램 수행의 분기가 가능합니다.
- ▶ 프로세스간의 분기는 허용되지 않습니다.

ETOS의 동작에 관한 모든 내용은 프로세스를 통해 프로그램이 가능하므로 사용자는 용도에 맞게 프로세스를 작성해야 합니다. 프로세스 등록은 ETOS-PD의 {프로세스} 창을 통해 할 수 있습니다. 프로세스는 아이템들의 집합이므로 본 장에서는 프로세스와 아이템의 사용법에 대해 설명합니다.

7.2. 프로세스 추가/편집

{프로세스 등록} 창에 마우스를 위치하고, 우측 버튼을 누릅니다. [프로세스를 추가]를 선택을 선택합니다. {프로세스 편집} 대화상자에서 이름을 입력하여 프로세스를 등록 합니다. <Enable> 체크박스는 프로세스를 수행할지 여부를 결정하는 항목이며, 기본값은 Enable입니다.

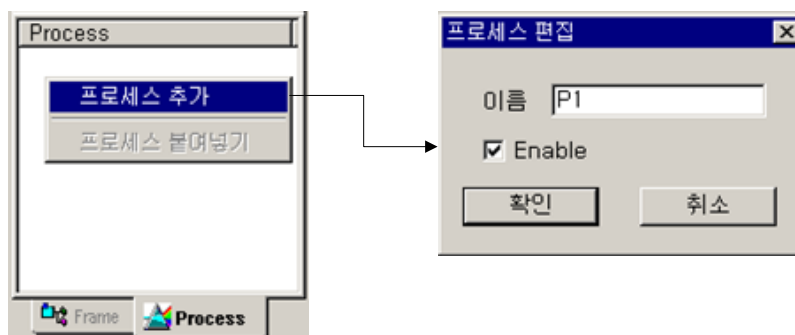


그림 7.2-1 프로세스 추가

프로세스 등록을 하면 ETOS-PD의 우측 화면에 다음 그림과 같은 {프로세스 편집} 창이 나타납니다. {프로세스 편집} 창에서 아이템을 추가, 삭제 및 수정 할 수 있습니다. 한 프로그램에서 등록 가능한 프로세스의 수는 최대 1024 개이며, 프로세스 당 최대 4096 개의 아이템을 입력할 수 있습니다.

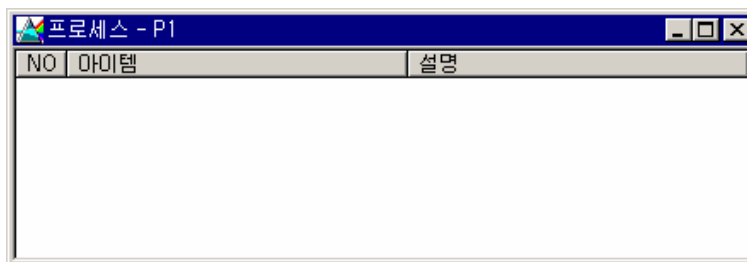


그림 7.2-2 프로세스 편집 창

프로세스 단위의 복사, 붙여넣기, 삭제 등이 가능합니다. {프로세스 등록} 창의 해당 프로세스에 마우스를 위치하고, 마우스 우측버튼을 누르면, 프로세스 편집 메뉴를 사용할 수 있습니다.

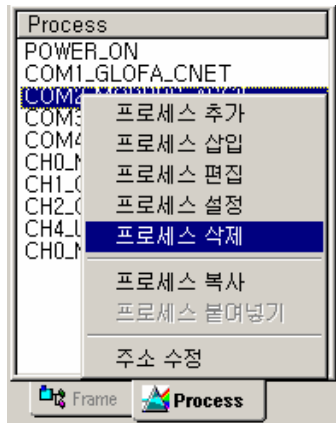


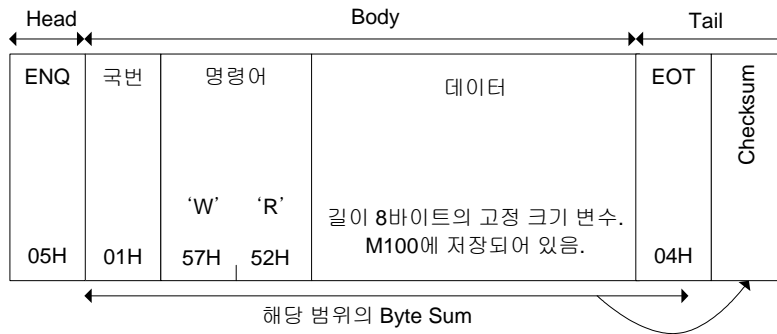
그림 7.2-3 프로세스 편집 메뉴

7.3. 프로세스 등록 예제

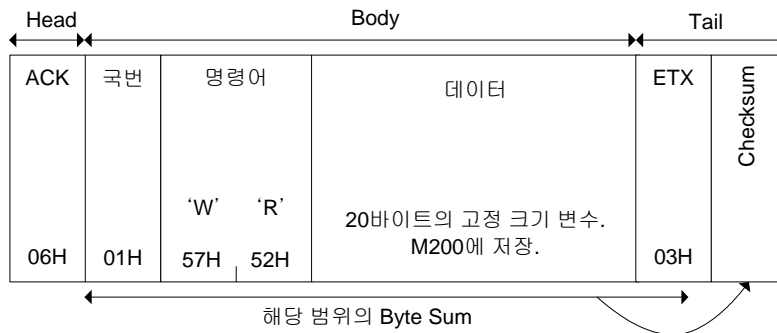
7.3.1. 예제 프레임

[6.4 프레임 등록 예제](#)에서 등록한 예제 프레임을 사용하여 프로세스를 작성하겠습니다. ETOS 가 1~10 번 국번의 상대기기에 차례로 데이터를 요청하고(송신 프레임) 응답을 받으면(수신 프레임) 메모리에 저장하는 예입니다. 송,수신 프레임 등록은 [6.4 프레임 등록 예제](#) 를 참조하시기 바랍니다.

송신 프레임 (ETOS → 상대기기)



수신 프레임 (상대기기 → ETOS)



7.3.2. 프로세스 추가

{프로세스 등록} 창에서 마우스 우측 버튼을 눌러 [프로세스 추가]를 선택 합니다.
{프로세스 편집} 대화상자에서 <이름>을 입력합니다.

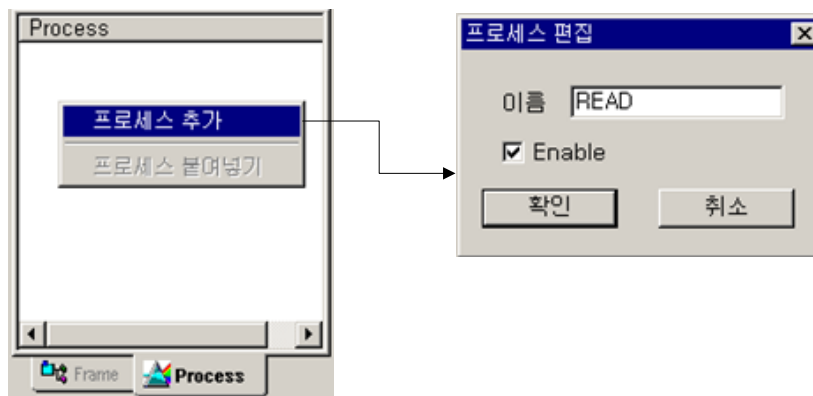


그림 7.3-1 예제 프로세스 추가

7.3.2.1. 아이템 추가

▶ 시작 아이템 추가

프로세스를 추가하면 시작 아이템을 입력하도록 {아이템 추가} 대화상자가 나타납니다.^(*80)

<분류>를 변경하지 말고 시작 아이템에서 원하는 아이템을 선택해서 입력해야 합니다.^(*81)

<아이템>에서 IF_TIMER 를 선택한 후, <주기>에 “100”을 입력합니다. 이렇게 만들어진 프로세스 ‘READ’는 1 초 주기로 실행되는 정주기 프로세스가 됩니다.

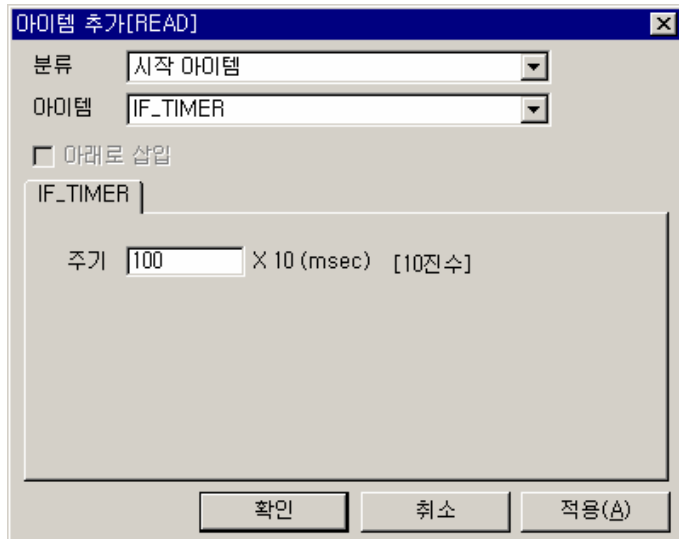


그림 7.3-2 시작 아이템 추가

다음 그림은 시작 아이템인 IF_TIMER 가 설정된 프로세스 ‘READ’의 편집 창 모습입니다. IF_TIMER 아이템을 추가하면 자동으로 END_IF 아이템이 함께 추가됨을 알 수 있습니다.

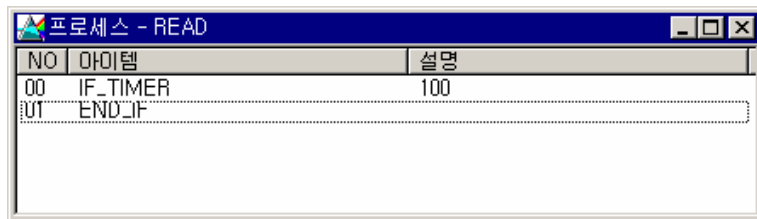


그림 7.3-3 정주기 프로세스

^(*80) ETOS-PD V2.3 이전 버전은 프로세스 추가 후, [아이템 추가] 메뉴를 실행해서 시작 아이템을 추가해 주어야 합니다.

^(*81) 프로세스 선두는 반드시 시작 아이템이어야 합니다.

▶ 국번 설정

등록한 송신 프레임의 2 번째 바이트는 국번입니다. 국번은 1 바이트 고정 크기 변수로 등록하였습니다.^(*82) 그러므로 송신 전에 그 값을 지정할 필요가 있습니다. 이 예에서는 1 번부터 10 번까지 국번을 바꿔가며 동일한 프레임을 1 ~ 10 번 국번의 상대기기에 송신하도록 하겠습니다. 국번이 매번 증가하도록 합니다. 다음 그림과 같이 END_IF 에 마우스를 위치하고, 우측 버튼을 눌러 [아이템 삽입] 메뉴를 선택합니다.

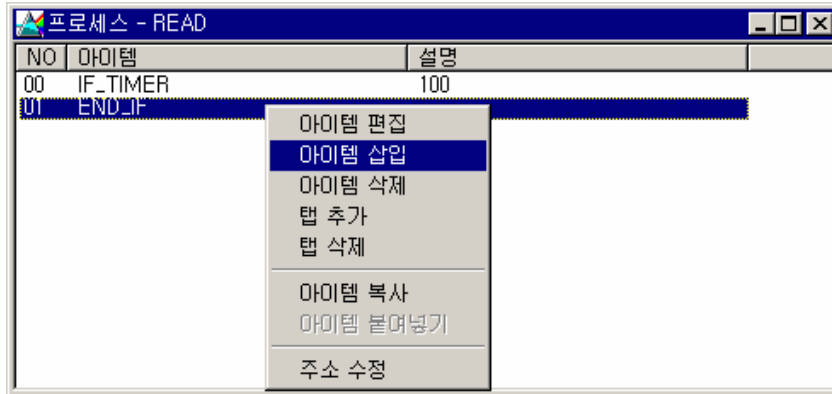


그림 7.3-4 ARITHMETIC 아이템 삽입

아이템으로 ARITHMETIC 을 선택하고 다음 그림과 같이 입력합니다. 의미는 국번(98 번지)을 1 증가시키는 것입니다.

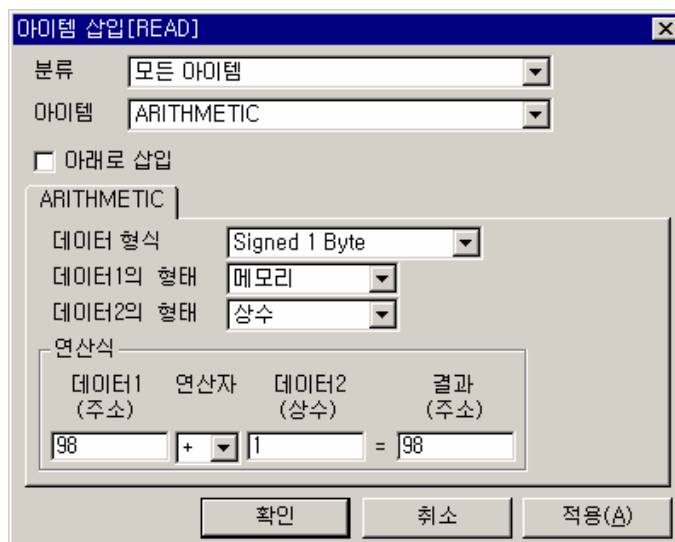


그림 7.3-5 ARITHMETIC 아이템을 이용한 국번 증가

^(*82) 6.4. 프레임 등록 예제를 참조하시기 바랍니다.

이제 국번은 1 초 마다 1 씩 증가하게 됩니다. 그러나, 10 번까지만 증가하고 다시 1 번으로 초기화되어야 합니다. 따라서, 국번이 10 번을 초과하는지 확인하는 아이템이 필요합니다.

다음 그림과 같이 END_IF 위치에서 아이템을 삽입합니다.

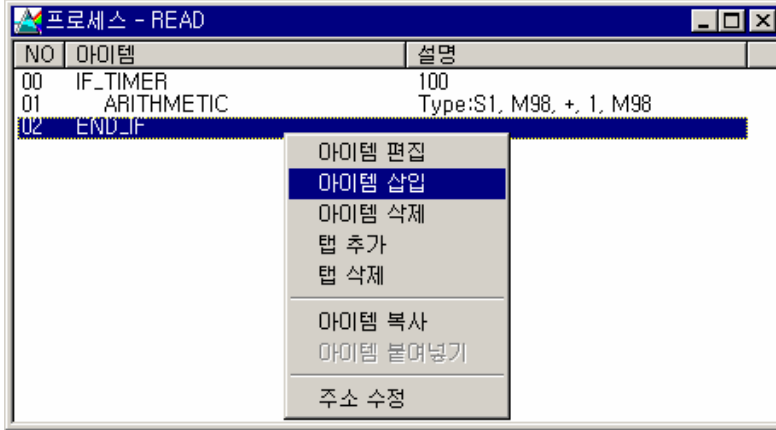


그림 7.3-6 IF_NUM_CMP 아이템 삽입

아이템으로 IF_NUM_CMP 를 선택하고 다음 그림과 같이 입력합니다. 98 번지는 앞서 프레임 등록 예에서 국번으로 지정한 번지입니다. 다음 그림은 98 번지 즉, 국번이 10 보다 커지면이라는 의미입니다.

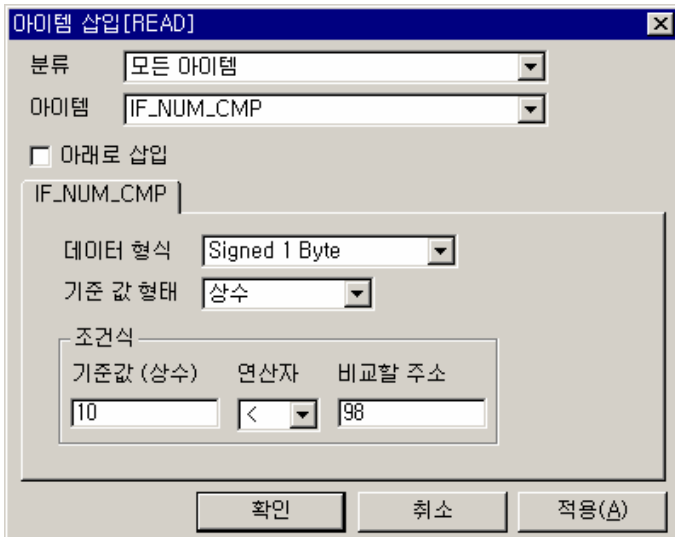


그림 7.3-7 IF_NUM_CMP 아이템을 이용한 국번 비교

이제 국번이 10 보다 커지면 국번을 다시 1로 돌려놓습니다.

다음 그림과 같이 END_IF에 마우스를 위치하고 아이템을 삽입합니다.

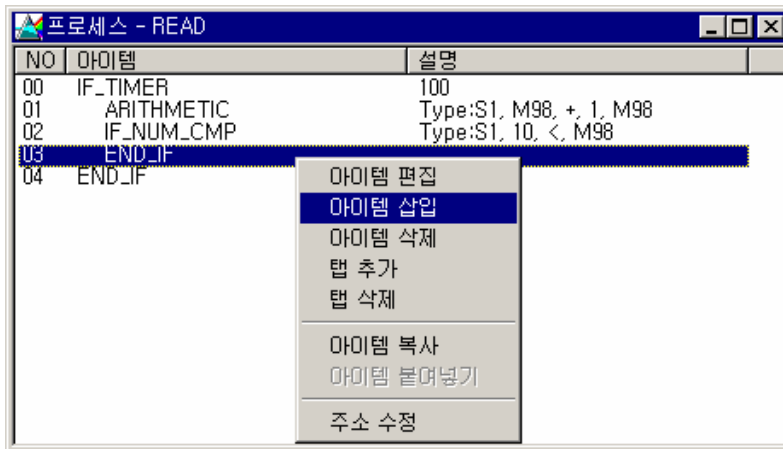


그림 7.3-8 MEM_SET 아이템 삽입

아이템에서 MEM_SET을 선택하고, 다음 그림과 같이 입력합니다. 이제 국번이 10보다 커지면 다시 1로 초기화 됩니다.

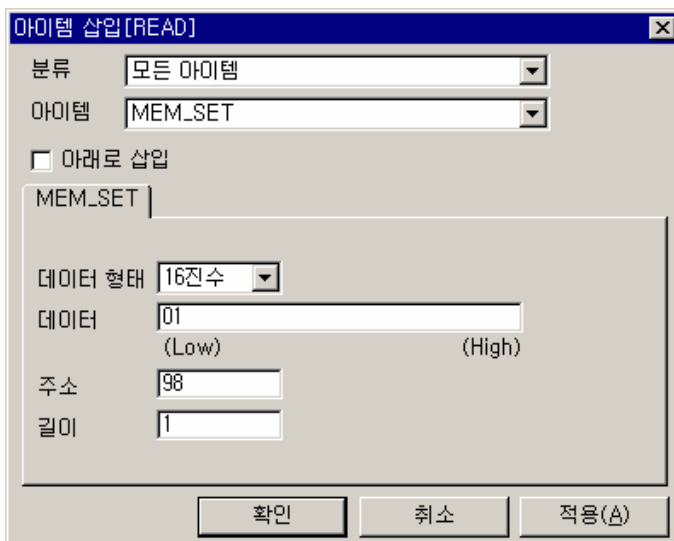


그림 7.3-9 MEM_SET 아이템을 이용한 국번 초기화

▶ 통신 아이템 추가

이제 실제 프레임을 송신하도록 아이템을 추가할 차례입니다. 다음 그림과 같이 END_IF 아이템에 마우스를 위치하고, 아이템을 삽입 합니다.

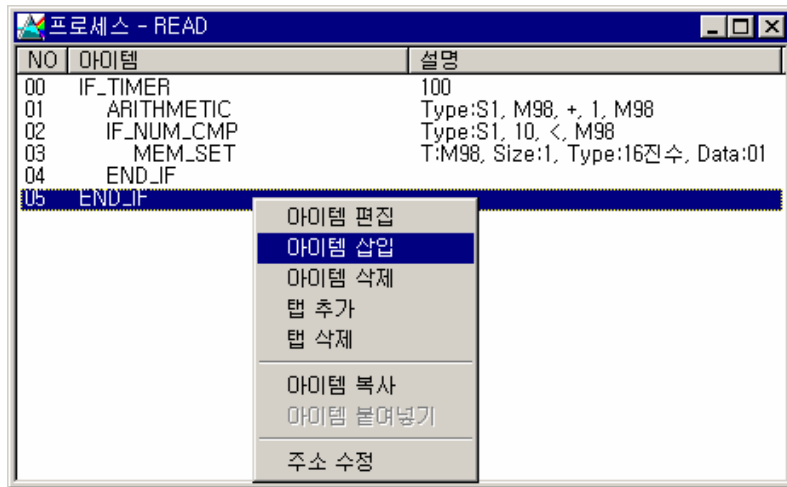


그림 7.3-10 IF_REQ_RESP 아이템 추가

아이템에서 IF_REQ_RESP 아이템을 선택합니다. 요구 프레임과 응답 프레임은 직접 입력하거나, {프레임 등록} 창에서 해당 프레임을 선택한 후, 드래그 앤 드롭(Drag & drop)할 수 있습니다. IF_REQ_RESP 아이템은 요구 프레임을 송신한 후, 응답 프레임을 기다리는 아이템입니다. 응답 프레임이 정상적으로 수신되면 다음 아이템으로 진행하고, 응답 프레임이 오지 않아 타임아웃이 발생하는 경우에는 ELSE 나 END_IF 다음 아이템으로 진행합니다.

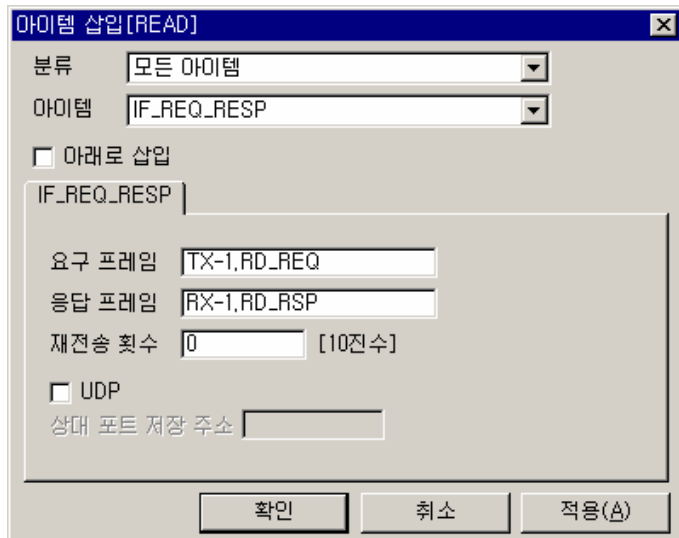
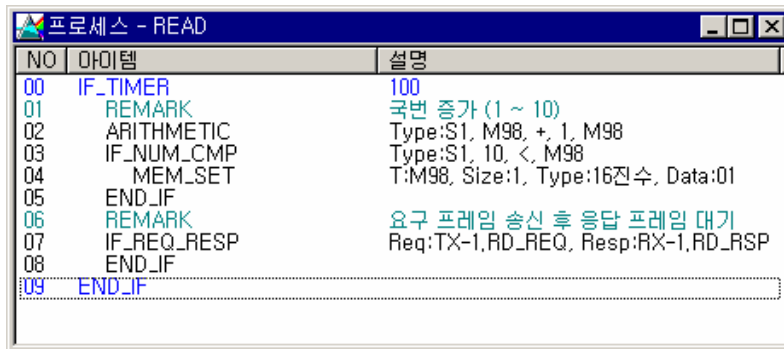


그림 7.3-11 IF_REQ_RESP 아이템을 이용한 송,수신 프레임 설정

다음 그림은 완성된 프로세스의 모습입니다. 1 초에 한번씩 국번을 바꿔가며(1 ~ 10 번) 프레임을 송신하고, 응답 프레임이 수신되면 해당 메모리에 데이터를 저장합니다. 수신 프레임의 데이터 저장은 별도의 아이템 필요없이 IF_REQ_RESP 아이템 등록만으로 OS가 알아서 처리합니다. 데이터 저장 주소는 수신 프레임 등록 시 지정한 200 번지입니다. 수신 프레임에 대해서는 [6.4 프레임 등록 예제](#) 를 참조하시기 바랍니다.



NO	아이템	설명
00	IF_TIMER	100
01	REMARK	국번 증가 (1 ~ 10)
02	ARITHMETIC	Type:S1, M98, +, 1, M98
03	IF_NUM_CMP	Type:S1, 10, <, M98
04	MEM_SET	T:M98, Size:1, Type:16진수, Data:01
05	END_IF	
06	REMARK	요구 프레임 송신 후 응답 프레임 대기
07	IF_REQ_RESP	Req:TX-1,RD_REQ, Resp:RX-1,RD_RSP
08	END_IF	
09	END_IF	

- * REMARK 아이템을 이용해서 프로그램에 대한 설명을 삽입할 수 있습니다. REMARK 아이템은 프로세스 수행에 영향을 미치지 않습니다.

7.4. 아이템 분류

7.4.1. 조건 분기 아이템

조건분기 아이템은 프로세스 수행 중 조건에 따라 분기를 하며 IF_XXX 아이템에서 시작하여 ELSE 또는 END_IF 로 끝나는 영역까지의 아이템을 수행합니다. 조건 분기 아이템은 END_IF 와 쌍(Pair)으로 사용되어야 합니다.

▶ ELSE 가 없는 경우

IF_XXX 아이템의 조건을 만족 할 경우 END_IF 까지의 아이템들을 수행합니다.

예) IF_MEM_CMP 아이템의 조건이 만족되면 BIT_WRITE 아이템을 수행합니다.

NO	아이템	설명
00	IF_MEM_CMP	M100, M200, Size:4
01	BIT_WRITE	T:M10, BitPos:3, Data:On
02	END_IF	

그림 7.4-1 IF 조건문 구성 예

▶ ELSE 가 있는 경우

IF_XXX 아이템의 조건을 만족 할 경우 IF_XXX ~ ELSE 사이의 아이템들을 수행하며, 만족치 않을 경우 ELSE ~ END_IF 사이의 아이템들을 수행합니다.

예) IF_MEM_CMP 아이템의 조건이 만족되면 01 번 아이템이 실행되며, 만족치 않을 경우 03 번 아이템이 실행됩니다.

NO	아이템	설명
00	IF_MEM_CMP	M100, M200, Size:4
01	BIT_WRITE	T:M10, BitPos:3, Data:On
02	ELSE	
03	BIT_WRITE	T:M10, BitPos:3, Data:Off
04	END_IF	

그림 7.4-2 IF 조건문 구성 예

* 위 예에서 프로세스 0 번 줄의 IF_MEM_CMP 아이템은 시작 아이템이 아닙니다. 이해를 돕기 위해 선두에 작성한 것이며, 실제 작성 시는 적절한 시작 아이템을 사용해야 합니다.

▶ IF_XXX 아이템의 중복 사용

IF_XXX 아이템은 중복 사용할 수 있습니다. 즉, IF_XXX ~ END_IF 사이에 다른 IF_XXX ~ END_IF 아이템을 여러 개 사용할 수 있으며, 이 경우 IF_XXX 와 END_IF 가 쌍을 이루도록 해야 합니다. 여러 개를 중복하는 경우, IF_XXX 에서 가장 가까운 END_IF 까지 쌍으로 동작합니다. 다음은 IF_XXX 아이템을 중복하여 사용한 예입니다.

NO	아이템	설명
00	IF_TIMER	100
01	IF_MEM_CMP	M100, M200, Size:4
02	BIT_WRITE	T:M10, BitPos:3, Data:On
03	IF_BIT_TEST	S:M10, BitPos:3, Data:On
04	ARITHMETIC	Type:S1, M20, +, 1, M20
05	END_IF	
06	END_IF	
07	END_IF	

그림 7.4-3 IF 아이템 중복사용 예

7.4.2. 시작 아이템

여러 개의 프로세스를 등록한 경우 각각의 프로세스는 독립적으로 동시에 실행(MULTI-TASKING) 됩니다. 프로세스의 등록순서는 실행 순서와 관계가 없으며 각 프로세스의 시작 아이템의 조건에 의해 동작 시점이 결정됩니다. 이는 하나의 프로세스가 다른 프로세스의 동작에 영향을 미치지 않음을 의미하며, 각각의 프로세스는 프로세스 선두에 등록된 아이템의 실행 조건이 충족되면 수행됩니다. 프로세스의 선두에 등록할 수 있는 시작 아이템은 다음과 같으며, 이 외의 다른 아이템은 프로세스 선두에 올 수 없습니다.

▶ 프로세스는 반드시 다음 중 한 가지의 아이템으로만 시작됩니다. 이 외의 다른 아이템은 프로세스 선두에 올 수 없습니다.

- * IF_POWER_ON
- * IF_TIMER
- * IF_FRAME_RCV
- * IF_SECS_MC_FRAME_RCV ^(*83)
- * IF_MEM_WRITE_RCV

프로세스는 마지막 아이템 또는 END 아이템을 만나면 수행을 종료합니다. 상기 아이템들의 동작에 대해서는 [7.6. 아이템 설명](#) 을 참조하시기 바랍니다.

▶ 시작 아이템

프로세스를 시작할 수 있는 아이템으로, 모든 프로세스는 반드시 이 중 하나로 시작해야 합니다. 또, IF_XXX 아이템은 해당 조건이 만족하지 않는 경우를 위해 ELSE 아이템을 사용할 수 있으나, 시작 아이템은 해당 조건이 만족하지 않는 경우, 프로세스 실행 자체가 되지 않으므로, ELSE 아이템을 사용할 수 없습니다.

▶ 종료 아이템

프로세스의 마지막 줄은 반드시 END_IF 아이템으로 끝나야 합니다.

^(*83) 드라이버가 SECS 일 경우에만 사용할 수 있습니다.

7.4.3. 통신 아이템

통신 아이템은 시리얼 또는 이더넷을 통해 프레임을 송,수신하는 송,수신 통신 아이템과 통신 환경 설정과 관련된 아이템으로 나눌 수 있습니다. 송,수신 통신 아이템은 사용자 정의 프레임 송, 수신 아이템과 내장 프로토콜 송,수신 아이템으로 다시 분류할 수 있습니다. 다음 표는 통신 아이템을 위의 기준으로 분류한 것입니다.

표 7.4-1 통신 아이템 분류

분류	아이템	내 용
통신 환경 설정	ENET_CLOSE	이더넷 채널 연결 해제
	ENET_DISABLE	이더넷의 드라이버의 통신을 잠시 중지(연결 유지)
	ENET_ENABLE	중단된 이더넷 드라이버 통신을 다시 시작
	ENET_OPEN	이더넷 채널 연결 요청
	MODEM_CONTROL	모뎀의 DTR/RTS/Break 신호를 제어
사용자 정의 프레임 송,수신	FRAME_SND	참조된 프레임의 송신명령
	IF_FRAME_RCV	참조된 프레임 수신 시 IF 문 실행
	IF_REQ_RESP	프레임 송신 후 응답 프레임 수신
내장 프로토콜 송,수신	IF_3964R_READ	Siemens 3964(R) 프로토콜로 읽기 요청 송신
	IF_3964R_WRITE	Siemens 3964(R) 프로토콜로 쓰기 요청 송신
	IF_MEM_WRITE_RCV	드라이버로 메모리 쓰기 프레임 수신 시 IF 문 수행
	IF_GLOFA_READ	GLOFA Cnet/Enet 프로토콜로 읽기 요청 송신
	IF_GLOFA_WRITE	GLOFA Cnet/Enet 프로토콜로 쓰기 요청 송신
	IF_MELSEC_READ	MELSEC 프로토콜로 읽기 요청 송신
	IF_MELSEC_WRITE	MELSEC 프로토콜로 쓰기 요청 송신
	IF_MODBUS_READ	MODBUS 프로토콜로 읽기 요청 송신
	IF_MODBUS_WRITE	MODBUS 프로토콜로 쓰기 요청 송신
	MEM_WRITE_RESP_SND	IF_MEM_WRITE_RCV 수행 후 응답 리턴

▶ 이더넷 전용 아이템

ENET_CLOSE, ENET_DISABLE, ENET_ENABLE, ENET_OPEN 은 이더넷 전용 아이템입니다.

▶ 시리얼 전용 아이템

MODEM_CONTROL, IF_3964R_READ, IF_3964R_WRITE 는 시리얼 전용 아이템입니다.

* 나머지 통신 아이템은 시리얼, 이더넷 공용 아이템입니다.

7.4.4. Queue 아이템

Queue 관리 아이템은 ETOS 메모리의 일정 영역을 사용자가 FIFO(First In First Out)메모리로 사용할 수 있도록 합니다. FIFO 메모리는 사용자가 직접 메모리 번지를 할당하여 관리하는 방식이 아니라, Queue 의 ID 를 이용하여 지정한 크기의 메모리 POOL 에 데이터를 저장했다가 필요한 경우 선입선출의 원칙에 의해 꺼내 쓸 수 있는 방식입니다. 이 아이템들을 이용하여 사용자는 임의의 크기로 Queue 를 생성 할 수 있으며, 생성된 Queue 에 데이터를 저장 및 호출할 수 있습니다.

Queue 하나 당 2Kbyte 의 메모리가 할당되어 Queue 의 데이터 길이 설정 값에 따라 n 개의 FIFO 메모리로 설정됩니다. 예를 들어 데이터 길이를 32byte 로 설정할 경우 그 Queue 는 32byte 크기의 64 개 FIFO 메모리를 가진 Queue 로 사용 가능합니다. (2Kbyte 를 32byte 로 나눈 결과가 Queue 에 할당된 FIFO 개수로 64 개가 됩니다.)

Queue 를 이용하기 위해서는 새로운 Queue 를 만들어야 하며 QUEUE_CREATE 를 사용하여 크기 및 ID 를 부여 합니다. QUEUE_CREATE 에 의해 생성된 Queue 로 데이터를 저장하기 위해서는 QUEUE_SND_TO 아이템을 사용하며 이때 Queue 생성시 사용한 ID 를 사용하여 해당 Queue 로 데이터를 보냅니다. Queue 에 저장된 데이터는 QUEUE_RCV_FROM 아이템을 이용하여 사용할 수 있으며 QUEUE_CLEAR 아이템을 사용하여 Queue 의 모든 FIFO 데이터를 지울 수 있습니다. Queue 는 4 개 까지만 생성 가능합니다.

표 7.4-2 Queue 아이템

아이템	내 용
QUEUE_CLEAR	Queue 의 FIFO 메모리를 지웁니다.
QUEUE_CREATE	Queue 메모리(FIFO)를 생성합니다.
QUEUE_RCV_FROM	설정된 Queue 로부터 데이터를 읽어냅니다.
QUEUE_SND_TO	설정된 Queue 에 데이터를 저장합니다.

7.4.5. 전용 아이템

ETOS 특수 모듈 전용 아이템입니다. ETOS 특수 모듈에는 IFOR 모듈, DIO8 모듈이 있습니다. 다음 아이템들은 전용 모듈을 옵션으로 장착한 경우에만 사용할 수 있습니다. IFOR 및 DIO8 모듈은 모두 ETOS-200/1000 에만 장착 가능합니다.

- * 다음 아이템을 편집하기 위해서는 ETOS-PD 에서 [옵션] -> [편집 옵션]을 선택한 후, {편집 옵션} 대화상자에서 해당 모듈 편집을 체크해야 합니다.

7.4.5.1. IFOR

ETOS-200/1000 용의 광링(Optic Ring) 제어 모듈입니다. 다음과 같이 3 가지의 전용 아이템이 있습니다. 다음 아이템은 ETOS-PD 에서 [옵션] -> [편집 옵션]을 선택한 후, {편집 옵션} 대화상자에서 <IFOR 편집>을 체크해야 나타납니다.

IFOR 모듈에 대한 보다 자세한 사항은 ETOS-IFOR 사용설명서를 참조하시기 바랍니다.

표 7.4-3 IFOR 전용 아이템

아이템	내 용
EVENT_READ_WRITE	IFOR 공유 메모리 중 이벤트 영역 읽기/쓰기
IF_EVENT_OCCUR	IFOR 공유 메모리 중 이벤트 영역 변경시 이벤트 발생
SLAVE_MEM_READ_WRITE	IFOR 공유 메모리 읽기/쓰기

7.4.5.2. DIO8

ETOS-200/1000 용의 디지털 입,출력 모듈입니다. 다음과 같이 2 가지의 전용 아이템이 있습니다. 다음 아이템은 ETOS-PD 에서 [옵션] -> [편집 옵션]을 선택한 후, {편집 옵션} 대화상자에서 <I/O 보드 편집>을 체크해야 나타납니다.

DIO 모듈에 대한 보다 자세한 사항은 ETOS-DIO8 사용설명서를 참조하시기 바랍니다.

표 7.4-4 I/O 전용 아이템

아이템	내 용
IF_IO_STATE_CHANGE	DIO보드 IO값 변경시 이벤트 발생
IO_READ_WRITE	DIO보드 IO값 읽기/쓰기

7.5. 아이템 일람

7.5.1. 아이템 일람 (기능별)

번호	분류	아이템	내 용	비 고
1	기본	ARITHMETIC	사칙 연산 기능	
2		ASC_TO_BIN	문자열로 표현된 2 진수를 실제 2 진수로 변환	
3		ASC_TO_HEX	ASCII 문자를 16 진수로 변환	
4		BIN_TO_ASC	2 진수값을 문자열로 표현	
5		BIT_MASK	BIT 단위의 메모리 조작	
6		BIT_WRITE	BIT 단위 데이터 쓰기	
7		BIT_WRITE_EX	BIT 단위 데이터 간접 지정 쓰기	
8		BITS_CPY	BIT 단위 데이터 복사	
9		BITS_OPERATION	메모리의 BIT AND, OR, NOT, XOR 연산	
10		DELAY	시간 지연 삽입	
11		END	프로세스 수행 종료	종료아이템
12		HEX_TO_ASC	16 진 숫자를 ASCII 문자로 변환	
13		HEX_TO_BCD	16 진 숫자를 BCD 숫자로 변환	
14		INT_TO_STR	정수나 실수의 형태로 표시되어 있는 데이터를 문자값으로 변환	
15		LOOKUP_TABLE	특정 메모리 내용을 Lookup Table 상에서 찾아 대응되는 값으로 변환	
16		MEM_CONVERT	메모리의 특정 데이터를 지정한 값으로 변환	
17		MEM_COPY	메모리간의 데이터복사, 특정 형식 데이터 쓰기	
18		MEM_SET	메모리를 지정한 데이터로 채우기	
19		REMARK	참고-설명문 삽입	
20		RESET	ETOS 를 리셋	
21		STR_TO_INT	문자열을 숫자로 변환	
22		SWAP	메모리의 데이터 위치 교환	
23		TIME_SETTING	메모리의 문자열값으로 현재 시각을 변경	
24		TIME_STR	ETOS 의 시간 정보를 메모리에 저장	
25		TYPE_CONVERT	데이터의 형태를 다른 형태로 변환	
26	조건 분기	ELSE	IF 문의 조건이 맞지 않을 경우 수행	
27		END_IF	IF 문의 종료 명령	종료아이템
28		END_LOOP	반복순환 동작의 종료	
29		IF_BIT_FIND	BIT 열에서 ON/OFF 비트의 위치를 검색	
30		IF_BIT_TEST	메모리의 특정 BIT 값 비교 후 분기	
31		IF_MEM_CMP	두 메모리 값이 동일한지 비교	
32		IF_NUM_CMP	메모리와 메모리/ 상수 값의 대.소 비교	
33		IF_STR_CMP	메모리와 문자열을 비교 후 분기	

번호	분류	아이템	내 용	비 고
34	조건 분기 (계속)	JUMP	현 위치에서 지정한 LABEL 위치로 분기	
35		LABEL	JUMP 명령으로 분기할 위치 지정	
36		LOOP	반복순환 동작의 시작	
37	통신	ENET_CLOSE	이더넷 채널 연결 해제	
38		ENET_DISABLE	이더넷의 드라이버의 통신을 잠시 중지	연결 유지
39		ENET_ENABLE	중단된 이더넷 드라이버 통신을 다시 시작	
40		ENET_OPEN	이더넷 채널 연결 요청	
41		FRAME_SND	참조된 프레임의 송신명령	
42		IF_3964R_READ	Siemens 3964(R) 프로토콜로 읽기 요청 송신	
43		IF_3964R_WRITE	Siemens 3964(R) 프로토콜로 쓰기 요청 송신	
44		IF_MEM_WRITE_RCV	메모리 쓰기 시 IF 문 수행	시작아이템
45		IF_FRAME_RCV	수신 프레임으로 정의된 프레임 수신	시작아이템
46		IF_GLOFA_READ	GLOFA Cnet/Enet 프로토콜로 읽기 요청 송신	
47		IF_GLOFA_WRITE	GLOFA Cnet/Enet 프로토콜로 쓰기 요청 송신	
48		IF_MELSEC_READ	MELSEC 프로토콜로 읽기 요청 송신	
49		IF_MELSEC_WRITE	MELSEC 프로토콜로 쓰기 요청 송신	
50		IF_MODBUS_READ	MODBUS 프로토콜로 읽기 요청 송신	
51		IF_MODBUS_WRITE	MODBUS 프로토콜로 쓰기 요청 송신	
52		IF_REQ_RESP	프레임 송신 후 응답 프레임 수신	
53		MEM_WRITE_RESP_SND	IF_MEM_WRITE_RCV 수행 후 응답 리턴	
54		MODEM_CONTROL	모뎀의 DTR 시그널을 제어	
55	시작	IF_POWER_ON	전원 'ON'시 1 회만 수행	시작아이템
56		IF_TIMER	정주기 실행	시작아이템
57	응용	ENCRYPTION	데이터를 암호화 변환	
58		EVENT_READ_WRITE	IFOR 공유 메모리 중 이벤트 영역 읽기/쓰기	IFOR 전용
59		IF_EVENT_OCCUR	IFOR 공유 메모리 중 이벤트 영역 변경시 이벤트 발생	IFOR 전용
60		IF_IO_STATE_CHANGE	DIO 보드 IO 값 변경시 이벤트 발생	DIO8 전용
61		IO_READ_WRITE	DIO 보드 IO 값 읽기/쓰기	DIO8 전용
62		IP_SAVE	현재 사용중인 IP 를 사용자 메모리에 기록	
63		QUEUE_CLEAR	해당 QUEUE 에 저장된 데이터를 모두 삭제	
64		QUEUE_CREATE	새로운 QUEUE 생성	
65		QUEUE_RCV_FROM	QUEUE 에서 데이터 복사	
66		QUEUE_SND_TO	QUEUE 로 데이터 복사	
67	SLAVE_MEM_READ_WRITE	IFOR 공유 메모리 읽기/쓰기	IFOR 전용	
68	SECS	IF_SECS_MC_FRAME_RCV	수신 프레임으로 정의한 SECS 프레임을 수신	시작아이템
69		IF_SECS_MC_REQ_RESP	SECS 프레임을 송신하고, 정의한 SECS 프레임 수신	
70		SECS_MC_FRAME_SND	SECS-I 이나 HSMS 로 지정한 프레임 송신	

7.5.2. 아이템 일람 (철자순)

번호	아이템	내 용	비 고
1	ARITHMETIC	사칙 연산 기능	
2	ASC_TO_BIN	문자열로 표현된 2 진수를 실제 2 진수로 변환	
3	ASC_TO_HEX	ASCII 문자를 16 진 숫자로 변환	
4	BIN_TO_ASC	2 진수값을 문자열로 표현	
5	BIT_MASK	BIT 단위의 메모리 조작	
6	BIT_WRITE	BIT 단위 데이터 쓰기	
7	BIT_WRITE_EX	BIT 단위 데이터 간접 지정 쓰기	
8	BITS_CPY	BIT 단위 데이터 복사	
9	BITS_OPERATION	메모리의 BIT AND, OR, NOT, XOR 연산	
10	DELAY	시간 지연 삽입	
11	ELSE	IF 문의 조건이 맞지 않을 경우 수행	
12	ENCRYPTION	데이터를 암호화 변환	
13	END	프로세스 수행 종료	종료아이템
14	END_IF	IF 문의 종료 명령	종료아이템
15	END_LOOP	반복순환 동작의 종료	
16	ENET_CLOSE	이더넷 채널 연결 해제	
17	ENET_DISABLE	이더넷의 드라이버의 통신을 잠시 중지(연결 유지)	
18	ENET_ENABLE	중단된 이더넷 드라이버 통신을 다시 시작	
19	ENET_OPEN	이더넷 채널 연결 요청	
20	EVENT_READ_WRITE	IFOR 공유 메모리 중 이벤트 영역 읽기/쓰기	IFOR 전용
21	FRAME_SND	참조된 프레임의 송신명령	
22	HEX_TO_ASC	16 진 숫자를 ASCII 문자로 변환	
23	HEX_TO_BCD	16 진 숫자를 BCD 숫자로 변환	
24	IF_3964R_READ	Siemens 3964(R) 프로토콜로 읽기 요청 송신	
25	IF_3964R_WRITE	Siemens 3964(R) 프로토콜로 쓰기 요청 송신	
26	IF_BIT_FIND	BIT 열에서 ON/OFF 비트의 위치를 검색	
27	IF_BIT_TEST	메모리의 특정 BIT 값 비교 후 분기	
28	IF_MEM_WRITE_RCV	메모리 쓰기 시 IF 문 수행	시작아이템
29	IF_EVENT_OCCUR	IFOR 공유 메모리 중 이벤트 영역 변경시 이벤트 발생	IFOR 전용
30	IF_FRAME_RCV	수신 프레임으로 정의된 프레임 수신	시작아이템
31	IF_GLOFA_READ	GLOFA Cnet/Enet 프로토콜로 읽기 요청 송신	
32	IF_GLOFA_WRITE	GLOFA Cnet/Enet 프로토콜로 쓰기 요청 송신	
33	IF_IO_STATE_CHANGE	DIO 보드 IO 값 변경시 이벤트 발생	DIO8 전용
34	IF_MELSEC_READ	MELSEC 프로토콜로 읽기 요청 송신	
35	IF_MELSEC_WRITE	MELSEC 프로토콜로 쓰기 요청 송신	
36	IF_MEM_CMP	두 메모리 값이 동일한지 비교	

번호	아이템	내 용	비 고
37	IF_MODBUS_READ	MELSEC 프로토콜로 읽기 요청 송신	
38	IF_MODBUS_WRITE	MELSEC 프로토콜로 쓰기 요청 송신	
39	IF_NUM_CMP	메모리와 메모리/ 상수 값의 대.소 비교	
40	IF_POWER_ON	전원 'ON'시 1 회만 수행	시작아이템
41	IF_REQ_RESP	프레임 송신 후 응답 프레임 수신	
42	IF_SECS_MC_FRAME_RCV	수신 프레임으로 정의한 SECS 프레임을 수신	시작아이템
43	IF_SECS_MC_REQ_RESP	SECS 프레임을 송신하고, 정의한 SECS 프레임 수신	SECS 전용
44	IF_STR_CMP	메모리와 문자열을 비교 후 분기	
45	IF_TIMER	정주기 실행	시작아이템
46	INT_TO_STR	정수나 실수의 형태로 표시되어 있는 데이터를 문자값으로 변환	
47	IO_READ_WRITE	DIO 보드 IO 값 읽기/쓰기	DIO8 전용
48	IP_SAVE	현재 사용중인 IP 주소를 사용자 메모리에 기록	
49	JUMP	현 위치에서 지정한 LABEL 위치로 분기	
50	LABEL	JUMP 명령으로 분기할 위치 지정	
51	LOOKUP_TABLE	특정 메모리 내용을 Lookup Table 상에서 찾아 대응되는 값으로 변환	
52	LOOP	반복순환 동작의 시작	
53	MEM_CONVERT	메모리의 특정 데이터를 지정한 값으로 변환	
54	MEM_COPY	메모리간의 데이터복사, 특정 형식 데이터 쓰기	
55	MEM_SET	메모리에 임의의 데이터로 채우기	
56	MEM_WRITE_RESP_SND	IF_MEM_WRITE_RCV 수행 후 응답 리턴	
57	MODEM_CONTROL	모뎀의 DTR 시그널을 제어	
58	QUEUE_CLEAR	해당 QUEUE 에 저장된 데이터를 모두 삭제	
59	QUEUE_CREATE	새로운 QUEUE 생성	
60	QUEUE_RCV_FROM	QUEUE 에서 데이터 복사	
61	QUEUE_SND_TO	QUEUE 로 데이터 복사	
62	REMARK	참고-설명문 삽입	
63	RESET	ETOS 를 리셋	
64	SECS_MC_FRAME_SND	SECS-I 이나 HSMS 로 지정한 프레임 송신	SECS 전용
65	SLAVE_MEM_READ_WRITE	IFOR 공유 메모리 읽기/쓰기	IFOR 전용
66	STR_TO_INT	문자열을 숫자로 변환	
67	SWAP	메모리의 데이터 교환	
68	TIME_SETTING	메모리의 문자열값으로 현재 시각을 변경	
69	TIME_STR	ETOS 의 시간 정보를 메모리에 저장	
70	TYPE_CONVERT	데이터의 형태를 다른 형태로 변환	

7.5.3. 데이터 형식

데이터 유형 별로 10 진수로 표현했을 최소 최대 범위 입니다.

표 7.5-1 형 별 데이터 범위

Key-word	데이터 형	비트	데이터 범위
Signed 1 Byte	Short Integer	8	-128 ~ 127
Unsigned 1 Byte	Unsigned Short Integer	8	0 ~ 255
Signed 2 Byte	Integer	16	-32768 ~ 32767
Unsigned 2 Byte	Unsigned Integer	16	0 ~ 65535
Signed 4 Byte	Double Integer	32	-2147483648 ~ 2147483647
Unsigned 4 Byte	Unsigned double Integer	32	0 ~ 4294967295
Float	Real Number	32	-1.797693e38 ~ -1.401298e-45 1.401298e-45 ~ 3.402823e38
Double	Long Real Number	64	-1.7976931e308 ~ -4.9406564e-324 4.9406564e-324 ~ 1.7976931e308

7.6. 아이템 설명

7.6.1. ARITHMETIC

- ▶ 기본 아이템 (*8 4)
- ▶ ETOS-50/100/150/200/1000 (*8 5)

7.6.1.1. 기능

사칙연산(+, -, *, /)과 모듈러스(Modulus) 연산(%)을 수행합니다. <연산자>에서 연산 기능을 선택하고 데이터의 형태 및 연산할 데이터의 형식을 지정하면 지정된 연산 방식에 따라 데이터 1 과 데이터 2 의 값을 연산하여 결과값을 결과(주소)에 저장합니다.

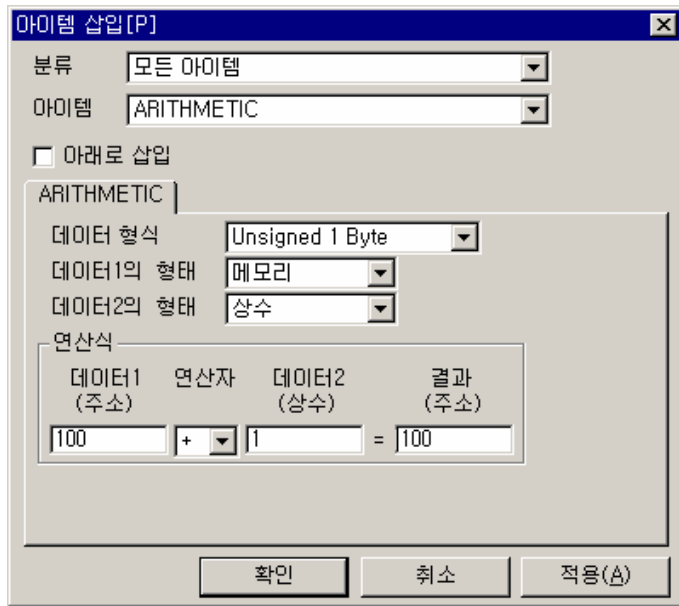


그림 7.6-1 ARITHMETIC 입력 예

표 7.6-1 ARITHMETIC 항목 설명

항 목	내 용
데이터 형식	데이터 1 과 데이터 2 의 형식을 지정 (*8 6)
데이터 1 의 형태	메모리/상수 중 하나를 선택, 데이터 1 의 입력값이 선택에 따라 메모리 주소나 상수 값으로 지정
데이터 2 의 형태	메모리/상수 중 하나를 선택, 데이터 2 의 입력값이 선택에 따라 메모리 주소나 상수 값으로 지정
데이터 1	데이터 1 의 형태 지정에 따라 사용자 메모리 바이트 주소나 상수
연산자	+(더하기), -(빼기), /(나누기), *(곱하기), %(나머지) 중 지정
데이터 2	데이터 2 의 형태 지정에 따라 사용자 메모리 바이트 주소나 상수
결과(주소)	연산 결과를 저장할 ETOS 의 사용자 메모리 바이트 주소를 지정(*8 7)

(*8 4) 아이템 종류

(*8 5) 아이템을 지원하는 제품명(또는 모델명)

(*8 6) 7.5.3 데이터 형식 참조

(*8 7) 특별한 언급이 없으면 주소는 모두 ETOS 의 사용자 메모리 바이트 주소를 가리킴

7.6.1.2. 예제

메모리 100 번지의 값을 0 에서 255 까지 1 초 주기로 1 씩 증가시킵니다.

▶ **예제 아이템 등록**

1 초 주기로 증가해야 하므로, 정주기 아이템인 IF_TIMER 를 먼저 추가합니다. <주기>는 “100” (100 X 10ms = 1000ms, 1 초)을 입력합니다. IF_TIMER 와 END_IF 사이에 ARITHMETIC 아이템을 삽입합니다. 이제 ARITHMETIC 아이템은 1 초에 한 번씩 실행됩니다.

ARITHMETIC 아이템의 각 항목은 다음과 같이 입력합니다.

항 목	입력 내용
데이터 형식	Unsigned 1 byte
데이터 1 의 형태	메모리
데이터 2 의 형태	상수
데이터 1	100
연산자	+(더하기)
데이터 2	1
결과(주소)	100

100 번지 값에 1 을 더해 다시 100 번지에 저장하게 됩니다. 초기값은 0 이며, 최대값인 255 에 다시 1 을 더하면 0 이 됩니다. 따라서, 100 번지가 0 에서 255 까지 1 초 주기로 증가하게 됩니다.

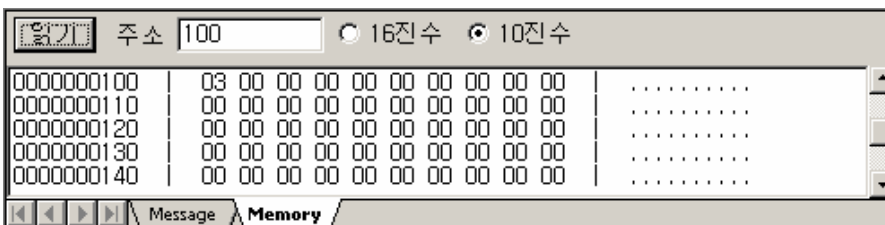
▶ **예제 프로세스**

지금까지 작성한 예제 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	100
01	ARITHMETIC	Type:U1, M100, +, 1, M100
02	END_IF	

▶ **결과**

프로그램 쓰기 후, 100 번지 주소가 1 초 주기로 0 ~ 255 까지 1 씩 증가하는 것을 볼 수 있습니다.



7.6.2. ASC_TO_BIN

- ▶ 기본 아이템 (*88)
- ▶ ETOS-50/100/150/200/1000 (*89)
- ▶ [BIN_TO_ASC](#) (*90)

7.6.2.1. 기능

2 진수(Binary) 문자열("11010011")을 역순의 실제 2 진수 값으로 변환합니다. 이때, 정수의 비트 위치가 역순으로 바뀝니다.

- ▶ "0101" -> 1010_B (= 10), 비트 위치가 역순으로 바뀜에 유의

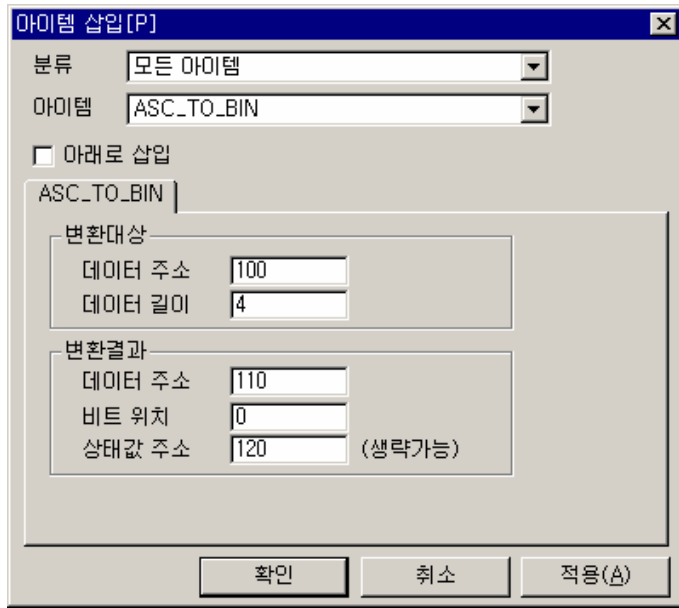


그림 7.6-2 ASC_TO_BIN 입력 예

표 7.6-2 ASC_TO_BIN 항목 설명

항 목		내 용
변환 대상	데이터 주소	변환할 데이터가 저장되어 있는 메모리 바이트 주소를 지정
	데이터 길이	변환할 문자열의 길이, 입력 범위는 1~126
변환 결과	데이터 주소	변환 결과 데이터를 저장할 주소를 지정합니다.
	비트 위치	결과 주소 몇 번째 비트부터 데이터를 저장할 지를 지정합니다.
	상태값 주소	변환 상태 값을 저장할 메모리의 주소 지정, 오류가 발생할 경우 0 이 저장되며, 오류가 없을 경우 변환한 문자열의 길이를 저장 (2 바이트 사용)(*91)

(*88) 아이템 종류

(*89) 아이템을 지원하는 제품명(또는 모델명)

(*90) 관련 아이템

(*91) 4 바이트를 변환한 경우, 저장값은 04 00 으로 저장됩니다. (00 04 가 아님)

7.6.2.2. 예제

문자열 “0101”을 2 진수 1010_B (= 0A_H, 10)으로 변환하여 110 번지에 저장합니다.

▶ **예제 아이템 등록**

[IF_POWER_ON](#) 아이템을 추가합니다.

먼저 변환될 문자열을 준비해야 합니다. MEM_SET 아이템을 삽입하고, <데이터 형태>에 ‘문자열’, <데이터>에 “0101”, <주소>에 “100”, <길이>에 “4”를 입력합니다.

이제 ASC_TO_BIN 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목		입력 내용
변환 대상	데이터 주소	100
	데이터 길이	4
변환 결과	데이터 주소	110
	비트 위치	0
	상태값 주소	120

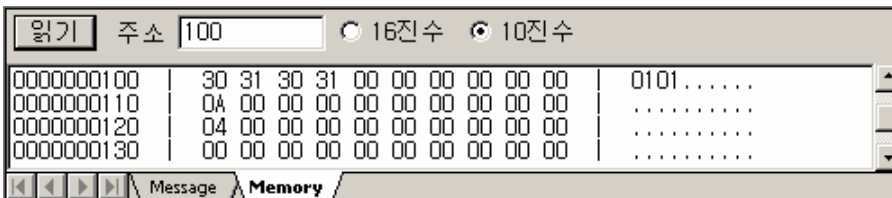
▶ **예제 프로세스**

편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:4, Type:문자열, Data:0101
02	ASC_TO_BIN	S:M100, Size:4, T:M110, BitPos:0, Status:M120
03	END_IF	

▶ **결과**

MEM_SET 아이템에 의해 100 번지에 30313031_H (문자열 = “0101”)이 표시됩니다. 110 번지에는 ASC_TO_BIN 아이템에 의해 정수로 변환된 값(0A_H = 1010_B)이 표시됩니다. 앞에서 강조했듯이, 각 비트의 위치가 역순으로 바뀔에 주의하시기 바랍니다. (0101_B 이 아니라 1010_B 으로 변환됩니다), 120 번지 상태값 주소에는 변환한 문자열의 길이가 저장되어 있습니다.(04 00)



7.6.3. ASC_TO_HEX

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [HEX_TO_ASC](#)

7.6.3.1. 기능

16 진수 문자열('1', ~ '9', 'A' ~ 'F' 로 이루어진 문자열)을 16 진수 값으로 변환합니다. 변환할 영역 내의 문자가 16 진수를 나타내는 문자가 아니면 변환을 하지 않습니다.

- ▶ "FA34"(46413334_H) -> FA34_H

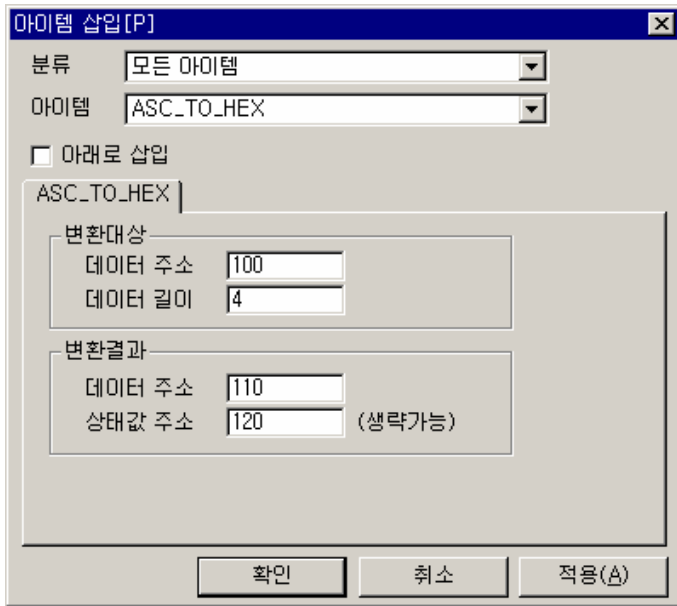


그림 7.6-3 ASC_TO_HEX 입력 예

표 7.6-3 ASC_TO_HEX 항목 설명

항 목		내 용
변환 대상	데이터 주소	변환할 데이터가 저장되어 있는 메모리 주소를 지정
	데이터 길이	변환할 문자열의 길이, 입력 범위는 2~32768 문자열 2 바이트 -> 정수 1 바이트로 변환하므로 항상 짝수로 입력
변환 결과	데이터 주소	변환 결과 저장 주소, 변환된 값은 길이가 절반으로 줄어듭니다.
	상태 값 주소	변환 상태 값을 저장할 메모리의 주소, 오류가 없을 경우에는 변환된 데이터의 길이가 저장 (2 바이트 사용) ^(*92)

(*92) 변환 결과의 길이가 저장됩니다. 위의 예에서는 2 바이트의 16 진수로 변환되었으므로 4 가 아닌 2 가 저장됩니다.

7.6.3.2. 예제

16 진수 문자열 “0123” (30313233_H)을 16 진수값 (0123_H)으로 변환합니다.

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 추가합니다.

먼저 변환할 문자열을 준비합니다. MEM_SET 아이템을 삽입하고, <데이터 형태>에 ‘문자열’, <데이터>에 “0123”, <주소>에 “100”, <길이>에 “4”를 입력합니다.

이제 ASC_TO_HEX 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목		입력 내용
변환 대상	데이터 주소	100
	데이터 길이	4
변환 결과	데이터 주소	110
	상태 값 주소	120

▶ **예제 프로세스**

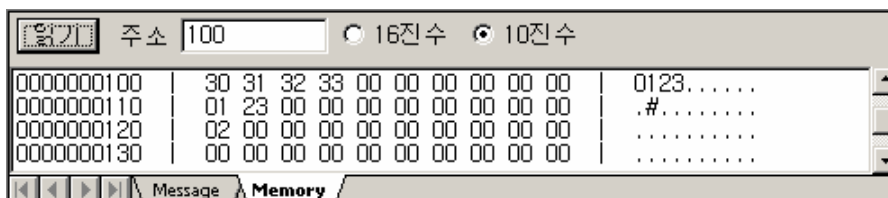
편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:4, Type:문자열, Data:0123
02	ASC_TO_HEX	S:M100, Size:4 T:M110, Status:M120
03	END_IF	

▶ **결과**

100 번지에 16 진수 문자열 “0123” (30313233_H)이 입력되어 있습니다. (우측에 문자로 0123 으로 표시)

110 번지에는 16 진수 0123_H 으로 변환된 결과를 볼수 있습니다. 4 바이트 문자열이 2 바이트 16 진수 값으로 바뀌었습니다. 변환된 16 진수의 길이는 120 번지에 표시됩니다. (02 00)



7.6.4. BIN_TO_ASC

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [ASC_TO_BIN](#)

7.6.4.1. 기능

2 진수(Binary)를 역순의 2 진수 문자열 (“10011011”) 형태로 변환합니다.

ASC_TO_BIN 아이템과 반대 기능의 아이템으로 역시 문자열의 각 비트 위치가 역순으로 바뀝니다.

- ▶ 0101_B (= 5) -> “1010”, 비트 위치가 역순으로 바뀔에 유의

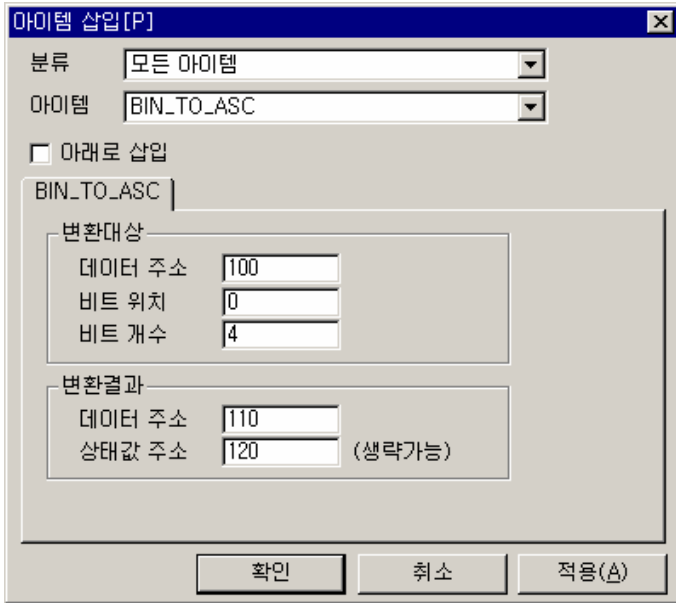


그림 7.6-4 ASC_TO_BIN 입력 예

표 7.6-4 ASC_TO_BIN 항목 설명

항 목	내 용	
변 환 대상	데이터 주소	변환할 데이터가 저장되어 있는 메모리 주소 지정
	비트 위치	변환할 데이터 주소의 몇 번째 비트부터 변환할 지 지정
	비트 개수	변환할 비트의 개수를 지정
변 환 결과	데이터 주소	변환된 데이터가 저장될 메모리의 주소를 지정, 저장될때 비트의 순서는 역순으로 저장
	상태값 주소	변환 상태 값을 저장할 메모리의 주소를 지정, 오류가 없을 경우 변환한 비트 개수를 저장 (2 바이트 사용)

7.6.4.2. 예제

100 번지에 저장되어 있는 16 진수값(5, 0101_B)을 역순의 2 진수 문자열로 변환하여 110 번지에 저장합니다.

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 추가합니다.

먼저 100 번지에 16 진수값 5 를 입력해야 합니다. MEM_SET 아이템을 삽입하고, <데이터 형태>는 '16 진수', <데이터>는 "05", <주소>는 "100", <길이>는 "1"을 입력합니다.

BIN_TO_ASC 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목		입력 내용
변환 대상	데이터 주소	100
	비트 위치	0
	비트 개수	4
변환 결과	데이터 주소	110
	상태값 주소	120

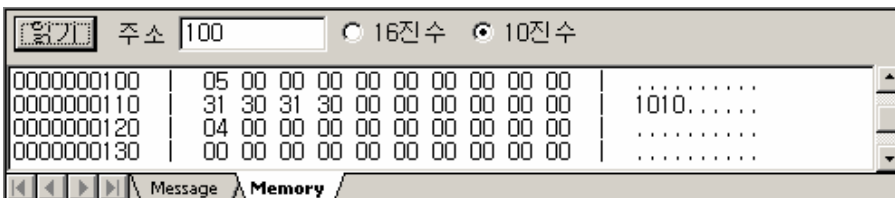
▶ **예제 프로세스**

편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:1, Type:16진수, Data:05
02	BIN_TO_ASC	S:M100, BitPos:0, BitSize:4, T:M110, Status:M120
03	END_IF	

▶ **결과**

MEM_SET 아이템에 의해 100 번지에 05 (= 0101_B)가 입력되어 있습니다. 110 번지에는 역순의 2 진수 문자열 "1010" (= 31303130_H) 이 변환 되어 있습니다. 상태값 주소 위치인 120 번지에는 변환한 비트의 개수가 저장됩니다. (04 00)



7.6.5. BIT_MASK

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [BITS_OPERATION](#)

7.6.5.1. 기능

지정한 주소의 데이터에 지정한 패턴으로 비트 마스크(AND/OR/XOR)를 씩습니다.

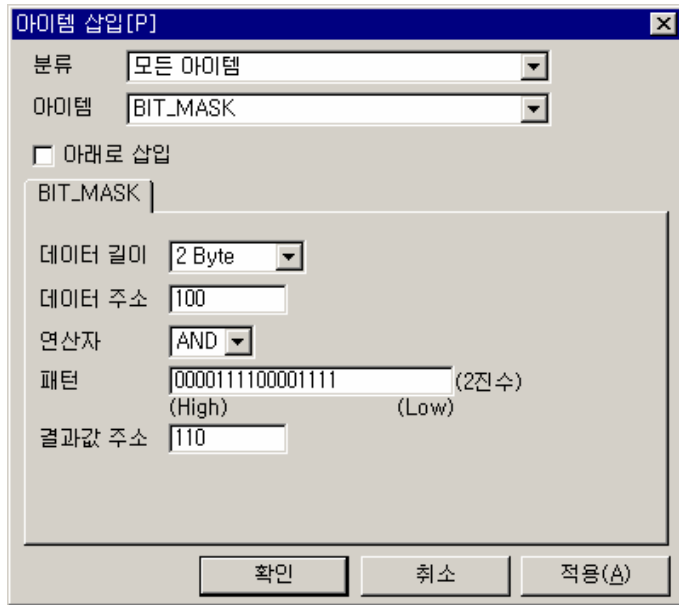


그림 7.6-5 BIT_MASK 입력 예

표 7.6-5 BIT_MASK 항목 설명

항 목	내 용
데이터 길이	변환할 데이터의 길이를 지정(1/2/4 Byte 중 선택), 데이터 길이와 패턴의 길이가 일치해야 합니다. 1BYTE = 8BIT
데이터 주소	비트 마스크 대상이 되는 데이터가 저장되어 있는 주소 지정합니다.
연산자	AND/OR/XOR 중 선택
패턴	비트 마스크 패턴을 2 진수로 입력합니다. <데이터 길이>에서 지정한 데이터의 길이만큼 입력해야 합니다. (최소 8 비트 ~ 최대 32 비트)
결과값 주소	연산 결과를 저장할 주소 지정합니다.

7.6.5.2. 예제

100 번지의 값 (A55A_H)를 패턴 '0000111100001111'(0F0F_H)로 AND 마스크를 해서 110 번지에 저장합니다.

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 추가합니다.

MEM_SET 아이템을 삽입하고 <데이터 형태>는 '16 진수', <데이터>는 "A55A", <주소>는 "100" 번지, <길이>는 "2"를 각각 입력 합니다.

이제 BIT_MASK 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
데이터 길이	2Byte 선택
데이터 주소	100
연산자	AND 선택
패턴	0000111100001111
결과값 주소	110

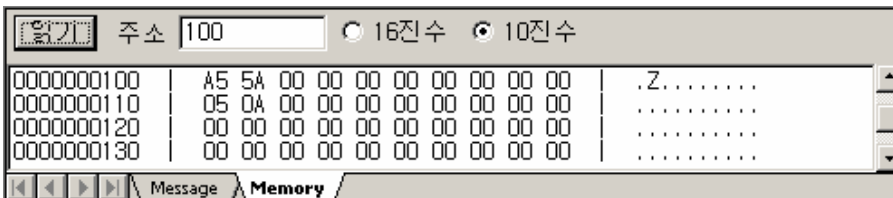
▶ **예제 프로세스**

편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:2, Type:16진수, Data:A55A
02	BIT_MASK	M100, AND, 0000111100001111, M110
03	END_IF	

▶ **결과**

100 번지에는 MEM_SET 아이템에 의해 A5 5A 가 입력되어 있습니다. 110 번지에는 BIT_MASK 결과 상위 4 비트는 0 이 되고, 하위 4 비트씩만 값이 유지되었습니다. (050A_H)



7.6.6. BIT_WRITE

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [BIT_WRITE_EX](#)

7.6.6.1. 기능

지정한 주소의 특정 비트를 온(on),오프(off)합니다. 비트의 위치는 바이트 주소와 비트 위치로 표현합니다. 예를 들어 100 번째 첫번째 비트는 100.0 또는 바이트 주소 100, 비트 위치 0으로 표현할 수 있습니다.

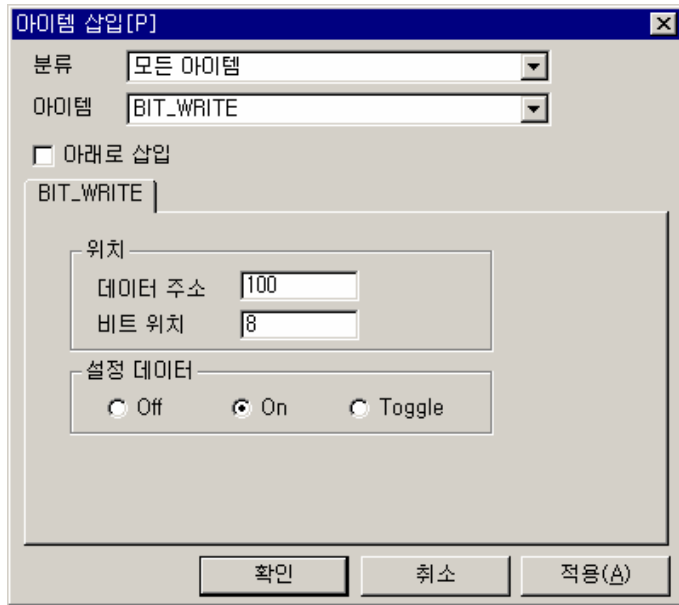


그림 7.6-6 BIT_WRITE 입력 예

표 7.6-6 BIT_WRITE 설정 항목

항 목	내 용
데이터 주소	비트 On/Off 할 메모리의 바이트 번지를 입력
비트 위치	위의 주소 항목에서 지정한 메모리의 몇 번 비트를 On/Off 할지 지정 ^(*93) 비트 위치는 0에서 시작하며, 하위 비트부터 카운트 됩니다. 예를 들어, 위와 같이 100 번째/8 번 비트이면, 101 번째 바이트의 1 번째(0 번)를 선택한 것입니다.
데이터	해당 비트의 결과값을 지정 Off -> 해당 비트 0 On -> 해당 비트 1 Toggle -> 해당 비트 반전 (현재값의 반대값으로 변경)

(*93) 비트 위치는 LSB(Least Significant Bit)가 0 번 비트이고, MSB(Most Significant Bit)가 7 번 비트입니다. 따라서 한 바이트는 하위 비트부터 차례로 0,1,2,...,7 번 비트가 됩니다. 8 번 비트는 그 다음 바이트의 0 번 비트가 됩니다.

7.6.6.2. 예제

ETOS 메모리 101 번지 첫 번째 비트를 On 시킵니다.

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.

특정 비트를 선택하는 주소 + 비트 위치 조합은 무수히 많습니다.

(100 번지 + 8 번 비트, 101 번지 + 0 번 비트 등)

BIT_WRITE 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
데이터 주소	100
비트 위치	8
데이터	On

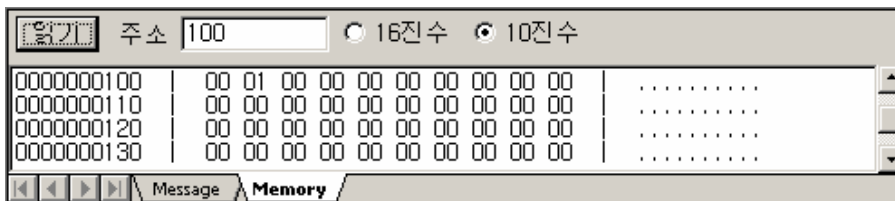
▶ 예제 프로세스

편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	BIT_WRITE	T:M100, BitPos:8, Data:On
02	END_IF	

▶ 결과

101 번지 첫 번째 비트가 On 되었습니다.



7.6.7. BIT_WRITE_EX

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [BIT_WRITE](#)

7.6.7.1. 기능

지정한 주소의 특정 비트를 온(on),오프(off)합니다. BIT_WRITE 와 동일한 기능에 주소나 비트 위치 지정시 간접 지정이 가능하도록 한 아이템입니다.

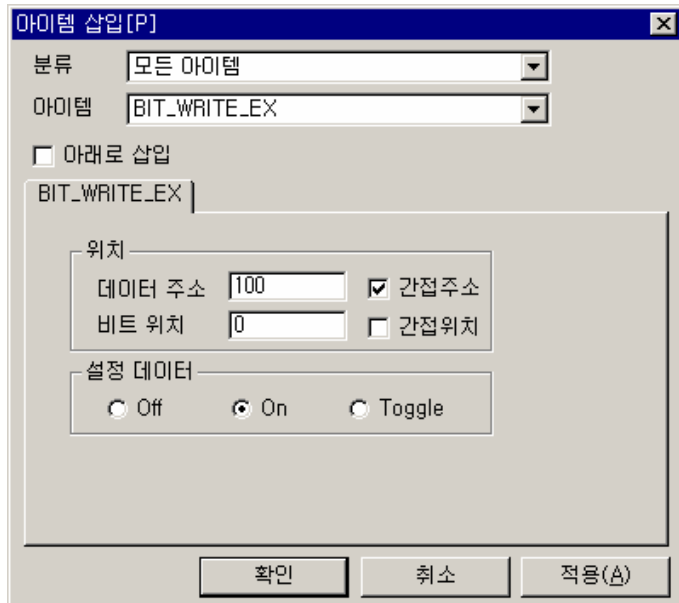


그림 7.6-7 BIT_WRITE_EX 입력 예

표 7.6-7 BIT_WRITE_EX 설정 항목

항 목	내 용
데이터 주소	비트 On/Off 할 메모리의 바이트 번지를 입력
간접주소	주소에 저장된 데이터값이 On/Off 할 메모리의 주소가 됩니다. (4 바이트 사용)
비트 위치	위의 주소 항목에서 지정한 메모리의 몇 번 비트를 On/Off 할지 지정 ^(*94) 비트위치는 0 에서 시작하며, 하위 비트부터 카운트 됩니다. 예를 들어, 위와 같이 100 번지/0 번 비트이면, 100 번지 바이트의 1 번째(0 번)를 선택한 것입니다.
간접위치	비트위치는 더 이상 비트 위치를 나타내지 않고, 메모리의 바이트 주소를 의미합니다. 해당 주소의 데이터가 비트 위치가 됩니다. (2 바이트 사용)
데이터	해당 비트의 결과값을 지정 Off -> 해당 비트 0 On -> 해당 비트 1 Toggle -> 해당 비트 반전 (현재값의 반대값으로 변경)

(*94) 비트 위치는 LSB(Least Significant Bit)가 0 번 비트이고, MSB(Most Significant Bit)가 7 번 비트 입니다. 따라서 한 바이트는 하위 비트부터차례로 0,1,2,...,7 번 비트가 됩니다. 8 번 비트는 그 다음 바이트의 0 번 비트가 됩니다.

7.6.7.2. 간접 지정

간접 지정에 대해서 알아보겠습니다.

앞 그림의 주소 항에 100 을 입력하고, 간접 주소를 체크한 경우를 가정하겠습니다. 이 경우 BIT_WRITE_EX 의 대상이 되는 주소는 100 번지가 아니고, 100 번지의 데이터가 가리키는 주소가 됩니다. 다음 그림을 참고하십시오.

주소를 간접 지정하는 경우 100 번지부터 4 바이트를 확인해야 합니다. (간접주소는 4 바이트로 표시되고, 간접크기는 2 바이트로 표시됩니다.) 4 바이트의 값이 아래 그림과 같다고 가정하면, 주소는 0x0000 0100 = 256 번지가 됩니다. 따라서, BIT_WRITE_EX 의 대상이 되는 주소는 100 번지가 아니라 256 번지가 됩니다. 직접 지정하는 경우와 달리, 특정 번지의 데이터값에 따라 여러 위치를 가리킬 수 있게 됩니다. 즉, 동작 중에 아이템의 실행 대상이 되는 주소를 프로그램에서 조정할 수 있게 됩니다.

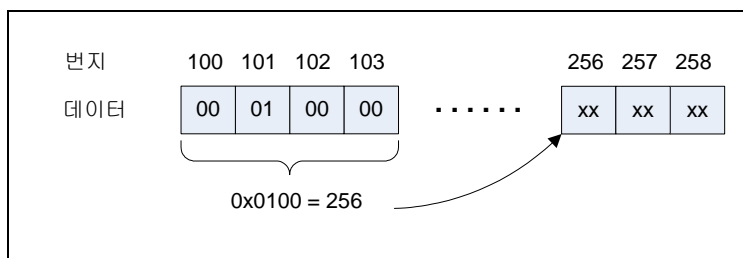


그림 7.6-8 주소 간접 지정

7.6.7.3. 예제

ETOS 메모리 100 번지의 데이터가 가리키는 번지의 0 번 비트를 ON 시킵니다. (간접주소를 사용하는 경우는 해당 번지의 데이터가 변하는 경우이나, 예에서는 그 값이 고정되어 있는것으로 합니다.)

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.

MEM_SET 아이템을 삽입하고 <데이터 형태>는 '16 진수', <데이터>는 "C8", <주소>는 "100" 번지, <길이>는 "1"을 각각 입력 합니다.

이제 BIT_WRITE_EX 를 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
데이터 주소	100
간접주소	체크
비트 위치	0
간접위치	체크안함
데이터	On 선택

▶ 예제 프로세스

편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	100
01	MEM_SET	T:M100, Size:1, Type:16진수, Data:C8
02	BIT_WRITE_EX	T:M100, BitPos:0, Data:On
03	END_IF	

▶ 결과

먼저 100 번지의 데이터가 가리키는 번지를 확인해 봅니다. 간접 주소 지정은 앞서도 설명한 바와 같이 지정 번지부터 4 바이트를 주소를 나타내는 데이터로 취급해야 합니다. 4 바이트의 데이터는 100 번지부터 차례대로 C8 00 00 00 이고, 실제 값은 순서를 뒤집은 00 00 00 C8 입니다. (000000C8_H = 200)

따라서 0 번 비트를 On 하는 대상은 100 번지가 아니라 200 번지가 됨을 알 수 있습니다. 0 번 비트, 즉 첫번째 비트를 On 하면 그 값은 01 이 되어야 합니다.

간접 주소 지정:
 0x0000 00C8 = 200번지

주소 100 16진수 10진수

0000000100	C8 00 00 00	00 00 00 00 00 00
0000000110	00 00 00 00	00 00 00 00 00 00
0000000120	00 00 00 00	00 00 00 00 00 00
0000000130	00 00 00 00	00 00 00 00 00 00
0000000140	00 00 00 00	00 00 00 00 00 00
0000000150	00 00 00 00	00 00 00 00 00 00
0000000160	00 00 00 00	00 00 00 00 00 00
0000000170	00 00 00 00	00 00 00 00 00 00
0000000180	00 00 00 00	00 00 00 00 00 00
0000000190	00 00 00 00	00 00 00 00 00 00
0000000200	01 00 00 00	00 00 00 00 00 00
0000000210	00 00 00 00	00 00 00 00 00 00

200번지 0번 비트 On

7.6.8. BITS_CPY

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [MEM_CPY](#)

7.6.8.1. 기능

비트 단위로 데이터를 복사합니다.

비트의 위치는 바이트 주소와 비트 위치로 표현합니다. 예를 들어 100 번째 첫번째 비트는 100.0 또는 바이트 주소 100, 비트 위치 0 으로 표현할 수 있습니다. 간접 지정에 대한 설명은 BIT_WRITE_EX 의 [7.6.7.2. 간접 지정](#)을 참조하시기 바랍니다.

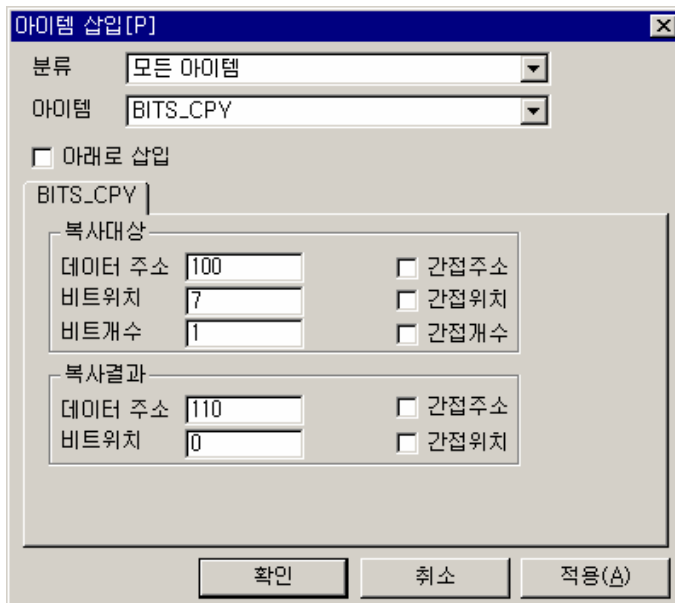


그림 7.6-9 BITS_CPY 입력 예

표 7.6-8 BITS_CPY 설정 항목

항 목	내 용	
복사 대상	데이터 주소	복사 대상의 바이트 번지를 입력합니다.
	간접주소	해당 주소에 저장된 데이터가 번지가 됩니다. (4 바이트)
	비트 위치	위의 바이트 주소의 몇 번 비트인지를 지정합니다. 하위 1 번째 비트가 0 번 비트이며, 최상위 비트는 7 번 비트가 됩니다.
	간접위치	비트위치는 바이트 주소가 되며, 바이트 주소의 데이터 값이 비트위치가 됩니다. (2 바이트)
	비트개수	복사할 비트의 수를 입력합니다.
	간접개수	비트개수는 바이트 주소가 되며, 바이트 주소의 데이터 값이 비트개수가 됩니다. (2 바이트)

표 7.6-8 BITS_CPY 설정 항목 (계속)

항 목		내 용
복사 결과	데이터 주소	복사되어 저장될 주소를 입력합니다.
	간접주소	해당 주소에 저장된 데이터가 번지가 됩니다. (4 바이트)
	비트위치	저장 주소가 위의 바이트 주소의 몇 번 비트인지를 지정합니다.
	간접위치	비트위치는 바이트 주소가 되며, 바이트 주소의 데이터 값이 비트위치가 됩니다. (2 바이트)

7.6.8.2. 예제

ETOS 메모리 100 번지의 최상위 비트인 7 번 비트를 110 번 0 번 비트에 복사합니다.

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 추가합니다.

먼저, BIT_WRITE 아이템으로 100 번지 7 번 비트를 On 시킵니다.

이제 BITS_CPY 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목		입력 내용
복사 대상	데이터 주소	100
	비트 위치	7
	비트개수	1
복사 결과	데이터 주소	110
	비트위치	0

간접주소, 간접위치, 간접개수는 모두 선택하지 않습니다.

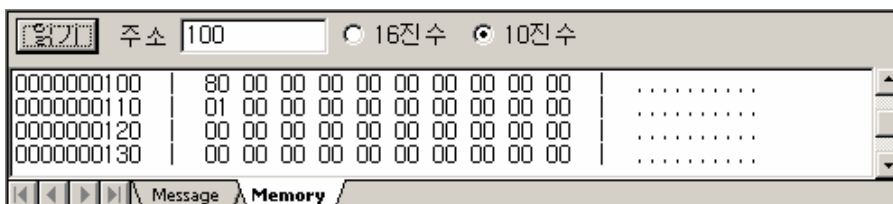
▶ **예제 프로세스**

편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	BIT_WRITE	T:M100, BitPos:7, Data:On
02	BITS_CPY	S:M100/7, BitNo:1, T:M110/0
03	END_IF	

▶ **결과**

100 번지 데이터는 80_H 로 최상위 비트가 On 되어 있습니다. 최상위 비트인 7 번 비트를 110 번지 0 번 비트에 복사한 결과, 110 번지 값이 01 이 되었습니다.



7.6.9. BITS_OPERATION

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [BIT_MASK](#)

7.6.9.1. 기능

<데이터 1 주소>와 <데이터 2 주소>에 있는 데이터를 AND, OR, XOR, NOT 연산을 하여 결과값 주소에 저장합니다.

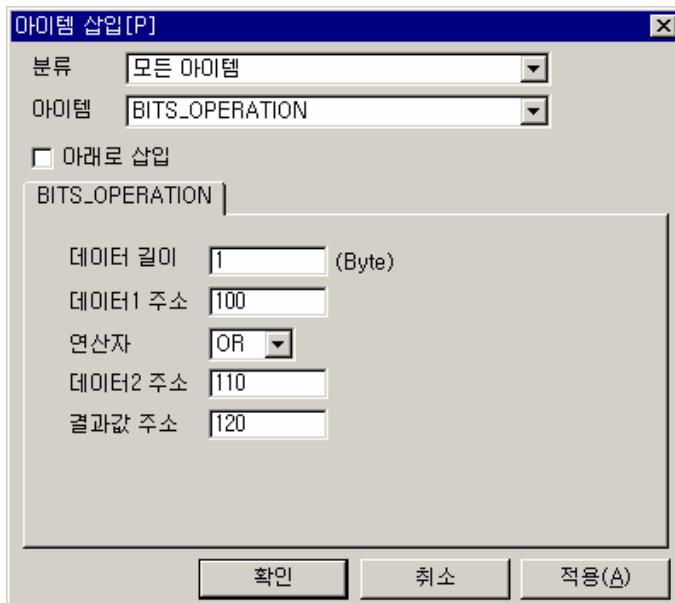


그림 7.6-10 BITS_OPERATION 입력 예

표 7.6-9 BITS_OPERATION 설정항목

항 목	내 용
데이터 길이	데이터 1 과 데이터 2 의 바이트 길이 입니다.
데이터 1 주소	연산 대상 1 의 주소를 입력합니다.
연산자	데이터 1 과 데이터 2 를 해당 연산자로 연산합니다. (AND, OR, XOR, NOT)
데이터 2 주소	연산 대상 2 의 주소를 입력합니다.
결과값 주소	결과값을 저장할 주소를 입력합니다.

7.6.9.2. 예제

ETOS 메모리 100 번지 1 바이트와 110 번지 1 바이트를 OR 연산하여 120 번지에 저장합니다.

▶ **예제 아이템 등록**

MEM_SET 아이템으로 100, 110 번지에 각각 16 진수로 01 과 02 를 입력합니다.

이제 BITS_OPERATION 을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
데이터 길이	1
데이터 1 주소	100
연산자	OR
데이터 2 주소	110
결과값 주소	120

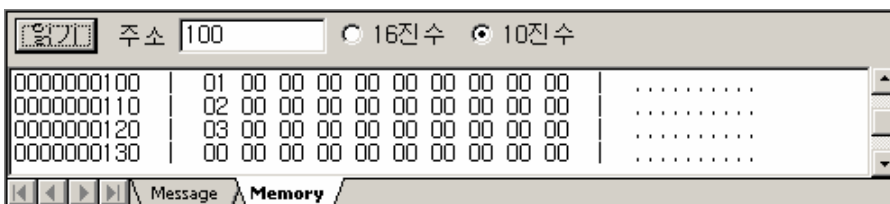
▶ **예제 프로세스**

편집을 마친 프로세스 화면입니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:1, Type:16진수, Data:01
02	MEM_SET	T:M110, Size:1, Type:16진수, Data:02
03	BITS_OPERATION	1Byte, M100, OR, M110, M120
04	END_IF	

▶ **결과**

120 번지에는 100 번지 01 과 110 번지 02 의 OR 값인 03 이 저장됩니다.



7.6.10. DELAY

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.10.1. 기능

지정한 시간 동안 프로세스 수행을 지연합니다.

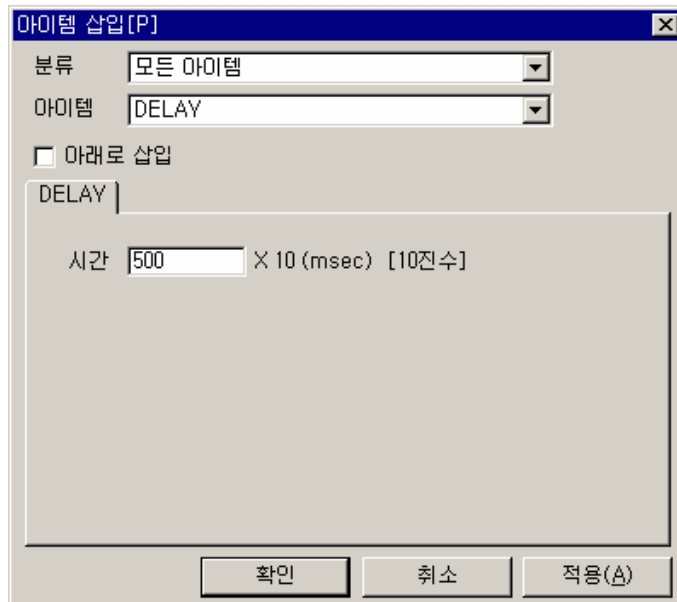


그림 7.6-11 DELAY 입력 예

표 7.6-10 DELAY 설정 항목

항 목	내 용
시간	지연 시간을 입력합니다. 입력 가능한 범위는 1~4294967295(4Byte)이며, 10msec 를 기본 단위로 합니다.(예: 100 ->100*10msec=1 초)

7.6.10.2. 예제

ETOS 메모리 100 번지 값을 10 초 주기로 1 씩 증가시킵니다.

▶ 예제 아이템 등록

먼저 시작아이템으로 IF_TIMER 를 설정하고, <주기>는 “500”으로 설정합니다.

DELAY 아이템을 삽입하고, <시간>은 “500” (500 X 10ms = 5 초)을 입력합니다.

ARITHMETIC 아이템을 삽입하고, 100 번지에 1 을 더해 100 번지에 저장하도록 입력합니다.

▶ 예제 프로세스

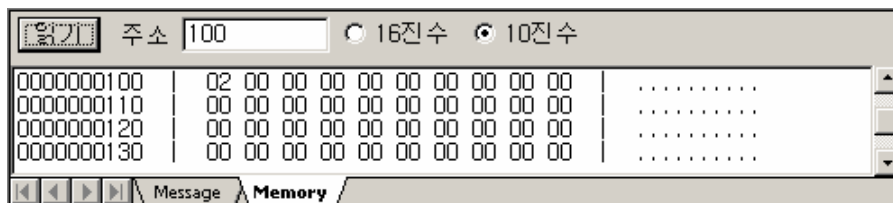
편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	500
01	DELAY	500
02	ARITHMETIC	Type:S1, M100, +, 1, M100
03	END_IF	

▶ 결과

IF_TIMER 에 의해 5 초 주기로 프로세스가 수행되리라고 예측할 수 있습니다. 그러나, IF_TIMER 다음의 아이템들을 수행하는데 5 초 이상 걸리는 경우, 프로세스는 그 수행 시간을 더한 주기로 실행됩니다. 예에서는 IF_TIMER 5 초, DELAY 5 초가 걸리므로, 프로세스는 10 초 주기로 실행됩니다.

100 번지의 값이 10 초 주기로 증가하는 것을 확인할 수 있습니다.



7.6.11. ELSE

- ▶ 조건 분기 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.11.1. 기능

IF 아이템과 함께 사용되어 IF 조건이 만족하지 못할 경우 ELSE 아이템 이후부터 수행하게 하는 분기 명령입니다. ELSE 아이템은 IF 아이템 중 프로세스 선두에 오는 시작 아이템의 IF 문에는 사용할 수 없습니다.

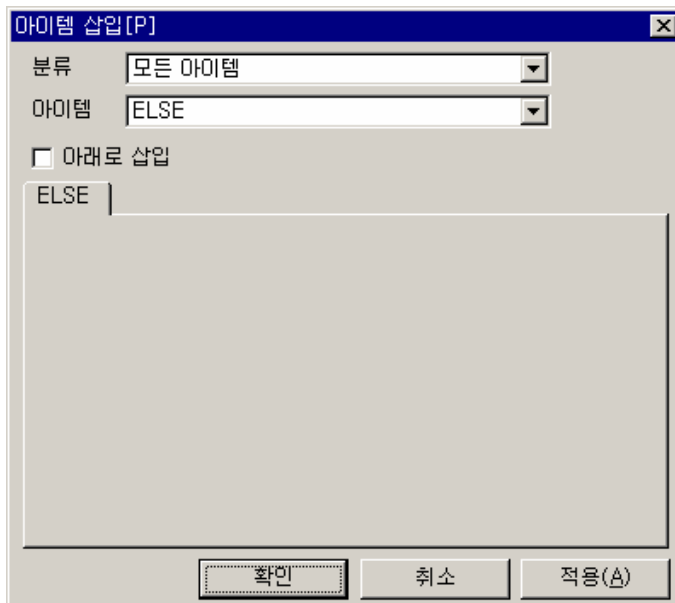


그림 7.6-12 ELSE 아이템

표 7.6-11 ELSE 설정 항목

항 목	내 용
없음	설정 항목이 없습니다

7.6.11.2. 예제

ETOS 메모리 100 번지의 값이 10 이면 110 번지를 증가시키고, 그렇지 않다면 120 번지를 증가 시킵니다.

▶ 예제 아이템 등록

IF_TIMER 아이템을 추가하고, <주기>를 “100”으로 입력합니다.(1 초 주기)

ARITHMETIC 아이템을 삽입하고, 100 번지에 1 을 더해 100 번지에 저장하도록 입력합니다.

IF_NUM_CMP 아이템에서 100 번지가 1 바이트가 10 이 되는지를 비교하도록 등록합니다.

<데이터 형식>은 ‘Unsigned 1Byte’, <기준값 형태>는 ‘상수’, <기준값(상수)>는 “10”, <연산자>는 ‘=’, <데이터 주소>는 “100”을 입력합니다.

ARITHMETIC 아이템을 삽입하고, 110 번지에 1 을 더해 110 번지에 저장하도록 입력합니다.

ELSE 아이템을 삽입합니다.

ARITHMETIC 아이템을 삽입하고, 120 번지에 1 을 더해 120 번지에 저장하도록 입력합니다.

▶ 예제 프로세스

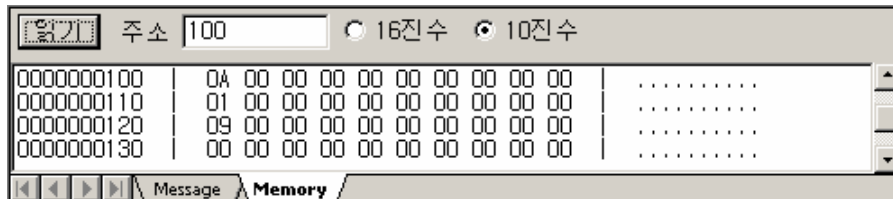
편집을 마친 프로세스 화면입니다. 동일한 순서로 등록했는지 확인하시기 바랍니다.

NO	아이템	설명
00	IF_TIMER	100
01	ARITHMETIC	Type:U1, M100, +, 1, M100
02	IF_NUM_CMP	Type:U1, 10, =, M100
03	ARITHMETIC	Type:U1, M110, +, 1, M110
04	ELSE	
05	ARITHMETIC	Type:U1, M120, +, 1, M120
06	END_IF	
07	END_IF	

▶ 결과

ARITHMETIC 아이템으로 100 번지가 1 초마다 1 씩 증가하는 것을 볼 수 있습니다. 9 까지 증가를 할때까지는 ELSE 아이템 다음인 05 ARITHMETIC 아이템이 실행되어, 120 번지가 1 씩 증가를 하는 것을 볼 수 있습니다. 이후 100 번지가 10 이 되면 IF_NUM_CMP 아이템에 의해 증가한 값이 10(A)인지 비교를 하게 되고 그 값이 10 을 만족하므로 ELSE 아이템 위의 03 ARITHMETIC 아이템이 실행되어 110 번지가 1 증가합니다. 이후에는 계속 05 ARITHMETIC 아이템이 실행되어 120 번지의 값이 증가 할 것입니다.

IF 아이템의 조건이 맞지 않는 경우 실행됨을 알 수 있습니다.



7.6.12. ENCRYPTION

- ▶ 응용 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.12.1. 기능

변환대상 데이터 주소에 저장되어 있는 데이터를 선택한 방법에 의해 암호화(Encryption)해서 변환결과 데이터 주소에 저장합니다. 현재 이 아이템은 ABB Alpha meter의 encryption 방법만 제공합니다. 암호화에 대한 보다 자세한 내용은 제조사로 문의 하시기 바랍니다.

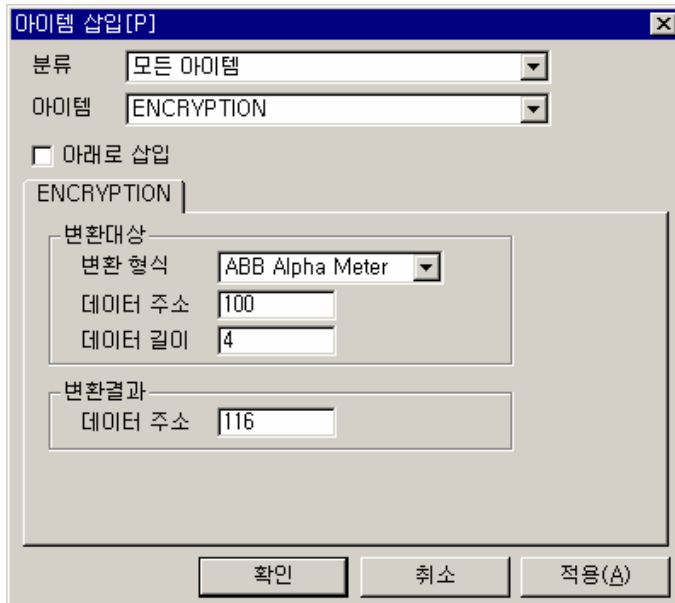


그림 7.6-13 ENCRYPTION 입력 예

표 7.6-12 ENCRYPTION 설정 항목

항 목	내 용	
변환 대상	변환 형식	ABB Alpha meter 만 지원 ^(*95)
	데이터 주소	암호화 하려는 데이터가 저장되어있는 주소를 입력합니다.
	데이터 길이	암호화 할 데이터의 길이를 바이트 단위로 입력합니다.
변환결과 데이터 주소	암호화된 데이터를 저장할 주소를 입력합니다.	

(*95) 추후 추가 예정

7.6.13. END

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [END_IF](#)

7.6.13.1. 기능

해당 프로세스를 종료합니다. 프로세스 수행 중, END 아이템을 만나면 프로세스가 종료됩니다.

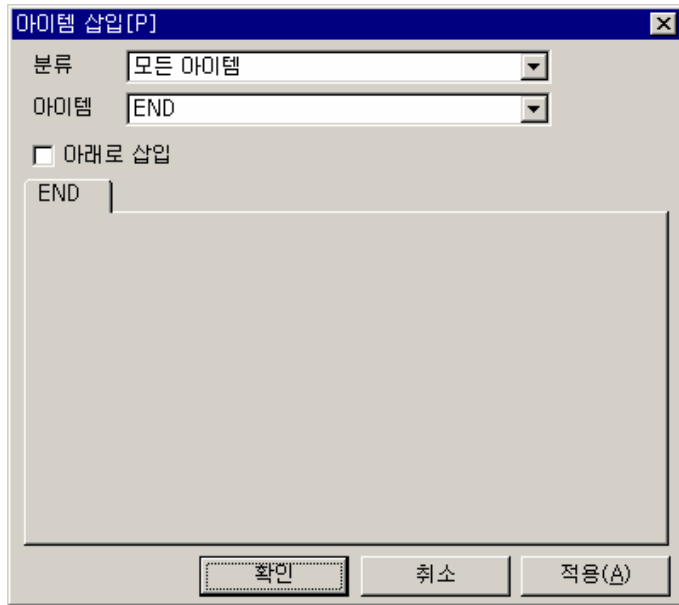


그림 7.6-14 END 아이템

표 7.6-13 END 설정 항목

항 목	내 용
없음	설정 항목이 없습니다

7.6.13.2. 예제

ETOS 메모리 100 번지와 110 번지가 값이 1 씩 증가하도록 아이템을 등록한 후 프로세스가 중간에서 중단되도록 합니다.

▶ 예제 아이템 등록

IF_TIMER 아이템을 추가하고, <주기>를 “100”으로 입력합니다.(1 초 주기)

ARITHMETIC 아이템을 삽입하고, 100 번지에 1 을 더해 100 번지에 저장하도록 입력합니다.
END 아이템을 등록합니다.

ARITHMETIC 아이템을 삽입하고, 110 번지에 1 을 더해 110 번지에 저장하도록 입력합니다.

▶ 예제 프로세스

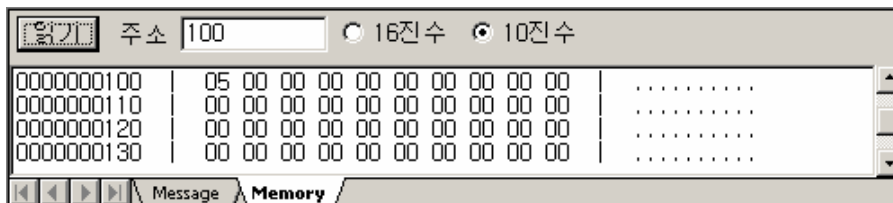
편집을 마친 프로세스 화면입니다 동일한 순서로 입력하였는지 확인하시기 바랍니다.

NO	아이템	설명
00	IF_TIMER	100
01	ARITHMETIC	Type:U1, M100, +, 1, M100
02	END	
03	ARITHMETIC	Type:U1, M110, +, 1, M110
04	END_IF	

▶ 결과

100 번지 값이 1 초 주기로 1 씩 증가하는 것을 볼 수 있습니다. 그러나 110 번지의 값은, END 아이템이 프로세스를 중간에서 종료하므로 증가하지 않습니다.

위 예제는 02 번 줄 END 와 03 번 줄 ARITHMETIC 을 모두 등록하지 않은 경우와 결과가 동일합니다.



7.6.14. END_IF

- ▶ 조건 분기 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [END](#)

7.6.14.1. 기능

IF 아이템 블록의 종료를 나타내며, IF 아이템과 쌍으로 사용되어야 합니다. 즉, IF 아이템은 가장 가까운 END_IF 사이의 명령을 수행하며 IF 아이템과 END_IF 아이템은 사용 개수가 동일해야 합니다. IF 아이템은 반드시 END_IF 아이템을 동반해야 하며, IF 아이템의 조건이 만족되지 않는 경우, END_IF 아이템의 다음을 수행합니다. 그러나, 시작 아이템은 END_IF 아이템 이후에 다른 아이템을 삽입할 수 없습니다.

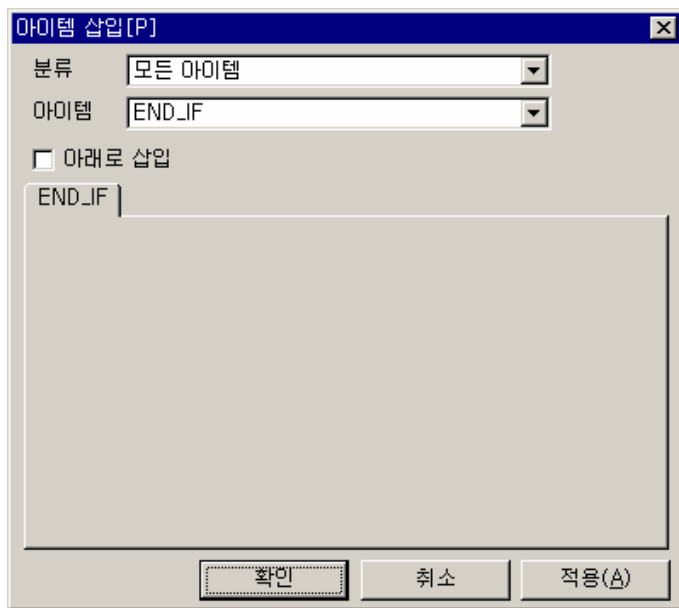


그림 7.6-15 END_IF 아이템

표 7.6-14 END_IF 설정 항목

항 목	내 용
없음	설정 항목이 없습니다

7.6.14.2. 예제

시작 아이템 IF_TIMER 를 등록한 후 END_IF 를 삭제하는 경우 어떠한 메시지가 나오는지 { 메시지 } 창을 활성화 시킨후 프로그램 검사를 실행합니다.

▶ 예제 아이템 등록

IF_TIMER 아이템을 1 초 주기가 되도록 등록합니다.

END_IF 아이템을 삭제합니다.

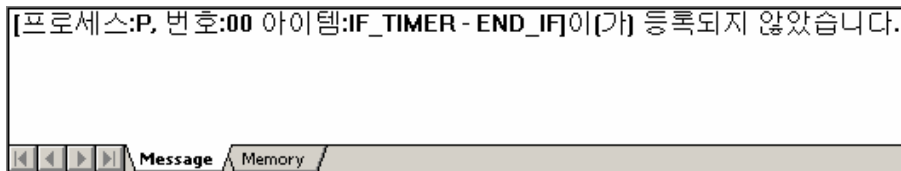
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	100

▶ 결과

[온라인] -> [편집 내용 검사]를 한 후 {메시지} 창을 보면 IF_TIMER 의 짝 아이템인 END_IF 가 등록되지 않았다는 메시지가 나타난 것을 볼 수 있습니다. IF 아이템과 짝을 이루어야 한다는 것을 알 수 있습니다.



7.6.15. END_LOOP

- ▶ 조건 분기 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [LOOP](#)

7.6.15.1. 기능

LOOP 아이템과 짝으로 사용되어 LOOP 블록의 끝을 나타냅니다. LOOP 카운터가 0 이 될때까지 LOOP 블록을 반복 수행하고, LOOP 카운터가 0 이 되면 END_LOOP 아이템의 다음을 수행합니다.

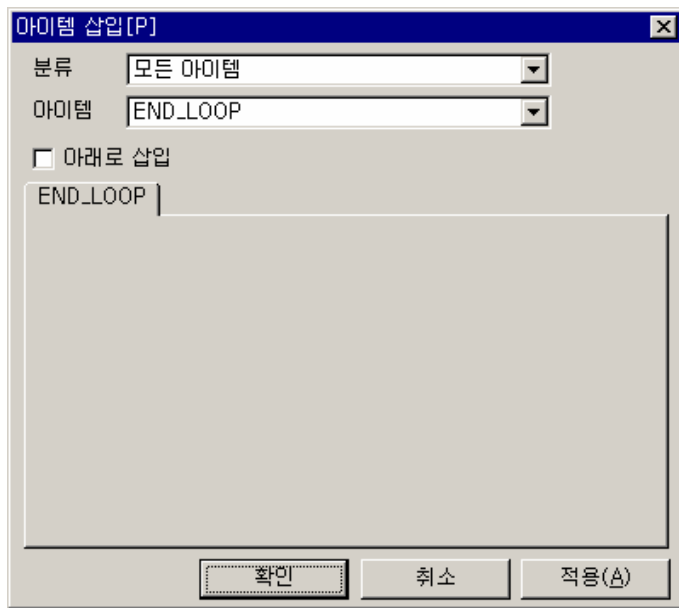


그림 7.6-16 END_LOOP 아이템

표 7.6-15 END_LOOP 설정 항목

항 목	내 용
없음	설정 항목이 없습니다

7.6.15.2. 예제

사용에는 [LOOP 아이템](#)을 참조하시기 바랍니다.

7.6.16. ENET_CLOSE

- ▶ 통신 아이템 (이더넷 전용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [ENET_OPEN](#)

7.6.16.1. 기능

이더넷 채널 연결을 끊습니다. 이더넷 채널 설정에서 설정한 채널 번호를 입력합니다.

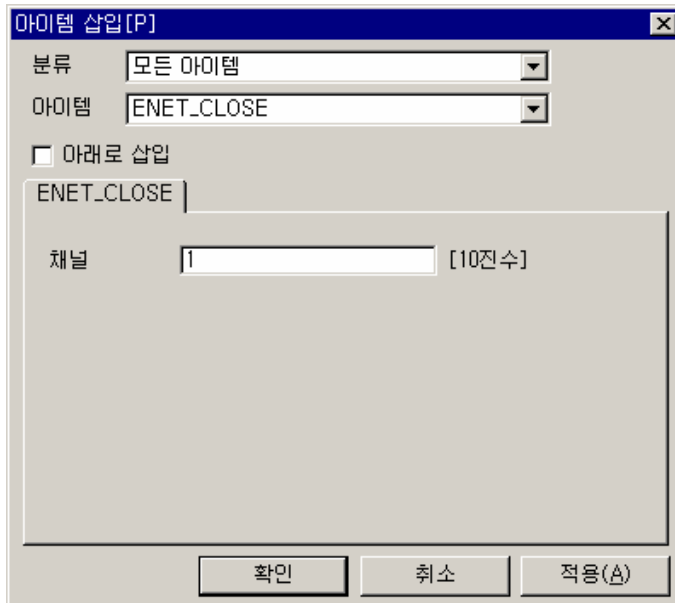


그림 7.6-17 ENET_CLOSE 입력 예

표 7.6-16 ENET_CLOSE 설정 항목

항 목	내 용
채널	설정된 이더넷 채널 번호 (0 ~ 31) ^(*96)

(*96) 입력가능한 채널 번호는 시리즈 별로 다릅니다.

ETOS-50 : 0 ~ 3
 ETOS-100/150 : 0 ~ 15
 ETOS-200 : 0 ~ 23
 ETOS-1000 : 0 ~ 31

7.6.17. ENET_DISABLE

- ▶ 통신 아이템 (이더넷 전용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [ENET_ENABLE](#)

7.6.17.1. 기능

이더넷 드라이버(슬레이브)의 통신을 잠시 중지시킵니다. 연결은 유지합니다.
통신을 재개하려면 ENET_ENABLE 아이템을 사용합니다.

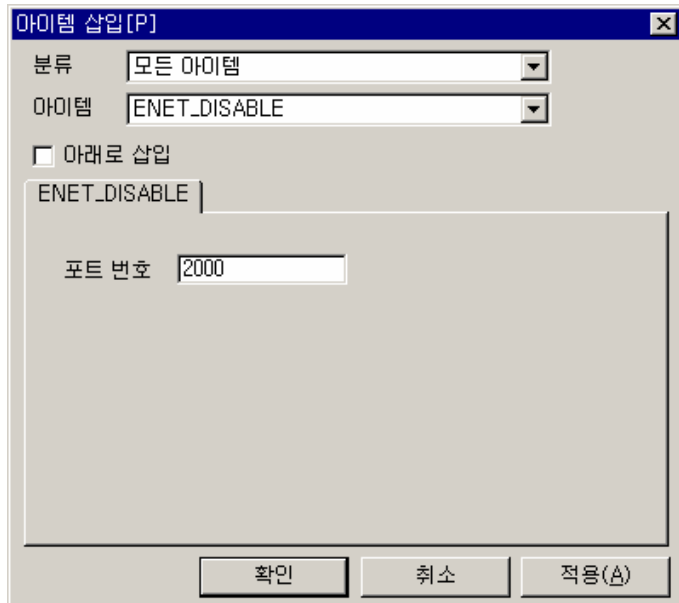


그림 7.6-18 ENET_DISABLE 입력 예

표 7.6-17 ENET_DISABLE 설정 항목

항 목	내 용
포트	이더넷 드라이버 설정에서 설정한 서버 포트 번호

7.6.18. ENET_ENABLE

- ▶ 통신 아이템 (이더넷 전용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [ENET_DISABLE](#)

7.6.18.1. 기능

중단된 이더넷 드라이버 통신을 다시 시작합니다. ENET_DISABLE 아이템과 짝으로 사용됩니다.

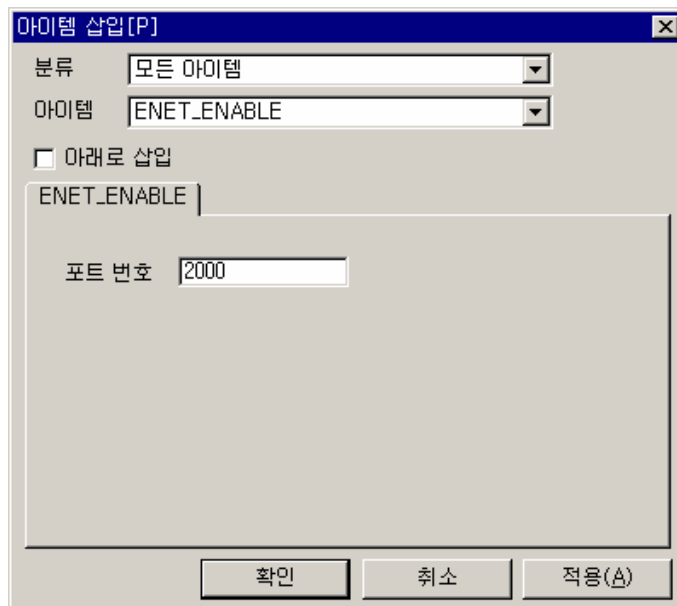


그림 7.6-19 ENET_ENABLE 입력 예

표 7.6-18 ENET_ENABLE 설정 항목

항 목	내 용
포트	이더넷 드라이버 설정에서 설정한 서버 포트 번호

7.6.19. ENET_OPEN

- ▶ 통신 아이템 (이더넷 전용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [ENET_CLOSE](#)

7.6.19.1. 기능

상대국에 이더넷 채널 연결 요청을 합니다. 상대국이 연결을 승낙하면 다음 아이템을 수행합니다. ETOS OS 버전 1.20 부터는 ENET_OPEN 아이템을 사용하지 않고, 통신 아이템을 사용하는 경우, 연결이 필요하면 자동으로 연결 요청을 합니다. 따라서, 특별한 경우를 제외하고는 사용하지 않습니다.

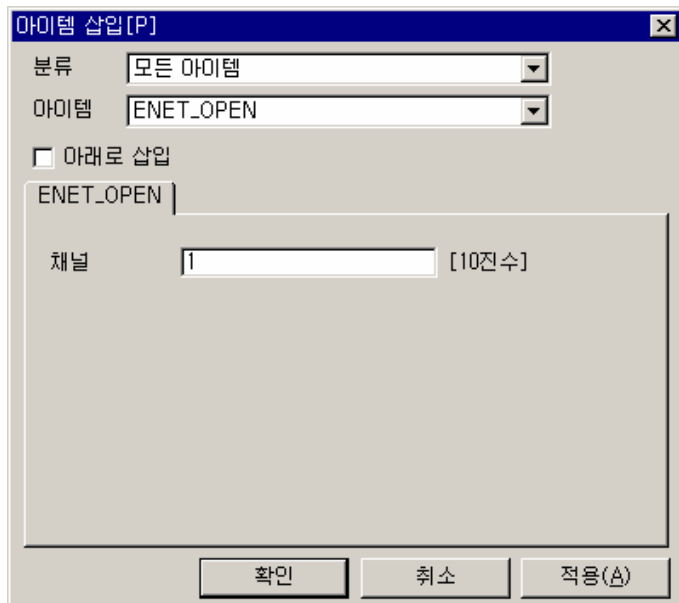


그림 7.6-20 ENET_OPEN 입력 예

표 7.6-19 ENET_OPEN 설정 항목

항 목	내 용
채널	이더넷 채널 설정에서 설정한 채널 번호 (0~31) ^(*97)

(*97) 입력가능한 채널 번호는 시리즈 별로 다릅니다.

- ETOS-50 : 0 ~ 3
- ETOS-100/150 : 0 ~ 15
- ETOS-200 : 0 ~ 23
- ETOS-1000 : 0 ~ 31

7.6.20. EVENT_READ_WRITE

- ▶ 응용 아이템 (IFOR 전용)
- ▶ ETOS-200/1000 IFOR
- ▶ [IF_EVENT_OCCUR](#), [SLAVE_MEM_READ_WRITE](#)

7.6.20.1. 기능

ETOS 시리즈의 IFOR(Intelligent Fiber Optic dual-Ring system) 모듈 전용 아이템으로, IFOR 모듈이 장착된 시스템에서만 사용합니다. IFOR는 ETOS-200/1000 용의 광링(Optic Ring) 제어 모듈입니다. IFOR 전용 아이템을 편집하기 위해서는 [옵션] -> [편집 옵션]에서, <IFOR 편집>을 체크해야 합니다.

아래는 Card 0 번 IFOR에서 Event 메모리 0 번지부터 8 바이트를 ETOS의 100 번지로 READ 하는 설정입니다.

그림 7.6-21 EVENT_READ_WRITE 입력 예

표 7.6-20 EVENT_READ_WRITE 설정 항목

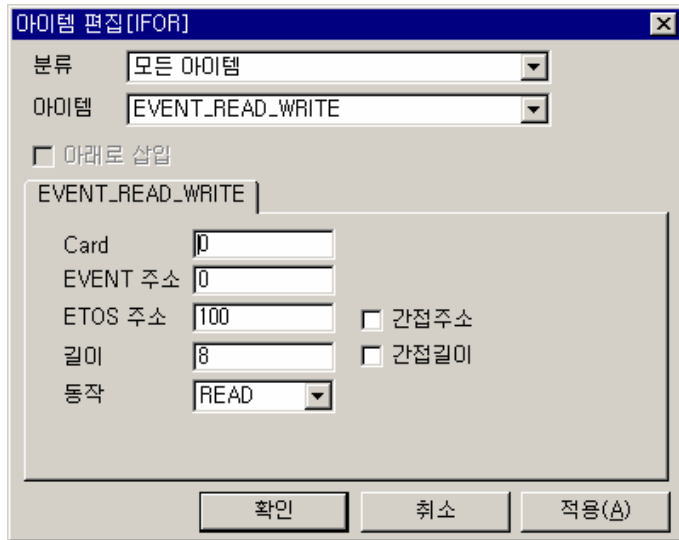
항 목	내 용
Card	IFOR CARD의 번호를 입력합니다.
EVENT 주소	이벤트 메모리내의 주소입니다.
ETOS 주소	ETOS 사용자 메모리내의 주소입니다.
간접 주소	ETOS 사용자 메모리내의 주소를 간접 지정할 경우 체크합니다.
길이	READ/WRITE 할 메모리 영역의 길이입니다.
길이	READ/WRITE 할 메모리 영역의 길이를 ETOS 사용자 메모리내의 주소로 간접 지정할 경우 체크합니다.
동작	READ : IFOR EVENT -> ETOS 사용자 메모리 WRITE : IFOR EVENT <- ETOS 사용자 메모리

7.6.20.2. 예제

Card 0 의 IFOR Event 메모리 1 번 비트가 Falling 되면 0~7 번까지의 Event 메모리를 ETOS 의 사용자 영역 100 번지로 복사합니다.

▶ **예제 아이템 등록**

EVENT_READ_WRITE 아이템을 추가하고, <Card>는 “0”으로, <EVENT 주소>는 “0”, <ETOS 주소>는 “100”번지로, <길이>는 “8”로, <동작>은 ‘READ’로 각각 설정합니다.



▶ **예제 프로세스**

예제에선 프로세스의 시작을 위해서 IF_EVENT_OCCUR 를 추가했습니다.

NO	아이템	설명
00	IF_EVENT_OCCUR	Card:0, EventBit:1, EventType:Falling
01	EVENT_READ_WRITE	Card:0, Event:0, ETOS:MI00, Size:8, Function:Read
02	END_IF	

▶ **결과**

IFOR Event 메모리 1 번 비트가 Falling 되면 0~7 번까지의 Event 메모리를 ETOS 의 사용자 메모리 100 번지로 복사합니다.

7.6.21. FRAME_SND

- ▶ 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_REQ_RESP](#)

7.6.21.1. 기능

시리얼 포트나 이더넷 채널로 지정한 송신 프레임을 송신합니다. 송신 프레임은 미리 등록해야 합니다.

다음 그림은 FRAME_SND 아이템으로 좌측 {프레임 등록} 창의 READ_요청 프레임을 송신하는 예입니다.

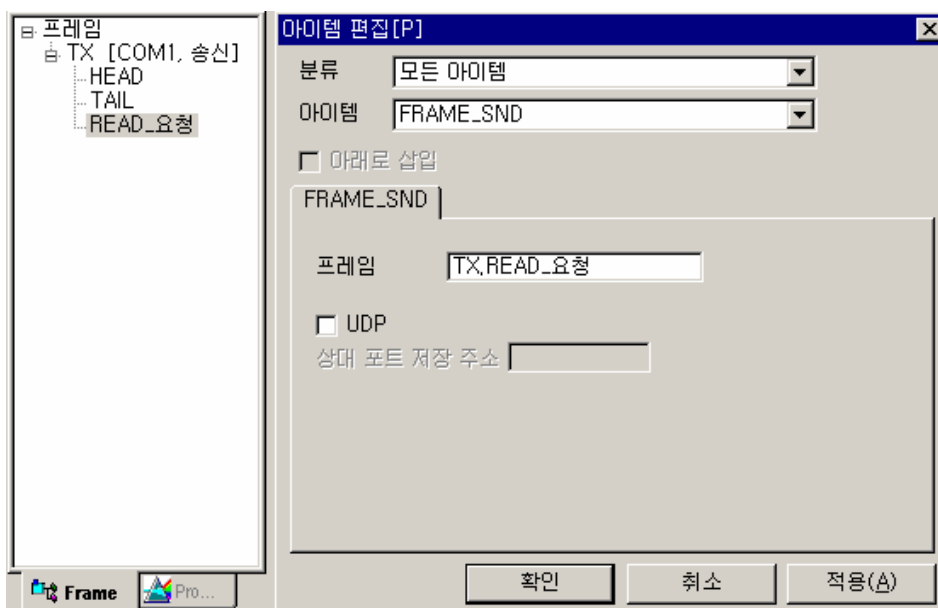


그림 7.6-22 FRAME_SND 입력 예

표 7.6-21 FRAME_SND 설정 항목

항 목	내 용
프레임	송신할 프레임 이름(그룹이름.BODY 프레임 이름)을 입력합니다. 키보드 입력 대신 프레임을 클릭한 후 마우스 왼쪽 버튼을 누른채로 프레임 입력란에 가져다 놓아도 됩니다.(DRAG & DROP)
UDP	이더넷 UDP 통신인 경우 지정할 수 있습니다. ^(*98)
상대 포트 저장 주소	UDP 를 체크한 경우 활성화 됩니다. 상대 포트를 파라미터에서 지정하지 않은 경우, 포트 번호를 메모리에서 참조할 수 있습니다. 해당 주소에는 포트 번호가 미리 입력되어 있어야 정상 동작합니다. (포트 번호는 2 바이트로 저장되어야 합니다)

(*98) 파라미터 설정에서 상대국 UDP 포트를 지정한 경우는 지정하지 않아도 됩니다. 파라미터에서 설정한 포트와 다른 포트로 보내고자 할 때만 설정합니다.

7.6.21.2. 예제

COM1 포트의 송신 프레임의 정의를 하고, 프레임의 송신합니다.

▶ 예제 아이템 등록

먼저, COM1 포트를 RS232C 로 설정합니다. 파라미터 설정에 대한 자세한 사항은 [5.3.1.1. Serial 포트](#)를 참조하시기 바랍니다.

{프레임 등록} 창에 COM1 포트의 송신 그룹을 등록하고, 송신 프레임의 HEAD, BODY, TAIL 을 등록합니다. (실제 등록 프레임은 결과 화면을 참조하여 입력)

IF_TIMER 아이템을 추가하고, 주기를 200 으로 입력합니다. (2 초 주기)

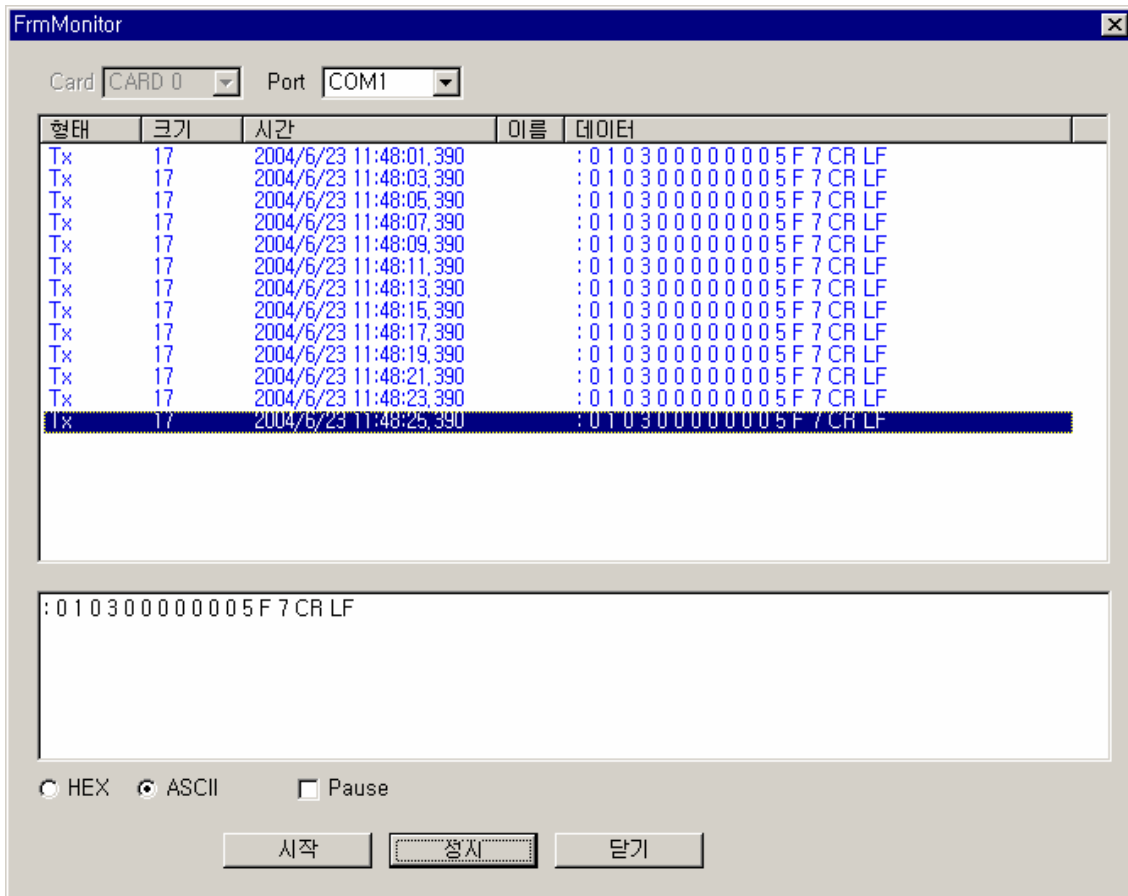
FRAME_SND 아이템을 추가하고, 프레임에 위에서 등록한 프레임의 이름을 입력합니다.

▶ 예제 프로세스

NO	아이템	설명
00	IF_TIMER	200
01	FRAME_SND	Frm:TX,READ_요청
02	END_IF	

▶ 결과

[온라인] -> [프레임 모니터]를 선택합니다. {프레임 모니터} 창에서 포트를 COM1 으로 선택하고, [시작] 버튼을 누르면, COM1 포트의 송,수신 프레임의 모니터할 수 있습니다. 등록된 프레임이 2 초 간격으로 송신되고 있음을 알 수 있습니다.



7.6.22. HEX_TO_ASC

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [ASC_TO_HEX](#)

7.6.22.1. 기능

지정한 주소의 16 진수 데이터를 16 진수 문자열('1', ~ '9', 'A' ~ 'F' 로 이루어진 문자열)로 변환합니다. ASC_TO_HEX 아이템의 기능과 정반대의 기능을 합니다.

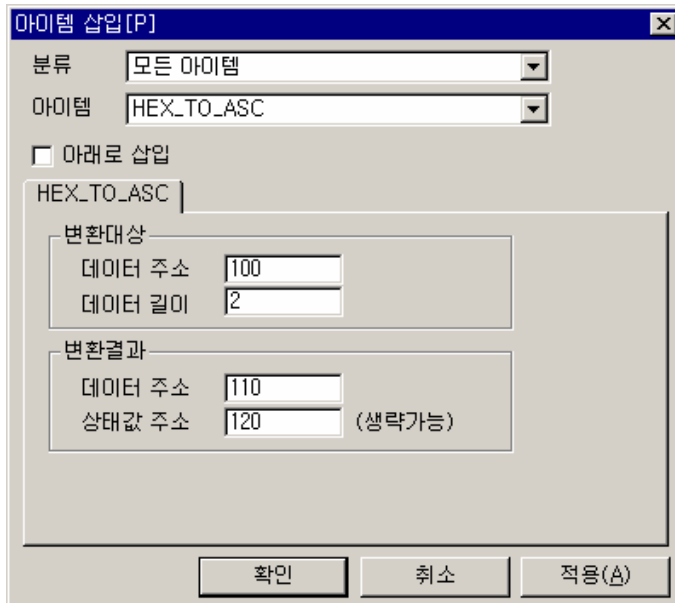


그림 7.6-23 HEX_TO_ASC 입력 예

표 7.6-22 HEX_TO_ASC 설정 항목

항 목	내 용	
변환 대상	데이터 주소	변환할 데이터가 저장되어 있는 주소를 지정합니다.
	데이터 길이	변환할 데이터의 길이를 지정합니다. 입력 가능한 데이터 길이의 범위는 1~16384 이며, 변환이 이루어 지면 데이터 길이는 2 배로 늘어 납니다.
변환 결과	데이터 주소	변환된 데이터가 저장될 메모리의 주소를 지정합니다.
	상태 값 주소	변환 결과값을 저장할 주소를 지정합니다. 정상 변환 시 변환된 문자열의 길이가 저장됩니다. (2 바이트 사용)

7.6.22.2. 예제

ETOS 메모리 100 번지, 101 번지 2 바이트 값을 16 진수 문자열로 변환합니다.

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.

변환 대상이 될 데이터를 MEM_SET 아이템으로 입력합니다. <데이터 형태>는 '16 진수', <데이터>는 "0123", <주소> "100", <길이> "2"를 각각 입력합니다.

HEX_TO_ASC 아이템을 삽입한 후 다음과 같이 입력합니다.

항 목		입력 내용
변환 대상	데이터 주소	100
	데이터 길이	2
변환 결과	데이터 주소	110
	상태 값 주소	120

▶ 예제 프로세스

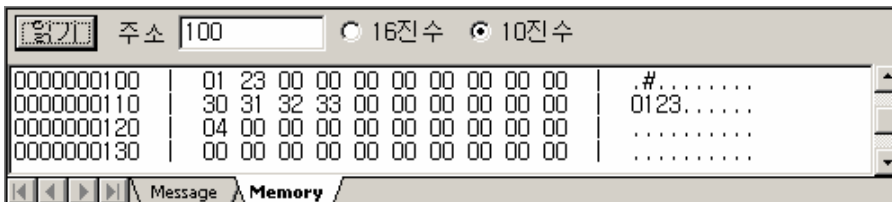
편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:2, Type:16진수, Data:0123
02	HEX_TO_ASC	S:M100, Size:2 T:M110, Status:M120
03	END_IF	

▶ 결과

100 번지, 101 번지 값은 각각 0123_H 입니다.

110 번지에는 위 16 진수 값의 문자열 변환 결과인 30313233_H (문자열 "0123")이 저장되었습니다. 120 번지 상태 값은 문자열의 길이인 04 00 (0004_H = 4 바이트) 이 저장되었습니다.



7.6.23. HEX_TO_BCD

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.23.1. 기능

지정한 주소의 16 진수 데이터를 BCD(Binary-Coded Decimal)로 변환합니다.

(예: FF_H -> BCD 로 0255)

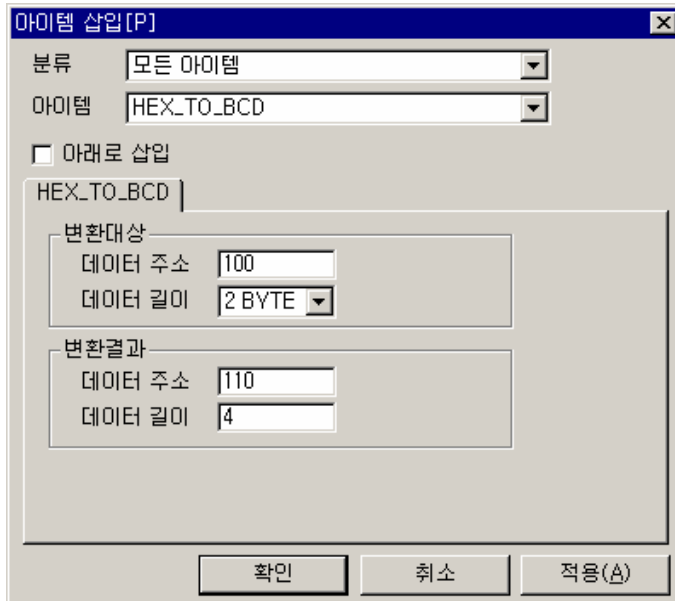


그림 7.6-24 HEX_TO_BCD 입력 예

표 7.6-23 HEX_TO_BCD 설정 항목

항 목		내 용
변환 대상	데이터 주소	변환 대상 데이터의 주소를 지정합니다.
	데이터 길이	변환 대상 데이터의 길이를 지정합니다. (1/2/4 바이트)
변환 결과	데이터 주소	변환 결과 데이터, 즉, BCD 값이 저장될 주소를 지정합니다.
	데이터 길이	변환 결과 데이터의 (최대)길이를 입력합니다. 변환된 길이는 16 진수 값보다는 더 커지는 차이가 있습니다. 그러므로 16 진수로 표현되는 최대값을 BCD 로 표현할 때의 최대길이를 지정해야 오류를 방지할 수 있습니다.

7.6.23.2. 예제

100 번지의 16 진수 값(0123_H)을 BCD(10 진수)로 변환하여 110 번지에 저장합니다.

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 추가합니다.

MEM_SET 아이템을 삽입하고, <데이터 형태>는 ‘16 진수’, <데이터>는 “0123”, <주소>는 “100”, <길이>는 “2”를 입력합니다.

HEX_TO_BCD 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목		입력 내용
변환 대상	데이터 주소	100
	데이터 길이	2 BYTE 선택
변환 결과	데이터 주소	110
	데이터 길이	4

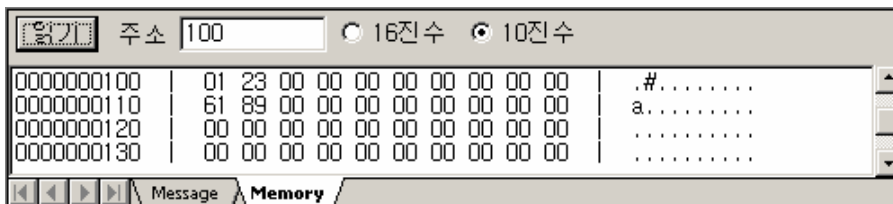
▶ **예제 프로세스**

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:2, Type:16진수, Data:0123
02	HEX_TO_BCD	S:M100, SSize:2, T:M110, TSize:4
03	END_IF	

▶ **결과**

100 번지 값 0123_H 이 BCD 로 변환된 결과값 6189_H 가 110 번지에 저장되었습니다. 다시말해 16 진수값 2301_H 은 10 진수로 변환하면 8961 이 됩니다.



7.6.24. IF_3964R_READ

- ▶ 통신 아이템 (시리얼 전용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_3964R_WRITE](#), ELSE, END_IF

7.6.24.1. 기능

Siemens 사의 시리얼 통신 프로토콜인 3964(R)을 사용하여 읽기를 수행합니다. 프레임을 별도로 등록할 필요없이 내장된 3964(R) 프로토콜을 사용합니다. 해당 포트 파라미터는 Siemens 3964(R)로 설정되어 있어야 합니다.

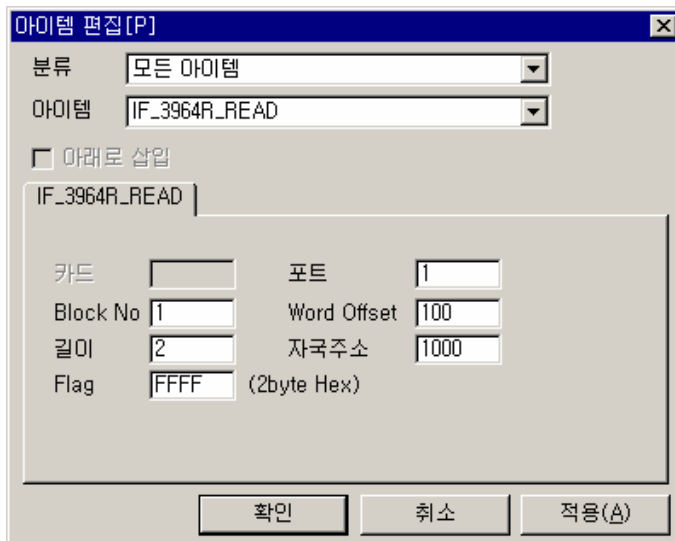


그림 7.6-25 IF_3964R_READ 입력 예

표 7.6-24 IF_3964R_READ 설정 항목

항 목	내 용
카드	해당 포트가 위치한 시리얼 통신 카드의 번호(0 ~ 7) ETOS-200/1000 일때만 활성화
포트	3964(R) 통신용 포트의 번호 (1 ~ 8)
Block No	데이터 블록 번호(DB) (1 ~ 255)
Word Offset	데이터 워드 오프셋(DW) (0 ~ 255)
길이	읽을 바이트 혹은 워드 길이 ^(*99)
자국주소	읽은 데이터를 저장할 ETOS 메모리 주소
Flag	coordination flag ^(*100)

(*99) 파라미터 설정시 결정한 데이터 형에 따라 바이트 혹은 워드 길이를 나타냅니다.

(*100) 프로토콜에 관한 사항은 Siemens 3964(R) 프로토콜 매뉴얼을 참조하시기 바랍니다.

7.6.25. IF_3964R_WRITE

- ▶ 통신 아이템 (시리얼 전용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_3964R_READ](#), ELSE, END_IF

7.6.25.1. 기능

Siemens 사의 시리얼 통신 프로토콜인 3964(R)을 사용하여 쓰기를 수행합니다.

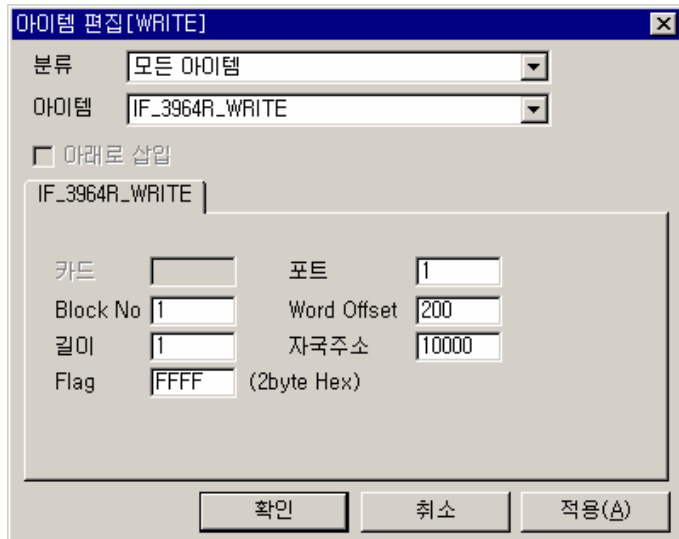


그림 7.6-26 IF_3964R_WRITE 입력 예

표 7.6-25 IF_3964R_WRITE 설정 항목

항 목	내 용
카드	해당 포트가 위치한 시리얼 통신 카드의 번호(0 ~ 7) ETOS-200/1000 일때만 활성화
포트	3964(R) 통신용 포트의 번호 (1 ~ 8)
Block No	데이터 블록 번호(DB) (1 ~ 255)
Word Offset	데이터 워드 오프셋(DW) (0 ~ 255)
길이	상대국에 보낼 데이터의 바이트 혹은 워드 길이 (*101)
자국주소	상대국에 보낼 데이터가 저장되어 있는 ETOS 메모리 주소
Flag	coordination flag (*102)

(*101) 파라미터 설정시 결정한 데이터 형에 따라 바이트 혹은 워드 길이를 나타냅니다.

(*102) 프로토콜에 관한 사항은 Siemens 3964(R) 프로토콜 매뉴얼을 참조하시기 바랍니다.

7.6.26. IF_BIT_FIND

- ▶ 조건 분기 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.26.1. 기능

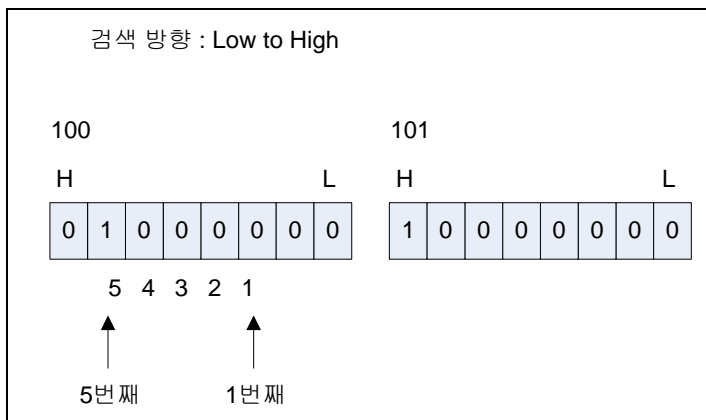
비트열에서 조건을 만족하는 비트의 위치를 찾습니다. 검색할 주소의 비트 위치부터 시작해서 검색방향을 따라 데이터에 설정한 조건, 즉 ON 또는 OFF 를 만족하는 비트의 위치를 찾아냅니다.

예를 들어보겠습니다.

100 번지, 101 번지에 다음 그림과 같이 각각 40_H, 80_H 이 저장되어 있다고 가정합니다.

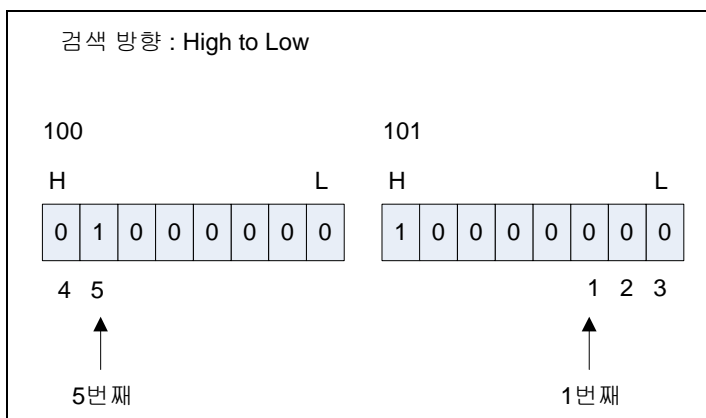
▶ Low To High

100 번지 3 번째 비트부터 Low To High 로 검색해서 데이터가 1(ON)인 비트의 위치를 찾습니다. 5 번째 만에 조건(1)을 만족하는 비트를 찾았습니다. 결과 주소에는 5 가 저장됩니다.



▶ High To Low

다음은 101 번지 3 번째 비트부터 High To Low 로 검색해서 데이터가 1(ON)인 비트의 위치를 찾습니다. Low 비트쪽으로 세어나가다가 1 번째 비트(LSB, Least Significant Bit) 까지 세면, 101 번 보다 낮은 메모리인 100 번지 8 번째 비트(MSB, Most Significant Bit)부터 Low 쪽으로 세어나갑니다. 결과 주소에는 5 가 저장됩니다.



주의할 점은 비트의 위치를 서수(Ordinal Numeral)로 표현한다는 것입니다. 따라서, 0 번 비트 (LSB)는 1 번째 비트이므로 1 을, 7 번 비트(MSB)는 8 번째 비트이므로 8 을 입력해야 합니다.

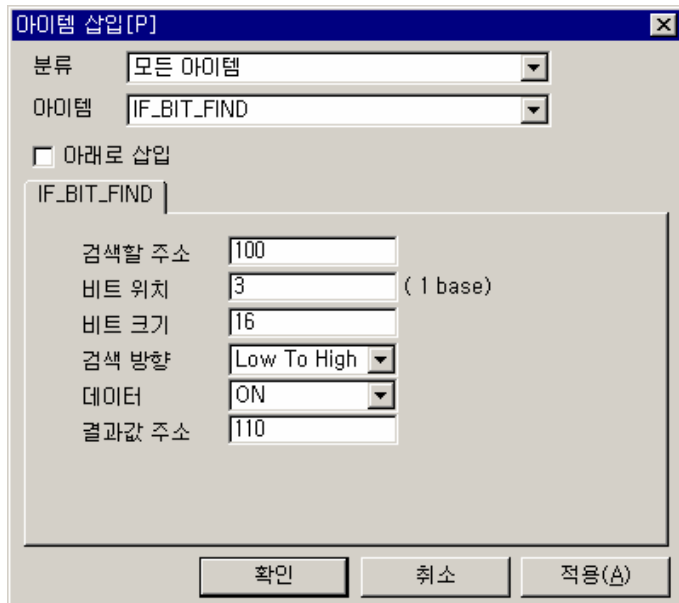


그림 7.6-27 IF_BIT_FIND 입력 예

표 7.6-26 IF_BIT_FIND 설정 항목

항 목	내 용
검색할 주소	검색 시작 위치의 바이트 주소를 입력합니다.
비트 위치	검색 시작 비트 위치를 서수로 입력 (1 ~ 8)
비트 크기	검색할 비트 개수를 지정합니다.
검색 방향	검색을 진행할 방향을 지정합니다. (Low To High / High To Low)
데이터	검색 조건을 지정합니다 (On (1) / Off (0))
결과 주소	검색 조건을 만족하는 비트 위치를 저장할 주소를 지정합니다.

7.6.26.2. 예제

100 번지와 101 번지에 각각 40H, 80H 이 저장되어 있습니다.

100 번지 3 번째 비트에서 Low To High 방향으로 검색해서 데이터가 1(ON)인 비트의 위치를 찾아 110 번지에 저장합니다.

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.

먼저 100 번지와 101 번지에 40H, 80H 을 저장합니다. MEM_SET 아이템을 추가하고, <데이터 형태>는 '16 진수', <데이터>는 "4080", <주소>는 "100", <길이>는 "2"를 입력합니다.

IF_BIT_FIND 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
검색할 주소	100
비트 위치	3
비트 크기	16
검색 방향	Low To High
데이터	ON
결과 주소	110

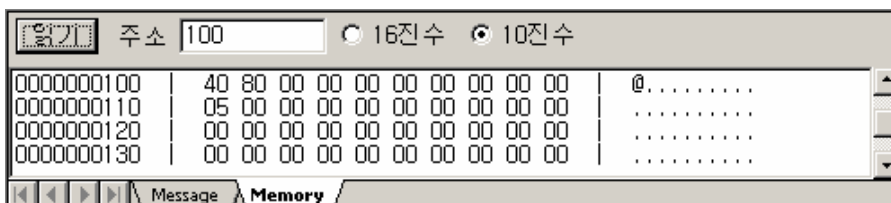
▶ 예제 프로세스

편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:2, Type:16진수, Data:4080
02	IF_BIT_FIND	S:M100, BitPos:3, BitSize:16, Dir:LowToHigh, Data:ON, Result:M110
03	END_IF	
04	END_IF	

▶ 결과

100 번지에는 4080H 이 저장되어 있습니다. 100 번지 3 번째 비트부터 Low To High 로 검색할 경우 5 번째만에 1(ON)인 비트를 찾을 수 있습니다. 110 번지에는 5 가 저장되어있습니다. 예제에 대한 자세한 설명은 기능 설명의 [▶ Low To High](#) 를 참조하시기 바랍니다.



7.6.27. IF_BIT_TEST

- ▶ 조건 분기 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.27.1. 기능

지정한 번지의 특정 비트가 1(ON)이거나 0(OFF)인지 테스트합니다. 조건을 만족하는 경우, IF_BIT_TEST 와 END_IF 사이에 등록된 아이템들이 실행됩니다.

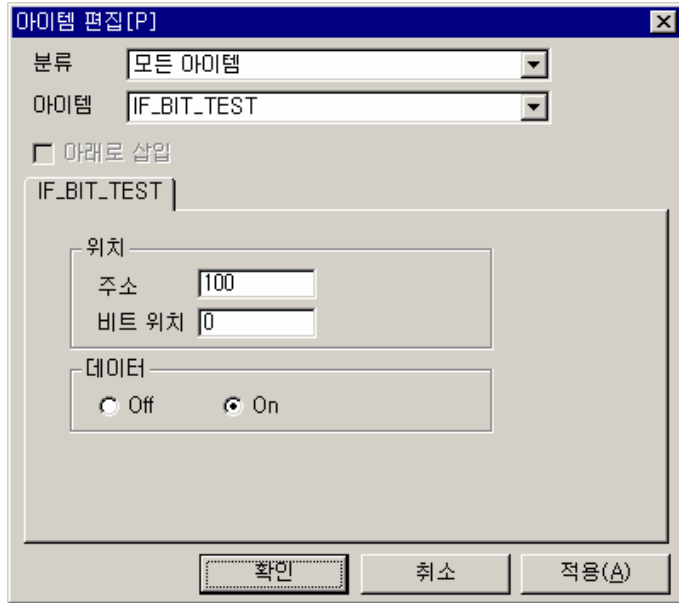


그림 7.6-28 IF_BIT_TEST 입력 예

표 7.6-27 IF_BIT_TEST 설정 항목

항 목	내 용
주소	테스트하고자 하는 바이트 주소를 입력합니다.
비트 위치	몇 번 비트를 테스트 할지 입력합니다.(0 (LSB) ~ 7 (MSB))
데이터	테스트 조건을 선택합니다. Off 를 선택하면 0 인지를, On 을 선택하면 1 인지를 테스트합니다.

7.6.27.2. 예제

ETOS 메모리 100 번지 0 번 비트(LSB)가 1 이면 110 번지에 문자열 “ON”을 저장합니다.

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 추가합니다.

먼저 100 번지값을 미리 지정합니다. MEM_SET 아이템을 삽입하고, <데이터 형태>는 ‘16 진수’, <데이터>는 “01”, <주소>는 “100”, <길이>는 “1”을 입력합니다.

IF_BIT_TEST 아이템을 추가하고, 다음과 같이 입력합니다.

항 목	입력 내용
주소	100
비트 위치	0
데이터	On 을 선택

IF_BIT_TEST 와 END_IF 사이에 MEM_SET 아이템을 추가하고, <데이터 형태>는 ‘문자열’, <데이터>는 “ON”, <주소>는 “110”, <길이>는 “2”를 입력합니다.

MEM_SET 아이템 다음에 ELSE 아이템을 삽입합니다.

ELSE 아이템 다음에 MEM_SET 아이템을 추가하고, <데이터 형태>는 ‘문자열’, <데이터>는 “OFF”, <주소>는 “110”, <길이>는 “3”을 입력합니다.

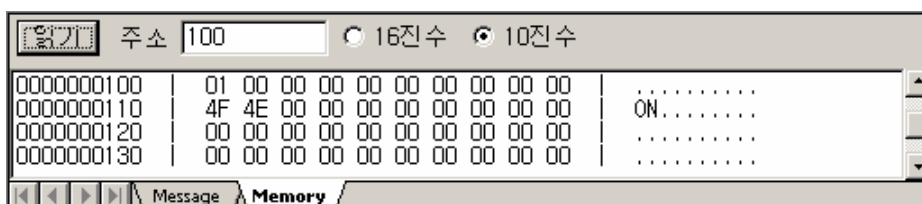
▶ **예제 프로세스**

편집을 마친 프로세스 화면입니다. 등록된 프로세스의 순서를 확인하시기 바랍니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:1, Type:16진수, Data:01
02	IF_BIT_TEST	S:M100, BitPos:0, Data:On
03	MEM_SET	T:M110, Size:2, Type:문자열, Data:ON
04	ELSE	
05	MEM_SET	T:M110, Size:3, Type:문자열, Data:OFF
06	END_IF	
07	END_IF	

▶ **결과**

100 번지 0 번 비트는 MEM_SET 아이템에 의해 1 로 설정되어 있습니다. IF_BIT_TEST 결과 테스트 조건을 만족하므로 03 MEM_SET 아이템이 실행되어 110 번지에는 문자열 “ON”이 쓰여졌습니다.



7.6.28. IF_EVENT_OCCUR

- ▶ 시작 아이템 (IFOR 전용)
- ▶ ETOS-200/1000 IFOR
- ▶ [EVENT_READ_WRITE](#), [SLAVE_MEM_READ_WRITE](#), END_IF

7.6.28.1. 기능

ETOS 시리즈의 IFOR(Intelligent Fiber Optic dual-Ring system) 모듈 전용 아이템으로, IFOR 모듈이 장착된 시스템에서만 사용합니다. IFOR는 ETOS-200/1000 용의 광링(Optic Ring) 제어 모듈입니다. IFOR 전용 아이템을 편집하기 위해서는 [옵션] -> [편집 옵션]에서, <IFOR 편집>을 체크해야 합니다.

IFOR는 공유 메모리(Shared Memory)^(*103) 외에 이벤트 영역을 공유할 수 있습니다. IFOR 이벤트 영역은 64Bytes(512Bits)로, 전체 국들이 공유하게 됩니다. 이벤트의 전송은 각 국마다 8Bits(1Byte)단위로 설정 가능합니다.

공유 메모리와는 달리 이벤트 영역은 IF_EVENT_OCCUR 라는 아이템을 사용해서 송수신이 모두 인터럽트(Interrupt)를 이용한 동기송수신으로 이루어집니다. 인접 두 국간에는 평균 800uSec 이내에 동기화가 가능합니다.

IF_EVENT_OCCUR 아이템은 프로세스의 선두에 올 수 있는 시작 아이템으로 카드번호, 비트 위치, 동작을 정의합니다.

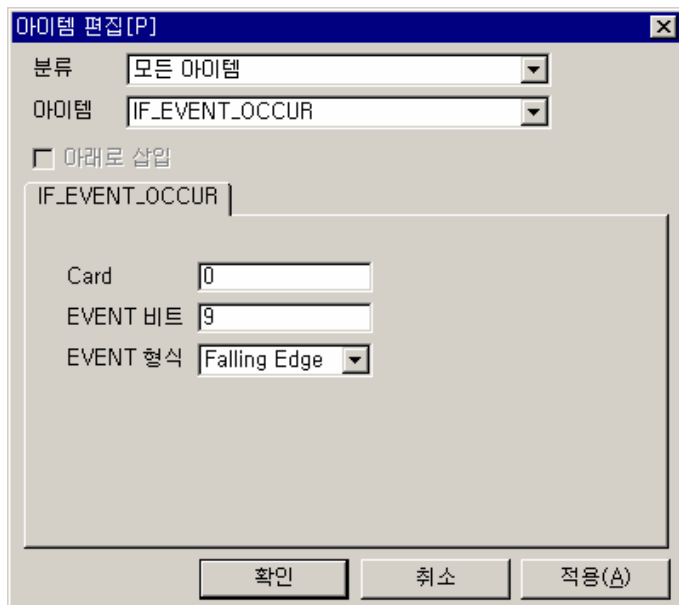


그림 7.6-29 IF_EVENT_OCCUR 입력 예

(*103) 각 국은 설정된 시간마다 전체 국으로 자기의 메모리를 Broadcasting 하여, 모든 국이 동일한 메모리를 가지고 있는 것처럼 보이게 합니다. 최대 약 50Kbytes의 메모리를 공유할 수 있습니다.

표 7.6-28 IF_EVENT_OCCUR 설정 항목

항 목	내 용
Card	IFOR 보드의 카드 번호를 입력합니다. (0 ~ 7) ^(*104)
EVENT 비트	1 바이트 단위로 동일 카드에 소속되도록 설정합니다. 예를 들어, 0 ~ 7 번 비트, 8 ~ 15 번 비트, 16 ~ 23 번 비트는 동일 IFOR 카드의 이벤트로 등록되어야 합니다.
EVENT 형식	Falling Edge : 해당 비트가 1 에서 0 으로 바뀔 때, 프로세스가 수행됩니다. Rising Edge : 해당 비트가 0 에서 1 로 바뀔 때, 프로세스가 수행됩니다. Both Edge : 해당 비트가 변경될 때, 프로세스가 수행됩니다.

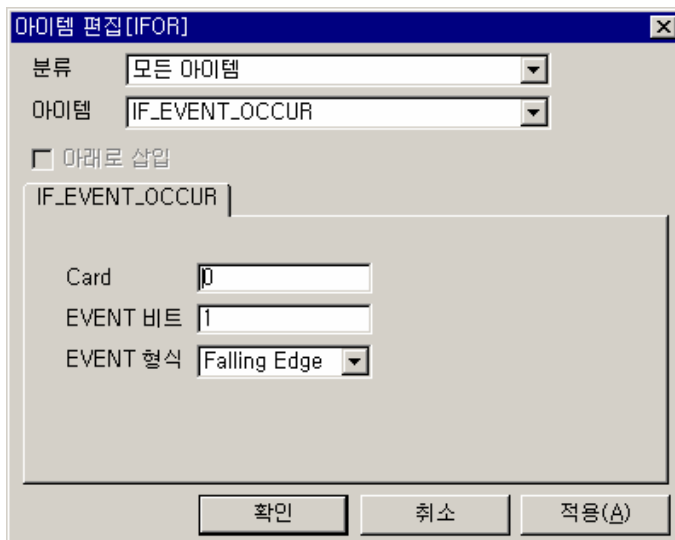
* 앞 페이지의 예는 9 번 비트가 1 에서 0 으로 바뀌면, IF_EVENT_OCCUR 프로세스가 실행됩니다.

7.6.28.2. 예제

Card 0 의 IFOR Event 메모리 1 번 비트가 Falling 되면 프로세스가 시작됩니다.

▶ 예제 아이템 등록

<Card>는 “0”으로, <EVENT 비트>는 “1”, <EVENT 형식>은 ‘Falling Edge’로 각각 설정합니다.



▶ 예제 프로세스

예제에선 프로세스가 시작되면 EVENT_READ_WRITE 를 수행하도록 했습니다.

NO	아이템	설명
00	IF_EVENT_OCCUR	Card:0, EventBit:1, EventType:Falling
01	EVENT_READ_WRITE	Card:0, Event:0, ETOS:M100, Size:8, Function:Read
02	END_IF	

▶ 결과

IFOR Event 메모리 1 번 비트가 Falling 되면 0~7 번까지의 Event 메모리를 ETOS 의 사용자 영역 100 번지로 복사합니다.

(*104) ETOS-200 의 경우 0/1, ETOS-1000 인 경우 0 ~ 7

7.6.29. IF_FRAME_RCV

- ▶ 시작 아이템, 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ END_IF

7.6.29.1. 기능

수신 프레임으로 정의한 프레임을 수신하고, 해당 프로세스를 실행합니다. 시리얼, 이더넷에서 모두 사용할 수 있습니다. 프레임을 정의하여 데이터를 주고 받는 경우, 프레임 정의 만으로는 수신 데이터가 자동으로 저장되지 않습니다. 즉, 아래 그림과 같이 이더넷 채널 0에 대해서 수신 프레임을 정의했다고 해서, 해당 프레임을 수신했을 때, 수신 데이터가 자동으로 ETOS 메모리에 저장되지는 않습니다. 프레임 정의는 수신을 위한 기초 단계일 뿐입니다. 수신 프레임의 데이터를 ETOS 메모리에 실제로 저장하기 위해서는 IF_FRAME_RCV 아이템을 사용해야 합니다.

이 아이템은 시작아이템으로 프로세스 중간에 사용할 수 없으며, 오직 프로세스 선두에만 올 수 있습니다. 이 아이템은 반드시 END_IF와 짝을 이루어야 하며, 중간에 ELSE 아이템은 사용할 수 없습니다.

다음 그림은 이더넷 채널 0의 수신 프레임으로 정의된 'READ_응답' 프레임을 수신하는 예입니다. 채널 0로 READ_응답 프레임을 수신하면 수신 데이터를 저장하고, 해당 프로세스 (IF_FRAME_RCV로 시작하는)를 수행합니다.

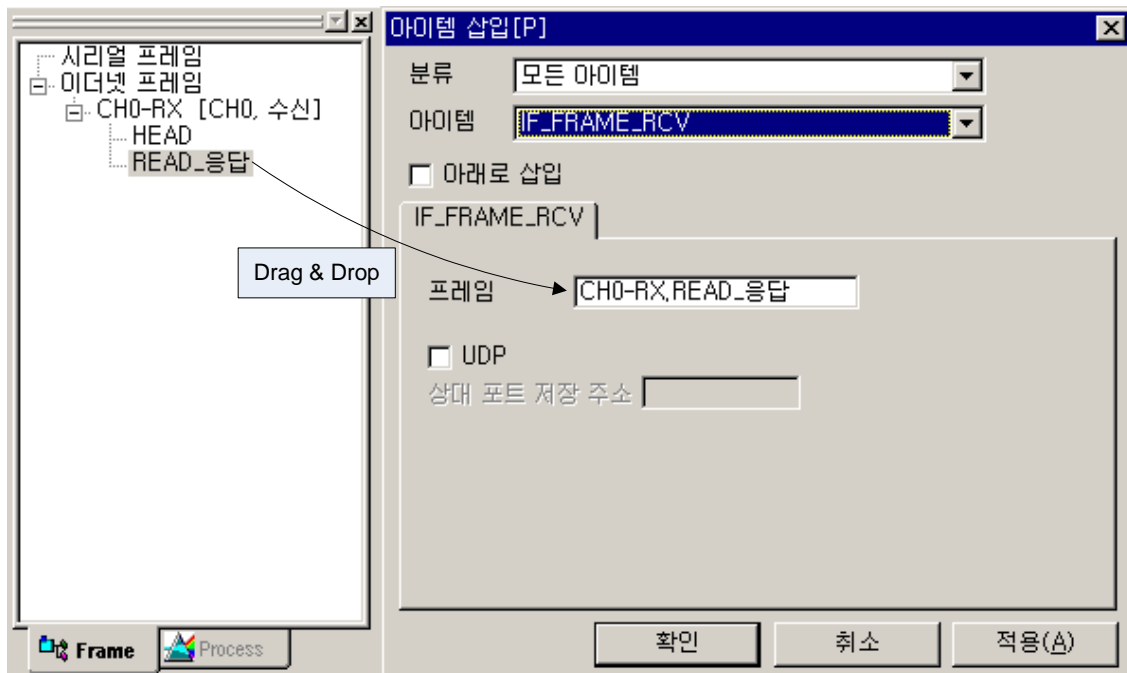


그림 7.6-30 IF_FRAME_RCV 입력 예

- * IF_FRAME_RCV 아이템은 상대국이 프레임을 보내면 수동적(Passive)으로 수신하는 역할을 합니다. IF_FRAME_RCV와 같이 수동적으로 동작하는 통신 아이템을 서버(Server)아이템이라고 합니다. 반대되는 개념은 클라이언트 또는 마스터(Client or Master)아이템으로 IF_REQ_RESP, FRAME_SND 등이 대표적인 예입니다.

표 7.6-29 IF_FRAME_RCV 설정 항목

항 목	내 용
프레임	수신할 프레임 이름(그룹이름.BODY 프레임 이름)을 입력합니다. 키보드 입력 대신 프레임을 선택한 후 마우스 왼쪽 버튼을 누른채로 프레임 입력란에 가져다 놓아도 됩니다.(DRAG & DROP)
UDP	이더넷 UDP 통신인 경우 지정할 수 있습니다. ^(*105)
상대 포트 저장 주소	UDP 를 체크한 경우 활성화 됩니다. 상대 포트를 파라미터에서 지정하지 않은 경우, 포트 번호를 메모리에서 참조할 수 있습니다. 해당 주소에는 포트 번호가 미리 입력되어 있어야 정상 동작합니다.

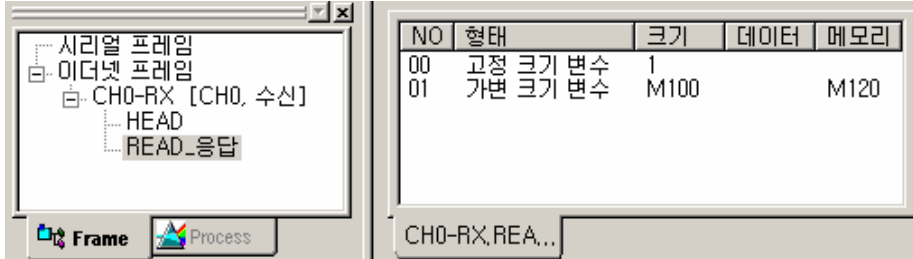
^(*105) 파라미터 설정에서 상대국 UDP 포트를 지정한 경우는 지정하지 않아도 됩니다. 파라미터에서 설정한 포트와 다른 포트로 보내고자 할 때만 설정합니다.

7.6.29.2. 예제

수신 프레임의 데이터를 메모리에 저장합니다.

▶ 예제 프레임 정의

먼저 예제 프레임을 다음과 같이 이더넷 채널 0에 대해서 정의합니다.



▶ 예제 아이템 등록

IF_FRAME_RCV 아이템을 등록하고, 프레임 이름에 위에서 정의한 수신 프레임의 이름(CH0-RX.READ_응답)을 입력합니다.

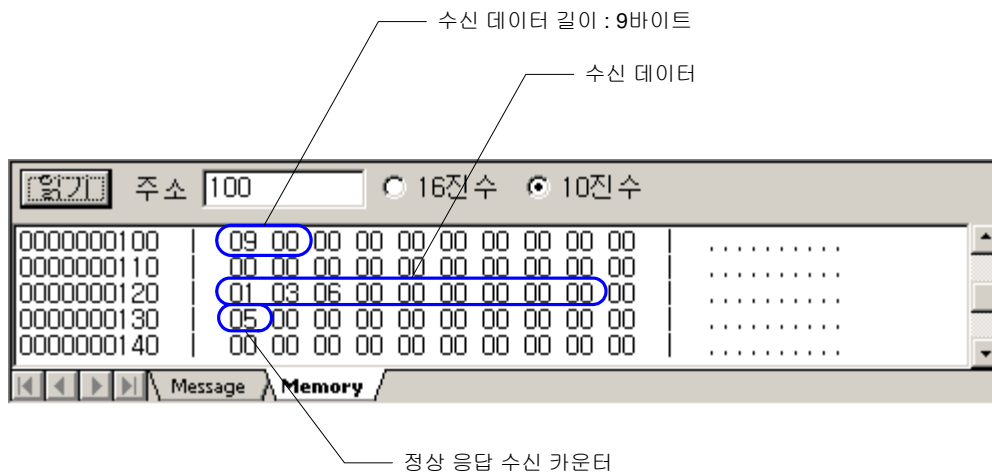
▶ 예제 프로세스

편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_FRAME_RCV	Frm:CH0-RX.READ_응답
01	END_IF	

▶ 결과

정의한 프레임을 수신하면, ETOS는 프레임 정의를 참조하여 해당 메모리 위치에 데이터를 저장합니다. 수신 데이터는 120번지에, 수신 데이터의 길이는 100번지에 저장되었습니다.^(*106)



(*106) 예제 프레임 참조

7.6.30. IF_GLOFA_READ

- ▶ 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_GLOFA_WRITE](#), ELSE, END_IF

7.6.30.1. 기능

LG 산전 GLOFA PLC 로부터 데이터를 읽어 ETOS 메모리에 저장합니다. 이더넷 및 시리얼에 모두 사용할 수 있습니다. (*107)

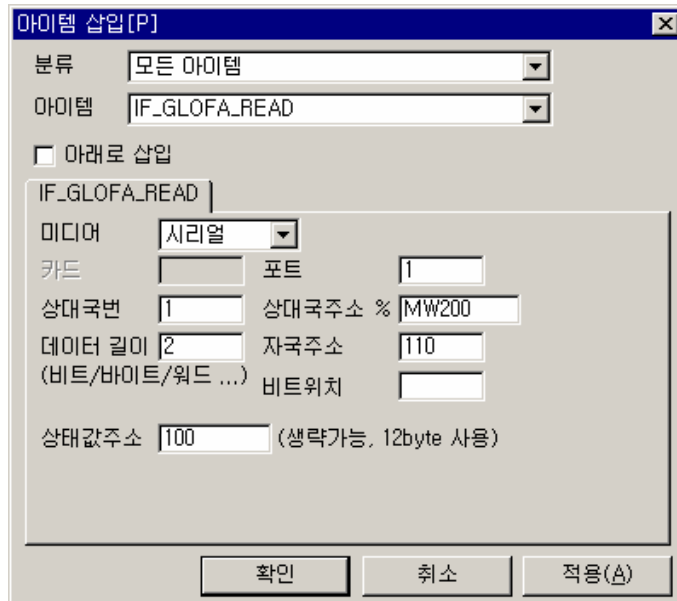


그림 7.6-31 IF_GLOFA_READ 입력 예

표 7.6-30 IF_GLOFA_READ 설정 항목

항 목	내 용
미디어	통신 포트의 종류를 선택합니다. (시리얼/이더넷)
카드	시리얼 카드의 번호 설정 (0 ~ 7) ^(*108)
포트	시리얼 포트 번호(1 ~ 8) 또는 이더넷 채널 번호(0 ~ 31)
상대국번	상대방의 국번(시리얼 통신인 경우만 입력)
상대국주소	데이터를 읽어올 상대국의 메모리를 직접 변수 형태로 입력합니다.
데이터 길이	읽을 데이터의 길이로 직접 변수의 타입에 따른 길이를 지정합니다. ^(*109)
자국주소	읽어온 데이터를 저장할 ETOS의 사용자 메모리 주소를 입력합니다.
비트위치	읽어온 데이터가 비트 형인 경우, 저장 비트 위치를 지정할 수 있습니다.

(*107) 시리얼은 GLOFA Cnet 프로토콜을, 이더넷은 GLOFA Enet 프로토콜을 사용합니다. 사용 프로토콜에 따라 읽을 수 있는 최대 길이 등의 사양이 조금씩 다릅니다. 자세한 사양은 GLOFA PLC 사용설명서를 참조하시기 바랍니다.

(*108) ETOS-200/1000 에서만 활성화 됨

(*109) %MW200 과 같이 직접 변수의 타입이 워드형이고 길이가 2 이면 2 워드, 즉, 4 바이트를 읽습니다.

표 7.6-30 IF_GLOFA_READ 설정 항목 (계속)

항 목	내 용
상태값주소	입력한 바이트 주소 위치에 성공 횟수, 실패 횟수, 에러 코드를 저장함. 입력한 위치로부터 차례대로, 4 바이트 - 성공 횟수, 4 바이트 - 실패 횟수, 4 바이트 - 에러 코드 (GLOFA PLC 에러 코드 표 참조) 를 의미.

* GLOFA 통신 설정에 대한 보다 자세한 사항은 8. 시리얼 통신 및 9. 이더넷 통신의 GLOFA Master 편을 참조하시기 바랍니다.

7.6.30.2. 예제

국번이 1 번인 GLOFA PLC 의 %MW200 번지에서 2 워드를 읽어 ETOS 메모리 110 번지에 저장합니다.

▶ 예제 아이템 등록

IF_GLOFA_READ 를 추가하고, 다음과 같이 입력합니다.

항 목	입력 내용
미디어	시리얼 선택
카드	N/A, (ETOS-50/100/150 을 선택한 경우)
포트	1
상대국번	1
상대국주소	MW200
데이터 길이	2
자국주소	110
비트위치	N/A, (비트 읽기가 아니므로 지정할 필요가 없음)
상태값주소	100

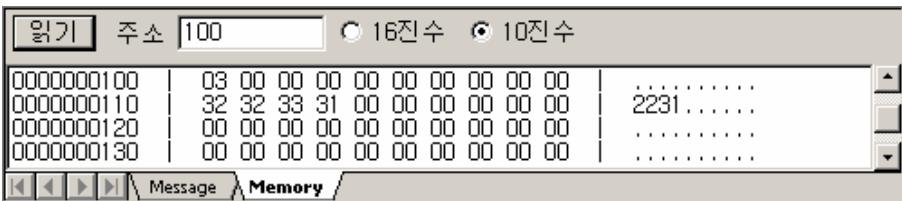
▶ 예제 프로세스

편집을 마친 프로세스 화면입니다.

NO	아이템	설명
00	IF_TIMER	100
01	IF_GLOFA_READ	Media:Serial, COM1, DestSt:1, DestAddr:MW200, Length:2, ...
02	END_IF	
03	END_IF	

▶ 결과

100 번지에는 읽기 성공 횟수(0x00000003 = 3 회)가 기록되고, 110 번지부터 4 바이트의 데이터가 저장되었습니다.



7.6.31. IF_GLOFA_WRITE

- ▶ 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_GLOFA_READ](#), ELSE, END_IF

7.6.31.1. 기능

ETOS 메모리의 데이터를 읽어 LG 산전 GLOFA PLC 디바이스 메모리에 씁니다. 이더넷 및 시리얼에 모두 사용할 수 있습니다.

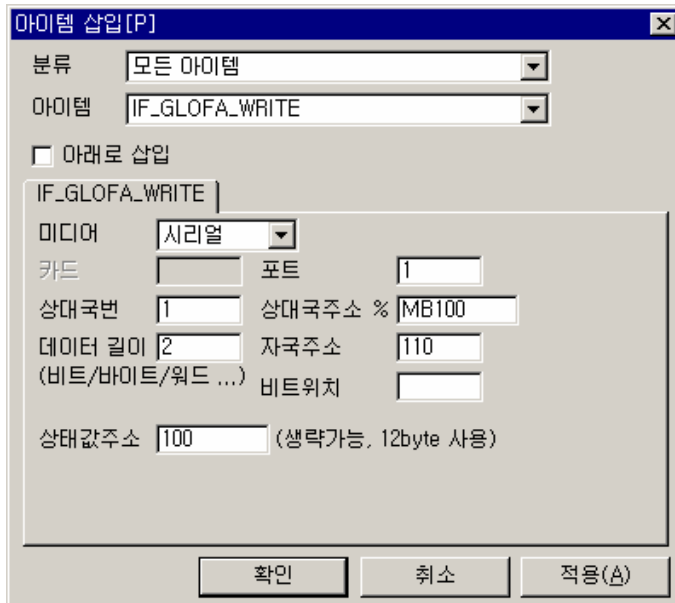


그림 7.6-32 IF_GLOFA_WRITE

표 7.6-31 IF_GLOFA_WRITE 설정 항목

항 목	내 용
미디어	통신 포트의 종류를 선택합니다. (시리얼/이더넷)
카드	시리얼 카드의 번호 설정 (0 ~ 7) ^(*110)
포트	시리얼 포트 번호(1 ~ 8) 또는 이더넷 채널 번호(0 ~ 31)
상대국번	상대방의 국번(시리얼 통신인 경우만 입력)
상대국주소	데이터를 저장할 상대국의 메모리를 직접 변수 형태로 입력합니다.
데이터 길이	쓰기 데이터의 개수로 직접 변수의 타입에 따른 개수를 지정합니다. ^(*111)
자국주소	쓰기 데이터가 저장되어 있는 ETOS의 사용자 메모리 주소를 입력합니다.
비트위치	쓰기 데이터가 비트 형인 경우, 읽을 비트 위치를 지정할 수 있습니다.

(*110) ETOS-200/1000에서만 활성화 됨

(*111) %MW200 과 같이 직접 변수의 타입이 워드형이고 길이가 2 이면 2 워드, 즉, 4 바이트를 씁니다.

표 7.6-31 IF_GLOFA_WRITE 설정 항목 (계속)

항 목	내 용
상태값주소	입력한 바이트 주소 위치에 성공 횟수, 실패 횟수, 에러 코드를 저장함. 입력한 위치로부터 차례대로, 4 바이트 - 성공 횟수, 4 바이트 - 실패 횟수, 4 바이트 - 에러 코드 (GLOFA PLC 에러 코드 표 참조) 를 의미.

* GLOFA 통신 설정에 대한 보다 자세한 사항은 8.시리얼 통신 및 9.이더넷 통신의 GLOFA Master 편을 참조하시기 바랍니다.

7.6.31.2. 예제

생략

7.6.32. IF_IO_STATE_CHANGE

- ▶ 시작 아이템, 응용 아이템 (DIO 전용)
- ▶ ETOS-200/1000 DIO8
- ▶ [IO_READ_WRITE](#), END_IF

7.6.32.1. 기능

IF_IO_STATE_CHANGE 아이템은 ETOS DIO8 모듈 전용 아이템입니다. ETOS DIO8 모듈은 8 점의 TR 출력과 24Volt DC 입력 8 점, 시리얼 2 포트를 가지고 있는 VME Bus 슬레이브 모듈로 ETOS-200/1000 에 장착할 수 있습니다.

IF_IO_STATE_CHANGE 아이템은 입력 8 점의 상태를 감시하여, 그 값이 변하는 즉시 해당 프로세스를 트리거(Trigger)합니다. 프로세스의 선두에 올 수 있는 시작 아이템으로 DIO 카드의 Card ID 와 비트 위치, 동작을 정의합니다.

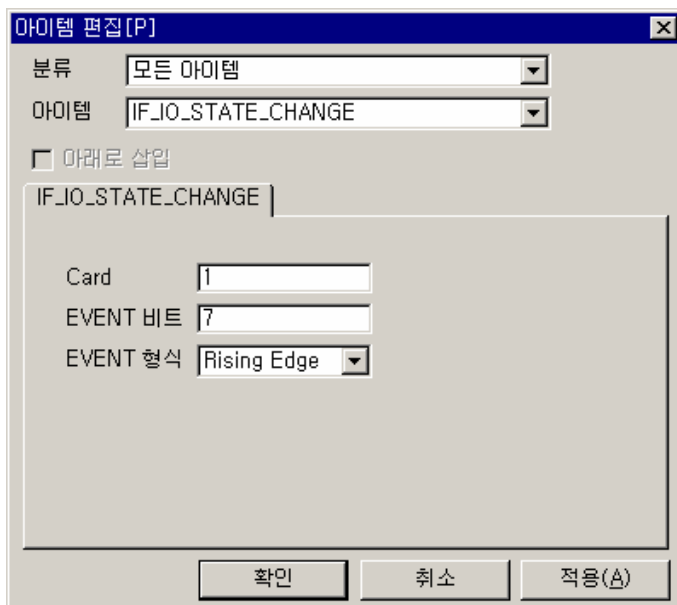


그림 7.6-33 IF_IO_STATE_CHANGE

표 7.6-32 IF_EVENT_OCCUR 설정 항목

항 목	내 용
Card	DIO 보드의 카드 번호를 입력합니다. (0 ~ 7) ^{(*)112}
EVENT 비트	입력 8 점 중 감시하고자 하는 비트를 설정합니다 (0 ~ 7)
EVENT 형식	Falling Edge : 해당 비트가 1 에서 0 으로 바뀔 때, 프로세스가 수행됩니다. Rising Edge : 해당 비트가 0 에서 1 로 바뀔 때, 프로세스가 수행됩니다. Both Edge : 해당 비트가 변경될 때, 프로세스가 수행됩니다.

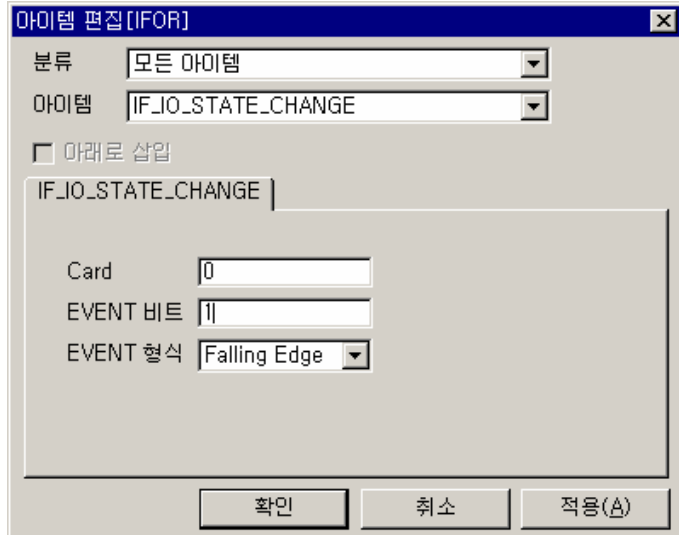
^{(*)112} ETOS-200 의 경우 0/1, ETOS-1000 인 경우 0 ~ 7

7.6.32.2. 예제

Card 0 의 입력중 비트가 Falling 되면 프로세스가 시작됩니다.

▶ 예제 아이템 등록

<Card>는 “0”으로, <EVENT 비트>는 “1”, <EVENT 형식>은 ‘Falling Edge’로 각각 설정합니다.



▶ 예제 프로세스

예제에선 프로세스가 시작되면 SLAVE_READ_WRITE 를 수행하도록 했습니다.

NO	아이템	설명
00	IF.IO.STATE.CHANGE	Card:0, EventBit:1, EventType:Falling
01	IO_READ_WRITE	Card:0, Type:Input, IO:0, ETOS:M100, Size:8, Fu...
02	END_IF	

▶ 결과

DIO 의 1 번 입력이 Falling 되면 프로세스가 시작됩니다.

7.6.33. IF_MELSEC_READ

- ▶ 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_MEMSEC_WRITE](#), ELSE, END_IF

7.6.33.1. 기능

MITSUBISHI MELSEC PLC로부터 데이터를 읽어서 ETOS 메모리에 저장합니다.

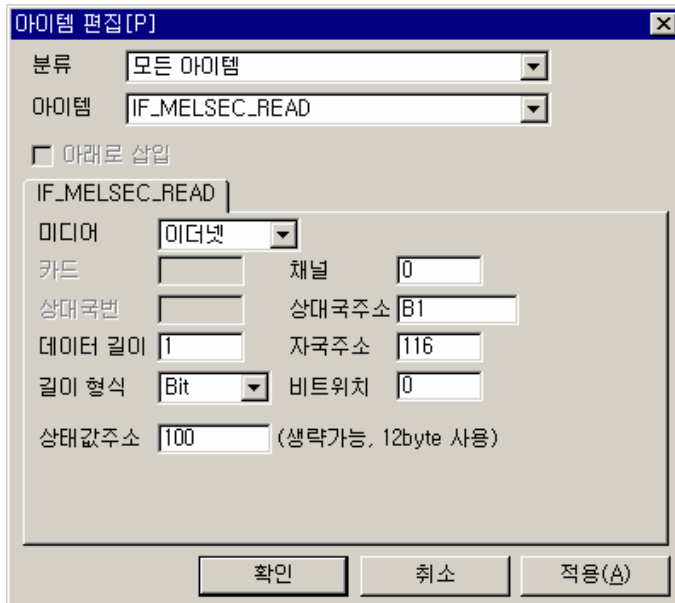


그림 7.6-34 IF_MELSEC_READ 입력 예

표 7.6-33 IF_MELSEC_READ 설정 항목

항 목	내 용
미디어	통신 포트의 종류를 선택합니다. (시리얼/이더넷)
카드	시리얼 카드의 번호 설정 (0 ~ 7) ^(*113)
채널	시리얼 포트 번호(1 ~ 8) 또는 이더넷 채널 번호(0 ~ 31)
상대국번	상대방의 국번(시리얼 통신인 경우만 입력)
상대국주소	데이터를 읽어올 상대국의 메모리를 지정합니다. (디바이스 + 오프셋)
데이터길이	읽을 데이터의 개수로 데이터 형에 따른 개수를 지정합니다. ^(*114)
자국주소	읽어온 데이터를 저장할 ETOS의 사용자 메모리 주소를 입력합니다.
길이 형식	Bit/Word 중에서 선택
비트위치	읽어온 데이터가 비트 형인 경우, 저장 비트 위치를 지정할 수 있습니다.
상태값 주소	입력한 바이트 주소 위치에 성공 횟수, 실패 횟수, 에러 코드를 저장함. 입력한 위치로부터 차례대로, 4 바이트 - 성공 횟수, 4 바이트 - 실패 횟수, 4 바이트 - 에러 코드 (0xCxxx - MELSEC 에러 코드 표 참조) 를 의미.

(*113) ETOS-200/1000에서만 활성화 됨

(*114) 데이터 형이 비트이면 길이 1은 1비트, 워드이면 길이 1은 1워드, 즉 2바이트가 됩니다.

7.6.34. IF_MELSEC_WRITE

- ▶ 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_MELSEC_READ](#), ELSE, END_IF

7.6.34.1. 기능

ETOS 메모리의 데이터를 읽어 MITSUBISHI MELSEC PLC 디바이스 메모리에 씁니다.

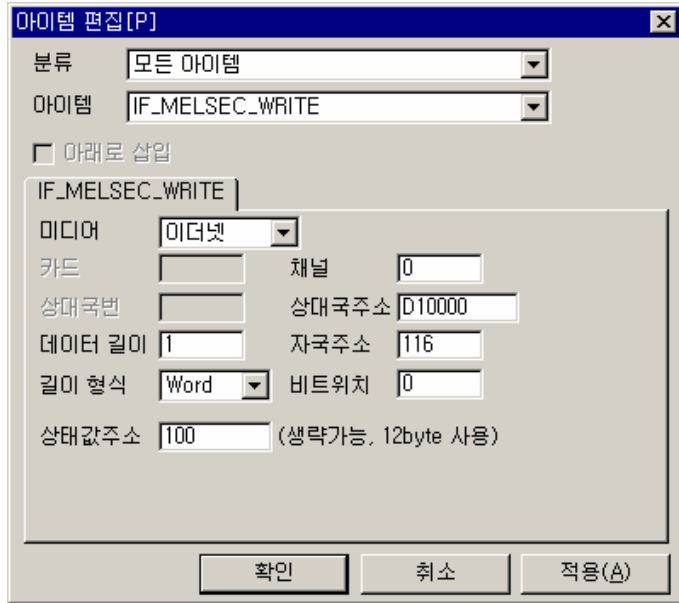


그림 7.6-35 IF_MELSEC_WRITE 입력 예

표 7.6-34 IF_MELSEC_READ 설정 항목

항 목	내 용
미디어	통신 포트의 종류를 선택합니다. (시리얼/이더넷)
카드	시리얼 카드의 번호 설정 (0 ~ 7) ^(*115)
채널	시리얼 포트 번호(1 ~ 8) 또는 이더넷 채널 번호(0 ~ 31)
상대국번	상대방의 국번(시리얼 통신인 경우만 입력)
상대국주소	데이터를 저장할 상대국의 메모리를 지정합니다. (디바이스 + 오프셋)
길이	쓰기 데이터의 개수로 데이터 형에 따른 개수를 지정합니다. ^(*116)
자국주소	쓰기 데이터가 저장되어 있는 ETOS의 사용자 메모리 주소를 입력합니다.
길이 형식	Bit/Word 중에서 선택
비트위치	쓰기 데이터가 비트 형인 경우, 읽어들 비트 위치를 지정할 수 있습니다.
상태값 주소	입력한 바이트 주소 위치에 성공 횟수, 실패 횟수, 에러 코드를 저장함. 입력한 위치로부터 차례대로, 4 바이트 - 성공 횟수, 4 바이트 - 실패 횟수, 4 바이트 - 에러 코드 (0xCxxx - MELSEC 에러 코드 표 참조) 를 의미.

(*115) ETOS-200/1000 에서만 활성화 됨

(*116) 데이터 형이 비트이면 길이 1은 1 비트, 워드이면 길이 1은 1 워드, 즉 2 바이트가 됩니다.

7.6.35. IF_MEM_CMP

- ▶ 조건분기 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_NUM_CMP](#), [IF_STR_CMP](#), ELSE, END_IF

7.6.35.1. 기능

사용자 메모리 두 곳의 데이터를 비교합니다. 지정한 두 곳의 데이터가 동일하면, IF_MEM_CMP ~ ELSE^(*117) 또는 IF_MEM_CMP ~ END_IF^(*118) 블록안의 아이템을 수행하고, 일치하지 않는 경우는 ELSE ~ END_IF^(*) 블록 아이템을 수행합니다.

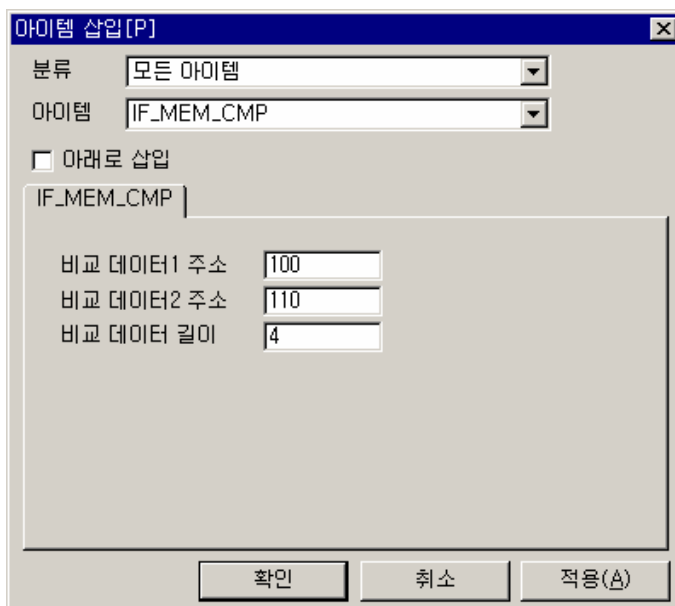


그림 7.6-36 IF_MEM_CMP 입력 예

표 7.6-35 IF_MEM_CMP 설정 항목

항 목	내 용
비교 데이터 1 주소	첫번째 비교 대상 주소를 입력합니다.
비교 데이터 2 주소	두번째 비교 대상 주소를 입력합니다.
비교 데이터 길이	비교할 데이터의 길이를 입력합니다. (1 ~ 65535)

(*117) ELSE 아이템을 IF_MEM_CMP 와 END_IF 사이에 삽입한 경우

(*118) ELSE 아이템을 삽입하지 않은 경우

7.6.35.2. 예제

ETOS 메모리 100 번지와 110 번지를 비교하여 데이터가 같은 경우 120 번지 값을 증가하도록 합니다.

▶ 예제 아이템 등록

IF_TIMER 아이템을 추가하고, 주기를 100 으로 입력합니다. (1 초 주기)

MEM_SET 아이템을 삽입하고, <데이터 형태> '16 진수', <데이터> "01", <주소> "100", <길이> "1"을 각각 입력합니다.

MEM_SET 아이템을 삽입하고, <데이터 형태> '16 진수', <데이터> "01", <주소> "110", <길이> "1"을 각각 입력합니다.

IF_MEM_CMP 아이템을 추가하고, 다음과 같이 입력합니다.

항 목	입력 내용
비교 데이터 1 주소	100
비교 데이터 2 주소	110
비교 데이터 길이	4

ARITHMETIC 아이템을 삽입하고, <데이터 형식> 'Unsigned 1 Byte', <데이터 1 의 형태> '메모리', <데이터 2 의 형태> '상수', <데이터 1>에 "120", <데이터 2>에 "1", <결과(주소)>에 "120"을 각각 입력합니다.

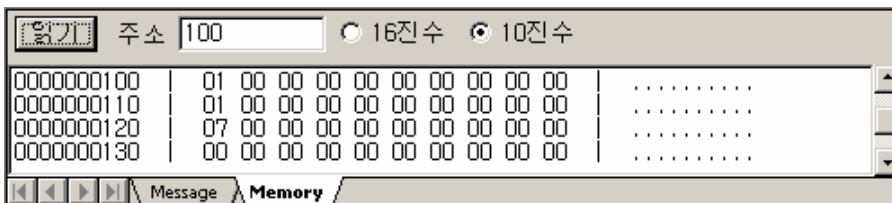
▶ 예제 프로세스

편집을 마친 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	100
01	MEM_SET	T:M100, Size:1, Type:16진수, Data:01
02	MEM_SET	T:M110, Size:1, Type:16진수, Data:01
03	IF_MEM_CMP	M100, M110, Size:4
04	ARITHMETIC	Type:U1, M120, +, 1, M120
05	END_IF	
06	END_IF	

▶ 결과

100 번지와 110 번지에는 각각 01 이 저장되어 있습니다. 두 번지를 IF_MEM_CMP 아이템으로 비교하면 같으므로 IF_MEM_CMP ~ END_IF 블록이 실행됩니다. 결과적으로 120 번지가 1 초 마다 1 씩 증가합니다.



7.6.36. IF_MEM_WRITE_RCV

- ▶ 시작 아이템, 통신 아이템 ((시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [MEM_WRITE_RESP_SND](#), END_IF

7.6.36.1. 기능

시리얼 슬레이브 또는 이더넷 드라이버로 지정된 포트의 메모리 쓰기 이벤트를 검출합니다. 통신 상대국이 ETOS의 사용자 메모리에 쓰기를 시도하는 경우(ETOS가 상대국의 메모리 쓰기 프레임임을 수신 하는 경우), 쓰기 이벤트를 발생시켜 원하는 동작을 취할 수 있습니다. 또, 드라이버 쓰기 이벤트가 발생할 메모리의 위치와 범위를 설정해서, 원하는 범위의 메모리만 감시할 수 있습니다. 즉, 수신 프레임의 쓰기 위치가 감시 범위안에 해당해야 이벤트가 발생합니다.

해당하는 시리얼 또는 이더넷 드라이버 목록^(*119)은 다음과 같습니다.

- ▶ 시리얼
 - * MODBUS RTU/ASCII 슬레이브
 - * GLOFA Cnet 슬레이브
- ▶ 이더넷
 - * MODBUS TCP 드라이버
 - * GLOFA Enet 드라이버

IF_MEM_WRITE_RCV 아이템은 내장 프로토콜 슬레이브로 동작하는 경우만 사용합니다. 반면에 또다른 프레임 수신 아이템인 IF_FRAME_RCV는 사용자 정의 프레임을 수신하는 경우에 사용됩니다.

또, IF_FRAME_RCV의 목적은 정의한 프레임을 수신하여 데이터를 저장하는 것이지만, IF_MEM_WRITE_RCV의 목적은 프레임 수신 및 데이터 저장이 아닙니다. 프레임 수신 및 데이터 저장은 해당 포트를 슬레이브로 설정하기만 하면(파라미터 설정만으로) 자동으로 이루어지는 기능이기 때문입니다. 프로세스 개입없이 파라미터 설정만으로 프레임 수신 및 데이터 저장이 이루어지므로 이번에는 언제 데이터가 저장되는지 또, 저장되는 데이터의 위치와 길이 등은 프로세스에서 알 수 없는 단점이 있습니다.

- ▶ 이 아이템의 목적은 슬레이브 포트로 수신되는 데이터의 수신 시점과, 위치, 길이를 프로세스에서 참조할 수 있도록 하는 것입니다.

조건을 만족하는 프레임을 슬레이브(또는 드라이버)가 수신하는 경우 이벤트를 발생시켜, IF_MEM_WRITE_RCV로 시작하는 프로세스를 기동시킵니다. 즉, 조건을 만족하는 프레임의 수신 시점을 알 수 있다는 것입니다. 또한, 프로세스가 기동될 때, 데이터의 위치 및 길이도 이미 저장되어 프로세스의 다른 아이템에서 참조할 수 있습니다.

- ▶ 슬레이브 포트가 수신하는 프레임 중 쓰기 명령에 해당하는 프레임만 이벤트를 발생시킬 수 있습니다. 즉, 상대국이 ETOS 메모리를 읽는 경우에는 이벤트가 발생하지 않고, 오직 ETOS의 메모리에 쓰기를 하는 경우만 그 데이터의 위치 및 크기, 시점등을 알 수 있습니다.

(*119) OS 버전 1.20 기준이며, 드라이버 목록은 추가될 수 있습니다.

IF_MEM_WRITE_RCV 는 시작 아이템입니다. 특정 감시 영역안에 쓰기가 발생할 경우, 해당 프로세스를 수행하게 됩니다. 다음은 IF_MEM_WRITE_RCV 아이템이 동작하는 개념을 설명한 그림입니다.

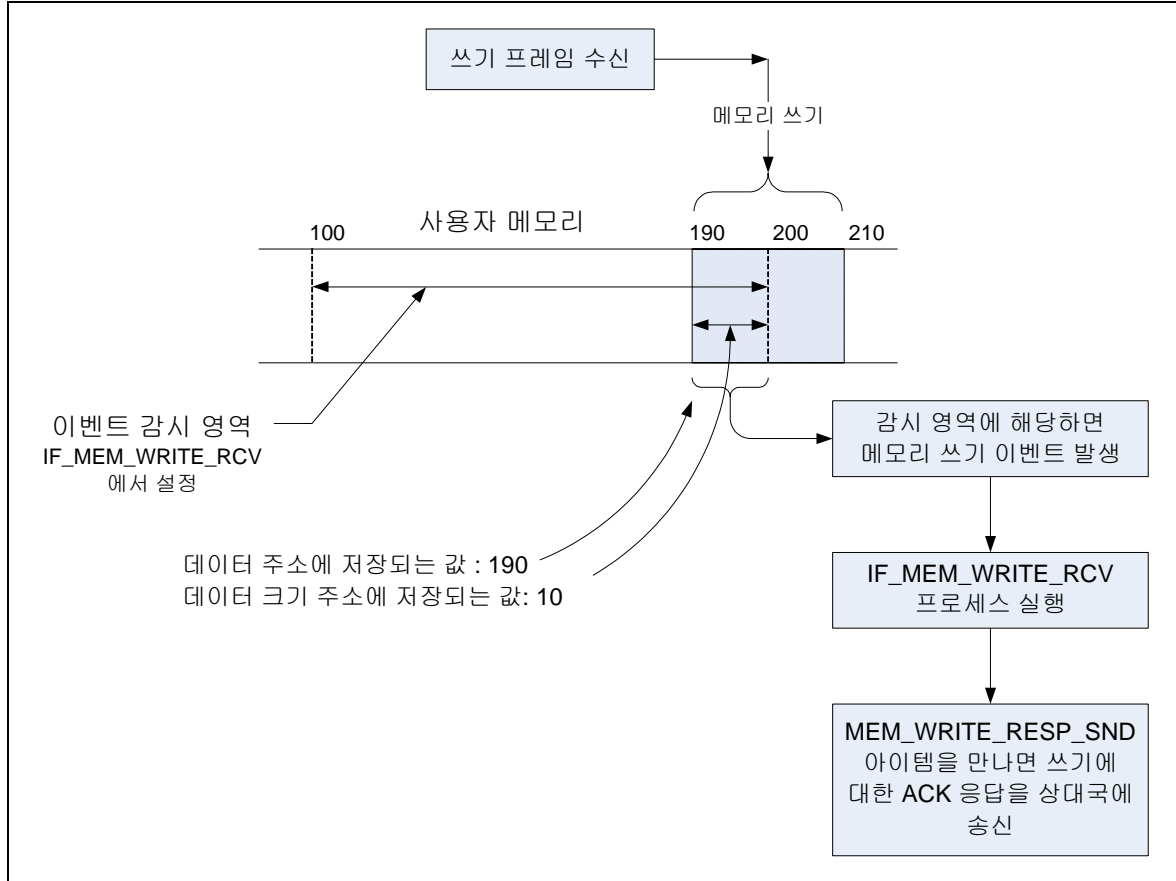


그림 7.6-37 IF_MEM_WRITE_RCV 동작 원리

IF_MEM_WRITE_RCV 아이템은 MEM_WRITE_RESP_SND 아이템과 반드시 짝을 이루어야 합니다. MEM_WRITE_RESP_SND 아이템은 드라이버 쓰기 수신 프레임에 대한 응답 프레임을 상대국에 전송하는 명령입니다. 만약, MEM_WRITE_RESP_SND 아이템이 빠지면, ETOS 메모리에 데이터가 써 지기는 하지만 응답을 보내지 않으므로, 상대국은 에러를 발생시킬 수 있습니다.

- * 감시 영역 밖에 쓰기를 할 경우, MEM_WRITE_RESP_SND 를 수행하지 않아도 자동으로 응답을 하고, 쓰기 데이터는 메모리에 써 집니다.
- * 감시 영역에 쓰기를 하는 경우, MEM_WRITE_RESP_SND 를 수행하지 않아도 메모리에 데이터가 써 집니다.
- * MEM_WRITE_RESP_SND 를 실행하지 않는 예
 상위장비에서 쓰기-> 하위장비에 쓰기 -> 응답 없음(쓰기 에러) -> MEM_WRITE_RESP_SND 생략 -> 상위장비에서 수신 타임아웃(쓰기 에러) 발생하여 에러 처리를 할 수 있음(다시 쓰기 시도를 한다던지 하는)

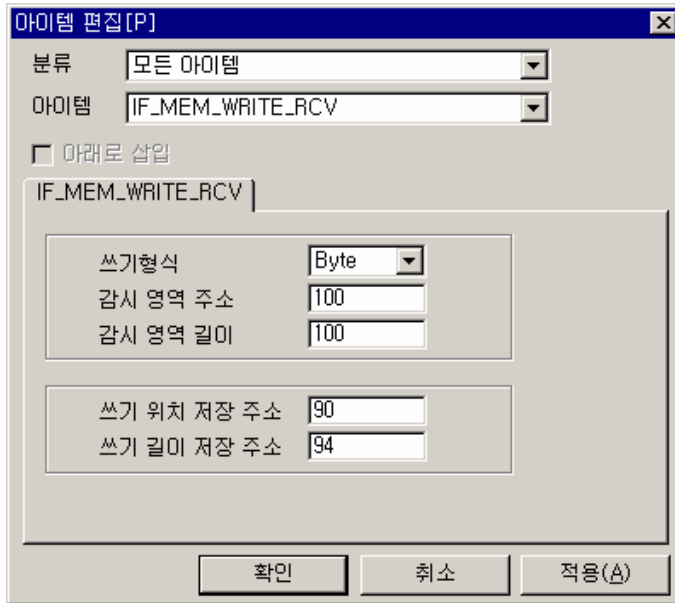


그림 7.6-38 IF_MEM_WRITE_RCV 입력 예

표 7.6-36 IF_MEM_WRITE_RCV 설정 항목

항 목	내 용
쓰기 형식	비트 쓰기/바이트 쓰기를 구분합니다.
감시 영역 주소	메모리 쓰기 이벤트를 발생시킬 감시 영역의 시작 주소를 입력합니다.
감시 영역 길이	감시 영역의 크기를 바이트 단위로 입력합니다.
쓰기 위치 저장 주소	수신 프레임의 데이터가 ETOS 메모리 어디에 저장되는지, 그 위치가 기록됩니다. (지정 번지부터 4 바이트에 걸쳐 위치가 기록됩니다.)
쓰기 길이 저장 주소	수신 프레임이 ETOS 메모리에 쓰는 데이터의 길이가 저장됩니다. ^(*120) 비트 쓰기일 경우는 비트의 위치가 저장됩니다. (지정 번지부터 2 바이트에 걸쳐 크기나 비트 위치가 기록됩니다.)

^(*120) 수신 프레임의 데이터 크기가 아니고, 수신 프레임의 데이터 중 감시 영역에 해당하는 부분의 크기입니다.

7.6.36.2. 예제

상대국이 ETOS 메모리에 쓰기를 시도하는 경우, 그 데이터는 ETOS 메모리의 어디인가에 저장되지만 프로세스는 그 위치와 크기를 알 수 없습니다. IF_MEM_WRITE_RCV 아이템을 이용하여 그 데이터가 저장되는 위치와 크기를 알아내어 특정 메모리 위치에 복사합니다. 결과적으로, 드라이버 수신 프레임의 데이터가 저장되는 위치와 크기를 이용하여, 수신 데이터에 대해서 변환 등의 원하는 조작을 할 수 있습니다.

▶ **예제 아이템 등록**

IF_MEM_WRITE_RCV 아이템을 추가하고, 다음과 같이 입력합니다.

항 목	입력 내용
쓰기 형식	Byte 선택
감시 영역 주소	100
감시 영역 길이	100
쓰기 위치 저장 주소	90 (4 바이트 사용)
쓰기 길이 저장 주소	94 (2 바이트 사용)

수신 데이터의 위치는 데이터 주소 90 번지에, 수신 데이터의 길이는 데이터 크기 주소 94 번지에 저장됩니다. 위치와 크기를 알 수 있으므로, 수신 데이터를 특정한 위치(50 번지)로 복사해 두겠습니다.

MEM_CPY 아이템을 추가하고, 다음과 같이 입력합니다.

항 목	입력 내용
형태	Memory 선택
복사할 데이터(주소)	90, 간접주소 체크
복사할 데이터 길이	94, 간접길이 체크
복사될 데이터 주소	50

마지막으로 MEM_WRITE_RESP_SND 아이템을 반드시 삽입해야 합니다. MEM_WRITE_RESP_SND 아이템은 앞에서 설명한 바와 같이 상대국의 쓰기에 대한 응답을 송신하는 역할을 합니다. 따라서, MEM_WRITE_RESP_SND 이 생략되면, 상대국의 쓰기 데이터는 정상적으로 ETOS 메모리에 저장되지만, ACK 응답을 하지 않으므로, 상대국은 에러를 발생할 수 있습니다. 주의하시기 바랍니다.

▶ **예제 프로세스**

편집을 마친 프로세스 화면입니다. 등록된 프로세스의 순서를 확인하시기 바랍니다.

NO	아이템	설명
00	IF_MEM_WRITE_RCV	Type:Byte, Address:M100, Size:100, DataAddr:M90, DataSize:M94
01	MEM_CPY	S:M90, T:M50, Size:M94
02	MEM_WRITE_RESP_SND	
03	END_IF	

▶ 결과

상대국 쓰기 프레임이 IF_MEM_WRITE_RCV 아이템의 조건을 만족하면 IF_MEM_WRITE_RCV 로 시작하는 해당 프로세스가 실행되고, 90 번지, 94 번지에 각각 수신 데이터의 위치와 길이를 자동으로 저장합니다.

다음 아이템인 MEM_CPY 가 실행됩니다. 간접 지정을 했으므로 복사 대상 데이터의 주소와 길이는 90 번지, 94 바이트가 아니라 90 번지, 94 번지에 저장되어 있는 값입니다. 그 값은 앞에서 IF_MEM_WRITE_RCV 아이템이 실행되면서 자동으로 저장된다고 이미 설명했습니다. 이 값을 참조하여 수신 데이터를 50 번지에 복사하게 됩니다.

90번지, 94번지 값을 참조하여
50번지에 수신 데이터 복사

90번지, 수신 데이터 주소(4바이트):
0x00000078 = 120번지

94번지, 수신 데이터 길이(2바이트):
0x0008 = 8바이트

주소	16진수	10진수
000000050	32 32 32 32 33 33 33 33 00 00	22223333..
000000060	00 00 00 00 00 00 00 00 00 00
000000070	00 00 00 00 00 00 00 00 00 00
000000080	00 00 00 00 00 00 00 00 00 00
000000090	78 00 00 00 08 00 00 00 00 00	x.....
000000100	00 00 00 00 00 00 00 00 00 00
000000110	00 00 00 00 00 00 00 00 00 00
000000120	32 32 32 32 33 33 33 33 00 00	22223333..
000000130	00 00 00 00 00 00 00 00 00 00

120번지, 실제 수신 데이터

7.6.37. IF_MODBUS_READ

- ▶ 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_MODBUS_WRITE](#), ELSE, END_IF

7.6.37.1. 기능

Modicon 사의 MODBUS 프로토콜을 사용하는 상대국으로부터 데이터를 읽어서, ETOS 메모리에 저장합니다. 이 아이템은 시리얼과 이더넷에 모두 사용할 수 있습니다. 시리얼의 경우, RTU(Binary) 모드와 ASCII 모드를 모두 지원합니다.

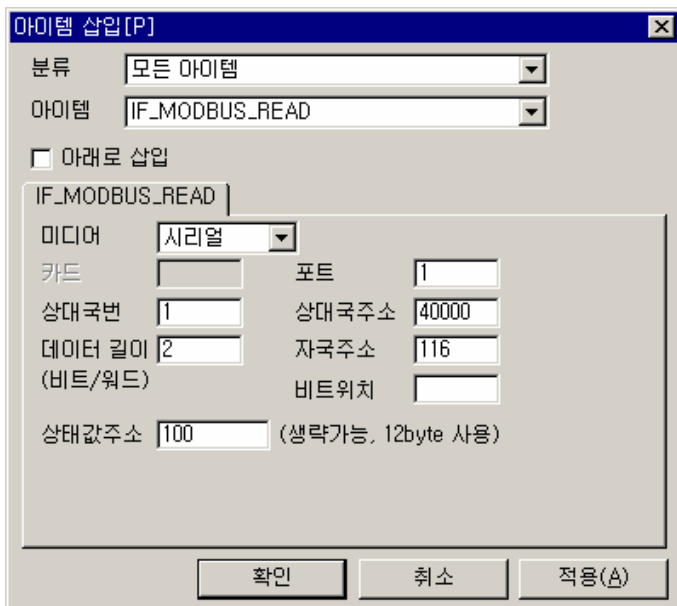


그림 7.6-39 IF_MODBUS_READ 입력 예

표 7.6-37 IF_MODBUS_READ 설정 항목

항 목	내 용
미디어	통신 미디어를 선택합니다. (시리얼 / 이더넷)
카드	시리얼 카드의 번호 설정 (0 ~ 7) ^(*121)
포트	시리얼 포트 번호(1 ~ 8) 또는 이더넷 채널 번호(0 ~ 31)
상대국번	상대방의 국번
상대국주소	데이터를 읽어올 상대국의 메모리를 지정합니다. 비트 출력 영역 : 0xxxx 비트 입력 영역 : 1xxxx 워드 입력 영역 : 3xxxx 워드 출력 영역 : 4xxxx 비트 출력 영역의 첫 번째 비트는 MODBUS 사양에서는 00001로 정의하고 있으나, ETOS는 이 번지를 00000으로 정의합니다. 마찬가지로 워드 입력 영역의 첫 워드 30001번지는 ETOS에서는 30000으로 표현됩니다.

^(*121) ETOS-200/1000에서만 활성화 됨

표 7.6-37 IF_MODBUS_READ 설정 항목 (계속)

항 목	내 용
길이	읽고자 하는 데이터의 개수를 설정합니다. 비트 영역 읽기인 경우에는, 입력한 수치는 비트의 개수를 의미하며, 워드 영역 읽기인 경우에는 워드의 개수를 의미합니다.
자국주소	상대국으로부터 읽어온 데이터를 저장할 ETOS 의 바이트 주소를 설정합니다. 응답 데이터를 저장할 ETOS 메모리 주소 를 입력합니다. (ETOS-50/100/150 : 0~256K, ETOS-200/1000 : 0~512K)
비트위치	상대국으로부터 읽어온 데이터를 저장할 비트 위치입니다. 상대국 주소가 비트 번지인 경우만 유효합니다. 비트 데이터를 저장할 ETOS 의 주소는 [자국주소 + 비트 위치] 형태로 지정합니다. 예를 들어 자국 주소에 5000 을 설정하고, 비트 위치에 3 을 설정했다면, 읽어온 데이터의 첫 번째 비트는 M5000.3 위치에 기록될 것입니다. 워드 읽기에서는 지정해도 의미가 없습니다.
상태값주소	입력한 바이트 주소 위치에 성공 횟수, 실패 횟수, 에러 코드를 저장함. 입력한 위치로부터 차례대로, 4 바이트 - 성공 횟수, 4 바이트 - 실패 횟수, 4 바이트 - 에러 코드 (MODBUS 에러 코드 표 참조) 를 의미.

7.6.37.2. 예제

내장 프로토콜 아이템인 IF_MODBUS_READ 아이템을 이용하여 Modicon PLC 의 데이터를 읽습니다.

▶ 예제 아이템 등록

내장 프로토콜 아이템을 사용하기 전에 파라미터 설정에서 관련 설정이 되어 있어야 합니다. 보다 자세한 사항은 [8.1.3. 내장 프로토콜 설정](#)을 참조하시기 바랍니다.

IF_TIMER 아이템을 추가하고, <주기>를 “100”으로 입력합니다. (1 초 주기)

IF_MODBUS_READ 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
미디어	시리얼 선택
카드	ETOS-200/1000 인 경우만 0 입력
포트	1
상대국번	1
상대국주소	40000
길이	2
자국주소	116
비트위치	
상태값주소	100

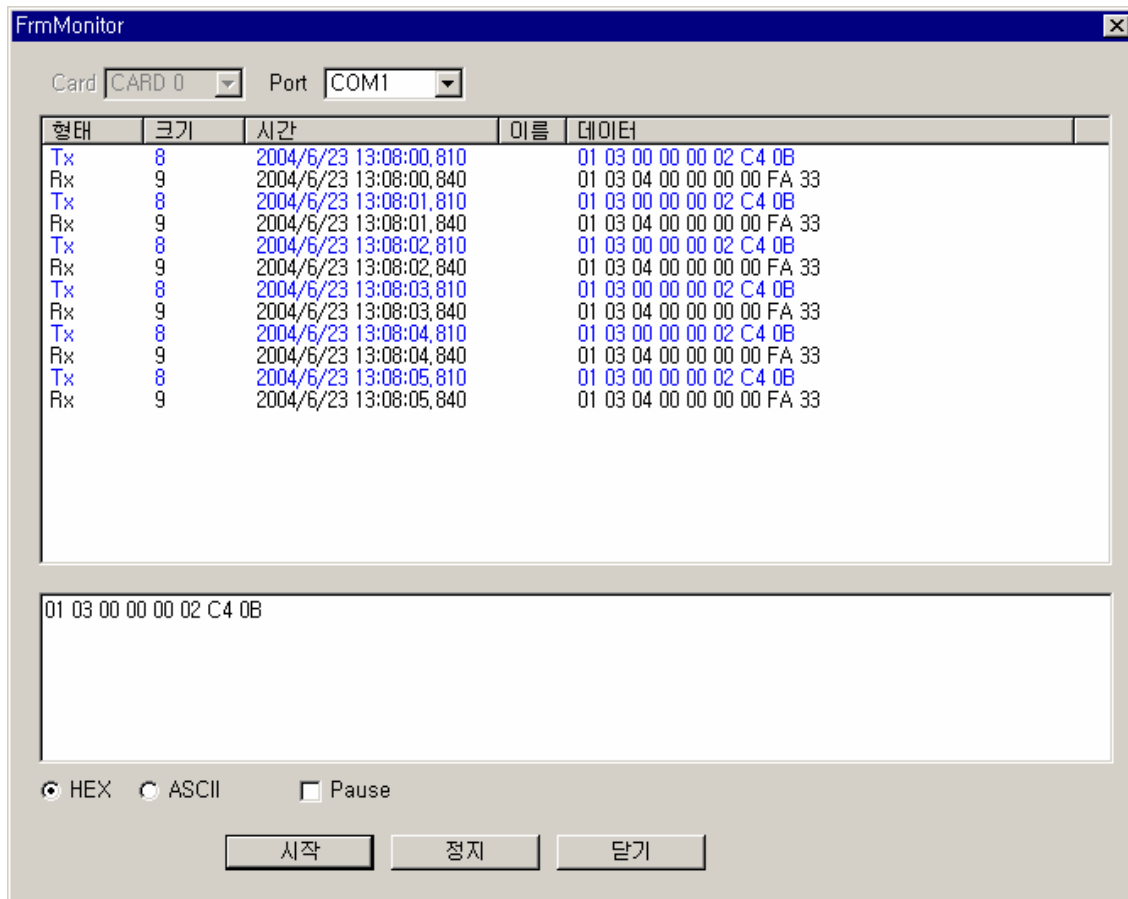
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	100
01	IF_MODBUS_READ	Media:Serial, COM1, DestSt:1, DestAddr:40000, Length:2, SelfAddr:M116,..
02	END_IF	
03	END_IF	

▶ 결과

[온라인] -> [프레임 모니터]를 선택합니다. <포트>를 'COM1'으로 선택하고 [시작] 버튼을 누릅니다. COM1 포트를 통해 8 바이트의 프레임이 1 초 주기로 송신 되고 상대방도 응답하고 있음을 알 수 있습니다.



- * 송신 프레임 설명
- 01 = 국번
- 03 = FUNCTION CODE
- 00 00 = ADDRESS
- 00 02 = SIZE
- C4 0B = MODBUS CRC

7.6.38. IF_MODBUS_WRITE

- ▶ 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_MODBUS_READ](#), ELSE, END_IF

7.6.38.1. 기능

ETOS 메모리의 데이터를 읽어 Modicon 사의 MODBUS 프로토콜을 사용하는 상대국 메모리에 씁니다. 이 아이템은 시리얼과 이더넷에 모두 사용할 수 있습니다. 시리얼의 경우, RTU(Binary) 모드와 ASCII 모드를 모두 지원합니다.

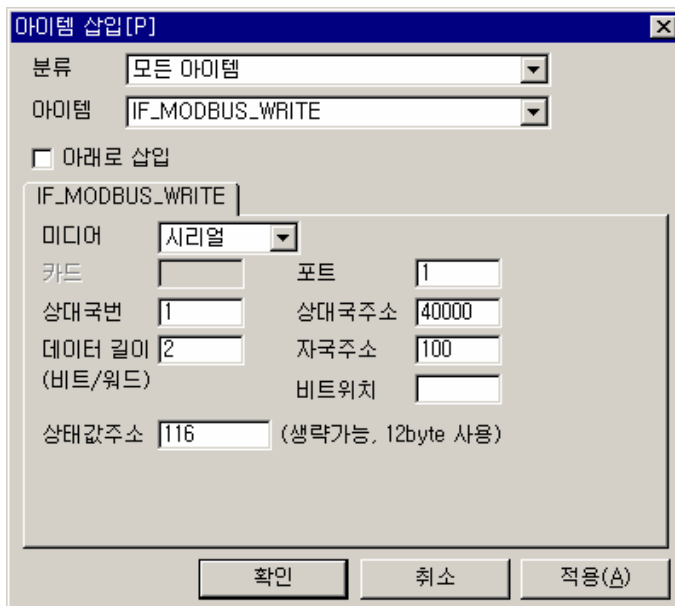


그림 7.6-40 IF_MODBUS_WRITE 입력 예

표 7.6-38 IF_MODBUS_WRITE 설정 항목

항 목	내 용
미디어	통신 미디어를 선택합니다. (시리얼 / 이더넷)
카드	시리얼 카드의 번호 설정 (0 ~ 7) ^(*122)
포트	시리얼 포트 번호(1 ~ 8) 또는 이더넷 채널 번호(0 ~ 31)
상대국번	상대방의 국번
상대국주소	데이터를 저장할 상대국의 메모리를 지정합니다. 비트 출력 영역 : 0xxxx 워드 출력 영역 : 4xxxx 쓰기는 출력 영역에만 할 수 있습니다. 비트 출력 영역의 첫 번째 비트는 MODBUS 사양에서는 00001 로 정의하고 있으나, ETOS 는 이 번지를 00000 으로 정의합니다. 마찬가지로 워드 출력 영역의 첫 워드 40001 번지는 ETOS 에서 40000 으로 표현됩니다.

(*122) ETOS-200/1000 에서만 활성화 됨

표 7.6-38 IF_MODBUS_READ 설정 항목 (계속)

항 목	내 용
길이	쓰기 데이터의 길이를 설정합니다. 비트 영역 쓰기인 경우에는, 입력한 수치는 비트의 개수를 의미하며, 워드 영역 쓰기인 경우에는 워드의 개수를 의미합니다.
자국주소	쓰기 데이터가 저장되어 있는 ETOS의 사용자 메모리 주소를 입력합니다.
비트위치	쓰기 데이터가 저장되어 있는 비트 위치입니다. 상대국 주소가 비트 번지인 경우만 유효합니다. 비트 데이터의 주소는 [자국주소 + 비트 위치] 형태로 지정합니다. 예를 들어 자국 주소에 5000 을 설정하고, 비트 위치에 3 을 설정했다면, 데이터의 첫 번째 비트는 M5000.3 위치에서 읽어옵니다. 워드 쓰기에서는 지정해도 의미가 없습니다.
상태값주소	입력한 바이트 주소 위치에 성공 횟수, 실패 횟수, 에러 코드를 저장함. 입력한 위치로부터 차례대로, 4 바이트 - 성공 횟수, 4 바이트 - 실패 횟수, 4 바이트 - 에러 코드 (MODBUS 에러 코드 표 참조) 를 의미.

7.6.38.2. 예제

생략

7.6.39. IF_NUM_CMP

- ▶ 조건분기 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_MEM_CMP](#), [IF_STR_CMP](#), ELSE, END_IF

7.6.39.1. 기능

지정된 위치(비교할 주소)의 값을 상수 또는 또다른 위치의 값과 비교합니다. IF_MEM_CMP 아이템이 메모리 대 메모리의 비교라면, IF_NUM_CMP는 상수 대 메모리 또는 메모리 대 메모리 비교입니다. 또, IF_MEM_CMP는 같은지 여부만 알 수 있지만, IF_NUM_CMP는 비교 연산자(=, >, <, <>, >=, <=)를 통해 크기 비교가 가능합니다. 이 아이템은 반드시 END_IF 아이템과 짝을 이루어야 하며, ELSE 아이템을 사용할 수 있습니다.

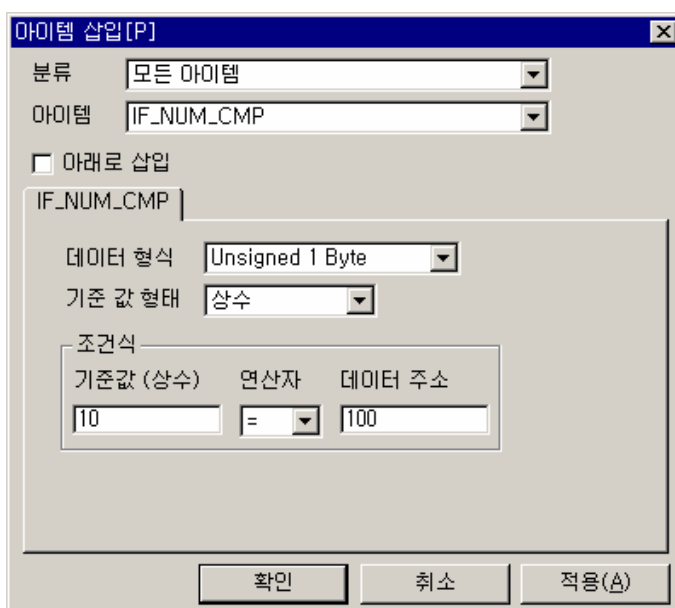


그림 7.6-41 IF_NUM_CMP 입력 예

표 7.6-39 IF_NUM_CMP 설정 항목

항 목	내 용
데이터 형식	비교할 데이터의 형식을 지정합니다.
기준 값 형태	첫번째 비교 대상의 형태를 지정합니다. (메모리/상수)
기준값	기준값 형태의 선택에 따라서, 메모리 : 입력한 값은 첫번째 비교 대상의 위치(주소)가 되며, 상수 : 입력한 값은 첫번째 비교 값(상수)가 됩니다.
연산자	비교 연산자(=, >, <, <>, >=, <=)를 선택합니다. 선택 된 연산자에 따라 = (같다), > (크다), < (작다), <> (같지 않다), >= (크거나 같다), <= (작거나 같다)의 동작을 수행합니다.
비교할 주소	두번째 비교 대상의 위치(번지)를 지정합니다.

7.6.39.2. 예제

메모리 100 번지를 1 씩 증가시키고 100 번지 값이 10 이상이면 다시 1 부터 증가하도록 합니다.

▶ 예제 아이템 등록

IF_TIMER 아이템을 추가하고, <주기>는 “100”을 입력합니다. (1 초 주기)

먼저 ARITHMETIC 아이템을 삽입하고, 100 번지에 1 을 더해 100 번지에 저장하도록 입력합니다.

IF_NUM_CMP 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
데이터 형식	Unsigned 1 Byte 선택
기준 값 형태	상수 선택
기준값	비교 기준값으로 10 입력
연산자	= 선택
비교할 주소	100

아이템의 의미는 100 번지 값이 10 과 같아지면... 의 의미입니다.

다시 MEM_SET 아이템을 삽입하고, <데이터 형태> '16 진수', <데이터> “01”, <주소> “100”, <길이> “1”을 입력합니다.

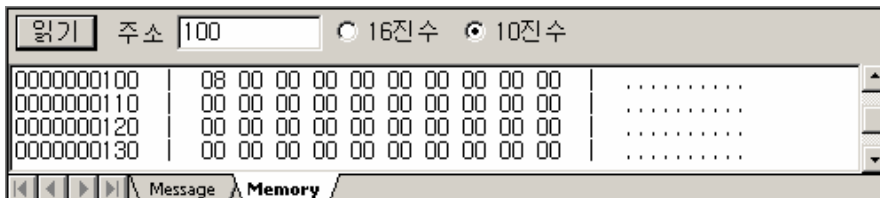
▶ 예제 프로세스

편집을 마친 프로세스 화면입니다.

NO	아이템	설명
00	IF_TIMER	100
01	ARITHMETIC	Type:U1, M100, +, 1, M100
02	IF_NUM_CMP	Type:U1, 10, =, M100
03	MEM_SET	T:M100, Size:1, Type:16진수, Data:01
04	END_IF	
05	END_IF	

▶ 결과

100 번지가 1 초마다 1 씩 증가하는 것을 볼 수 있습니다. IF_NUM_CMP 아이템은 그 값이 10(0A) 이 되면 MEM_SET 아이템으로 100 번지에 1 을 쓰게 합니다. 그러므로 100 번지는 1~9 까지 증가를 반복하며 100 번지 값은 10 을 넘지 못하게 됩니다.



7.6.40. IF_POWER_ON

- ▶ 시작 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ END_IF

7.6.40.1. 기능

ETOS의 전원을 켜 후 또는 파라미터나 프로그램 쓰기 등으로 리셋된 후 다른 프로세스에 앞서 한번만 실행되는 아이템입니다. 다른 프로세스는 IF_POWER_ON으로 시작하는 프로세스(이하 IF_POWER_ON 프로세스)가 끝날때까지 실행되지 않습니다. 따라서, IF_POWER_ON 프로세스 안에 통신 아이템(ENET_OPEN, IF_REQ_RESP) 등을 넣어놓는 경우, 통신 상대방과 통신이 두절되는 상황(예를 들면, 상대방의 전원이 꺼져 있는 경우 등)이 발생하면, 통신 지연이 발생하고, 다른 프로세스의 수행에도 지장을 초래할 우려가 있습니다.

주로 메모리 초기화 등의 용도로만 사용할 것을 권장합니다.

이 아이템은 반드시 END_IF 아이템과 짝을 이루어야 하며, ELSE 아이템은 사용할 수 없습니다. 또한, 프로세스의 선두에만 올 수 있는 시작 아이템으로 프로세스 중간에는 사용할 수 없으며, 프로그램을 통털어 IF_POWER_ON 프로세스는 1개만 등록할 수 있습니다.

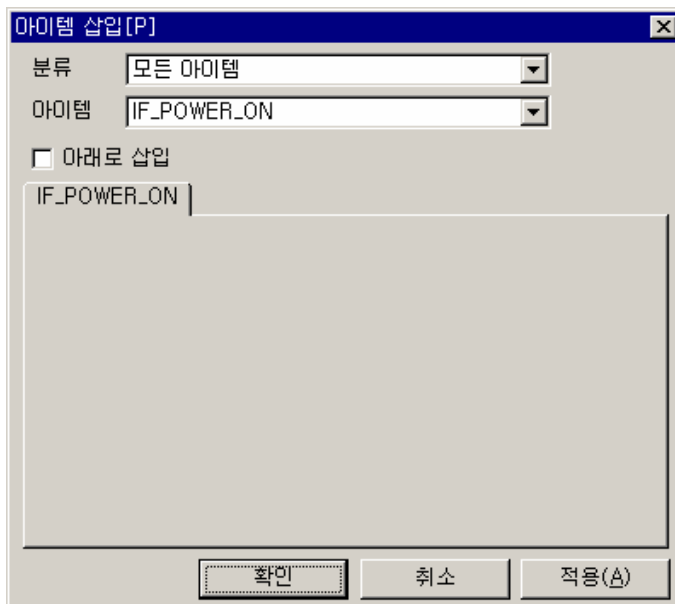


그림 7.6-42 IF_POWER_ON 아이템

표 7.6-40 IF_POWER_ON 설정 항목

항 목	내 용
없음	설정 항목이 없습니다

7.6.40.2. 예제

ETOS 전원을 켜는 경우, 메모리를 초기화하여 다른 프로세스에서 사용할 수 있도록 합니다.
 (메모리 초기화는 일반적으로 IF_POWER_ON 프로세스에서 수행합니다.)

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 시작아이템으로 등록합니다.

MEM_SET 아이템으로 100 번지부터 10 바이트를 31_H 로, 110 번지부터 10 바이트를 32_H 로 설정합니다.

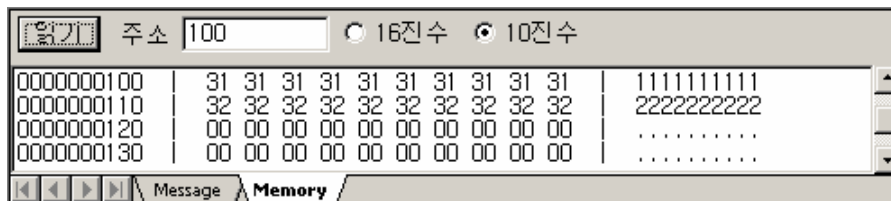
▶ **예제 프로세스**

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:10, Type:문자열, Data:1
02	MEM_SET	T:M110, Size:10, Type:문자열, Data:2
03	END_IF	

▶ **결과**

100 번지부터 10 바이트, 110 번지부터 10 바이트가 각각 31_H, 32_H로 초기화 되었습니다. 앞에서 설명한 바와 같이 IF_POWER_ON 에서 초기화하면, 다른 모든 프로세스보다 우선하여 실행되므로, 이후 어떤 프로세스에서도 초기화된 메모리 값을 참조할 수 있습니다.



7.6.41. IF_REQ_RESP

- ▶ 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [FRAME_SND](#), [IF_FRAME_RCV](#), ELSE, END_IF

7.6.41.1. 기능

ETOS 가 마스터가 되어 상대국과 사용자 정의 프레임으로 통신을 하는 경우에 사용합니다. (내장 프로토콜에 의한 통신은 프로토콜 별로 통신 아이템이 지정되어 있습니다.) 이더넷 및 범용 시리얼 통신에 모두 사용할 수 있으며, 프로세스 작성 전에 프레임을 미리 정의해야 합니다.

ETOS 가 상대국에 요구 프레임을 송신하고, 응답 프레임을 수신타임아웃 시간 동안 기다립니다. 수신타임아웃 시간안에 응답이 오면, IF_REQ_RESP ~ END_IF 블록^(*123) 또는 IF_REQ_RESP ~ ELSE 블록^(*124) 사이의 아이템을 실행하며, 등록된 응답 프레임과 실제 응답 프레임이 다르거나 응답이 전혀 없어서 타임아웃이 발생하는 경우에는 ELSE ~ END_IF^(*2) 또는 END_IF 다음^(*1) 아이템을 수행합니다.

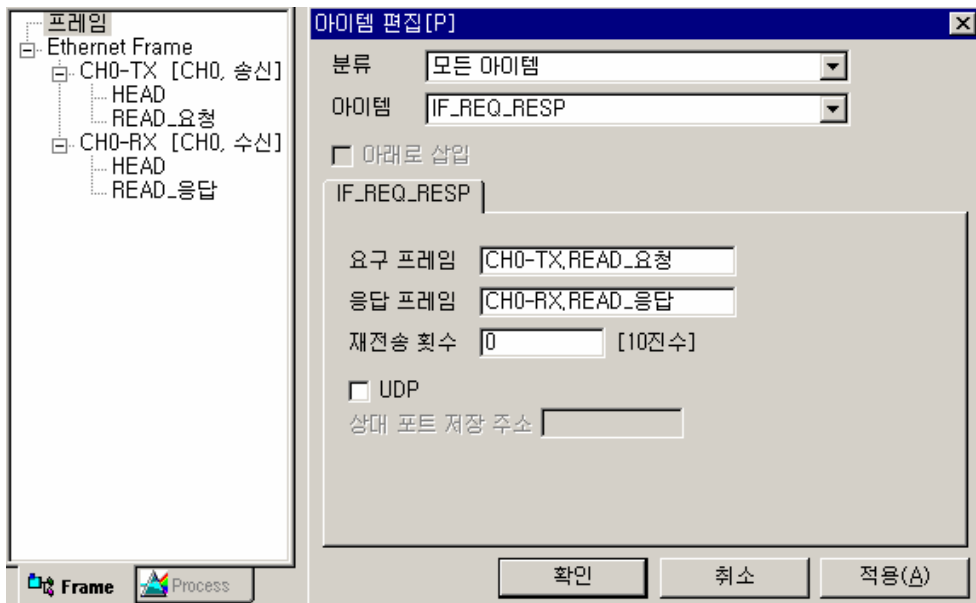


그림 7.6-43 IF_REQ_RESP 입력 예

IF_REQ_RESP 아이템이 범용 시리얼 포트에 사용되면 오픈 트랜잭션(Open Transaction)을 지원하지 않습니다. 동일 포트에 두 개의 프로세스에서 IF_REQ_RESP 아이템을 사용하여 통신을 시도하면, 먼저 송신 프레임을 보낸 프로세스가 응답 프레임을 받거나, 타임아웃이 발생할 때까지, 다른 프로세스의 IF_REQ_RESP 아이템에 의한 요구 프레임이 송신되지 않습니다. 응답을 받기 전에 다른 요구 프레임을 반드시 송신해야 하는 경우라면 IF_REQ_RESP 아이템 대신에 FRAME_SND 와 IF_FRAME_RCV 아이템을 사용할 수 있습니다. 이때에는 요구 프레임 송신과 응답 프레임 수신을 각각 별도의 프로세스에서 해야 합니다.

IF_REQ_RESP 아이템이 이더넷에 사용되면 오픈 트랜잭션을 지원합니다.

(*123) IF_REQ_RESP ~ END_IF 블록 사이에 ELSE 아이템이 없는 경우

(*124) IF_REQ_RESP ~ END_IF 블록 사이에 ELSE 아이템이 삽입된 경우

표 7.6-41 IF_REQ_RESP 설정 항목

항 목	내 용
요구 프레임	송신할 프레임 이름(그룹이름.BODY 프레임 이름)을 입력합니다. 키보드 입력 대신 프레임을 선택한 후 마우스 왼쪽 버튼을 누른채로 프레임 입력란에 가져다 놓아도 됩니다.(DRAG & DROP 지원)
요구 프레임	요구 프레임에 대한 응답 프레임의 이름을 입력합니다. 요구 프레임과 마찬가지로 DRAG & DROP 을 지원합니다.
재전송 횟수	재전송 횟수를 1 회 이상으로 설정하면, 응답 프레임을 받지 못한 경우 요구 프레임을 다시 송신하고 응답 프레임을 기다립니다. 기본값은 0 으로 재전송하지 않습니다.
UDP	이더넷 UDP 통신인 경우 지정할 수 있습니다. ^(*125)
상대 포트 저장 주소	UDP 를 체크한 경우 활성화 됩니다. 상대 포트를 파라미터에서 지정하지 않은 경우, 포트 번호를 메모리에서 참조할 수 있습니다. 해당 주소에는 포트 번호가 미리 입력되어 있어야 정상 동작합니다. (포트 번호는 2 바이트로 저장되어야 합니다)

(*125) 파라미터 설정에서 상대국 UDP 포트를 지정한 경우는 지정하지 않아도 됩니다. 파라미터에서 설정한 포트와 다른 포트로 송신하고자 하는 경우만 설정합니다.

7.6.41.2. 예제

송,수신 프레임을 등록하여 상대국과 통신을 시도합니다. 상대국으로부터 응답을 받으면, 130 번지를 1 증가시키고, 응답을 받지 못하면, 140 번지를 1 증가시킵니다.

▶ 예제 아이템 등록

IF_TIMER 아이템을 추가하고, <주기>를 “100”으로 입력합니다. (1 초 주기)

IF_REQ_RESP 아이템을 삽입하고, 요구 프레임과 응답 프레임 이름을 각각 입력합니다.

IF_REQ_RESP 아이템 직후에 ELSE 아이템을 삽입하고, 그 아래 위로 ARITHMETIC 아이템을 각각 삽입합니다. 각각 130 번지와 140 번지를 1 증가시키도록 입력합니다.

▶ 응답 프레임

응답 프레임(CH0-RX.READ_응답)은 다음과 같이 등록하였습니다.

NO	형태	크기	데이터	메모리
00	고정 크기 변수	2		
01	가변 크기 변수	M100		M120

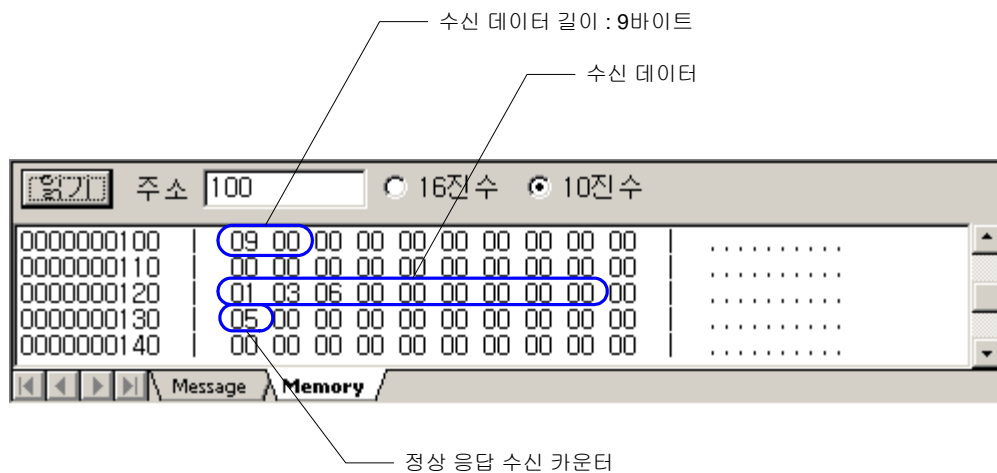
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	100
01	IF_REQ_RESP	Req:CH0-TX,READ_요청, Resp:CH0-RX,READ_응답
02	ARITHMETIC	Type:U1, M130, +, 1, M130
03	ELSE	
04	ARITHMETIC	Type:U1, M140, +, 1, M140
05	END_IF	
06	END_IF	

▶ 결과

100 번지에는 응답 프레임의 데이터 길이가, 120 번지에는 응답 프레임 데이터가 저장되었습니다. 130 번지가 증가한 것으로 보아 상대국이 정상적으로 응답하였음을 알 수 있습니다.



7.6.42. IF_SECS_MC_FRAME_RCV

- ▶ 통신 아이템 (SECS 통신 전용)
- ▶ ETOS-150-HSMS, ETOS-150-Dual-HSMS , ETOS-200-HSMS
- ▶ END_IF

7.6.42.1. 기능

수신 프레임으로 정의한 SECS 프레임을 수신하고, 해당 프로세스를 실행합니다. SECS-I, HSMS 에서 모두 사용할 수 있습니다. 프레임을 정의하여 데이터를 주고 받는 경우, 프레임 정의 만으로는 수신 데이터가 자동으로 저장되지 않습니다. 즉, 아래 그림과 같이 S1F2 프레임을 정의했다고 해서, 해당 프레임을 수신했을 때, 수신 데이터가 자동으로 ETOS 메모리에 저장되지 않습니다. 프레임 정의는 수신을 위한 기초 단계일 뿐 입니다. 수신 프레임의 데이터를 ETOS 메모리에 실제로 저장하기 위해서는 IF_SECS_MC_FRAME_RCV 아이템을 사용해야 합니다.

이 아이템은 시작아이템으로 프로세스 선두에 올 수 있습니다. 이 아이템은 반드시 END_IF 와 짝을 이루어야 하며, 중간에 ELSE 아이템을 사용할 수 있습니다. 또한 이 아이템은 프로세스의 중간에서도 사용할 수 있습니다.

다음 그림은 정의된 S1F1_RCV 프레임을 COM2 로 수신하는 예입니다. COM2 로 S1F1_RCV 프레임을 수신하면 수신 데이터를 저장하고, 해당 프로세스(IF_SECS_MC_FRAME_RCV 로 시작하는)를 수행합니다.

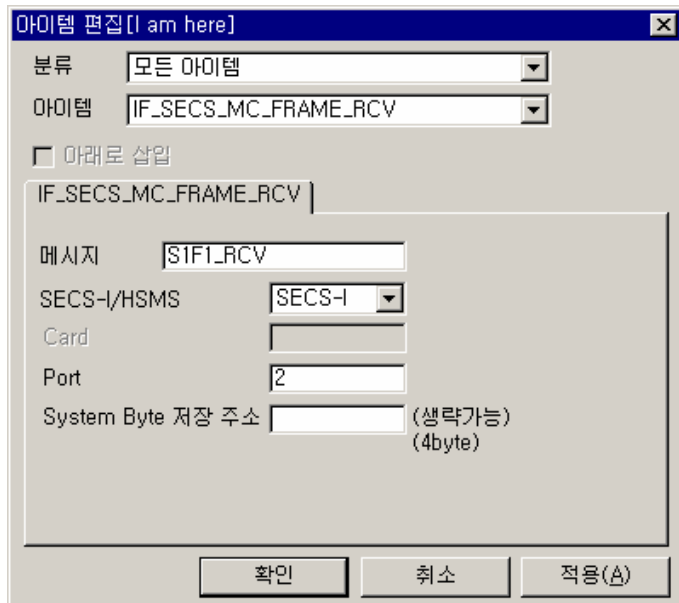


그림 7.6-44 IF_SECS_MC_FRAME_RCV 입력 예

- * IF_SECS_MC_FRAME_RCV 아이템은 상대국이 프레임을 보내면 수동적(Passive)으로 수신하는 역할을 합니다. IF_SECS_MC_FRAME_RCV 와 같이 수동적으로 동작하는 통신 아이템을 서버(Server)아이템이라고 합니다. 반대되는 개념은 클라이언트 또는 마스터(Client or Master)아이템으로 IF_SECS_MC_REQ_RESP, SECS_MC_FRAME_SND 등이 대표적인 예입니다.

표 7.6-42 IF_SECS_MC_FRAME_RCV 설정 항목

항 목	내 용
메시지	수신할 SECS 프레임 이름을 입력합니다. 키보드 입력 대신 프레임을 선택한 후 마우스 왼쪽 버튼을 누른채로 프레임 입력란에 가져다 놓아도 됩니다.(DRAG & DROP)
SECS-I/HSMS	RS-232 를 사용하는 SECS-I 과 이더넷을 사용하는 HSMS 를 선택합니다.
Card	ETOS-200 에서 SECS-I 을 사용하는 경우, CARD 번호를 설정합니다.
Port	Port 번호(SECS-I:COM1,COM2..., HSMS:0,1...)를 설정합니다.
System Byte 저장 주소	수신한 프레임의 System Byte 가 필요할 경우, 따로 저장할 수 있습니다. 생략 가능합니다. 생략시엔 같은 프로세스안에서 IF_SECS_MC_FRAME_RCV 후 SECS_MC_FRAME_SND 를 사용하면 자동으로 동일한 System Byte 를 사용해서 전송합니다.

7.6.42.2. 예제

COM2 로 S1F1 를 수신합니다.

▶ 예제 프레임 정의

먼저 예제 프레임을 다음과 같이 정의합니다.



▶ 예제 아이템 등록

IF_SECS_MC_FRAME_RCV 아이템을 등록하고, 메시지에 S1F1_RCV 를 입력합니다.

▶ 예제 프로세스

편집을 마친 프로세스 화면입니다. 예제에는 SECS_MC_FRAME_SND 아이템도 추가 되어 있습니다.

NO	아이템	설명
00	IF_SECS_MC_FRAME_RCV	S1F1_RCV, Port:COM2
01	SECS_MC_FRAME_SND	S1F2_SND, Port:COM2
02	END_IF	

▶ 결과

정의된 S1F1_RCV 프레임을 수신하면, ETOS 는 프레임 정의를 참조하여 해당 메모리 위치에 데이터 저장하고, S1F2 를 송신 합니다.

7.6.43. IF_SECS_MC_REQ_RESP

- ▶ 통신 아이템 (SECS 통신 전용)
- ▶ ETOS-150-HSMS, ETOS-150-Dual-HSMS , ETOS-200-HSMS
- ▶ ELSE, END_IF

7.6.43.1. 기능

SECS 프레임 송신하고, 정의한 SECS 프레임 수신되면, 해당 프로세스를 실행합니다. SECS-I, HSMS 에서 모두 사용할 수 있습니다.

이 아이템은 시작아이템으로 프로세스 선두에 올 수 있습니다. 이 아이템은 반드시 END_IF 와 짝을 이루어야 하며, 중간에 ELSE 아이템을 사용할 수 있습니다. 또한 이 아이템은 프로세스의 중간에서도 사용할 수 있습니다.

다음 그림은 SECS-I(COM1)으로, S1F1 을 송신, S1F2 를 수신하는 예입니다. COM1 으로 S1F1_SND 를 송신하고, S1F2_RCV 를 수신하면 수신 데이터를 저장합니다.

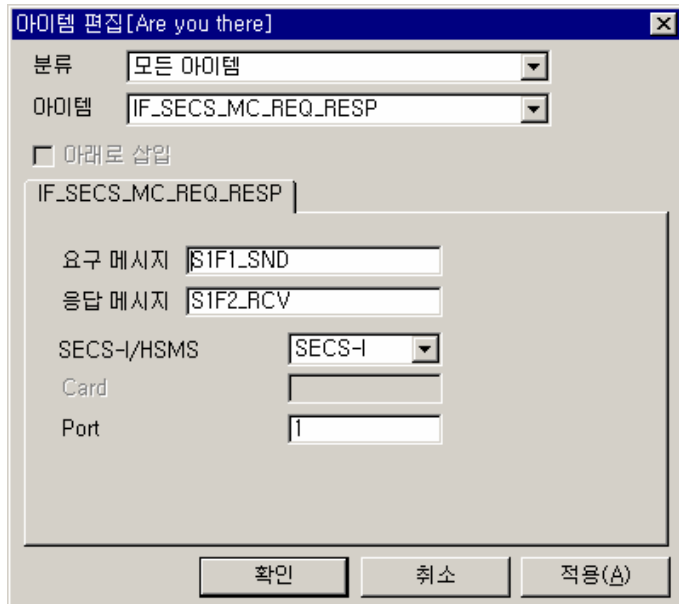


그림 7.6-45 IF_SECS_MC_REQ_RESP 입력 예

표 7.6-43 IF_SECS_MC_REQ_RESP 설정 항목

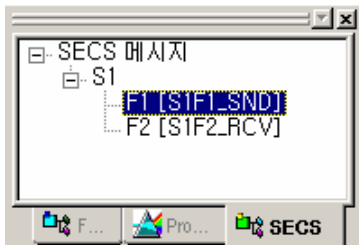
항 목	내 용
요구 메시지	송신할 SECS 프레임 이름을 입력합니다. 키보드 입력 대신 프레임을 선택한 후 마우스 왼쪽 버튼을 누른채로 프레임 입력란에 가져다 놓아도 됩니다.(DRAG & DROP)
응답 메시지	수신할 SECS 프레임 이름을 입력합니다.
SECS-I/HSMS	RS-232 를 사용하는 SECS-I 과 이더넷을 사용하는 HSMS 를 선택합니다.
Card	ETOS-200 에서 SECS-I 을 사용하는 경우, CARD 번호를 설정합니다.
Port	Port 번호(SECS-I:COM1,COM2..., HSMS:0,1...)를 설정합니다.

7.6.43.2. 예제

COM1 으로 10 초마다 S1F1 을 송신하고, S1F2 를 수신, 프레임의 데이터를 메모리에 저장합니다.

▶ **예제 프레임 정의**

먼저 예제 프레임을 다음과 같이 정의합니다.



▶ **예제 아이템 등록**

IF_SECS_MC_REQ_RESP 아이템을 등록하고, <요구 메시지>에 S1F1 을, <응답 메시지>에 S1F2 를 입력합니다.

▶ **예제 프로세스**

편집을 마친 프로세스 화면입니다. 예제에는 프로세스의 시작을 위해서 IF_TIMER 가 추가되어 있습니다.

NO	아이템	설명
00	IF_TIMER	1000
01	IF_SECS_MC_REQ_RESP	Req:S1F1_SND, Resp:S1F2_RCV, Port:COM1
02	END_IF	
03	END_IF	

▶ **결과**

COM1 으로 10 초마다 S1F1_SND 프레임을 송신하고, S1F2_RCV 를 수신하면, ETOS 는 프레임 정의를 참조하여 해당 메모리 위치에 데이터를 저장합니다

7.6.44. IF_STR_CMP

- ▶ 조건분기 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_MEM_CMP](#), [IF_NUM_CMP](#), ELSE, END_IF

7.6.44.1. 기능

입력한 문자열과 데이터 주소에 저장되어 있는 문자열 데이터를 비교합니다. 입력한 문자열의 길이 만큼 비교합니다. 이 아이템은 반드시 END_IF 아이템과 짝을 이루어야 하며, ELSE 아이템을 사용할 수 있습니다.

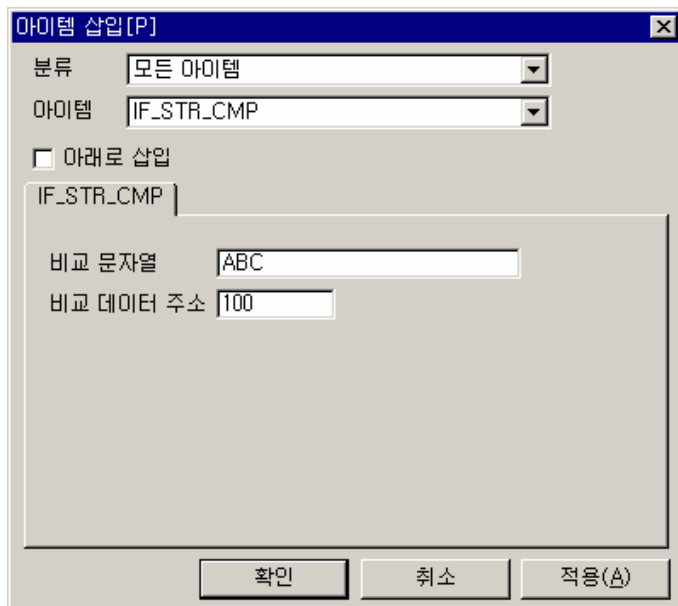


그림 7.6-46 IF_STR_CMP 입력 예

표 7.6-44 IF_STR_CMP 설정 항목

항 목	내 용
비교 문자열	비교할 문자열을 입력합니다. 최대 80 자까지 입력 가능합니다.
비교 데이터 주소	문자열과 비교할 데이터가 저장되어 있는 ETOS 메모리의 주소를 입력합니다.

7.6.44.2. 예제

ETOS 메모리 100 번지에 문자열 ABC 가 저장되어 있는지 확인하고 만약 그렇다면 110 번지가 1 증가가 하도록 합니다.

▶ **예제 아이템 등록**

먼저 100 번지 데이터를 준비합니다. MEM_SET 아이템을 삽입하고, <데이터 형태> '문자열', <데이터> "ABC", <주소> "100", <길이> "3"을 입력합니다.

IF_STR_CMP 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
비교 문자열	비교 기준 데이터인 문자열 ABC 입력
비교 데이터 주소	비교 대상 주소인 100 번지 입력

ARITHMETIC 아이템을 삽입하고, 110 번지에 1 을 더해서 110 번지에 저장하도록 입력합니다.

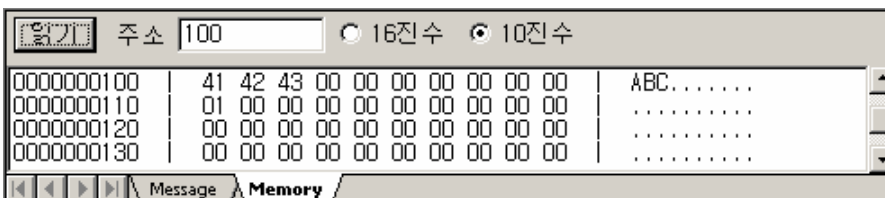
▶ **예제 프로세스**

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:3, Type:문자열, Data:ABC
02	IF_STR_CMP	Str:ABC, S:M100
03	ARITHMETIC	Type:U1, M110, +, 1, M110
04	END_IF	
05	END_IF	

▶ **결과**

IF_STR_CMP 아이템으로 100 번지의 데이터가 문자열 "ABC"와 동일한 지 비교합니다. 100 번지에 문자열 "ABC"(414243H)가 저장되어 있어, 비교 기준 문자열과 동일하므로 ARITHMETIC 아이템이 실행되어 110 번지가 1 증가하는 것을 볼 수 있습니다.



7.6.45. IF_TIMER

- ▶ 시작 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ END_IF

7.6.45.1. 기능

프로세스 선두에만 올 수 있는 시작 아이템으로 지정 주기로 프로세스를 동작시킵니다. 그러나 프로세스 안에서 통신 지연(상대국이 응답하지 않아서 타임아웃이 발생하는 상황)이나 DELAY 아이템에 의한 프로세스 지연 등이 발생하면, 지정한 주기에 지연 시간을 더한 시간이 새로운 주기가 됩니다. 따라서 다음 그림과 같이 주기에 100 을 입력한 경우, 주기는 1 초이지만, 프로세스 안에서 시간 지연이 발생하면 그 주기는 (1 + N(시간 지연))초가 됩니다. 주의하시기 바랍니다.

이 아이템은 반드시 END_IF 아이템과 짝을 이루어야 하며, ELSE 아이템은 사용할 수 없습니다. 또한, 등록 가능한 TIMER 프로세스의 개수가 ETOS 제품 별로 다음과 같이 제한되어 있습니다. (*126)

항 목	ETOS-50A	ETOS-100A/150	ETOS-200	ETOS-1000
타이머 프로세스	16	32	64	128

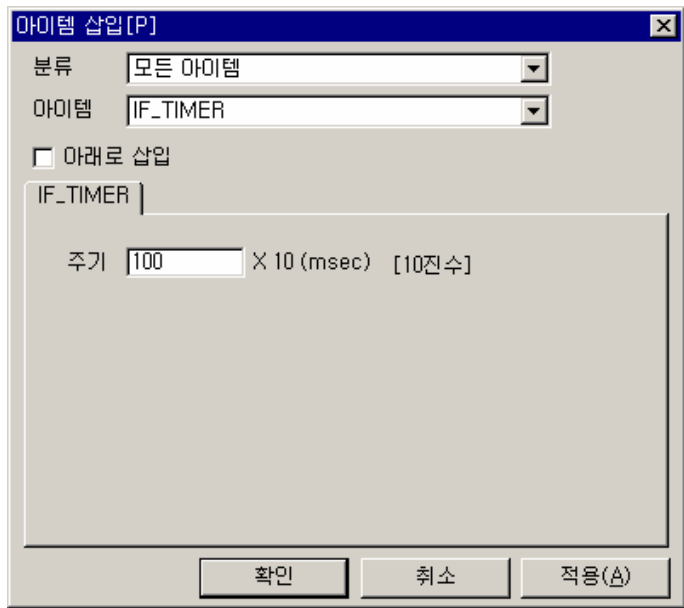


그림 7.6-47 IF_TIMER 입력 예

표 7.6-45 IF_TIMER 설정 항목

항 목	내 용
주기	최소 10(msec)단위로 동작을 합니다. 100 을 입력하면 주기는 100 x 10(msec)이므로 타이머의 주기는 1 초가 됩니다.

(*126) Enable/Disable 에 관계없이 등록된 모든 IF_TIMER 프로세스의 수입니다.

7.6.45.2. 예제

ETOS 사용자 메모리 100 번지를 1 초마다 1 씩 증가시킵니다.

▶ 예제 아이템 등록

IF_TIMER 아이템을 추가하고, <주기> 입력란에 “100” (X10 msec)을 입력합니다.

ARITHMETIC 을 삽입하고, 100 번지에 1 을 더하여 100 번지에 저장합니다.

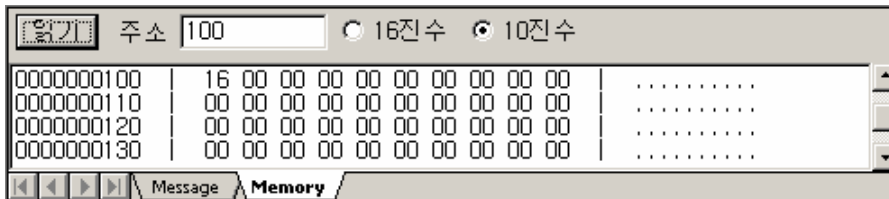
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	100
01	ARITHMETIC	Type:U1, M100, +, 1, M100
02	END_IF	

▶ 결과

1 초 주기로 ARITHMETIC 아이템이 동작하여 100 번지가 1 씩 증가하는 것을 볼 수 있습니다.



7.6.46. INT_TO_STR

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [STR_TO_INT](#), [HEX_TO_ASC](#)

7.6.46.1. 기능

정수나 실수 형태의 데이터를 문자열로 변환합니다. 정수 데이터를 진수형태에 따라 10 진수 문자열(0,1,2,3, ... 9로 표현) 또는 16 진수 문자열(0,1,2,3 ... 9, A, ... F로 표현)로 변환합니다. 변환 데이터를 실수(Float, Double)로 선택한 경우에는 문자열 형식을 지정할 필요가 없습니다.

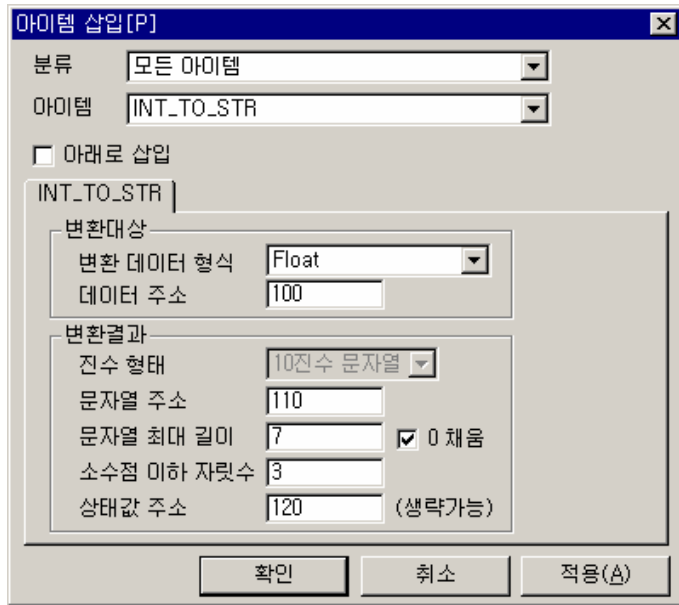


그림 7.6-48 INT_TO_STR 입력 예

표 7.6-46 INT_TO_STR 설정 항목

항 목	내 용
변환 대상	변환 데이터 형식 정수: Signed 1 Byte/ Unsigned 1 Byte/ Signed 2 Byte/ Unsigned 2 Byte/ Signed 4 Byte/ Unsigned 4 Byte 실수: Float/ Double
	데이터 주소 변환 대상 데이터의 위치를 입력합니다.
변환 결과	진수 형태 변환 대상이 정수인 경우, 결과 문자열을 10 진수 또는 16 진수 문자열로 표현합니다. (Float 나 Double 에서는 비활성화 됩니다.)
	문자열 주소 변환 결과 문자열을 저장할 주소를 입력합니다.
	문자열 최대길이 변환 결과 문자열의 최대 길이를 지정합니다. 문자열의 길이는 일정하지 않을 수 있으므로 최대 길이에 제한을 두는 것입니다.
	소수점 이하 자릿수 변환데이터 형식을 Float 나 Double 로 선택하는 경우 이 창이 활성화 됩니다. 소수점 이하의 데이터를 문자열로 변경하는 경우에 그 개수에 제한을 두어 무한대로 길어지는 것을 방지합니다.
	상태값 주소 정상적으로 변환되면 변환된 문자열의 길이가 저장됩니다

7.6.46.2. 예제

부동소수점 상수(Float) 1.23 을 문자열 “1.23”으로 변환합니다.

100 번지 : 1.23 (float, 부동소수점 상수) -> 110 번지 : “1.23” (ASCII, 문자열)

▶ **예제 아이템 등록**

먼저 100 번지에 부동소수점 상수 1.23.을 준비해야 합니다.

ARITHMETIC 아이템을 삽입하고 다음과 같이 입력합니다.

항 목	입력 내용
데이터 형식	Float
데이터 1 의 형태	상수
데이터 2 의 형태	상수
데이터 1	123
연산자	/(나누기)
데이터 2	100
결과(주소)	100

INT_TO_STR 아이템을 삽입하고 다음과 같이 입력합니다.

항 목		입력 내용
변환 대상	변환 데이터 형식	Float
	데이터 주소	100
변환 결과	문자열 주소	110
	문자열 최대길이	7
	0 채움	체크
	소수점이하 자릿수	3
	상태값 주소	120

▶ **예제 프로세스**

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	ARITHMETIC	Type:Float, 123, /, 100, M100
02	INT_TO_STR	Type:Float, Radix:10, S:M100, T:M110, MaxSize:7, FillZero, Precision:3, Status:M120
03	END_IF	

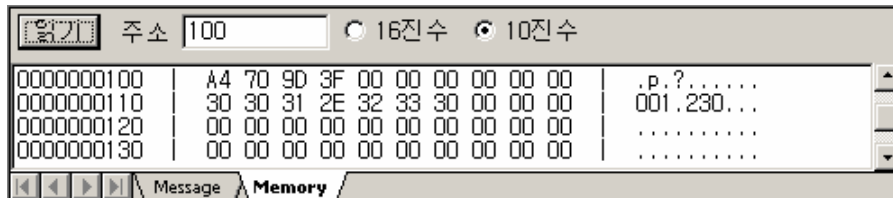
▶ 결과

100 번지에 부동소수점 상수 1.23 이 IEEE 포맷으로 저장되어 있습니다.(A4709D3F_H)

110 번지에는 INT_TO_STR 아이템에 의해 10 진수 문자열로 변환된 데이터가 저장되었습니다.

("001.230", 3030312E323330_H)

1.23 은 문자열로 4 바이트이나, 문자열 최대 길이를 7 바이트로 지정하였고, 또, 0 채움을 체크했기 때문에, 나머지 3 바이트가 0 으로 채워졌습니다.



7.6.47. IO_READ_WRITE

- ▶ 응용 아이템 (DIO 전용)
- ▶ ETOS-200/1000 DIO8
- ▶ [IF_IO_STATE_CHANGE](#)

7.6.47.1. 기능

IO_READ_WRITE 아이템은 ETOS DIO8 모듈 전용 아이템입니다. ETOS DIO8 모듈은 8점의 TR 출력과 24Volt DC 입력 8점, 시리얼 2 포트를 가지고 있는 VME Bus 슬레이브 모듈로 ETOS-200/1000 에 장착할 수 있습니다. DIO 모듈의 입력, 출력 8 점을 ETOS 메모리로 읽어오거나, ETOS 메모리 값을 출력 8 점에 내보내는 아이템입니다.

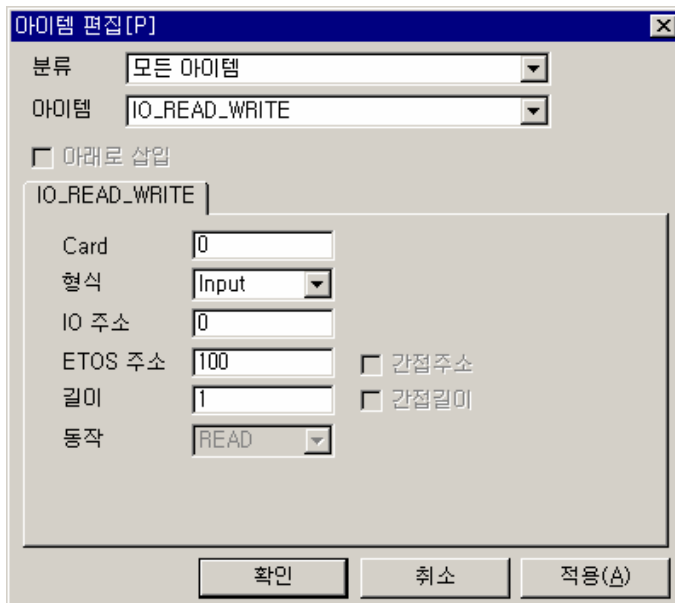


그림 7.6-49 IO_READ_WRITE 입력 예

표 7.6-47 IO_READ_WRITE 설정 항목

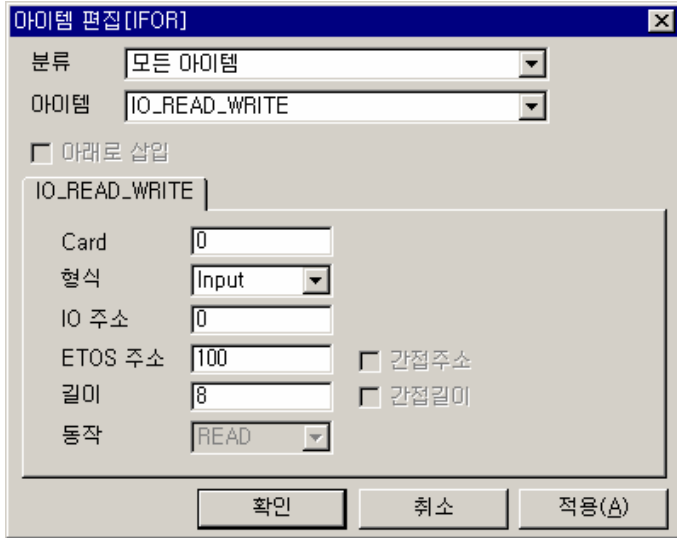
항 목	내 용
CARD	DIO 모듈의 카드 번호를 입력합니다. (0 ~ 7)
형식	입력점(Input)이나 출력점(Output)을 선택합니다.
IO 주소	입력, 또는 출력점의 위치를 지정합니다. (0 ~ 7)
ETOS 주소	형식이 Input 인 경우 : 해당 점점을 읽어서 저장할 사용자 메모리 주소 형식이 Output 인 경우: 해당 점점의 출력값을 저장하고 있는 사용자 메모리 주소
길이	읽어오거나 쓰려는 데이터의 길이를 입력합니다.
동작	형식이 OUTPUT 으로 선택되면 활성화 됩니다. 기능은 출력점점의 데이터를 읽거나 쓰려는 경우 해당 기능을 선택할 수 있습니다.
간접주소	동작이 'WRTIE'로 선택되면 활성화 됩니다. 기능은 ETOS 의 사용자 메모리 주소를 간접주소로 사용할 수 있습니다.
간접길이	동작이 'WRTIE'로 선택되면 활성화 됩니다. 기능은 길이에 입력되어있는 값이 ETOS 의 메모리 주소가 되며 그 주소를 간접주소로 사용할 수 있습니다.

7.6.47.2. 예제

Card 0 의 입력 8 점을 ETOS 의 사용자 메모리 100 번지로 읽어 옵니다.

▶ **예제 아이템 등록**

<Card>는 “0”으로, <형식>은 ‘Input’으로, <IO 주소>는 “0”번 비트부터, <ETOS 주소>는 “100”번지로, <길이>는 “8”비트로 설정합니다.



▶ **예제 프로세스**

예제에선 프로세스를 시작하기 위해 IF_IO_STATE_CHANGE 를 추가했습니다.

NO	아이템	설명
00	IF_IO_STATE_CHANGE	Card:0, EventBit:1, EventType:Falling
01	IO_READ_WRITE	Card:0, Type:Input, IO:0, ETOS:M100, Size:8, Fu...
02	END_IF	

▶ **결과**

프로세스가 시작되면 Card 0 의 입력 8 점을 ETOS 의 사용자 메모리 100 번지로 읽어 옵니다.

7.6.48. IP_SAVE

- ▶ 응용 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.48.1. 기능

ETOS의 IP 주소를 지정 주소에 16진수 값으로 저장합니다. ETOS의 IP 주소는 [온라인] -> [TCP/IP 설정] 에서 확인할 수 있습니다.

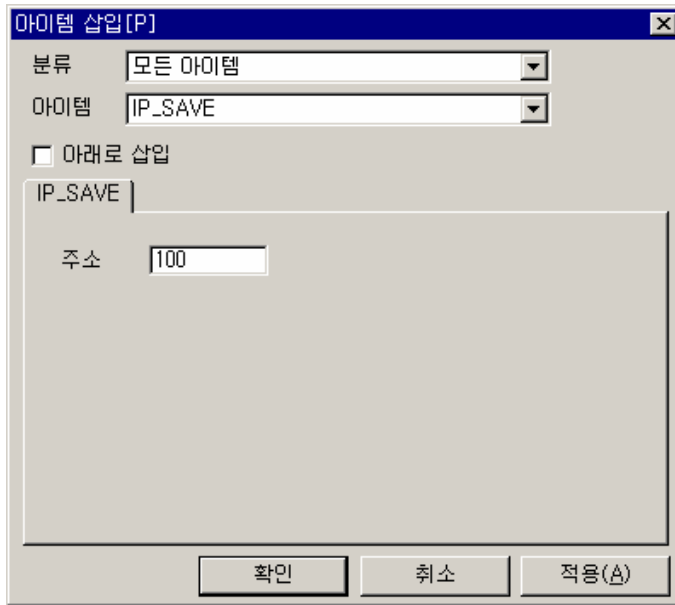


그림 7.6-50 IP_SAVE 입력 예

표 7.6-48 IP_SAVE 설정 항목

항 목	내 용
주소	IP 주소를 저장할 메모리 위치를 지정합니다.

7.6.48.2. 예제

ETOS 의 IP 주소를 100 번지에 저장합니다.
 ETOS 의 IP 주소가 192.168.1.2 라고 가정합니다.

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 추가합니다.
 IF_POWER_ON 아이템 맞으로 IP_SAVE 아이템을 삽입하고, <주소>에 “100”을 입력합니다.

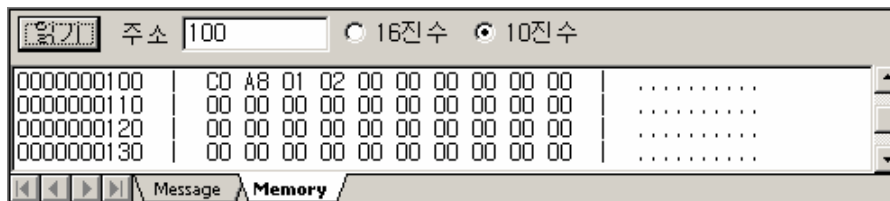
▶ **예제 프로세스**

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	IP_SAVE	M100
02	END_IF	

▶ **결과**

100 번지에 IP 주소가 저장되었습니다.
 192.168.1.2 (10 진수) -> C0.A8.01.02 (16 진수)



7.6.49. JUMP

- ▶ 조건분기 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [LABEL](#)

7.6.49.1. 기능

프로세스 수행 중 이 아이템을 만나면, 지정한 레이블 아이템 위치로 프로세스 수행 위치를 변경합니다. 조건에 의한 분기가 아니라 무조건 분기 명령입니다. 동일 프로세스 내에 반드시 지정한 LABEL 아이템이 존재해야 합니다.

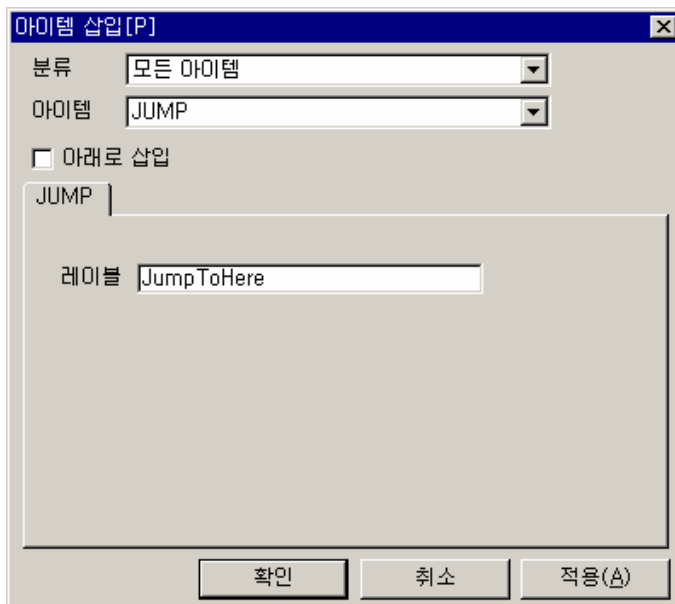


그림 7.6-51 JUMP 입력 예

표 7.6-49 JUMP 설정 항목

항 목	내 용
레이블	분기할 위치의 레이블 아이템을 입력합니다. 레이블 아이템은 프로세스에 미리 등록되어 있어야 합니다.

7.6.49.2. 예제

ETOS 메모리 100 번지가 1 씩 증가하여 10 이 될 때 110 번지는 1 이 증가가 되도록 합니다.
(LOOP 와 END_LOOP 아이템으로도 같은 결과를 만들 수 있습니다.)

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 시작아이템으로 등록합니다.
 LABEL 아이템을 삽입하고, <레이블>에 “JumpToHere” 라고 입력합니다.
 ARITHMETIC 아이템을 삽입하고, 100 번지에 1 을 더해 100 번지에 저장하도록 합니다.
 IF_NUM_CMP 아이템을 삽입하고 100 번지가 10 이 되는지를 비교하도록 등록합니다.
 ARITHMETIC 아이템을 삽입하고, 110 번지에 1 을 더해 110 번지에 저장하도록 합니다.
 END 아이템을 삽입합니다.

JUMP 아이템을 삽입하고, 위에서 등록한 LABEL 아이템의 이름 “JumpToHere”를 입력합니다.
 앞서 등록한 JumpToHere 레이블로 분기할 것입니다.

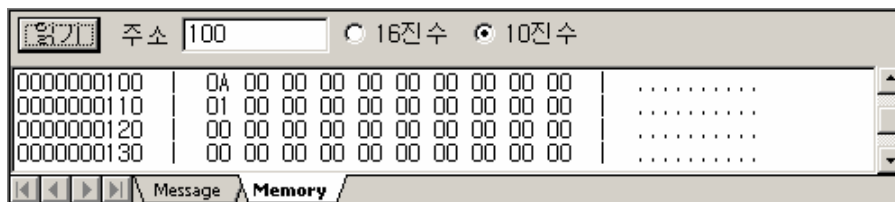
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	LABEL	JumpToHere
02	ARITHMETIC	Type:U1, M100, +, 1, M100
03	IF_NUM_CMP	Type:U1, 10, =, M100
04	ARITHMETIC	Type:U1, M110, +, 1, M110
05	END	
06	END_IF	
07	JUMP	JumpToHere
08	END_IF	

▶ 결과

100 번지 값이 10(0AH)가 될 때까지 LABEL ~ JUMP 사이를 반복실행합니다.
 100 번지 값이 10 과 같아지면(IF_NUM_CMP), 110 번지를 1 증가 시키고, END 아이템에 의해 프로세스가 종료됩니다. IF_POWER_ON 프로세스이므로 전원이 켜질 때 1 회만 동작합니다.



7.6.50. LABEL

- ▶ 조건분기 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [JUMP](#)

7.6.50.1. 기능

JUMP 아이템과 같이 사용되어, JUMP 아이템으로부터 분기해 올 위치를 지정하는데 사용됩니다.

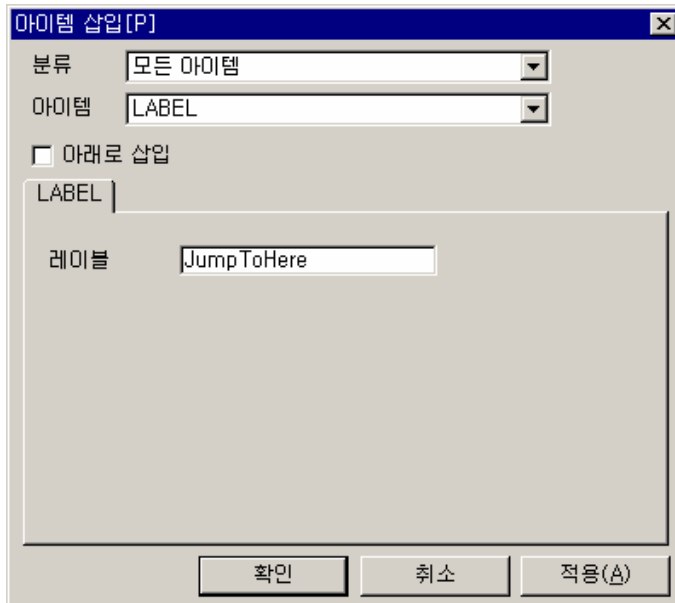


그림 7.6-52 LABEL 입력 예

표 7.6-50 LABEL 설정 항목

항 목	내 용
레이블	사용자가 원하는 레이블 이름을 입력할 수 있습니다. (최대 영문 10 자, 한글 5 자) 레이블 이름은 JUMP 아이템에서 등록된 이름과 짝을 이루어야 합니다.(동일해야 합니다.)

7.6.50.2. 예제

사용예는 [JUMP 아이템](#)을 참조하시기 바랍니다.

7.6.51. LOOKUP_TABLE

- ▶ 응용 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [MEM_CONVERT](#)

7.6.51.1. 기능

특정 위치(입력 데이터 주소)의 내용을 Lookup Table 상에서 찾아 대응되는 값으로 변환하여 지정 위치(출력 데이터 주소)에 저장합니다.

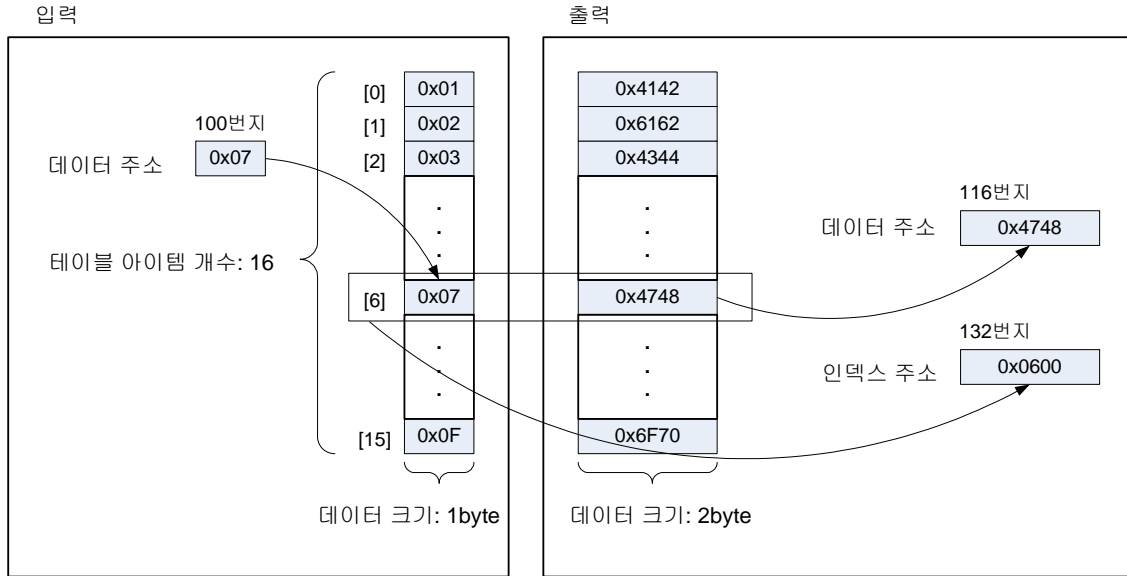


그림 7.6-53 LOOKUP_TABLE 동작 개념

Lookup 테이블은 입력 테이블과 출력 테이블 2 개를 미리 메모리 특정 위치에 마련해 두어야 합니다. 입력 데이터 값을 입력 테이블에서 찾아 동일 위치(인덱스)의 출력 테이블 값으로 변환합니다.

그림에서와 같이 입력 데이터 주소 100 번지에 0x07 이란 값이 저장되어 있을 경우, 입력 테이블에서 0x07 을 찾습니다. 검색 결과, 7 번째에서 값을 찾았다면, 출력값은 출력 테이블의 7 번째 값, 즉, 0x4748 이 됩니다. 이때, 인덱스 주소로 지정한 132 번지에는 인덱스 값 0x0600 (6)이 저장됩니다.

예에서와 같이, LOOKUP_TABLE 아이템은 변환 값간의 관계를 다른 아이템으로 표현하기 어렵거나 불가능한 경우 유용하게 사용할 수 있습니다. 앞에서 설명한 바와 같이, 입, 출력 테이블은 MEM_SET 아이템 등으로 미리 만들어 두어야 합니다.

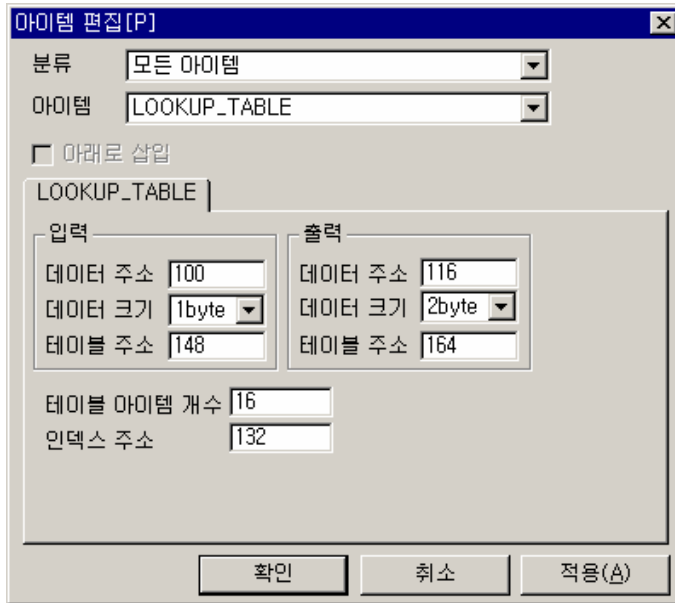


그림 7.6-54 LOOKUP_TABLE 입력 예

표 7.6-51 LOOKUP_TABLE 설정 항목

항 목	내 용	
입력	데이터 주소	변환 대상 데이터의 위치를 입력합니다.
	데이터 크기	변환 대상 데이터의 바이트 수를 입력합니다.(1/2/4 바이트) 입력 테이블의 데이터와 동일 크기여야 합니다.
	테이블 주소	입력 테이블의 시작 주소를 입력합니다.
출력	데이터 주소	변환 결과 데이터의 저장 위치를 입력합니다.
	데이터 크기	출력 테이블의 각 데이터의 크기를 입력합니다. (1/2/4 바이트) 입력 테이블의 데이터 크기와 달라도 상관없습니다.
	테이블 주소	출력 테이블의 시작 주소를 입력합니다.
테이블 아이템 개수	테이블의 아이템(원소) 개수를 입력합니다. 입,출력 테이블의 아이템 개수는 동일해야 합니다.	
인덱스 주소	변환 대상 데이터가 입력 테이블 어느 위치에 있는지를 저장하는 주소입니다.	

7.6.51.2. 예제

100 번째 입력 값 07 을 변환 테이블상에서 찾아 출력값으로 변환하여 116 번째에 저장합니다.
 (입,출력 테이블은 임의로 정의하겠습니다. 실제에서는 사용자가 그 값을 지정해야 합니다.)

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.

100 번째 값을 07 로 가정하기로 하였으므로, MEM_SET 아이템을 삽입하고, <데이터 형태> ‘16진수’, <데이터> “07”, <주소> “100”, <길이> “1”을 각각 입력합니다.

입, 출력 테이블을 MEM_SET 아이템을 이용하여 정의합니다.

입력 테이블의 데이터는 16진수 “0102030405060708090A0B0C0D0E0F”로 정의합니다.

출력 테이블의 데이터는 문자열 “ABabCDcdEFefGHghIjijKLklMNmnOPop”로 정의합니다.

LOOKUP_TABLE 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목		입력 내용
입력	데이터 주소	100
	데이터 크기	1byte
	테이블 주소	148
출력	데이터 주소	116
	데이터 크기	2byte
	테이블 주소	164
테이블 아이템 개수		16
인덱스 주소		132

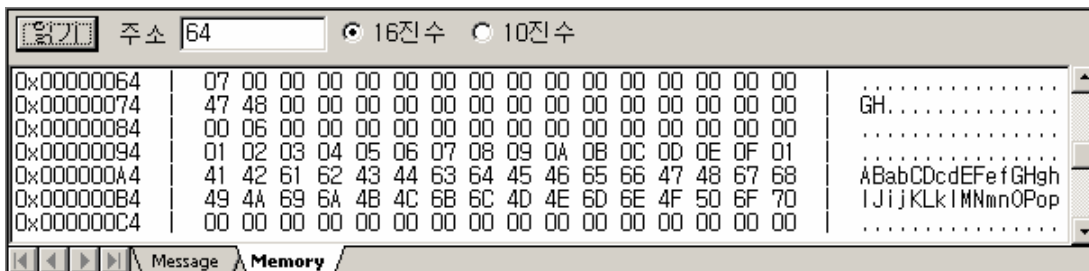
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:1, Type:16진수, Data:07
02	REMARK	<<< 입력 테이블 >>>
03	MEM_SET	T:M148, Size:16, Type:16진수, Data:0102030405060708090A0B0C...
04	REMARK	<<< 출력 테이블 >>>
05	MEM_SET	T:M164, Size:32, Type:문자열, Data:ABabCDcdEFefGHghIjijKLkl...
06	LOOKUP_TABLE	InData:M100, InDataSize:1, InTable:M148, OutData:M116, OutDat...
07	END_IF	

▶ 결과

입력 데이터 07 은 4748_H로 변환되어 116 번째에 저장되었습니다. 결과에 대한 자세한 내용은 [그림 7.6-52 LOOKUP_TABLE 동작 개념](#) 을 참조하시기 바랍니다.



7.6.52. LOOP

- ▶ 조건분기 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [END_LOOP](#)

7.6.52.1. 기능

END_LOOP 아이템과 함께 사용되어 LOOP ~ END_LOOP 블록을 반복하여 수행합니다. 반복 횟수는 카운터 주소에 저장된 값을 참조하며, LOOP 가 1 번 반복될 때마다 카운터 값이 1 씩 감소하여, 카운터 값이 0 이 되면 LOOP 를 빠져 나옵니다.

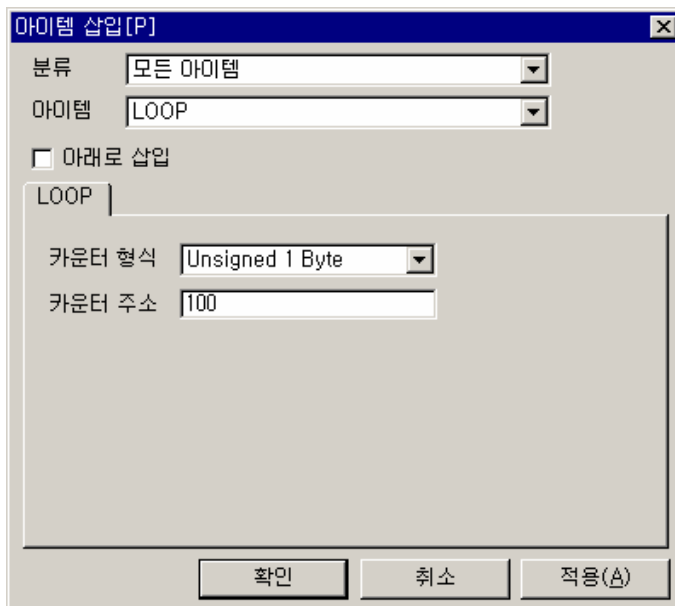


그림 7.6-55 LOOP 입력 예

표 7.6-52 LOOP 설정 항목

항 목	내 용
카운터 형식	카운터의 데이터 형을 지정합니다. 데이터 형에 따라서 카운트 횟수의 범위가 달라집니다. Signed 1 Byte/Unsigned 1 Byte/Signed 2 Byte/Unsigned 2 Byte/ Signed 4 Byte/Unsigned 4 Byte
카운터 주소	카운터 값이 저장되어 있는 메모리의 주소를 지정합니다.

7.6.52.2. 예제

ETOS 메모리 100 번지가 1 씩 증가하여 10 이 될 때 110 번지는 1 이 증가가 되도록 합니다.
(JUMP, LABEL 의 예와 동일한 내용을 LOOP, END_LOOP 아이템으로 구현합니다.)

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 시작아이템으로 등록합니다.

MEM_SET 아이템을 삽입하고, <데이터 형태> '16 진수', <데이터> "0A", <주소> "120", <길이> "1"을 각각 입력합니다.

LOOP 아이템을 추가하고, 다음과 같이 입력합니다.

항 목	입력 내용
카운터 형식	Signed 1 Byte
카운터 주소	120

LOOP 와 END_LOOP 사이에 ARITHMETIC 아이템을 삽입하고, 100 번지에 1 을 더해서 100 번지 에 저장하도록 입력합니다.

END_LOOP 다음에 ARITHMETIC 아이템을 삽입하고, 110 번지에 1 을 더해서 110 번지에 저장하 도록 입력합니다.

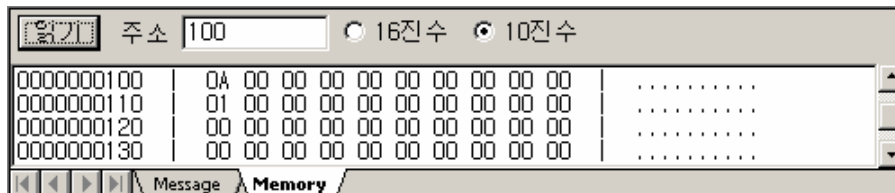
▶ **예제 프로세스**

프로세스에서 지금까지 작성한 아이템 들은 다음과 같은 형태로 등록됩니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M120, Size:1, Type:16진수, Data:0A
02	LOOP	M120, Type:S1
03	ARITHMETIC	Type:S1, M100, +, 1, M100
04	END_LOOP	
05	ARITHMETIC	Type:S1, M110, +, 1, M110
06	END_IF	

▶ **결과**

120 번지의 카운터 값이 10 이므로, 10 회 반복하면서 100 번지를 1 씩 증가시킵니다. 반복이 끝 나면 120 번지 카운터 값은 0 이 되고, 100 번지는 10(0AH)가 되었습니다. 110 번지는 1 증가하였 습니다.



7.6.53. MEM_CONVERT

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [LOOKUP_TABLE](#)

7.6.53.1. 기능

지정한 범위의 메모리에서 특정 값(찾을 데이터)을 찾아 다른 값(바꿀 데이터)으로 치환합니다.

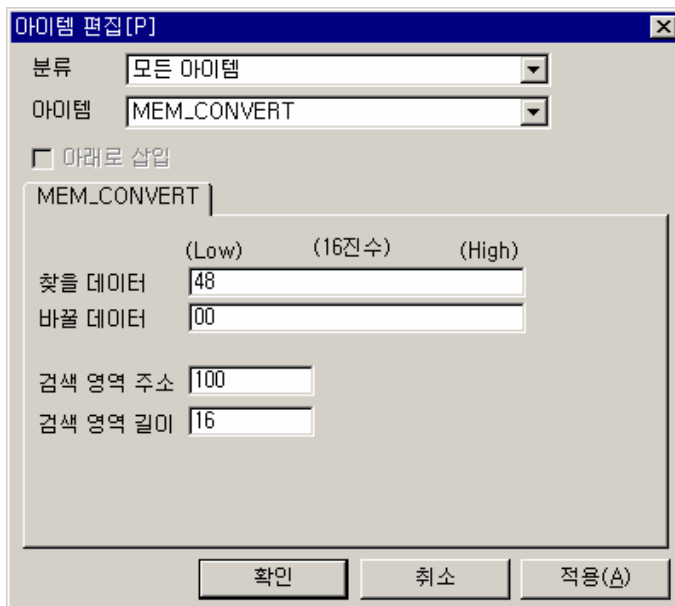


그림 7.6-56 MEM_CONVERT 입력 예

표 7.6-53 MEM_CONVERT 설정 항목

항 목	내 용
찾을 데이터	지정한 영역에서 치환 대상 값을 입력합니다. 해당 값은 다른 값(바꿀 데이터)으로 변경됩니다. 16 진수 형태로만 지정 가능합니다.
바꿀 데이터	치환 대상을 대신할 값을 입력합니다. 치환 대상(찾을 데이터)은 입력 값으로 바뀝니다. 16 진수 형태로만 지정 가능합니다.
검색 영역 주소	검색 영역의 선두 번지를 지정합니다.
검색 영역 길이	검색 영역의 길이를 지정합니다.

7.6.53.2. 예제

100 번지부터 16 바이트를 검색해서 문자 'H'(48H)를 0 으로 변경합니다.
 (100 번지에는 문자열 "ABCDEFGHJKLMNOP" 가 저장되어 있다고 가정합니다.)

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.
 MEM_SET 아이템을 삽입하고, 100 번지에 문자열 "ABCDEFGHJKLMNOP"을 입력합니다.

MEM_CONVERT 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
찾을 데이터	48 (문자 'H' 의 아스키 값)
바꿀 데이터	00
검색 영역 주소	100
검색 영역 길이	16

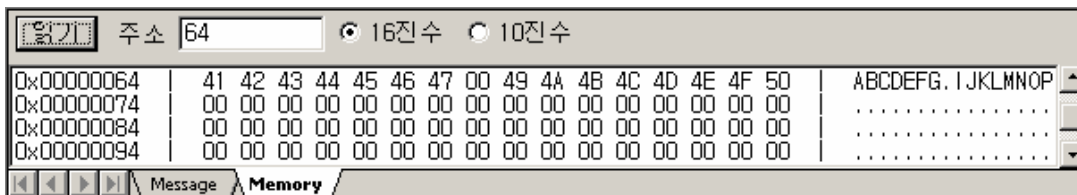
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:16, Type:문자열, Data:ABCDEFGHJKLMNOP
02	MEM_CONVERT	M100, Size:16, S:48, T:00
03	END_IF	

▶ 결과

100 번지부터 16 바이트를 검색해서 치환 대상(찾을 데이터) 48H (문자 'H')을 0 으로 변경했습니다.



7.6.54. MEM_CPY

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [BITS_CPY](#)

7.6.54.1. 기능

바이트 단위로 데이터를 복사합니다.

상수를 메모리로, 또는 메모리를 메모리로 복사할 수 있습니다.

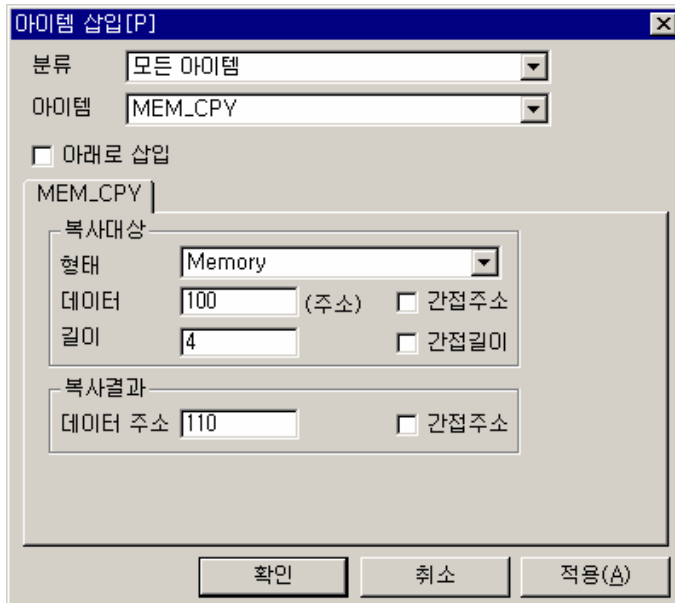


그림 7.6-57 MEM_CPY 입력 예

표 7.6-54 MEM_CPY 설정 항목

항 목	내 용	
복사 대상	형태	복사 대상을 상수나 메모리중에서 선택합니다. 메모리를 선택하는 경우 복사 길이를 자유롭게 설정할 수 있고, 상수인 경우에는 그 크기가 정해집니다.
	데이터	메모리를 선택한 경우에는 복사 대상 메모리의 위치를 입력하고, 상수인 경우에는 상수값(10 진수)을 입력합니다.
	간접주소	체크하면 데이터 주소에 쓰여있는 값(4 바이트로 저장)이 실제 주소가 됩니다. ^(*127)
	길이	데이터의 길이를 입력합니다. ^(*1)
	간접길이	체크하면 길이에 입력한 값은 길이가 아닌 주소를 의미. 그 위치의 값(2 바이트로 저장)이 실제 길이가 됩니다. ^(*1)
복사 결과	데이터 주소	복사 결과 데이터가 위치할 주소를 입력합니다.
	간접주소	체크하면 주소에 쓰여 있는 값(4 바이트로 저장)이 실제 주소가 됩니다.

^(*127) 복사 대상의 형태를 메모리로 선택한 경우만 활성화됩니다.

7.6.54.2. 예제

100 번지에서 110 번지로 4 바이트를 복사합니다.
 (100 번지부터 4 바이트에는 01 02 03 04 가 저장되어 있다고 가정합니다.)

▶ **예제 아이템 등록**

MEM_SET 아이템을 삽입하고, <데이터 형태> '16 진수', <데이터> "01020304", <주소> "100", <길이> "4"를 각각 입력합니다.

MEM_CPY 아이템을 삽입하고 다음과 같이 입력합니다.

항 목		입력 내용
복사대상	형태	메모리
	데이터	100
	길이	4
복사결과	데이터 주소	110

간접 주소 및 간접 길이는 모두 체크하지 않습니다.

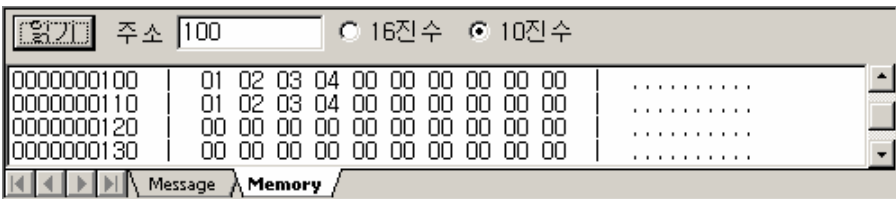
▶ **예제 프로세스**

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:4, Type:16진수, Data:01020304
02	MEM_CPY	S:M100, T:M110, Size:4
03	END_IF	

▶ **결과**

100 번지 데이터 4 바이트를 110 번지에 복사했음을 확인할 수 있습니다.



7.6.55. MEM_SET

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.55.1. 기능

16 진수 또는 문자열로 지정한 위치의 메모리를 채웁니다. 데이터에 입력한 16 진수 또는 문자열의 실제 길이보다 길이에 입력한 값이 큰 경우 데이터를 반복해서 저장합니다.
 예를 들어, 문자열 "ABC" (실제 문자열 길이: 3)을 입력하고, 길이를 5로 입력한 경우 지정 위치에는 문자열 "ABCAB"가 저장됩니다.

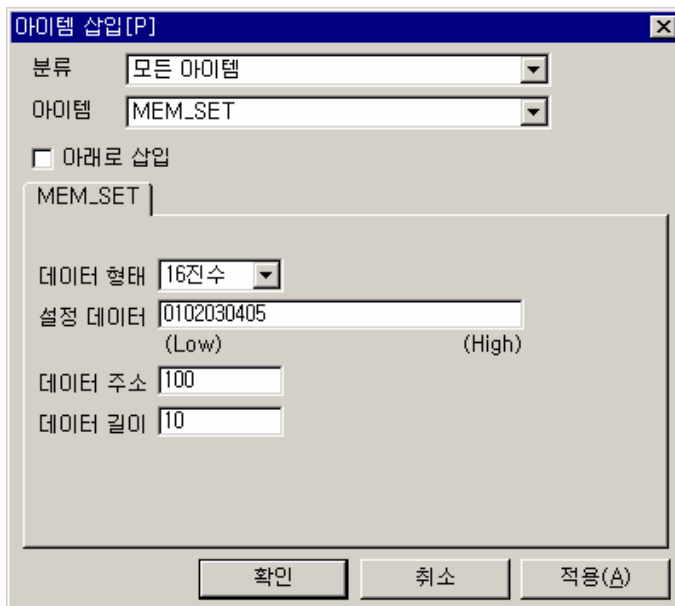


그림 7.6-58 MEM_SET 입력 예

표 7.6-55 MEM_SET 설정 항목

항 목	내 용
데이터 형태	데이터의 형태를 선택합니다. (16 진수/문자열)
설정 데이터	메모리에 저장할 데이터를 입력합니다. 낮은 메모리 위치부터 입력합니다.
데이터 주소	데이터를 저장할 메모리 위치를 지정합니다.
데이터 길이	데이터의 길이를 지정합니다. 단, 이 값은 입력한 데이터의 실제 길이보다 크거나 같아야 합니다.

* 16 진수 데이터를 입력하는 경우 0은 00으로 1은 01과 같이 입력해야 합니다. 또, 16 진수이므로, 01은 길이가 2가 아닌 1임을 주의하시기 바랍니다.

7.6.55.2. 예제

100 번지부터 10 바이트를 차례대로 01 02 03 04 05 01 02 03 04 05 값으로 설정합니다.

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.

MEM_SET 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
데이터 형태	16 진수 선택
설정 데이터	0102030405
데이터 주소	100
데이터 길이	10

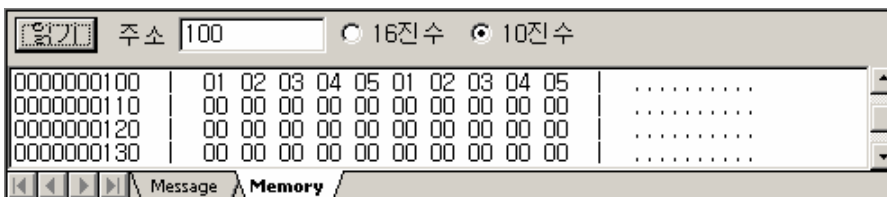
▶ 예제 프로세스

프로세스에서 지금까지 작성한 아이템 들은 다음과 같은 형태로 등록됩니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:10, Type:16진수, Data:0102030405
02	END_IF	

▶ 결과

100 번지부터 10 바이트가 차례대로 01 02 03 04 05 01 02 03 04 05 값으로 설정되었습니다. 01 02 03 04 05 가 두번 반복되는 패턴이므로 모두 입력하지 않고, 길이를 10 으로 설정하여 10 바이트가 저장되도록 하였습니다.



7.6.56. MEM_WRITE_RESP_SND

- ▶ 통신 아이템 (시리얼, 이더넷 공용)
- ▶ ETOS-50/100/150/200/1000
- ▶ [IF_MEM_WRITE_RCV](#)

7.6.56.1. 기능

IF_MEM_WRITE_RCV 와 함께 사용하여, 드라이버 쓰기 수신 프레임에 대한 응답 프레임을 상대국에 전송합니다. 시작 아이템인 IF_MEM_WRITE_RCV 와 반드시 함께 사용해야 하며, 만약, MEM_WRITE_RESP_SND 아이템이 빠지면, ETOS 메모리에 데이터가 써 지기는 하지만 응답을 보내지 않으므로, 상대국은 에러를 발생시킬 수 있습니다

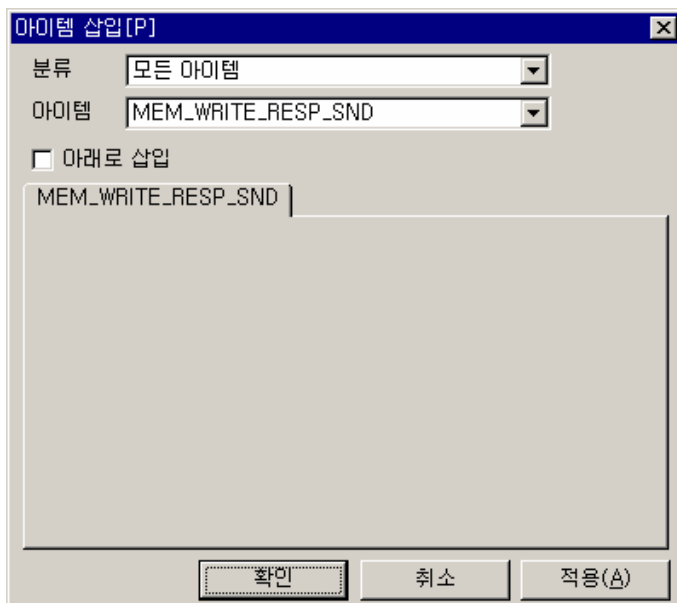


그림 7.6-59 MEM_WRITE_RESP_SND 아이템

표 7.6-56 MEM_WRITE_RESP_SND 설정 항목

항 목	내 용
없음	설정 항목이 없습니다

7.6.56.2. 예제

사용예는 [IF_MEM_WRITE_RCV 예제](#)를 참조하시기 바랍니다.

7.6.57. MODEM_CONTROL

- ▶ 통신 아이템 (시리얼 전용)
- ▶ ETOS-50/100/150/200/1000

7.6.57.1. 기능

시리얼 포트의 DTR, RTS, Break 제어를 수행합니다. 전용모뎀, 널모뎀에 관계없이 사용할 수 있으나, DTR, RTS 결선이 되어 있어야 합니다. 결선에 대한 자세한 사항은 [3 장, 설치 및 배선](#) 편을 참조하시기 바랍니다.

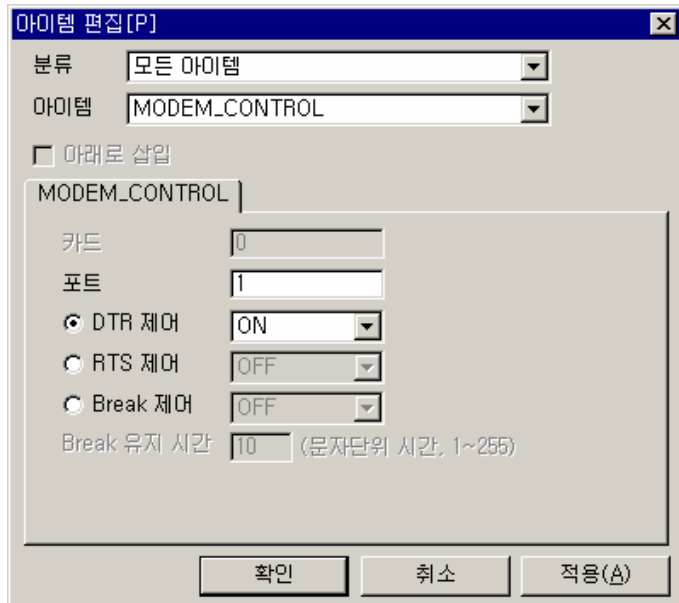


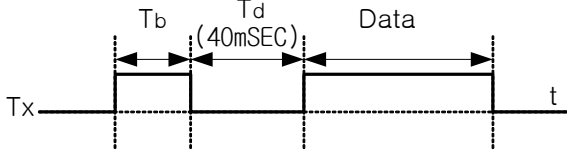
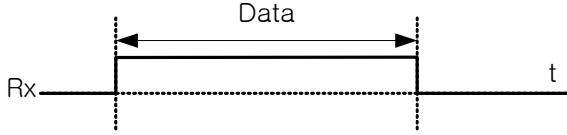
그림 7.6-60 MODEM_CONTROL 입력 예

표 7.6-57 MODEM_CONTROL 설정 항목

항 목	내 용
카드	제어하고자 하는 포트의 카드 번호 ^(*128) (0 ~ 7)
포트	제어하고자 하는 포트의 번호를 지정합니다. (1 ~ 8)
DTR 제어	DTR 신호를 On / Off 할 수 있습니다.
RTS 제어	RTS 신호를 On / Off 할 수 있습니다.
Break 제어	ETOS 에서 데이터를 송신하기 전 설정된 시간만큼 Break 신호를 삽입하여 수신국으로 하여금 데이터 수신준비를 할 수 있도록 합니다. 수신장비에서 Break 신호를 요구하는 경우에만 사용해야 합니다. (예 GE 90/70 PLC) 기본값은 Off 로 되어 있습니다. ON: Break 신호를 송신 전에 보낸 후 40mSEC 후에 데이터 송신 OFF: Break 신호를 보내지 않음(기본값)

^(*128) ETOS-200/1000 인 경우만 활성화됩니다.

표 7.6-57 MODEM_CONTROL 설정 항목 (계속)

항 목	내 용
Break 유지 시간	<p>얼마동안 Break 신호를 출력할 지 시간을 지정합니다. 시간의 단위는 문자 단위 출력 시간입니다. 즉, 5 라고 입력하면 5 문자를 출력하는데 필요한 시간을 의미합니다. 실제 시간은 통신 속도에 따라 달라집니다.</p> <p>다음 그림은 Break 제어의 타이밍도입니다. 송신데이터를 보내기 전 Break 제어가 ON 되어 있으면 Tb 시간동안 Break 신호를 먼저 보내고 Td 시간 경과후 데이터를 보냅니다. Break 유지 실제 시간 Td는 다음과 같이 설정됩니다. $T_b(\text{Break_Control_Time}) = T_c(\text{문자 단위 시간}) \times n(\text{설정값})$ $T_d(\text{Delay Time}) = 40 \text{ mSEC}(\text{고정})$ 위에서 Tc(문자 단위 시간)는 1 문자를 송신하는데 걸리는 시간으로 통신속도에 따라 변하므로 상대기기에서 필요로 하는 시간에 맞게 계산값을 설정해 주어야 합니다. 예를들어 9600bps 의 속도에서 $T_c = 1/9600 \times 10 = 1\text{mSEC}$ 입니다.</p> <p>a) Break 제어를 On 한 경우의 송신 타이밍</p>  <p>b) Break 제어를 Off 한 경우의 송신 타이밍(기본값)</p> 

7.6.58. QUEUE_CLEAR

- ▶ 응용 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [QUEUE_CREATE](#), [QUEUE_RCV_FROM](#), [QUEUE_SND_TO](#)

7.6.58.1. 기능

해당 Queue 의 FIFO 메모리를 지웁니다. Queue 메모리를 사용하기 위해서는 먼저, QUEUE_CREATE 아이템으로 Queue 를 생성해야 합니다.

Queue 관련 아이템에 대한 보다 자세한 사항은 [7.4.4. Queue 아이템](#) 을 참조하시기 바랍니다.

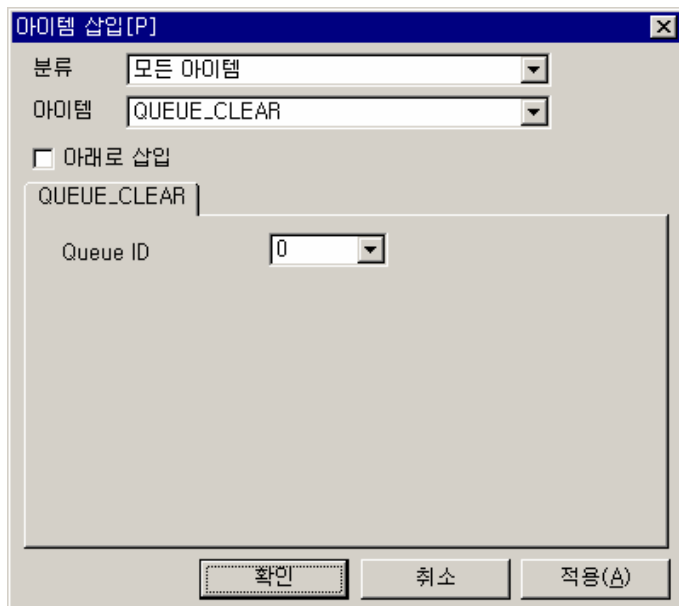


그림 7.6-61 QUEUE_CLEAR 입력 예

표 7.6-58 QUEUE_CLEAR 설정 항목

항 목	내 용
Queue ID	메모리를 초기화할 Queue 의 ID (0 ~ 3)

7.6.59. QUEUE_CREATE

- ▶ 응용 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [QUEUE_CLEAR](#), [QUEUE_RCV_FROM](#), [QUEUE_SND_TO](#)

7.6.59.1. 기능

Queue 메모리(FIFO)를 생성합니다. 최대 4 개까지 Queue 를 생성할 수 있으며, 각 Queue 별로 데이터의 길이를 달리 지정할 수 있습니다. 각각의 Queue 는 2Kbyte 의 내부 메모리를 차지하며, 하나의 Queue 는 데이터 길이에 따라 n 개(2000/데이터 길이)의 FIFO 메모리로 설정됩니다. Queue 관련 아이템에 대한 보다 자세한 사항은 [7.4.4. Queue 아이템](#) 을 참조하시기 바랍니다.

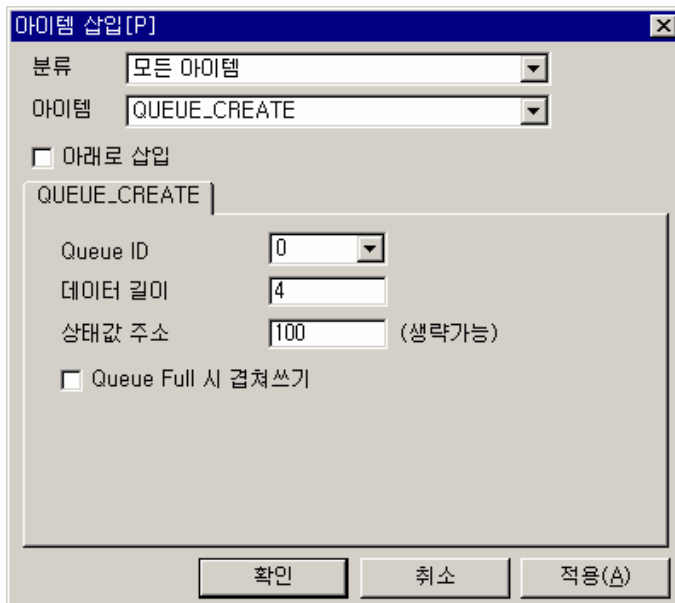


그림 7.6-62 QUEUE_CREATE 입력 예

표 7.6-59 QUEUE_CREATE 설정 항목

항 목	내 용
Queue ID	생성할 Queue 의 ID (0 ~ 3)
데이터 길이	Queue 메모리에 저장되는 데이터의 길이 데이터 길이에 따라 2KByte 의 Queue 는 설정된 크기를 갖는 n 개의 FIFO 메모리로 동작합니다. FIFO 개수는 2KByte 를 설정된 데이터 길이로 나눈 결과값이 됩니다. 최대 2Kbyte 까지 크기를 정할 수 있습니다.
상태값 주소	Queue 가 오류 없이 생성 되었는지의 정보를 저장 Queue 생성이 성공하면 0, 오류가 발생하면 1 을 저장합니다. 1 바이트 크기로 생략 가능합니다
Queue Full 시 겹쳐쓰기	Queue 메모리가 모두 사용되었을 경우, 즉, 데이터 저장 개수가 최대 용량을 초과한 경우 새로운 데이터의 저장을 허용할 것인지를 선택합니다. 체크하지 않으면 Queue 메모리가 모두 사용된 경우 새로운 데이터는 저장되지 않으며 QUEUE_RCV_FROM 아이템에 의해 Queue 에서 데이터를 읽어가야 새로운 데이터의 입력이 가능합니다. 체크하면 새로운 데이터는 가장 오래된 데이터를 지우고 그 자리에 써집니다.

7.6.60. QUEUE_RCV_FROM

- ▶ 응용 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [QUEUE_CLEAR](#), [QUEUE_CREATE](#), [QUEUE_SND_TO](#)

7.6.60.1. 기능

설정된 Queue 로부터 데이터를 읽어냅니다. 해당 Queue 의 FIFO 메모리에서 데이터 개수만큼 데이터 저장주소로 옮깁니다. FIFO 메모리의 특성대로 먼저 저장된 데이터가 먼저 읽히게 되며 Queue 가 비어있는 경우(읽을 데이터가 없는 경우) 상태값 주소에 에러 정보를 기록합니다. Queue 관련 아이템에 대한 보다 자세한 사항은 [7.4.4. Queue 아이템](#) 을 참조하시기 바랍니다.

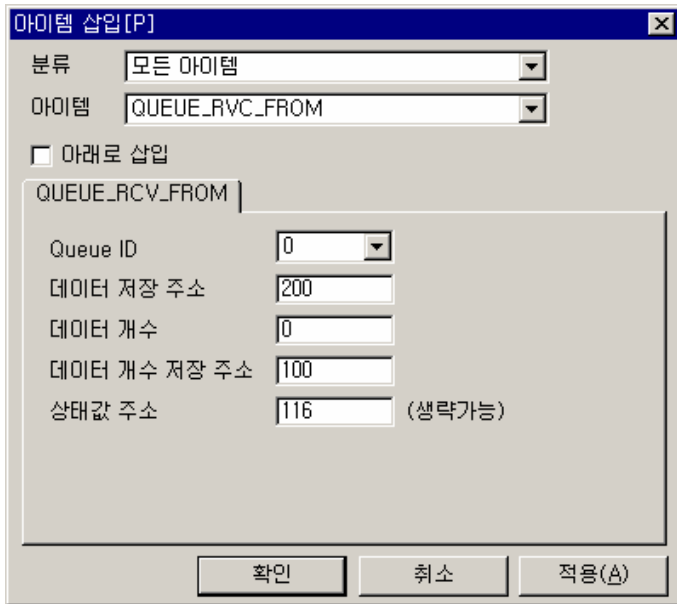


그림 7.6-63 QUEUE_RCV_FROM 입력 예

표 7.6-60 QUEUE_RCV_FROM 설정 항목

항 목	내 용
Queue ID	데이터를 읽어올 Queue 의 ID (0 ~ 3)
데이터 저장 주소	Queue 로부터 읽은 데이터를 저장할 주소를 지정합니다.
데이터 개수	Queue 로부터 읽을 FIFO 데이터의 최대 개수를 지정합니다. 0 으로 설정하면 저장된 데이터를 모두 가져옵니다.
데이터 개수 저장 주소	Queue 로부터 읽어온 FIFO 데이터의 개수를 저장할 주소를 지정합니다.
상태값 주소	Queue 에서 읽어온 결과를 저장하며 생략 가능합니다. '0'일 경우 에러 없이 지정한 개수를 읽어온 것을 의미하며 '01'일 경우 지정한 ID 의 Queue 가 없는 것을 의미하며 '02' 일 경우는 Queue 에 지정한 개수의 데이터가 없는 것을 의미합니다.

7.6.61. QUEUE_SND_TO

- ▶ 응용 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [QUEUE_CLEAR](#), [QUEUE_CREATE](#), [QUEUE_RCV_FROM](#),

7.6.61.1. 기능

설정된 Queue 에 데이터를 저장합니다. Queue 메모리가 모두 차는 경우는 옵션에 따라 데이터가 저장되지 않을 수 있습니다. 데이터 주소에서 해당 Queue 의 데이터 길이만큼 읽어 Queue 의 FIFO 메모리에 저장합니다.

Queue 관련 아이템에 대한 보다 자세한 사항은 [7.4.4. Queue 아이템](#) 을 참조하시기 바랍니다.

그림 7.6-64 QUEUE_SND_TO 입력 예

표 7.6-61 QUEUE_SND_TO 설정 항목

항 목	내 용
Queue ID	데이터를 저장할 Queue 의 ID (0 ~ 3)
데이터 주소	Queue 로 보낼 데이터가 있는 사용자 메모리 주소를 설정합니다. 설정된 주소의 데이터가 Queue 의 FIFO 메모리에 저장되며 데이터 길이는 Queue 생성시 설정한 데이터 길이로 고정됩니다.
상태값 주소	FIFO 메모리로 데이터를 보낸 결과를 표시합니다. 성공적으로 저장한 경우 '0' 이며 해당 Queue 가 없을 경우 '01'이고 해당 Queue 의 FIFO 메모리가 모두 사용되어 데이터를 옮길 수 없는 Queue-Full 의 경우 '02'의 에러를 표시합니다.

- * Queue Full 이라는 상태 값은 해당 Queue 를 생성할 때, QUEUE_CREATE 아이템의 <Queue Full 시 겹쳐 쓰기> 가 체크되지 않은 경우만 발생합니다.

7.6.62. REMARK

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.62.1. 기능

프로세스 내에 주석을 추가합니다. 이 아이템은 프로세스의 수행에 아무런 영향을 주지 않으며, 단지 프로그램의 크기를 증가시킵니다. 프로그램의 이해를 돕기 위한 설명을 덧붙이는데 사용됩니다.

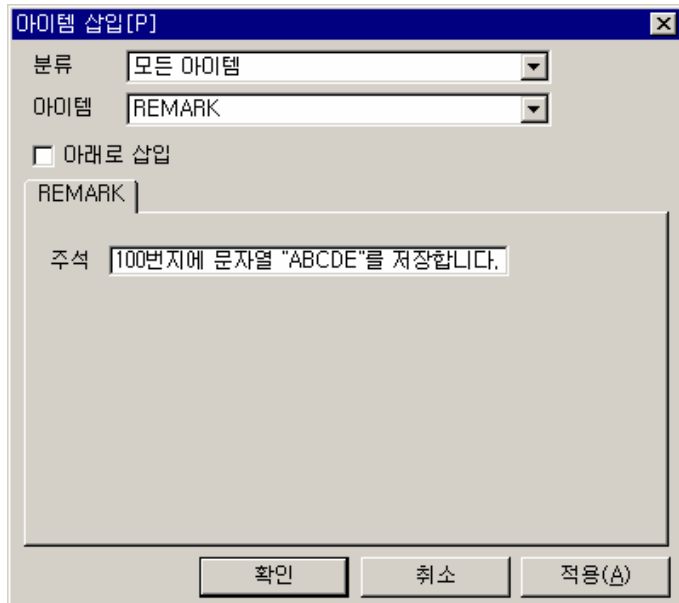


그림 7.6-65 REMARK 입력 예

표 7.6-62 REMARK 설정 항목

항 목	내 용
주석	프로세스 내에 표시할 주석문을 입력합니다. (최대 영문 40 자, 한글 20 자)

7.6.62.2. 예제

특정 아이템에 대한 주석을 입력해 보겠습니다.

▶ 예제 아이템 등록

MEM_SET 아이템을 삽입하고, <데이터 형태> 16 진수, <데이터> "01020304", <주소> "100", <길이> "4"를 각각 입력합니다.

REMARK 아이템을 삽입하고, 주석 입력란에 다음과 같이 입력합니다.

"100 번지에 4BYTE '01020304'를 저장합니다."

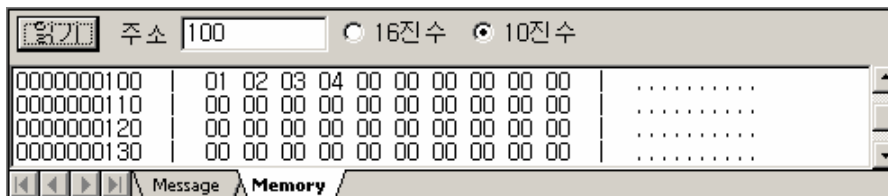
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	100
01	REMARK	100번지에 4BYTE '01020304'를 저장합니다.
02	MEM_SET	T:M100, Size:4, Type:16진수, Data:01020304
03	END_IF	

▶ 결과

100 번지에 01 02 03 04 가 저장되었으며, REMARK 아이템의 주석문은 메모리에 아무런 영향을 주지 않습니다.



7.6.63. RESET

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.63.1. 기능

ETOS 를 리셋하여 초기화합니다. 무한루프에 빠지거나 에러 상황이 발생하여 정상적으로는 복구가 불가능한 경우 사용하여, 정상 동작을 유도합니다.

ETOS-1000 이중화 모델은 Active/Standby CPU 가 절체되어, 역할을 서로 바꾸는 기능을 합니다.

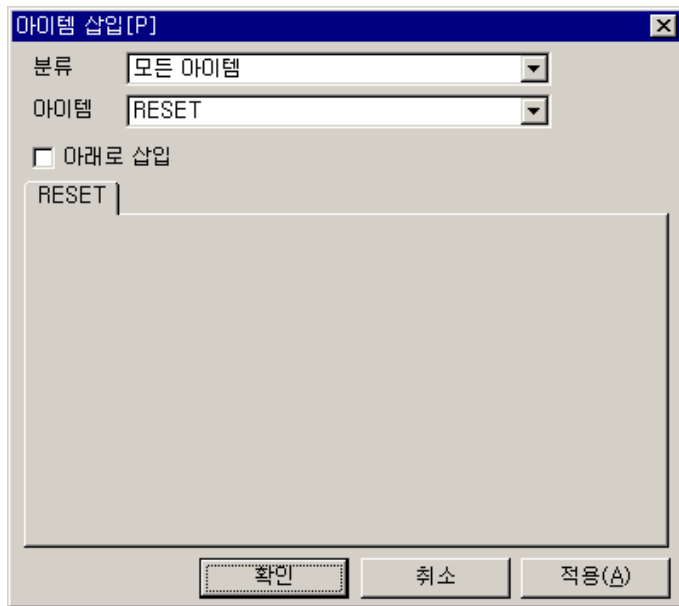


그림 7.6-66 RESET 아이템

표 7.6-63 RESET 설정 항목

항 목	내 용
없음	설정 항목이 없습니다

7.6.63.2. 예제

ETOS 메모리 100 번지를 1 초마다 1 증가시켜, 30 이 되는 경우에 ETOS 를 초기화 합니다.

▶ 예제 아이템 등록

IF_TIMER 아이템을 추가하고, <주기>에 “100”을 입력합니다.

ARITHMETIC 아이템을 삽입하고, 100 번지에 1 을 더해 100 번지에 저장합니다.

IF_NUM_CMP 아이템을 삽입하고, 100 번지가 상수 30 과 같은지를 체크합니다.

RESET 아이템을 IF_NUM_CMP 아이템과 END_IF 아이템 사이에 삽입합니다.

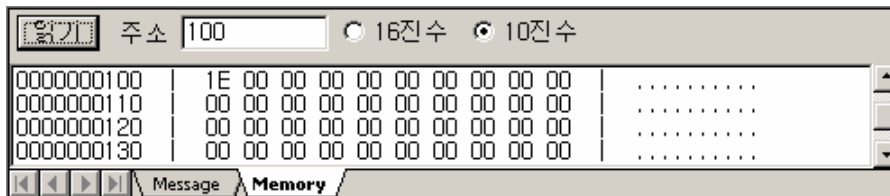
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_TIMER	100
01	ARITHMETIC	Type:U1, M100, +, 1, M100
02	IF_NUM_CMP	Type:U1, 30, =, M100
03	RESET	
04	END_IF	
05	END_IF	

▶ 결과

100 번지값이 1 초마다 증가하여 1E_H(30)가 되면 리셋되어 메모리가 0 으로 초기화 되고, 100 번지는 다시 1 씩 증가합니다.



* 리셋 후, 수 초 간은 시스템 초기화 중이므로, 메모리 읽기 등이 불가능할 수 있습니다.

7.6.64. SECS_MC_FRAME_SND

- ▶ 통신 아이템 (SECS 통신 전용)
- ▶ ETOS-150-HSMS, ETOS-150-Dual-HSMS , ETOS-200-HSMS

7.6.64.1. 기능

SECS-I 이나 HSMS 로 지정한 프레임을 송신합니다. 송신 프레임은 미리 등록해야 합니다.

다음 그림은 SECS_MC_FRAME_SND 아이템으로 S1F2_SND 프레임을 송신하는 예입니다.

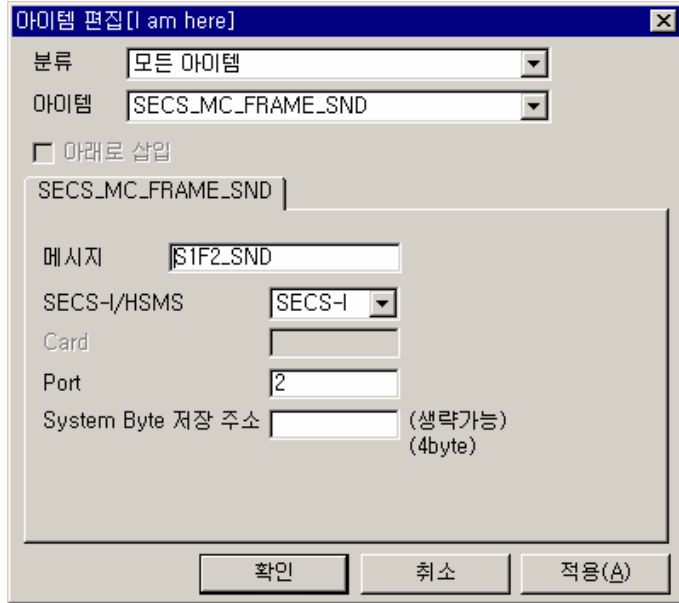


그림 7.6-67 SECS_MC_FRAME_SND 입력 예

표 7.6-64 SECS_MC_FRAME_SND 설정 항목

항 목	내 용
메시지	송신할 SECS 프레임 이름을 입력합니다. 키보드 입력 대신 프레임을 선택한 후 마우스 왼쪽 버튼을 누른채로 프레임 입력란에 가져다 놓아도 됩니다.(DRAG & DROP)
SECS-I/HSMS	RS-232 를 사용하는 SECS-I 과 이더넷을 사용하는 HSMS 를 선택합니다.
Card	ETOS-200 에서 SECS-I 을 사용하는 경우, CARD 번호를 설정합니다.
Port	Port 번호(SECS-I:COM1,COM2..., HSMS:0,1...)를 설정합니다.
System Byte 저장 주소	송신할 프레임의 System Byte 가 따로 있는 경우, 메모리의 위치를 지정할 수 있습니다. 생략 가능합니다. 생략시엔 ETOS 가 자동으로 값을 할당합니다. 10. SECS 통신의 예제를 참조하십시오.

7.6.64.2. 예제

COM2 로 S1F2 를 송신합니다..

▶ 예제 프레임 정의

먼저 예제 프레임을 다음과 같이 정의합니다.



▶ 예제 아이템 등록

SECS_MC_FRAME_SND 아이템을 등록하고, 메시지에 S1F2_SND 를 입력합니다.

▶ 예제 프로세스

편집을 마친 프로세스 화면입니다. 예제에는 프로세스의 시작을 위해서 IF_SECS_MC_FRAME_RCV 아이템도 추가 되어 있습니다.

NO	아이템	설명
00	IF_SECS_MC_FRAME_RCV	S1F1_RCV, Port:COM2
01	SECS_MC_FRAME_SND	S1F2_SND, Port:COM2
02	END_IF	

▶ 결과

정의된 S1F1_RCV 프레임을 수신하면, ETOS 는 프레임 정의를 참조하여 해당 메모리 위치에 데이터를 저장하고, S1F2 를 송신 합니다.

7.6.65. SLAVE_MEM_READ_WRITE

- ▶ 응용 아이템 (IFOR 전용)
- ▶ ETOS-200/1000 IFOR
- ▶ [EVENT_READ_WRITE](#), [IF_EVENT_OCCUR](#)

7.6.65.1. 기능

ETOS 시리즈의 IFOR(Intelligent Fiber Optic dual-Ring system) 모듈 전용 아이템으로, IFOR 모듈이 장착된 시스템에서만 사용합니다. IFOR는 ETOS-200/1000 용의 광링(Optic Ring) 제어 모듈입니다. IFOR 전용 아이템을 편집하기 위해서는 [옵션] -> [편집 옵션]에서, IFOR 편집을 체크해야 합니다.

ETOS의 사용자 메모리와 공유 메모리^(*129)간의 데이터를 교환하기 위한 아이템입니다. IFOR 보드가 광 링을 이용해서 공유하고 있는 메모리를 사용자 메모리로 가져오거나 반대로 사용자 메모리의 데이터를 공유 메모리로 보낼 때 사용합니다.

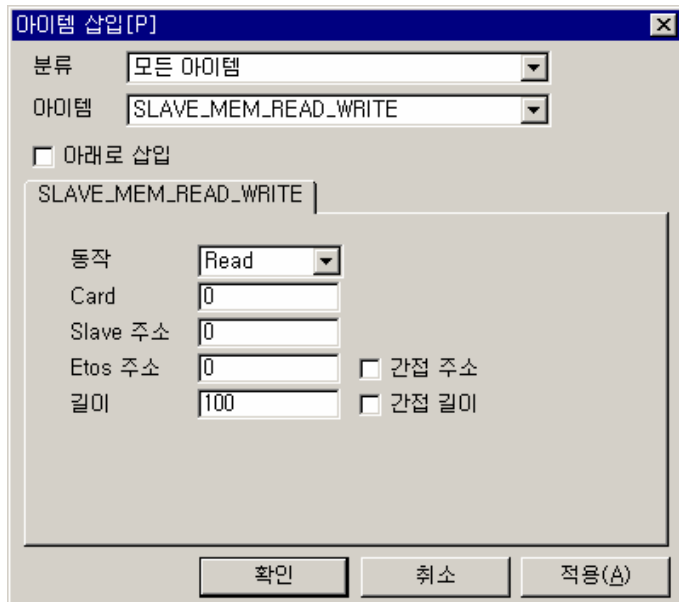


그림 7.6-68 SLAVE_MEM_READ_WRITE 입력 예

표 7.6-65 SLAVE_MEM_READ_WRITE 설정 항목

항 목	내 용
동작	Read : 공유 메모리를 읽어 사용자 메모리에 저장합니다. Write : 사용자 메모리의 데이터를 공유 메모리에 저장합니다.
카드	시리얼 카드의 번호 설정 (0 ~ 7)(*130)
Slave 주소	공유 메모리 주소를 의미합니다. [온라인] -> [IFOR Shared Memory 설정]에서 설정한 주소를 입력합니다.

(*129) 각 국은 설정된 시간마다 전 국으로 자기의 메모리를 Broadcasting 하여, 모든 국이 동일한 메모리를 가지고 있는 것처럼 보이게 합니다. 최대 약 50Kbytes의 메모리를 공유할 수 있습니다.

(*130) ETOS-200/1000에서만 활성화 됨

표 7.6-65 SLAVE_MEM_READ_WRITE 설정 항목 (계속)

항 목	내 용
Etos 주소	공유 메모리를 저장할 ETOS 사용자 메모리 주소이거나,(Read 동작일 경우) 공유 메모리에 저장할 데이터가 위치한 ETOS 사용자 메모리 주소를 의미합니다.(Write 동작일 경우)
간접 주소	주소를 간접으로 입력합니다 입력한 위치의 데이터 값이 주소가 됩니다.(4 바이트)
길이	읽거나 쓸 데이터의 바이트 길이입니다. IFOR Shared Memory 블록 설정에서 설정한 블록의 길이보다 작거나 같아야 합니다.(반드시 일치하지 않아도 상관없습니다)
간접 길이	길이를 간접 지정합니다 길이에 입력한 데이터는 길이가 아닌 주소가 되며, 그 위치의 데이터 값이 길이가 됩니다. (2 바이트)

* 사용예 등의 보다 자세한 사항은 ETOS-IFOR 사용설명서를 참조하시기 바랍니다.

7.6.66. STR_TO_INT

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [INT_TO_STR](#)

7.6.66.1. 기능

10 진수 또는 16 진수 문자열을 정수, 또는 실수로 변환합니다.

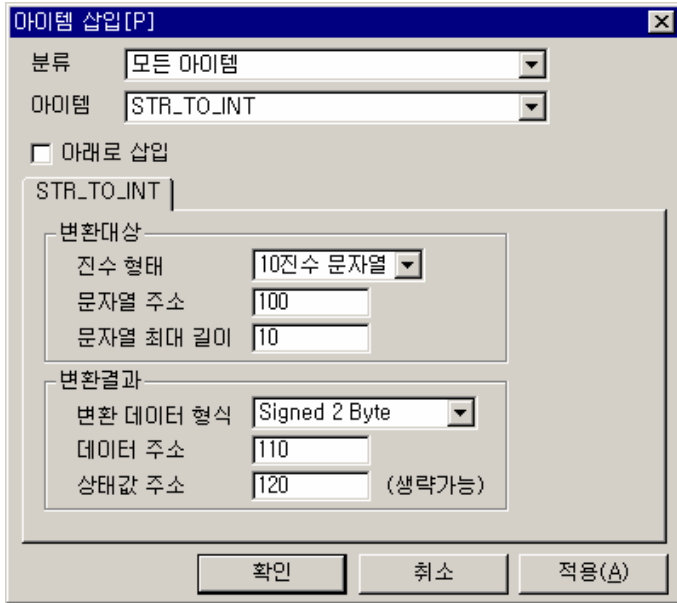


그림 7.6-69 STR_TO_INT 입력 예

표 7.6-66 STR_TO_INT 설정 항목

항 목	내 용	
변환 대상	진수 형태	변환 대상 데이터의 진수 형태를 선택합니다. 10 진수 문자열: 대상 문자열은 10 진수 값을 나타냅니다. 16 진수 문자열: 대상 문자열은 16 진수 값을 나타냅니다.
	문자열 주소	변환 대상 문자열이 저장되어 있는 위치를 입력합니다.
	문자열 최대 길이	변환 대상 문자열의 최대 길이를 지정합니다. 메모리의 데이터 중 NULL(Ox 00) 전까지를 문자열로 보며, 실제 문자열의 길이가 최대 길이를 초과할 경우에는 최대 길이만큼만 변환합니다. 최대 길이는 20 바이트입니다.
변환 결과	변환 데이터 형식	변환 결과 데이터의 형식을 선택합니다.
	데이터 주소	변환 결과값을 저장할 메모리의 주소를 입력합니다.
	상태값 주소	정상 변환되면 변환된 데이터의 길이가 저장됩니다.

7.6.66.2. 예제

10 진수 문자열 “100”을 상수로 변환합니다.

100 번지: “100”(문자열) -> 110 번지: 64 00 (Unsigned 2 byte 정수)

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 추가합니다.

MEM_SET 아이템을 삽입하고, <데이터 형태> ‘문자열’, <데이터> “100”, <주소> “100”, <길이> “3”을 각각 입력합니다.

STR_TO_INT 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목		입력 내용
변환 대상	진수 형태	10 진수 문자열 선택
	문자열 주소	100
	문자열 최대 길이	10
변환 결과	변환 데이터 형식	Unsigned 2 Byte 선택
	데이터 주소	110
	상태값 주소	120

▶ **예제 프로세스**

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:3, Type:문자열, Data:100
02	STR_TO_INT	S:M100, T:M110, Type:S2, Radix:10, MaxSize:10, Status:M120
03	END_IF	

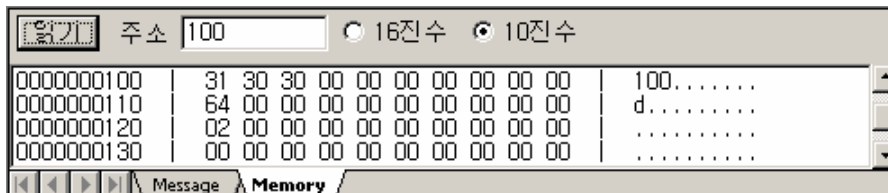
▶ **결과**

100 번지의 문자열 100 은 10 진수 문자열이므로 정수 100 을 의미합니다. 변환 결과는 110 번지에 부호없는 2 바이트 정수값으로 저장되었습니다.

6400_H = (100)

120 번지 상태값 주소에는 변환 결과 데이터의 길이가 저장되었습니다

02 00 = (2 바이트)



7.6.67. SWAP

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.67.1. 기능

워드(2 바이트) 또는 더블워드(4 바이트) 데이터의 바이트 위치를 서로 교환합니다. 워드 스왑(2 BYTE Swap)인 경우, 워드의 상, 하위 바이트 위치를 교환합니다. 또, 더블워드 스왑(4 BYTE Swap)인 경우, 더블 워드 1,2,3,4 번 바이트의 위치를 4,3,2,1 위치로 교환합니다.

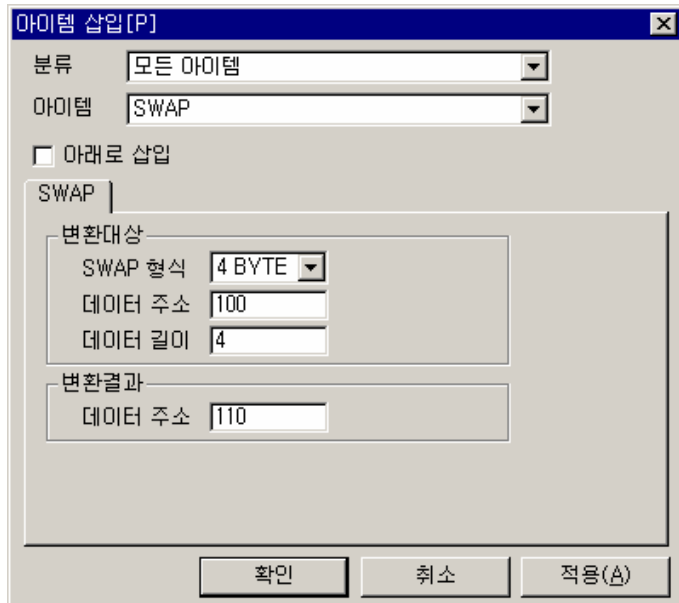


그림 7.6-70 SWAP 입력 예

표 7.6-67 SWAP 설정 항목

항 목	내 용
SWAP 형식	교환 형식을 지정합니다. 워드 단위(2 BYTE) / 더블 워드 단위(4 BYTE)
데이터 주소	변환 대상 데이터가 저장되어 있는 메모리 위치를 입력합니다.
데이터 길이	변환 대상 데이터의 길이를 입력합니다. 데이터 길이는 SWAP 형식의 배수로 설정을 합니다.
데이터 주소	변환 결과 데이터가 저장될 메모리 위치를 지정합니다.

7.6.67.2. 예제

100 번지값 11 22 33 44 를 4BYTE Swap 해 보겠습니다.

▶ **예제 아이템 등록**

IF_POWER_ON 아이템을 추가합니다.

MEM_SET 아이템을 삽입하고, <데이터 형태> '16 진수', <데이터> "11223344", <주소> "100", <길이> "4"를 각각 입력합니다.

SWAP 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목	입력 내용
SWAP 형식	4 BYTE
데이터 주소	100
데이터 길이	4
데이터 주소	110

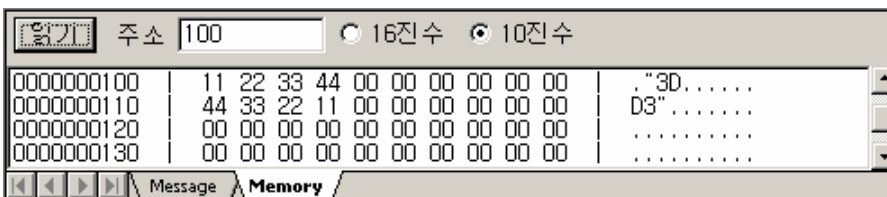
▶ **예제 프로세스**

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:4, Type:16진수, Data:11223344
02	SWAP	S:M100, T:M110, Size:4, Type:4
03	END_IF	

▶ **결과**

100 번지의 값 11 22 33 44 는 SWAP 에 의해 110 번지에 44 33 22 11 로 저장되었습니다.



7.6.68. TIME_SETTING

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [TIME_STR](#)

7.6.68.1. 기능

지정 위치의 문자열을 참조하여, ETOS 현재 시각을 변경합니다. 문자열은 `yyyymmddHHMMSS` 형식으로 저장되어 있어야 합니다. (예, “20100110142034”)

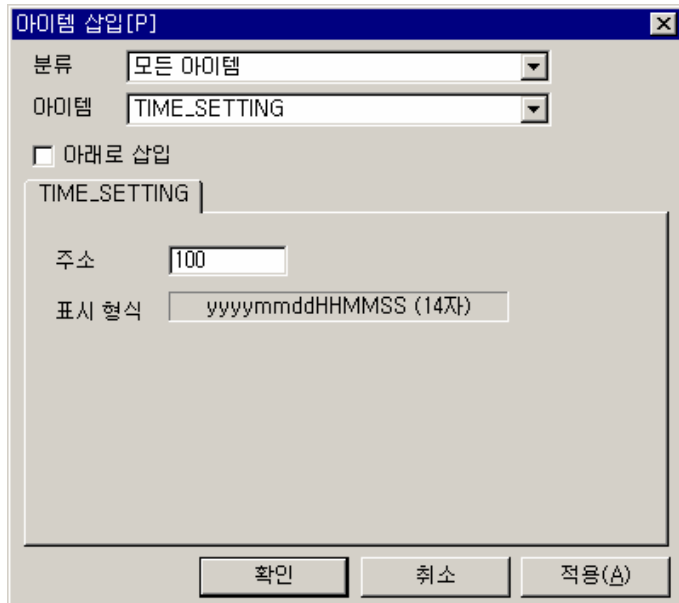


그림 7.6-71 TIME_SETTING 입력 예

표 7.6-68 TIME_SETTING 설정 항목

항 목	내 용
주소	시각을 나타내는 문자열의 위치를 지정합니다.
표시형식	문자열의 표시형식을 보여줍니다.

7.6.68.2. 예제

ETOS 의 시스템 시각을 변경합니다.

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.

MEM_SET 아이템을 삽입하고, <데이터 형태> '문자열', <데이터> "20040619142000", <주소> "100", <길이> "14"를 각각 입력합니다.

TIME_SETTING 아이템을 삽입하고, <주소>에 "100"을 입력합니다.

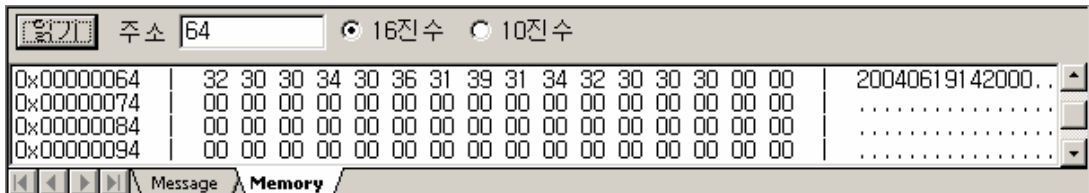
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

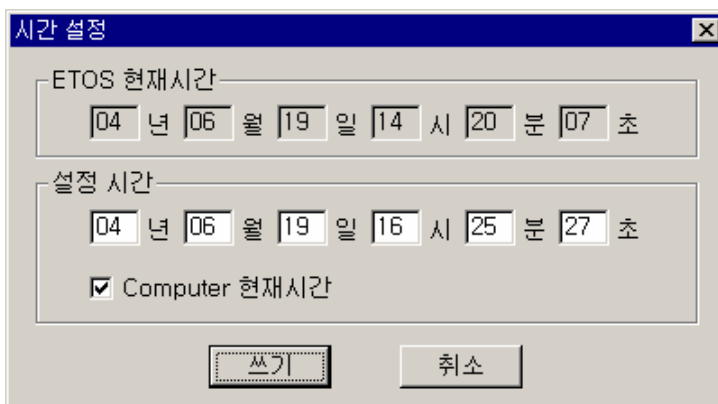
NO	아이템	설명
00	IF_POWER_ON	
01	MEM_SET	T:M100, Size:14, Type:문자열, Data:20040619142000
02	TIME_SETTING	M100
03	END_IF	

▶ 결과

100 번지에서 14 바이트의 시각을 의미하는 문자열을 볼 수 있습니다.



[온라인] -> [시간 설정]을 선택해서, 현재 시각이 지정 시각(문자열이 표현하는 시각)으로 변경되었는지 확인할 수 있습니다.



7.6.69. TIME_STR

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000
- ▶ [TIME_SETTING](#)

7.6.69.1. 기능

ETOS의 시스템 시각을 문자열 형태로 메모리에 저장합니다. TIME_SETTING의 기능과 반대의 기능을 합니다. 그러나 표시 형식은 “yyyy/mm/dd HH:MM:SS”로 조금 다릅니다. (TIME_SETTING은 “yyyymmddHHMMSS”의 14 문자 형식)

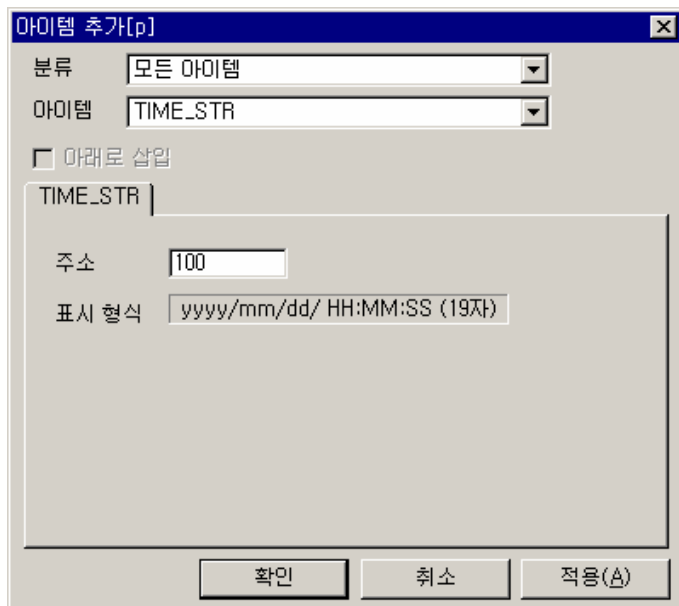


그림 7.6-72 TIME_STR 입력 예

표 7.6-69 TIME_STR 설정 항목

항 목	내 용
주소	시각을 나타내는 문자열(yyyy/mm/dd HH:MM:SS)을 저장할 위치를 지정합니다.
표시형식	저장 문자열의 형식을 보여줍니다.

7.6.69.2. 예제

100 번지에 Power On 시의 시각을 표시합니다.

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.

TIME_STR 아이템을 삽입하고, <주소>에 “100”을 입력합니다.

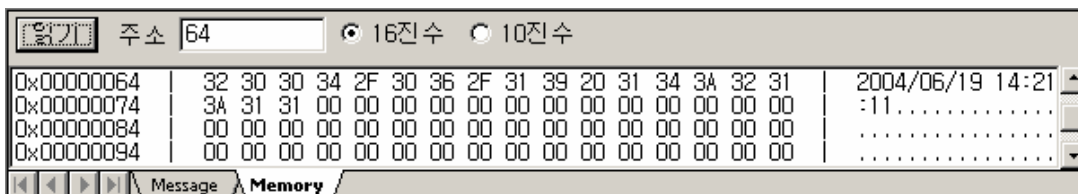
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	TIME_STR	M100
02	END_IF	

▶ 결과

100 번지에 ETOS 가 Power On 된 시각이 기록됩니다.
 (“2004/06/19 14:23:04”)



7.6.70. TYPE_CONVERT

- ▶ 기본 아이템
- ▶ ETOS-50/100/150/200/1000

7.6.70.1. 기능

정수나 실수를 다른 형식의 정수나 실수로 변환합니다.

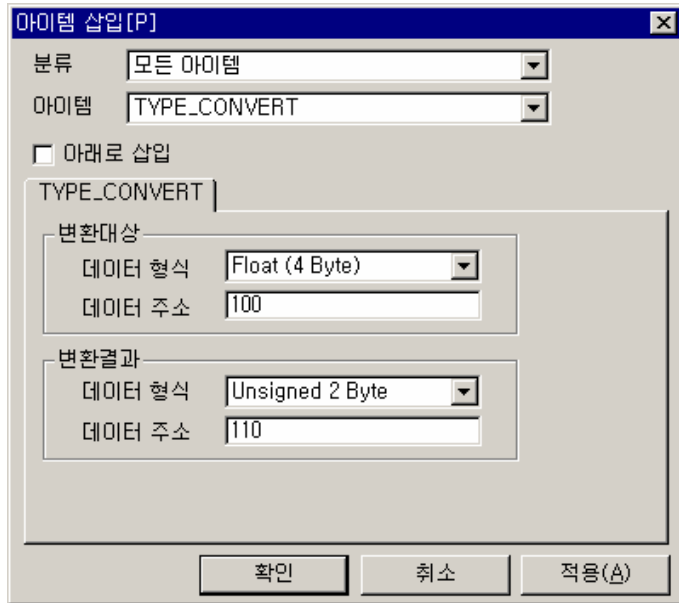


그림 7.6-73 TYPE_CONVERT 입력 예

표 7.6-70 TYPE_CONVERT 설정 항목

항 목		내 용
변환 대상	데이터 형식	변환 대상 데이터의 형식을 선택합니다. 정수는 signed, unsigned 1/2/4 바이트가 있습니다. 실수는 float/ Double/brabender float 가 있습니다.
	데이터 주소	변환 대상 데이터의 주소를 지정합니다.
변환 결과	데이터 형식	변환 결과 데이터의 형식을 선택합니다. 정수는 signed, unsigned 1/2/4 바이트가 있습니다. 실수는 float/ Double/brabender float 가 있습니다.
	데이터 주소	변환 결과 데이터가 저장될 위치를 지정합니다.

7.6.70.2. 예제

부동소수점 상수(Float) 123 을 부호없는 2 바이트 정수로 변환합니다.

▶ 예제 아이템 등록

IF_POWER_ON 아이템을 추가합니다.

ARITHMETIC 아이템을 삽입하고, 상수 123 에 상수 1 을 곱해서 100 번지에 Float 형식으로 저장합니다.

TYPE_CONVERT 아이템을 삽입하고, 다음과 같이 입력합니다.

항 목		입력 내용
변환 대상	데이터 형식	Float(4 Byte)
	데이터 주소	100
변환 결과	데이터 형식	Unsigned 2 Byte
	데이터 주소	110

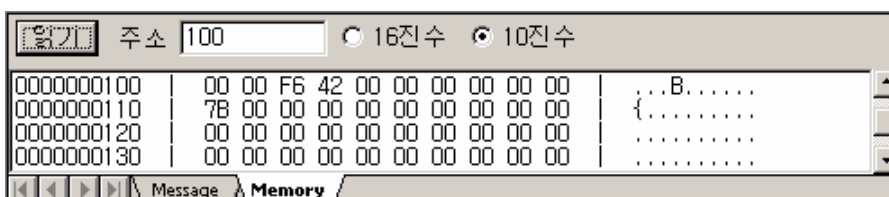
▶ 예제 프로세스

지금까지 작성한 프로세스는 다음과 같습니다.

NO	아이템	설명
00	IF_POWER_ON	
01	ARITHMETIC	Type:Float, 123, *, 1, M100
02	TYPE_CONVERT	S:Float,M100, T:U2,M110
03	END_IF	

▶ 결과

100 번지에는 부동소수점 상수 123 이 FLOAT 형식(00 00 F6 42) 으로 저장되어 있습니다. 110 번지에는 TYPE_CONVERT 아이템에 의해 2 바이트 정수로 변환된 결과 7B 00 (007BH = 123) 가 저장되었습니다.



7.7. 단종 아이템

더 이상 사용되지 않는 아이템으로, 호환성을 위해 존재하나 사용해서는 안되는 아이템입니다.
주의 하시기 바랍니다.

▶ [온라인] -> [편집 옵션]에서 <Legacy 아이템 편집>을 체크해야 나타납니다.

아이템	기능	대체 아이템
ENET_FRM_SND	이더넷 프레임 송신	FRAME_SND
EVENT_NOTIFY	이더넷 사용자 정의 이벤트	
IF_ENET_FRM_RCV	이더넷 프레임 수신	IF_FRAME_RCV
IF_ENET_REQ_RESP	이더넷 프레임 송신 후, 응답 대기	IF_REQ_RESP
IF_MEM_WRITE	이더넷 드라이버에 의해 메모리 변경시 이벤트 발생	IF_MEM_WRITE_RCV
IF_SECS_FRAME_RCV	수신 프레임으로 정의한 SECS 프레임을 수신	IF_SECS_MC_FRAME_RCV
IF_SECS_REQ_RESP	SECS 프레임을 송신하고, 정의한 SECS 프레임 수신	IF_SECS_MC_REQ_RESP
IP_SETTING	메모리값으로 현재 IP 주소를 변경	
SECS_ALM_SND	S6F11 송신	SECS_MC_FRAME_SND
SECS_FRAME_SND	SECS-I이나 HSMS로 지정한 프레임 송신	SECS_MC_FRAME_SND

시리얼 통신

제 8 장

범용 시리얼 통신(RS232C/RS422/RS485) 방식을 사용하는 장비와의 통신을 설명합니다.

	내용	위치
8.1	포트 설정	310
8.2	MODBUS Master	316
8.3	MODBUS Slave	319
8.4	GLOFA Master	322
8.5	GLOFA Slave	324
8.6	사용자 정의 프로토콜	326

8. 시리얼 통신

RS232C/RS422/RS485 방식을 사용하는 장비와의 통신을 설명합니다.

ETOS는 몇 가지의 내장 프로토콜^(*131)과 사용자 정의 프로토콜을 사용하여 시리얼 장비와 데이터를 주고 받을 수 있습니다. 내장 프로토콜을 사용하는 경우, 사용자는 파라미터 또는 파라미터와 프로세스 정의만으로 상대 기기와 통신이 가능합니다. 물론, 내장 프로토콜이 아닌 경우에도 상대 장비가 사용하는 프로토콜의 프레임 정의를 하면 데이터 통신이 가능합니다.

8.1. 포트 설정

먼저 사용자는 상대장비와의 통신에서 누가 마스터 역할을 할지 결정해야 합니다. 각 포트는 포트별로 마스터나 슬레이브로 동작할 수 있습니다. 따라서, COM1은 슬레이브인 동시에, COM2는 마스터인 구성이 가능합니다.(그림 참조)

다음 그림의 예를 설명하겠습니다. ETOS는 COM1 포트를 통해 상위 MMI PC와 연결되어 있고, COM2 포트를 통해 하위 PLC와 연결되어 있습니다. ETOS는 PLC 데이터를 ETOS 사용자 메모리로 가져오며, 상위 MMI PC는 ETOS 사용자 메모리를 참조하여, PLC 정보를 화면에 표시하게 됩니다.

ETOS는 하위 PLC에 대해 마스터로 동작하여, PLC 데이터를 ETOS 사용자 메모리로 가져옵니다. (어떤 데이터를 언제 얼마만큼 가져올지는 ETOS의 프로그램이 결정하며, PLC 프로그램은 필요 없습니다.) 따라서 COM2 포트는 마스터 포트에 설정되며, ETOS는 상대방에 데이터 읽기/쓰기 요청을 하고, 상대방은 수동적으로 응답합니다. COM2 포트의 파라미터 설정과, 읽기/쓰기 요청에 대한 프로세스 정의가 필요합니다.

COM1 포트는 PC에 대해 슬레이브 포트에 설정되어, 상대 PC의 읽기/쓰기 요청에 대해 수동적으로 응답합니다. 일반적으로, 슬레이브 포트는 프로세스 없이 파라미터 설정만으로 동작이 가능합니다.

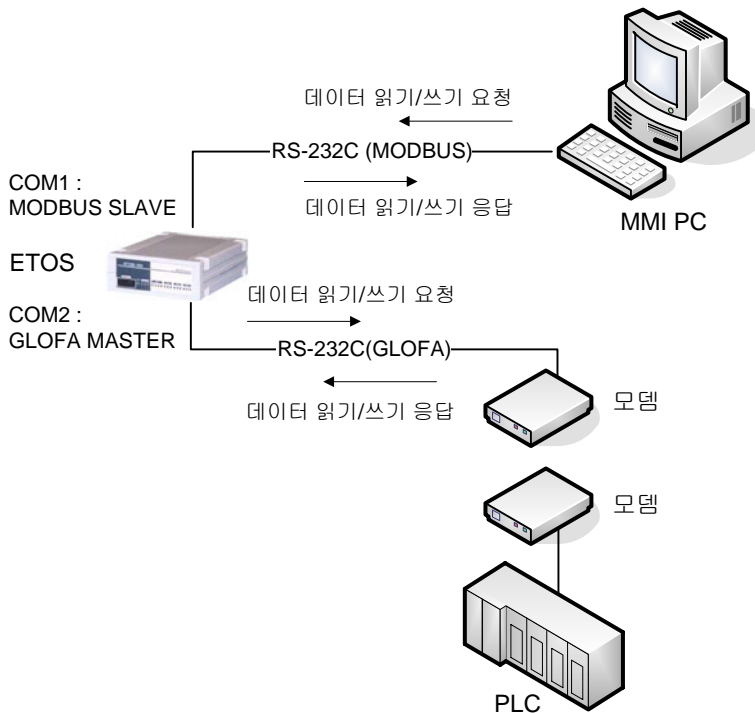


그림 8.1-1 시리얼 통신 시스템 구성 예

^(*131) OS version 1.20을 기준으로 MODBUS, GLOFA Cnet을 제공. 이하 버전에서는 지원하지 않음.

ETOS-PD 의 [편집] -> [파라미터 설정]을 선택합니다.

이제, 사용하고자 하는 COM 포트를 선택한 후 더블클릭 또는 [편집] 버튼을 누르고, 포트의 종류를 'Serial'로 설정합니다.

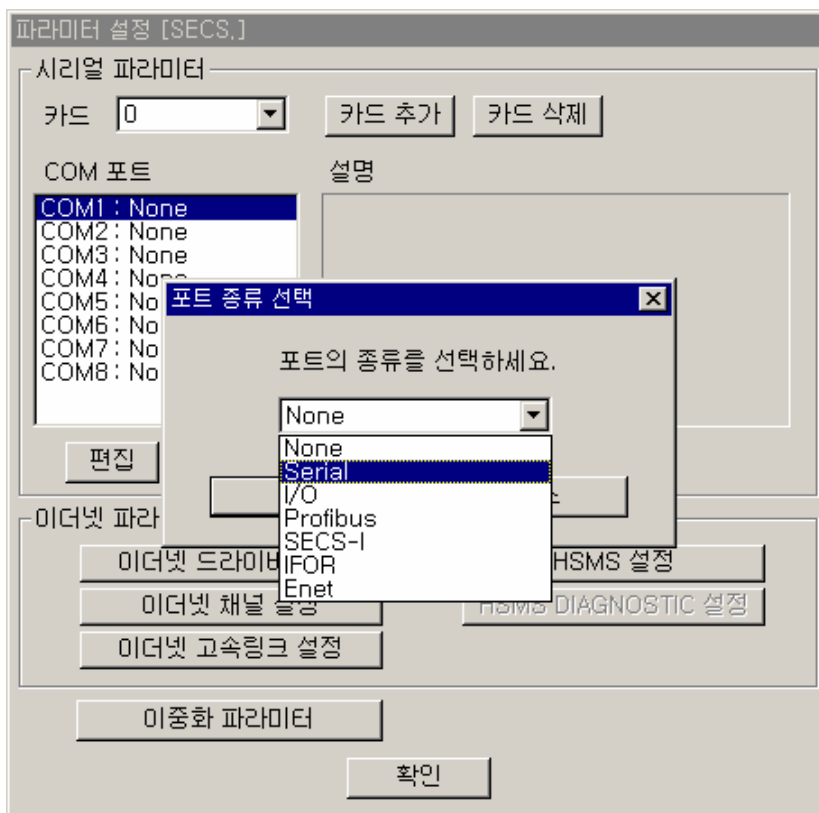


그림 8.1-2 포트 종류 선택(Serial)

화면에 표시되는 포트의 종류가 그림보다 적을 수 있습니다. [옵션] -> [편집 옵션]에서 옵션을 선택해야 나타나는 포트^(*132)가 있으며, SECS-1 포트는 드라이버의 종류가 SECS 인 경우에만 화면에 표시됩니다.

(*132) IFOR, Enet 은 [옵션] -> [편집 옵션]의 <IFOR 편집>을 체크한 경우만 화면에 표시됩니다.

8.1.1. 시리얼 통신 설정

'Serial' 로 설정한 후 [확인] 버튼을 누르면 {COM 포트 설정 (Serial)} 대화상자가 나타납니다. 이때, <통신형태>, <통신속도>, <데이터 비트>, <정지비트>, <패리티> 등이 기본값^(*133)으로 설정되어 있습니다. 통신하려고 하는 범용 시리얼 통신 장비와 통신 형태, 속도 등의 통신 파라미터를 일치시켜야 합니다.

또, 상대국과의 통신에서 사용할 프로토콜을 선택해야 합니다. ETOS 가 제공하지 않는 프로토콜을 사용해야 하는 경우에는, <프로토콜>을 'User Defined' 으로 설정하고, 프레임을 정의해서 사용해야 합니다. 다음 그림은 <그림 8.1-1. 시리얼 통신 시스템 구성 예>의 COM1 포트와 같이 프로토콜을 MODBUS 로 설정하고, 동시에 슬레이브 포트로 설정한 예입니다.

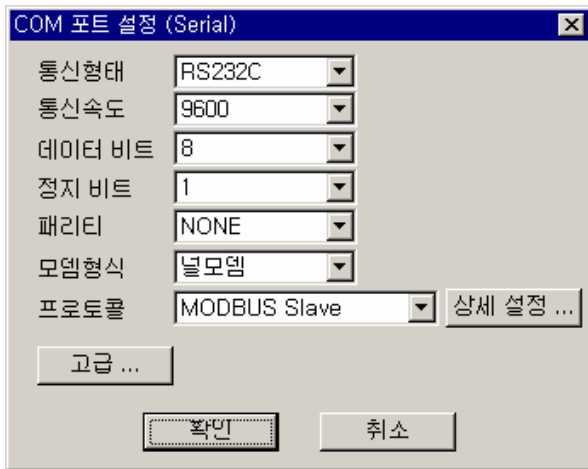


그림 8.1-3 COM 포트 설정 예

표 8.1-1 시리얼 포트 설정 항목

항 목	내 용
통신형태	RS232C/RS422/RS485/LOG 중에서 선택합니다.
통신속도	통신 속도(baud rate, bps)를 설정합니다.
데이터비트	7bit/ 8bit
정지비트	1bit/ 2bit
패리티	NONE(없음)/EVEN(짝수)/ODD(홀수)/SPACE(1)/MARK(0)
모뎀 형식	널 모뎀/전용 모뎀 에서 선택 ^(*134)
프로토콜	내장 프로토콜 ^(*135) 을 사용하는 경우 선택합니다.
상세 설정	내장 프로토콜 별 상세 설정 ^(*136)
고급	시리얼 통신 고급 설정 참조

(*133) RS232C/9600/8/1/None/널모뎀

(*134) 다이얼업 모뎀은 전용 모뎀 선택

(*135) OS version 1.20 을 기준으로 MODBUS, GLOFA Cnet 을 제공. 이하 버전에서는 지원하지 않음.

(*136) 프로토콜 별로 다릅니다. 해당 프로토콜 편을 참조하시기 바랍니다.

8.1.2. 내장 프로토콜 설정

<프로토콜>에서 사용하고자 하는 프로토콜과 마스터/슬레이브 역할을 설정할 수 있습니다.

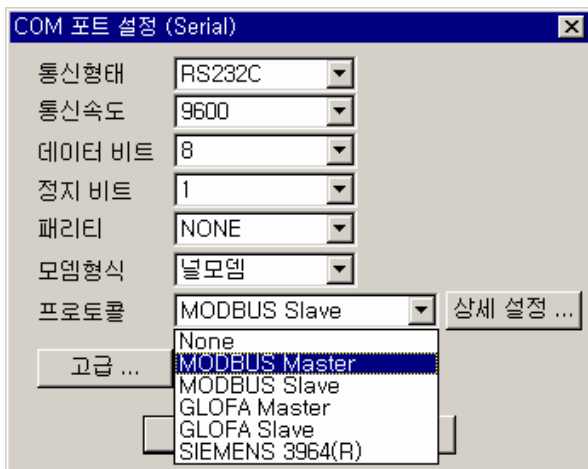


그림 8.1-4 COM 포트 프로토콜 설정

현재 사용 가능한 프로토콜 종류는 다음과 같습니다.

표 8.1-2 ETOS 지원 시리얼 통신 프로토콜

프로토콜	설 명	비 고
User Defined	아래 열거한 프로토콜 이외의 프로토콜을 사용하는 경우	프레임을 정의해야 합니다.
MODBUS Master	MODBUS RTU/ASCII 마스터	OS 버전 1.20 이후 지원
MODBUS Slave	MODBUS RTU/ASCII 슬레이브	OS 버전 1.20 이후 지원
GLOFA Master	GLOFA Cnet 마스터	OS 버전 1.20 이후 지원
GLOFA Slave	GLOFA Cnet 슬레이브	OS 버전 1.20 이후 지원
SIEMENS 3964(R)	SIEMENS 3964(R) 마스터/슬레이브	추후 지원 예정

사용하고자 하는 프로토콜이 있는 경우엔 해당 프로토콜을 선택하고, 그렇지 않은 경우엔 'User Defined'를 선택하시기 바랍니다. 각 프로토콜 별 상세설정 등의 내용은 8.2. MODBUS Master 에서부터 각 프로토콜별로 설명하겠습니다.

8.1.3. 시리얼 통신 고급 설정

[고급...]버튼을 누르면 고급 통신설정을 할 수 있는 {시리얼 통신 고급 설정} 대화상자가 나타납니다. 시리얼 통신 환경을 잘 이해하고 있는 사용자만 변경해야 합니다.

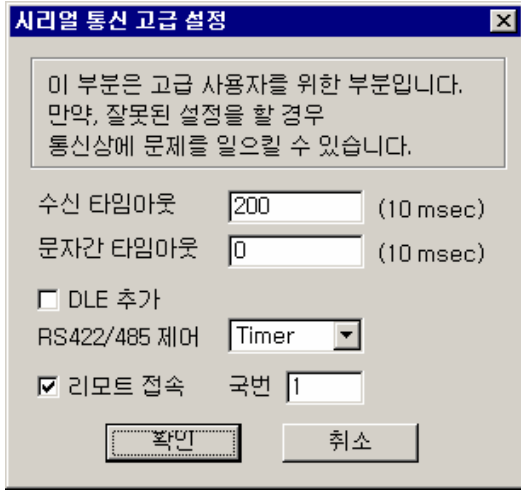


그림 8.1-5 시리얼 통신 고급 설정 예

표 8.1-3 시리얼 통신 고급 설정 항목

항 목	내 용
수신 타임아웃	IF_REQ_RESP 아이템 사용 시 수신 프레임 기다리는 최대 시간입니다. 정상 응답 프레임이 수신되면, 수신 즉시 다음 아이템으로 진행하며, 응답이 없을 경우, 수신 타임 아웃 시간만큼 해당 아이템에서 대기합니다.(수신 프레임을 기다립니다.)
문자간 타임아웃	프레임 수신 중 이 시간 동안 다른 데이터가 수신되지 않는 경우 여기까지 한 프레임으로 인식. 0 으로 설정되어 있으면 4 문자 전송시간이 프레임 구분 시간이 됩니다. (9600bps 에서 약 4ms)
DLE 추가	프레임 송신시에는 프레임 중에 DLE(0x10) 이 존재하면 DLE 를 추가하고, 프레임 수신시에는 추가되어 있는 DLE(0x10)를 삭제합니다.
RS422/485 제어	COM 포트의 <통신형태>를 'RS422' 또는 'RS485'로 설정한 경우만 활성화되어, 송,수신 타이밍을 제어 합니다. 자세한 사항은 8.1.2.1. RS422/485 제어방법 을 참조하시기 바랍니다.
리모트 접속	리모트 접속용 포트로 설정합니다. 사용자는 PGM 포트를 통해 연결되어 있는 ETOS 와 마찬가지로(로컬 접속), 원격지에 모뎀 등으로 연결되어 있는 ETOS 도 리모트 접속이 설정되어 있으면 온라인 작업을 할 수 있습니다. (원격지의 ETOS 는 리모트 접속이 설정되어 있고, 리모트 접속이 설정된 COM 포트는 모뎀 등을 통해 로컬 ETOS 와 연결되어 있어야 합니다.) 4.3.4.3. [온라인(O)] -> [온라인 연결 설정]의 리모트 접속 을 참조하시기 바랍니다.
국번	리모트 접속용 국번을 설정합니다. 로컬 ETOS 와의 통신이 모뎀을 이용한 1:1 통신일 경우에는 국번은 특별한 의미가 없습니다. 아무값이나 설정하고, [온라인] -> [온라인 연결 설정]에서 여기에 설정한 국번과 동일한 값을 설정하기만 하면 됩니다. 그러나, 로컬 ETOS 와 1:N 통신을 하는 경우(무선 통신 등)에는, 사용자는 원격지 ETOS 에 국번을 겹치지 않게 설정하고, [온라인] -> [온라인 연결 설정]에서도 통신하고자 하는 ETOS 의 국번을 정확하게 지정해야 합니다.

8.1.3.1. RS422/485 제어방법

RS422/485 포트를 사용하는 경우, 몇가지 포트 제어 옵션을 설정할 수 있습니다.

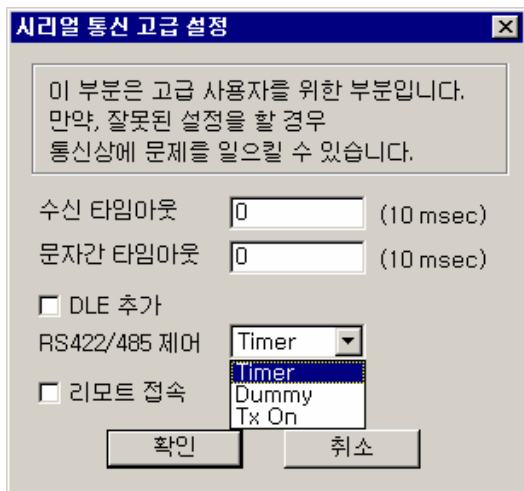


그림 8.1-6 RS422/485 제어방법 선택

▶ Timer

- * RS422/485 공통으로 사용가능하고, Default 로 동작합니다. 보통의 경우, 이 옵션을 사용합니다.
- * Tx Disable 시점을 ETOS 내부 타이머가 제어하게 됩니다.
- * 일반적으로 통신 속도 9600 bps 이하에서는 두 바이트 정도 여유를 두고 Tx Enable 이 끊어 집니다.
- * 통신 속도가 9600bps 이상인 경우에는 보통 2msec 정도 뒤에 Tx Disable 됩니다.

▶ Dummy

- * 위의 Timer 로 설정했을 때 ETOS 가 데이터를 송신한 경우, 약 두 바이트 전송 시간 (통신 속도가 9600bps 인 경우 약 2msec) 뒤에 상대방에서 응답해야 하나, 경우에 따라 상대방이 이시간보다 더 빨리 응답하는 경우가 있습니다. 이 경우, 수신 바이트가 손상되어 정상 수신을 하지 못합니다. 이 경우에 사용합니다.
- * ETOS 에서 프레임을 보낼 때 Dummy 로 한 바이트 더 보내게 하는 옵션입니다.
- * RS485 인 경우에만 사용가능 합니다.
- * RS422 인 경우 선택하면, Timer 방식으로 동작합니다.

▶ TxOn

- * 이 옵션을 선택한 경우, Tx 는 항상 Enable 됩니다.
- * RS422 인 경우에만 적용됩니다.
- * RS485 인 경우 선택하면, Timer 방식으로 동작하게 됩니다.

8.2. MODBUS Master

▶ 포트 파라미터 설정 + 프로세스 정의 필요

다음 그림은 MODBUS Master 포트를 사용하는 예입니다. ETOS 는 MODBUS Slave 장비와 RS232C 로 연결되어 있으며, MODBUS 프로토콜을 사용합니다. ETOS 는 MODBUS Slave 장비에 데이터 읽기/쓰기 요청을 하며, MODBUS Slave 장비는 요청한 동작을 수행하고, 응답합니다. 읽기/쓰기 요청은 MODBUS 전용 아이템을 사용하여, 프로세스에서 정의합니다.

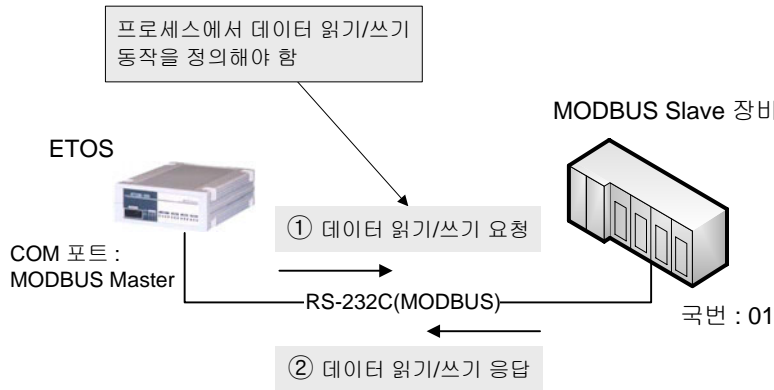


그림 8.2-1 MODBUS Master 포트 사용 예

8.2.1. 프로토콜 상세 설정

▶ {COM 포트 설정} 대화상자의 <프로토콜>을 'MODBUS Master'로 설정

설정된 포트는 MODBUS 프로토콜을 사용하는 장비와 통신합니다. 설정한 포트는 마스터로, 상대 장비는 슬레이브로 동작합니다. 'MODBUS Master'를 선택하면, {MODBUS Master} 대화상자가 나타납니다. 나중에라도, {COM 포트 설정} 대화상자의 [상세설정] 버튼을 누르면 수정이 가능합니다.

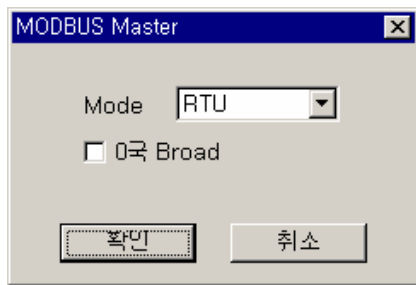


그림 8.2-2 MODBUS Master 상세설정

표 8.2-1 MODBUS Master 상세설정 항목

항 목	내 용
Mode	RTU/ASCII 모드 중 하나를 선택합니다. 상대 장비와 동일한 모드로 설정해야 합니다.
0국 Broad	체크하면 0번 국을 브로드캐스트(Broadcast) 국번으로 취급합니다.

* MODBUS 사양상, 0번 국번은 브로드캐스트 국번입니다. 그러나 ETOS 는 0번 국을 브로드캐스트 국번/개별 국번의 두가지로 취급할 수 있습니다. ETOS 기본값은 0번 국번을 개별 국번으로 취급하는 것입니다.

▶ ASCII 모드

- * Ascii data 를 이용하여 통신
- * LRC 를 이용하여 에러체크

구분	START (:)	ADDRESS	FUNCTION	DATA	LRC	END(CR LF)
크기	1 byte	2 byte	2 byte	n byte	2 byte	2 byte

그림 8.2-3 MODBUS ASCII 모드 프레임 구조

▶ RTU 모드

- * Hexa data 를 이용하여 통신
- * CRC-16 을 이용하여 에러체크

구분	START	ADDRESS	FUNCTION	DATA	CRC	END
크기	T4 idle ^(*137)	1 byte	1 byte	n byte	2 byte	T4 idle ^(*1)

그림 8.2-4 MODBUS RTU 모드 프레임 구조

8.2.2. MODBUS Master 아이템

마스터 포트에 설정하였으므로, 파라미터 설정 외에 프로세스를 정의해야 합니다. MODBUS 전용의 마스터 아이템은 다음과 같습니다. 각 아이템에 대한 설명 및 사용 예는 [7.6.37. IF_MODBUS_READ](#) 및 [7.6.38. IF_MODBUS_WRITE](#) 를 참조하시기 바랍니다.

종 류	내 용
IF_MODBUS_READ	상대국의 특정 영역을 읽어 ETOS 메모리에 저장
IF_MODBUS_WRITE	ETOS 메모리를 상대국 영역에 쓰기

8.2.3. 읽기/쓰기 최대 길이

IF_MODBUS_READ, IF_MODBUS_WRITE 아이템을 사용해서 한 번에 읽거나 쓸 수 있는 데이터의 최대 길이는 다음과 같습니다. 비트의 경우 읽기는 2000 비트까지, 워드는 125 워드, 쓰기의 경우 비트는 1600 비트까지 워드는 100 워드까지 한번에 요청 가능합니다. 그러나, 슬레이브로 동작할 상대기기가 지원하는 범위 안에서 요청해야 합니다.

아이템	데이터 형	Address	최대 길이
IF_MODBUS_READ	Coils(Bits)	0XXXX, 1XXXX	2000 Coils
	Registers(Words)	3XXXX, 4XXXX	125 Registers
IF_MODBUS_WRITE	Coils(Bits)	0XXXX	1600 Coils
	Registers(Words)	4XXXX	100 Registers

(*137) 4 문자 전송 시간을 의미합니다. 통신 속도가 9600bps 인 경우 약 4ms 가 됩니다. (통신 속도에 따라 달라짐)

8.2.4. 프로그램 예제

다음 예는 1 초 주기로 상대기기(국번 1)에 1 워드를 쓰고, 다시 읽어서 그 값이 제대로 쓰여졌는지 확인하는 프로세스입니다. 예에서는 COM2 포트를 MODBUS Master 포트로 설정하였으며, IF_MODBUS_READ 및 IF_MODBUS_WRITE 아이템을 사용하여 데이터를 읽고, 쓰고 있습니다.

NO	아이템	설명
00	IF_TIMER	100
01	REMARK	-----
02	REMARK	<<< 워드 읽기/쓰기 (4XXXX) >>>
03	REMARK	-----
04	REMARK	<<< 1 워드 >>>
05	REMARK	< 수신 영역 Clear >
06	MEM_SET	T:M25000, Size:200, Type:16진수, Data:00
07	IF_MODBUS_WRITE	Media:Serial, COM2, DestSt:1, DestAddr:40000, Length:1, SelfAddr:M5000, ...
08	END_IF	
09	IF_MODBUS_READ	Media:Serial, COM2, DestSt:1, DestAddr:40000, Length:1, SelfAddr:M25000,...
10	END_IF	
11	REMARK	< 송,수신 데이터 비교 >
12	IF_MEM_CMP	M5000, M25000, Size:2
13	ARITHMETIC	Type:U1, 1, +, M20018, M20018
14	ELSE	
15	ARITHMETIC	Type:U1, 1, +, M20034, M20034
16	END_IF	
17	ENDIF	

그림 8.2-5 프로그램 예

8.3. MODBUS Slave

▶ 포트 파라미터 설정 필요 (프로세스 정의 필요 없음)

다음 그림은 MODBUS Slave 포트를 사용하는 예입니다. ETOS 는 PC 와 RS232C 로 연결되어 있으며, MODBUS 프로토콜을 사용합니다. PC 는 ETOS 에 데이터 읽기/쓰기 요청을 하며, ETOS 는 PC 의 요청에 수동적으로 응답합니다. 읽기/쓰기 요청에 대한 응답은 프로세스 없이, 파라미터 설정만으로 가능합니다.

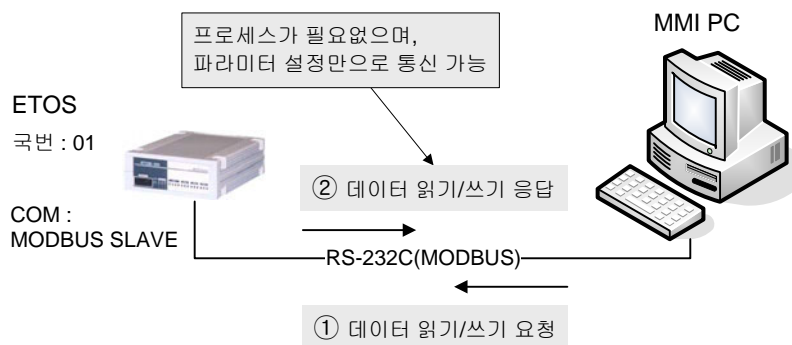


그림 8.3-1 MODBUS Slave 포트 사용 예

8.3.1. 프로토콜 상세 설정

▶ {COM 포트 설정} 대화상자의 <프로토콜>을 'MODBUS Slave'로 설정

설정된 포트는 MODBUS 프로토콜을 사용하는 장비와 통신합니다. 설정한 포트는 슬레이브로, 상대 장비는 마스터로 동작하게 됩니다. 다음은 {MODBUS Slave} 설정 대화상자입니다.

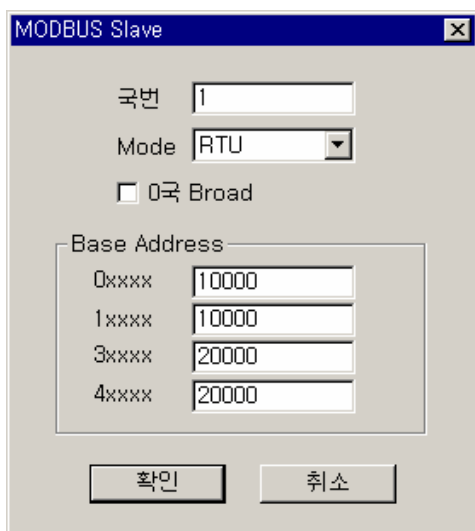


그림 8.3-2 MODBUS Slave 상세설정

표 8.3-1 MODBUS Slave 상세설정 항목

항 목	내 용	
국번	ETOS의 국번을 설정 (0 ~ 247)	
Mode	RTU/ASCII 모드 중에서 선택합니다. 통신하고자 하는 상대장비의 모드와 동일한 모드를 선택합니다.	
0국 Broad ^(*138)	체크하면 0국번을 브로드캐스트 국번으로 인식합니다.	
Base Address ^(*139)	0xxxx	비트 출력(Coil status) 영역의 위치를 설정합니다.
	1xxxx	비트 입력(Input status) 영역의 위치를 설정합니다.
	3xxxx	워드 입력(Input Registers) 영역의 위치를 설정합니다.
	4xxxx	워드 출력(Holding Registers) 영역의 위치를 설정합니다.

^(*138) MODBUS 사양상, 0번 국번은 브로드캐스트(Broadcast)국번입니다. 그러나 ETOS는 0번 국을 브로드캐스트 국번/개별 국번의 두가지로 취급할 수 있습니다. 기본값은 0번 국번을 개별 국번으로 취급하는 것입니다. 0번 국을 브로드캐스트 국번으로 사용하려면 <0국 Broad>옵션을 체크하시기 바랍니다.

^(*139) ETOS의 비트, 워드 입,출력 영역을 설정합니다. 예를 들어, 0xxxx에 10000이라고 설정하면, ETOS의 비트 출력 영역은 10000(바이트) 번지부터가 되는 것입니다.

8.3.2. ETOS Address Mapping

ETOS는 %M(내부 메모리) 영역만 존재합니다. 따라서 사용자는 MODBUS 입,출력 영역을 모두 ETOS의 %M 영역에 대응시켜야 합니다. ETOS가 지원하는 %M 영역의 크기는 다음 표를 참조하시기 바랍니다.

제품	%M 영역(바이트 주소)
ETOS-50	0 ~ 65535 (64Kbytes)
ETOS-100/150	0 ~ 262143 (256Kbytes)
ETOS-200/1000	0 ~ 524287 (512Kbytes)

Base Address 입력값은 유효한 %M 영역내에 있어야 합니다. MODBUS의 주소는 1 ~ 9999 (십진수) 이므로 비트 입,출력 영역의 크기는 $9999/8 = 1249.875$ 바이트가 됩니다. 또, 워드 입,출력 영역의 크기는 $9999*2 = 19998$ 바이트 입니다. 따라서 Base Address 를 지정할 때, %M 영역의 크기에서 해당 영역의 바이트 수를 뺀 주소가 입력할 수 있는 가장 큰 Base Address 가 됩니다.

예를 들어, ETOS-100A 를 기준으로 MODBUS의 비트 출력(0XXXX) 영역의 Base Address 는 0에서 $262143 - 1250 = 260893$ 범위 안에서 지정할 수 있습니다.

ETOS의 메모리는 바이트 형만을 지원합니다. 따라서, 비트 입,출력 영역은 바이트 주소와 비트 오프셋으로 표현됩니다. 만약, 사용자가 비트 출력(0XXXX) 영역의 Base Address 로 0을 설정했다면, MODBUS 비트 영역 00001은 0번째 바이트, 0번 비트(M0.0)에 대응되고, 00002은 0번째 바이트, 1번 비트(M0.1)에 대응하게 됩니다.

8.3.3. 지원 Function Code

ETOS 의 MODBUS RTU/ASCII 드라이버가 지원하는 Function code 는 다음과 같습니다. 각 Function code 에 대한 자세한 설명은 MODBUS 프로토콜 매뉴얼을 참조하시기 바랍니다.

표 8.3-2 ETOS 지원 MODBUS Function Code

Code	Name	Address
01	Read Coil Status	0XXXX(비트출력)
02	Read Input Status	1XXXX(비트입력)
03	Read Holding Registers	4XXXX(워드출력)
04	Read Input Registers	3XXXX(워드입력)
05	Force Single Coil	0XXXX(비트출력)
06	Preset Single Register	4XXXX(워드출력)
15	Force Multiple Coils	0XXXX(비트출력)
16	Preset Multiple Registers	4XXXX(워드출력)

8.3.4. 읽기/쓰기 최대 길이

ETOS 의 MODBUS RTU/ASCII 드라이버가 지원하는 응답 데이터의 최대 개수입니다. 상대 마스터 기기는 다음표의 범위안에서만 요청을 해야 합니다. 예를 들어, 비트 읽기 요청은 최대 2000 비트까지, 비트 쓰기 요청은 최대 1600 비트까지 가능합니다.

표 8.3-3 영역별 읽기/쓰기 가능 최대 길이

Code	Name	Address	Response
01	Read Coil Status	0XXXX	2000 Coils
02	Read Input Status	1XXXX	2000 Coils
03	Read Holding Registers	4XXXX	125 Registers
04	Read Input Registers	3XXXX	125 Registers
05	Force Single Coil	0XXXX	1 Coil
06	Preset Single Register	4XXXX	1 Register
15	Force Multiple Coils	0XXXX	1600 Coils
16	Preset Multiple Registers	4XXXX	100 Registers

8.4. GLOFA Master

▶ 포트 파라미터 설정 + 프로세스 정의 필요

다음 그림은 GLOFA Master 포트를 사용하는 예입니다. ETOS 는 PLC 와 RS232C 로 연결되어 있으며, GLOFA 프로토콜을 사용합니다. ETOS 는 PLC 에 데이터 읽기/쓰기 요청을 하며, PLC 는 요청한 동작을 수행하고, 응답합니다. 읽기/쓰기 요청은 GLOFA 전용 아이템을 사용하여, 프로세스에서 정의합니다.

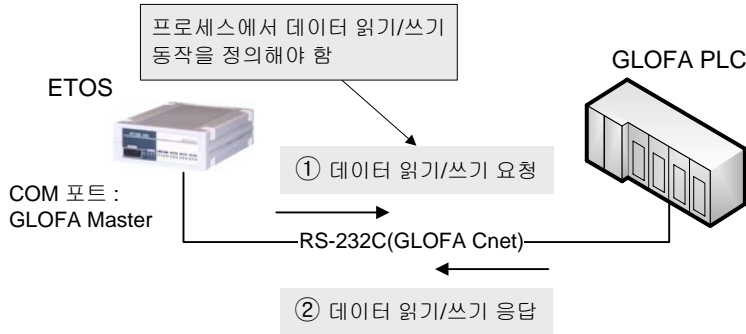


그림 8.4-1 GLOFA Master 포트 사용 예

8.4.1. 프로토콜 상세 설정

▶ {COM 포트 설정} 대화상자의 <프로토콜>을 'GLOFA Master'로 설정

상세 설정 내용이 없습니다.

8.4.2. GLOFA Cnet 마스터 아이템

마스터 포트로 설정하였으므로, 파라미터 설정 외에 프로세스를 정의해야 합니다. GLOFA 전용의 마스터 아이템은 다음과 같습니다. 각 아이템에 대한 설명 및 사용 예는 [7.6.31. IF_GLOFA_READ](#) 및 [7.6.32. IF_GLOFA_WRITE](#) 를 참조하시기 바랍니다.

아이템	내 용
IF_GLOFA_READ	상대국의 특정 영역을 읽어 ETOS 메모리에 저장
IF_GLOFA_WRITE	ETOS 메모리를 상대국 영역에 쓰기

8.4.3. 읽기/쓰기 최대 길이

IF_GLOFA_READ, IF_GLOFA_WRITE 아이템을 사용해서 한 번에 읽거나 쓸 수 있는 데이터의 최대 길이는 다음과 같습니다. 비트 읽기를 제외하고, 최대 120 바이트의 데이터를 한번에 읽거나 쓸 수 있습니다. 비트 읽기 및 쓰기는 항상 1 비트 단위로만 가능합니다. 다음 표를 참조하시기 바랍니다.

아이템	데이터 형	최대 길이
IF_GLOFA_READ IF_GLOFA_WRITE	%MX, %IX, %QX	1비트
	%MB, %IB, %QB	120바이트
	%MW, %IW, %QW	60워드
	%MD, %ID, %QD	30 더블워드
	%ML, %IL, %QL	15 롱워드

8.4.4. 프로그램 예제

다음의 예는 1 초 주기로 상대기기의 메모리에 쓰고, 다시 읽어서 그 값이 제대로 쓰여졌는지 확인하는 프로세스입니다. 예에서는 COM1 포트를 GLOFA Master 포트로 설정하였으며, IF_GLOFA_READ 및 IF_GLOFA_WRITE 아이템을 사용하여 데이터를 읽고, 쓰고 있습니다.

NO	아이템	설명
00	IF_TIMER	100
01	REMARK	<< 120BYTE >>
02	MEM_SET	T:M33000, Size:120, Type:16진수, Data:FF
03	REMARK	120 바이트 쓰기
04	IF_GLOFA_WRITE	Media:Serial, Card0, COM1, DestSt:1, DestAddr:MB31000, Length:120, Self,
05	END_IF	
06	REMARK	120 바이트 읽기
07	IF_GLOFA_READ	Media:Serial, Card0, COM1, DestSt:1, DestAddr:MB31000, Length:120, Self,
08	END_IF	
09	END_IF	

그림 8.4-2 프로그램 예

8.5. GLOFA Slave

▶ 포트 파라미터 설정 필요 (프로세스 정의 필요 없음)

다음 그림은 GLOFA Slave 포트를 사용하는 예입니다. ETOS 는 PC 와 RS232C 로 연결되어 있으며, GLOFA 프로토콜을 사용합니다. PC 는 ETOS 에 데이터 읽기/쓰기 요청을 하며, ETOS 는 PC 의 요청에 수동적으로 응답합니다. 읽기/쓰기 요청에 대한 응답은 프로세스 없이, 파라미터 설정만으로 가능합니다.

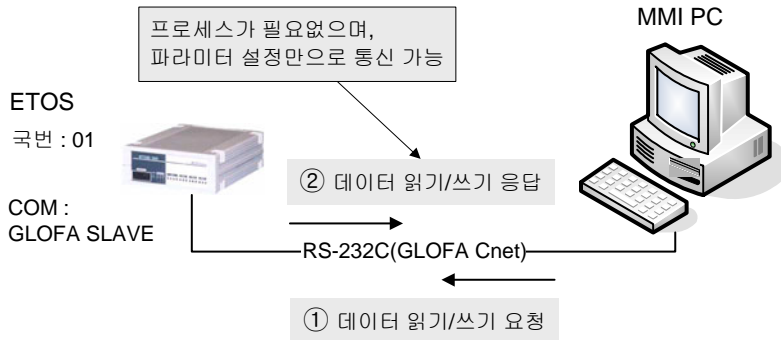


그림 8.5-1 GLOFA Slave 포트 사용 예

8.5.1. 프로토콜 상세 설정

▶ {COM 포트 설정} 대화상자의 <프로토콜>을 'GLOFA Slave'로 설정

다음과 같이 국번을 설정하고 [확인] 버튼을 누릅니다. 추후에, {COM 포트 설정} 대화상자의 [상세설정] 버튼을 눌러 변경할 수 있습니다.

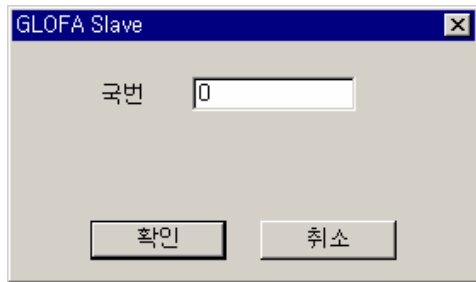


그림 8.5-2 GLOFA Slave 상세설정

표 8.5-1 GLOFA Slave 상세설정 항목

항목	설정값
국번	ETOS 국번을 설정합니다.

디바이스는 %M 만 존재하며 ETOS 사용자 메모리 0 번지부터 %MB0 에 매핑되어 있습니다. 따라서, 시리얼 포트별로 각각 다른 국번의 슬레이브를 설정하더라도, 내부메모리 영역은 모든 슬레이브 포트가 공유하게 됩니다.

8.5.2. 지원 명령

다음은 ETOS 가 지원하는 GLOFA Cnet 명령의 목록입니다. 직접변수 읽기와 직접변수 쓰기명령만을 지원합니다. GLOFA Cnet 명령 등에 대한 보다 자세한 사항은 GLOFA Cnet 프로토콜 매뉴얼을 참조하시기 바랍니다.

표 8.5-2 ETOS 지원 GLOFA Cnet 명령

명령			내용
직접 변수 읽기	연속 읽기	r(R)SB	Byte, Word, Dword, Lword형의 직접 변수를 블록 단위로 읽어옵니다. (비트 연속 읽기는 허용되지 않습니다) ^(*140)
	개별 읽기	r(R)SS	Bit, Byte Word, Dword, Lword형의 직접 변수를 읽어옵니다.
직접 변수 쓰기	연속 쓰기	w(W)SB	Byte, Word, Dword, Lword형의 직접 변수에 블록 단위로 씁니다. (비트 연속 쓰기는 허용되지 않습니다) ^(*)
	개별 쓰기	w(W)SS	Bit, Byte, Word, Dword, Lword형의 직접 변수에 데이터를 씁니다.

^(*140) 최대 120 바이트, GLOFA Cnet 사양

8.6. 사용자 정의 프로토콜 (User Defined)

▶ 포트 파라미터 설정 + 프로세스 정의 + 프레임 정의 필요

ETOS 가 제공하지 않는 프로토콜을 사용하는 기기와 통신하려고 하는 경우, 사용자는 해당 프로토콜의 프레임을 정의하여 상대기기와 통신 할 수 있습니다. 사용자 정의 프로토콜의 경우 마스터/슬레이브 포트의 구분이 없습니다. 다만, 사용하는 아이템에 따라 마스터 역할을 할 수도 있고, 슬레이브 역할을 할 수도 있습니다.^(*141) 상대기기로부터 데이터를 읽거나 쓰기 위해서는 포트 파라미터 설정뿐만 아니라, 프레임 및 프로세스 정의도 필요합니다.

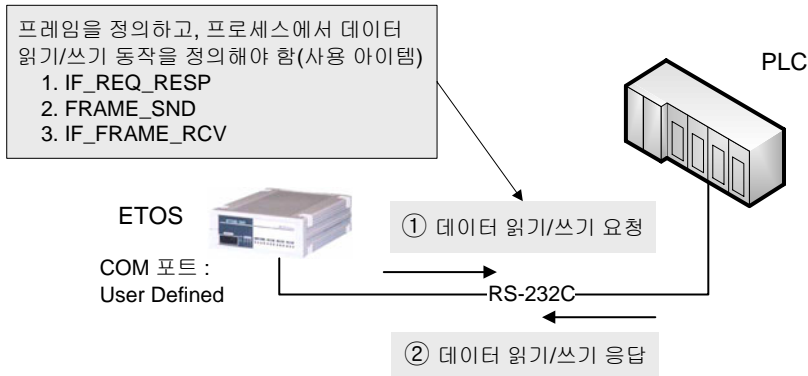


그림 8.6-1 사용자정의 프로토콜 사용 예

8.6.1. 포트 설정

▶ {COM 포트 설정} 대화상자의 <프로토콜>을 'User Defined' 으로 설정

상세 설정 내용이 없습니다.

8.6.2. 프레임 정의

{프레임 등록} 창에서 '시리얼 프레임' 또는 'Card #N'을 선택하고, 마우스 오른쪽 버튼을 눌러 [그룹 추가]를 선택합니다. 적당한 그룹명을 입력하고, 위에서 설정한 포트를 선택합니다.^(*142)

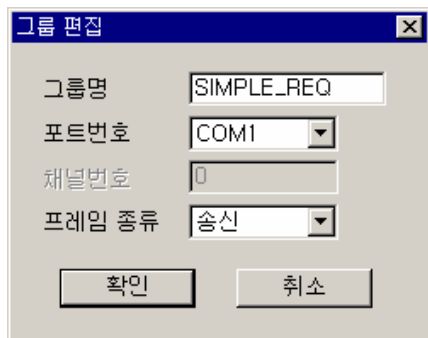


그림 8.6-2 그룹 등록 예

(*141) 8.6.3. 사용자 정의 프로토콜 아이템 참조

(*142) 프레임 등록 방법에 대한 자세한 사항은 6.4. 프레임 등록 예제를 참조하시기 바랍니다.

이제 추가된 그룹에 마우스를 클릭하고, 다시 오른쪽 버튼을 눌러 [프레임 추가]를 선택합니다.



그림 8.6-3 프레임 등록

<종류>에는 HEAD, TAIL, BODY 가 있습니다. 사용하려는 프로토콜 자료를 참조하여 각각을 설정합니다. BODY 는 항상 설정해야 하고, 경우에 따라 HEAD, TAIL 은 없을 수도 있습니다.

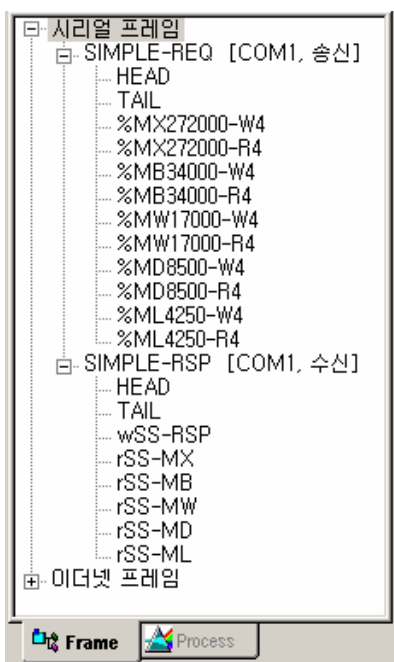


그림 8.6-4 시리얼 프레임 등록 예

8.6.3. 사용자 정의 프로토콜 아이템

사용자 정의 프로토콜은 프레임을 정의한 후, 동작을 정의하는 프로세스도 프로그램해야 합니다. 동작을 정의하는 아이템은 다음과 같습니다.

아이템	설 명
IF_REQ_RESP	상대방에 특정 프레임을 송신하고, 응답을 수신. 송,수신 프레임을 모두 참조
FRAME_SND	상대방에 특정 프레임을 송신, 송신 프레임만 참조
IF_FRAME_RCV	상대방으로부터 특정 프레임을 수신, 수신 프레임만 참조

프로세스에서 사용하는 아이템에 따라 해당 포트가 마스터인지 슬레이브인지가 결정됩니다. IF_REQ_RESP 아이템을 사용하는 경우, 마스터 포트가 되며, IF_FRAME_RCV 와 FRAME_SND 를 사용하면, 슬레이브 역할을 할 수 있습니다.

▶ 마스터 포트 구성 예

프로세스가 IF_TIMER 아이템 등으로 시작해서 IF_REQ_RESP 아이템을 사용하는 경우, 해당 포트는 마스터 역할을 합니다.

```
00 IF_TIMER
01     IF_REQ_RESP
02     END_IF
03 END_IF
```

▶ 슬레이브 포트 구성 예

프로세스가 IF_FRAME_RCV 아이템으로 시작해서 응답 프레임을 FRAME_SND 아이템으로 송신하는 경우, 해당 포트는 슬레이브 역할을 합니다.

```
00 IF_FRAME_RCV
01     FRAME_SND
02 END_IF
```

각 아이템의 사용법은 7.6. 아이템 설명을 참조하시기 바랍니다.

8.6.4. 프로그램 예제

다음 예는 1 초 주기로 상대기기의 특정 영역을 읽어오는 프로세스입니다. (상대 장비에 대해서 마스터로 동작하고 있습니다.)

NO	아이템	설명
00	IF_TIMER	100
01	REMARK	-----
02	REMARK	<< BYTE >>
03	REMARK	< 수신 영역 CLEAR >
04	MEM_SET	T:M38000, Size:120, Type:16진수, Data:FF
05	IF_REQ_RESP	Req:SIMPLE-REQ.%MB36000-W4, Resp:SIMPLE-RSP.wSS-RSP
06	ARITHMETIC	Type:U1, 1, +, M30048, M30048
07	ELSE	
08	ARITHMETIC	Type:U1, 1, +, M30064, M30064
09	END_IF	
10	IF_REQ_RESP	Req:SIMPLE-REQ.%MB36000-R4, Resp:SIMPLE-RSP.rSS-MB
11	ARITHMETIC	Type:U1, 1, +, M30049, M30049
12	ELSE	
13	ARITHMETIC	Type:U1, 1, +, M30065, M30065
14	END_IF	
15	REMARK	-----
16	END_IF	

그림 8.6-5 프로그램 예

이더넷 통신

제 9 장

ETOS 의 이더넷 통신 기능에 대해서 설명합니다.

	내용	위치
9.1	개요	330
9.2	이더넷 드라이버	332
9.3	이더넷 채널	340
9.4	이더넷 고속링크	353

9. 이더넷 통신

9.1. 개요

ETOS의 이더넷 통신 기능에 대해서 설명합니다. ETOS는 몇 가지의 내장 프로토콜과 사용자 정의 프로토콜을 사용하여 이더넷 장비와 데이터를 주고 받을 수 있습니다. 내장 프로토콜을 사용하는 경우, 사용자는 파라미터 또는 파라미터와 프로세스 정의만으로 상대 기기와 통신이 가능합니다. 물론, 내장 프로토콜이 아닌 경우에도 상대 장비가 사용하는 프로토콜의 프레임을 정의하면 데이터 통신이 가능합니다.

이더넷을 사용하기 위한 파일의 종류를 선택합니다. [파일] -> [새파일] 메뉴를 선택하고, {새파일} 대화상자에서 <드라이버>를 'Ethernet'으로 선택합니다.

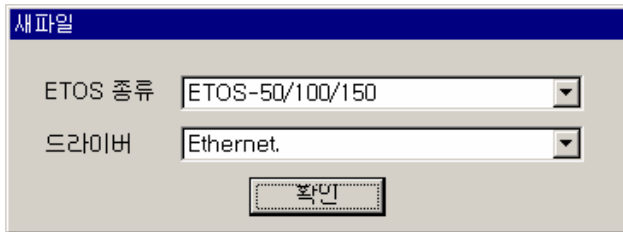


그림 9.1-1 새파일 설정

[확인]을 누르면 {파라미터 설정} 대화상자가 나타납니다.

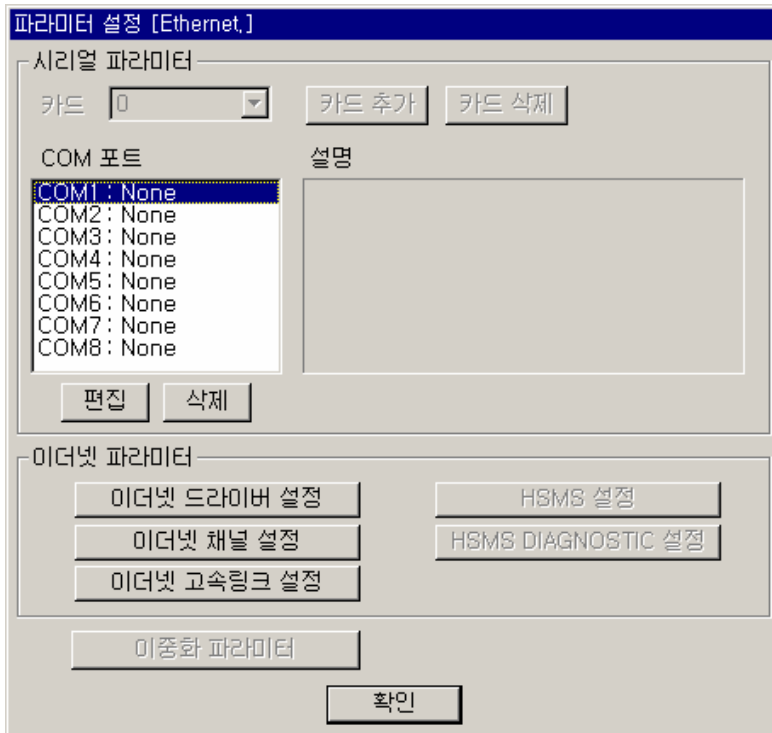


그림 9.1-2 파라미터 설정 화면

드라이버 종류가 Ethernet인 경우, 이더넷 파라미터 중에서 [이더넷 드라이버 설정], [이더넷 채널 설정], [이더넷 고속링크 설정] 버튼만이 활성화 됩니다. [HSMS 설정] 및 [HSMS DIAGNOSTIC 설정]은 드라이버를 SECS로 설정한 경우만 활성화 됩니다. HSMS 설정은 [10. SECS 통신](#)을 참조하시기 바랍니다.

9.1.1. 이더넷 드라이버 설정

이더넷 드라이버는 ETOS 가 내장된 프로토콜^(*143)을 가지고 슬레이브(또는 서버^(*144))로 동작하는 것을 말합니다. 내장된 프로토콜은 MODBUS, GLOFA Enet 등이 있습니다. 슬레이브로 동작하므로 상대국이 요구하는 정보에 대해 수동적으로 응답하는 형태로 동작합니다. 주로 MMI(또는 HMI)에서 ETOS 의 메모리를 참조하려고 할 경우 사용합니다.

9.1.2. 이더넷 채널 설정

이더넷 채널은 ETOS 가 내장한 프로토콜^(*145)을 가지고 마스터(또는 클라이언트^(*2))로 동작해야 하는 경우, 또는 내장 프로토콜 이외의 프로토콜(사용자 정의 프로토콜)로 통신을 해야 하는 경우에 사용합니다.

9.1.3. 이더넷 고속링크 설정

고속링크는 ETOS 와 GLOFA PLC 통신모듈, 또는 ETOS 와 ETOS 간의 통신 방법으로 특정 시간마다 주기적으로 상대국의 데이터나 정보를 교환할 때 사용합니다. 자신 또는 상대국의 변화되는 데이터를 서로 주기적으로 참조하여 운전하는 시스템에 효과적으로 사용할 수 있고, 파라미터 설정만으로 통신을 수행할 수 있습니다. 이더넷 파라미터의 [이더넷 고속링크 설정]에서 송수신하려는 상대국 영역과 자기 영역을 지정하고 데이터 크기, 국번을 지정하여 통신을 수행합니다. 송, 수신 데이터의 최대 크기는 200 워드입니다.

간단한 파라미터 설정만으로 상대국과 통신이 가능하고, 내부 데이터 처리 또한 고속이므로 많은 데이터를 한꺼번에 주기적으로 처리하는데 유용하게 사용할 수 있습니다.

^(*143) OS 버전 1.20 을 기준으로 MODBUS, GLOFA Enet 을 지원합니다.

^(*144) Server/Client 용어는 TCP 연결 설정시의 동작에서도 나오므로 주로 Slave/Master 로 칭하도록 합니다.

^(*145) OS 버전 1.20 을 기준으로 MODBUS, GLOFA Enet 을 지원하며, OS 버전 2.4 부터 MELSEC 을 추가 지원합니다.

9.2. 이더넷 드라이버

이더넷 드라이버는 내장 프로토콜을 가지고 슬레이브로 동작하는 채널을 말합니다. 사용자가 별도의 통신 프로그램을 작성하지 않고, 포트 파라미터 설정만으로 상대 기기와 통신이 가능합니다.

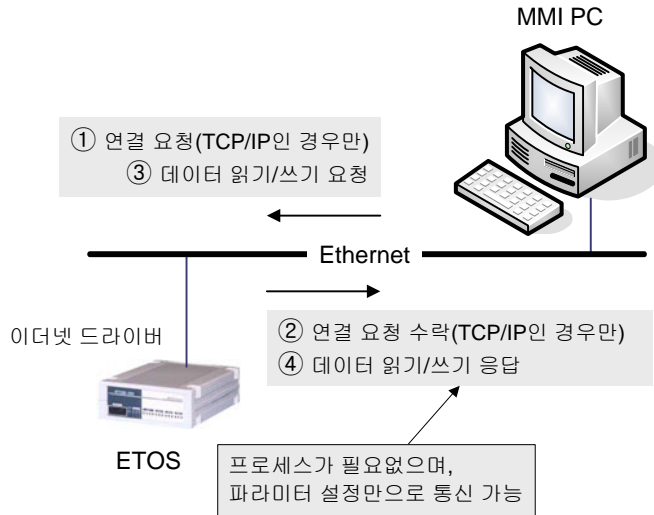


그림 9.2-1 이더넷 드라이버 사용 예

▶ 이더넷 드라이버 종류

지원하는 드라이버(프로토콜) 종류는 다음과 같습니다. (OS 버전 1.20 기준)

드라이버	설 명
MODBUS/TCP	Modicon사의 오픈 프로토콜
GLOFA Enet	LG산전의 GLOFA PLC 이더넷 전용 프로토콜
ETOS	AC&T의 ETOS 이더넷 전용 프로토콜

▶ 이더넷 드라이버 설정 방법

이더넷 파라미터의 [이더넷 드라이버 설정]을 누르면, 다음 그림과 같은 {이더넷 드라이버 설정} 대화상자가 나타납니다. 드라이버를 추가하지 않았기 때문에 설정 창이 비어있습니다.

이제 [추가] 버튼을 누르고, 사용하고자 하는 프로토콜에 따라 드라이버 포트를 설정합니다.

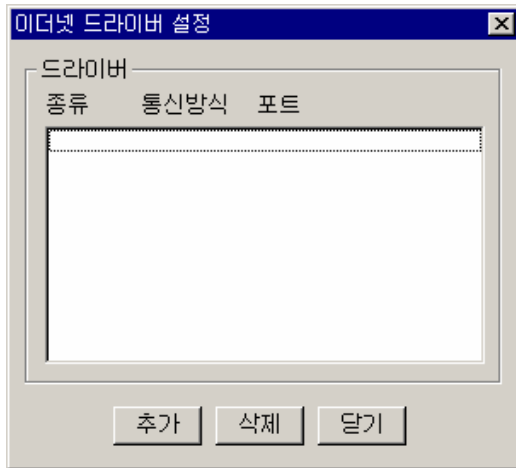


그림 9.2-2 이더넷 드라이버 설정 대화상자

ETOS 는 여러개의 드라이버를 동시에 지원할 수 있습니다. 지원할 수 있는 드라이버의 수는 모델에 따라 상이합니다. 다음 표를 참조하시기 바랍니다.

제품	지원가능한 드라이버 수(연결 수)
ETOS-50A	4
ETOS-100A/150	16
ETOS-200	24
ETOS-1000	32

단, 이때 설정하는 자국 포트 번호는 서로 달라야 합니다.

다음 그림은 여러 개의 이더넷 드라이버 포트를 설정한 화면입니다. 각각의 드라이버 포트는 서로 다른 프로토콜을 지원하고 있습니다.

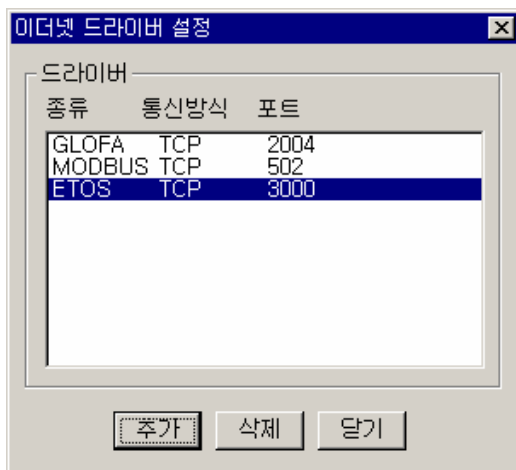


그림 9.2-3 이더넷 드라이버 설정 예

9.2.1. MODBUS/TCP

상대 기기가 MODBUS 를 사용하는 마스터일 경우에 사용합니다. ETOS 는 MODBUS 슬레이브로 설정되어, 상대 기기의 요청에 응답하는 형식으로 동작합니다. 다음과 같이 설정합니다.

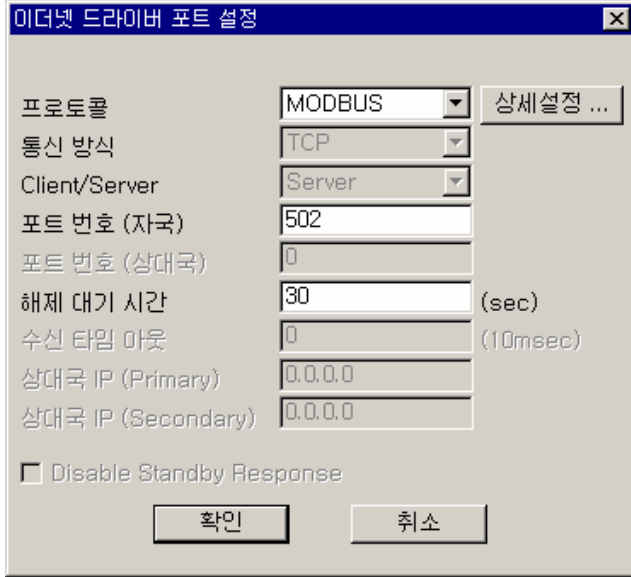


그림 9.2-4 MODBUS 드라이버 포트 설정

표 9.2-1 MODBUS 드라이버 포트 설정 항목

항 목	내 용
포트 형식	MODBUS 를 선택
포트 번호(자국)	서버 포트의 번호, 보통 502번 포트를 사용
해제 대기 시간	초 단위로 설정(일반적으로, 수 초 ~ 수십 초)

포트 번호는 MODBUS 드라이버가 사용하는 소켓의 포트 번호를 말합니다. 통상 MODBUS 는 502 번을 사용합니다. 해제 대기 시간은 설정한 시간 동안 통신이 없으면 해당 연결을 종료하게 됩니다. 초 단위로 설정합니다.

9.2.1.1. 프로토콜 상세 설정

포트 형식을 MODBUS 로 선택하면, 다음과 같은 {MODBUS Slave} 설정 대화상자가 나타납니다. 추후에, [상세설정] 버튼을 눌러 수정할 수 있습니다.

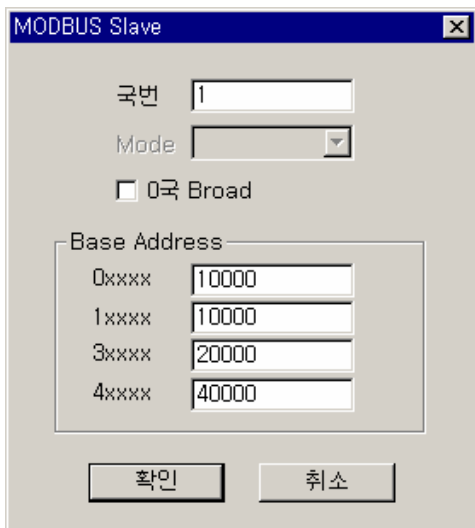


그림 9.2-5 MODBUS Slave 상세설정 예

표 9.2-2 MODBUS Slave 상세설정 항목

항 목		내 용
국번		ETOS의 국번 (0~247)
0국 Broad		0번 국을 브로드캐스트 국번으로 취급
Base Address	0XXXX	디지털(비트) 출력 영역에 해당하는 ETOS 바이트 주소
	1XXXX	디지털(비트) 입력 영역에 해당하는 ETOS 바이트 주소
	3XXXX	아날로그(워드) 입력 영역에 해당하는 ETOS 바이트 주소
	4XXXX	아날로그(워드) 출력 영역에 해당하는 ETOS 바이트 주소

0번 국은 MODBUS 프로토콜에서 브로드캐스트 국번으로 사용하나, ETOS에서는 0번 국을 개별 국번으로 사용할 수 있습니다. 기본값은 0번 국을 개별 국번으로 사용하는 것입니다. 0번 국을 브로드캐스트 국번으로 사용하려면 [0국 Broad] 옵션을 체크하시기 바랍니다.

Base Address는 ETOS의 비트, 워드 입, 출력 영역을 설정합니다. 예를 들어, 0xxxx에 10000이라고 설정하면, ETOS의 비트 출력 영역은 10000(바이트) 번지가 되는 것입니다.

9.2.1.2. ETOS Address Mapping

ETOS 는 %M(내부 메모리) 영역만 존재합니다. 따라서 사용자는 MODBUS 입,출력 영역을 모두 ETOS 의 %M 영역에 대응시켜야 합니다. ETOS 가 지원하는 %M 영역의 크기는 다음 표를 참조하시기 바랍니다.

표 9.2-3 제품별 사용자 메모리 영역

제품	%M 영역(바이트 주소)
ETOS-50A	0 ~ 65535 (64Kbytes)
ETOS-100A/150	0 ~ 262143 (256Kbytes)
ETOS-200/1000	0 ~ 524287 (512Kbytes)

Base Address 입력값은 유효한 %M 영역내에 있어야 합니다. MODBUS 의 주소는 1 ~ 9999 (십진수) 이므로 비트 입,출력 영역의 크기는 $9999/8 = 1249.875$ 바이트가 됩니다. 또, 워드 입,출력 영역의 크기는 $9999*2 = 19998$ 바이트 입니다. 따라서 Base Address 를 지정할 때, %M 영역의 크기에서 해당 영역의 바이트 수를 뺀 주소가 입력할 수 있는 가장 큰 Base Address 가 됩니다.

예를 들어, ETOS-100A 를 기준으로 MODBUS 의 비트 출력(0XXXX) 영역의 Base Address 는 0 에서 $262143 - 1250 = 260893$ 범위 안에서 지정할 수 있습니다.

ETOS 의 메모리는 바이트 형만을 지원합니다. 따라서, 비트 입,출력 영역은 바이트 주소와 비트 오프셋으로 표현됩니다. 만약, 사용자가 비트 출력(0XXXX) 영역의 Base Address 로 0 을 설정했다면, MODBUS 비트 영역 00001 은 0 번째 바이트, 0 번 비트(M0.0)에 대응되고, 00002 은 0 번째 바이트, 1 번 비트(M0.1)에 대응하게 됩니다.

9.2.1.3. 지원 Function Code

ETOS 의 MODBUS/TCP 드라이버가 지원하는 Function code 는 다음과 같습니다.

Code	Name	Address
01	Read Coil Status	0XXXX(비트출력)
02	Read Input Status	1XXXX(비트입력)
03	Read Holding Registers	4XXXX(워드출력)
04	Read Input Registers	3XXXX(워드입력)
05	Force Single Coil	0XXXX(비트출력)
06	Preset Single Register	4XXXX(워드출력)
15	Force Multiple Coils	0XXXX(비트출력)
16	Preset Multiple Registers	4XXXX(워드출력)

9.2.1.4. 읽기/쓰기 최대 길이

ETOS 의 MODBUS/TCP 드라이버가 지원하는 응답 데이터의 최대 개수입니다. 상대 마스터 기기는 다음표의 범위안에서만 요청을 해야 합니다. 예를 들어, 비트 읽기 요청은 최대 2000 비트까지이며, 비트 쓰기 요청은 최대 1600 비트까지 가능합니다.

Code	Name	Address	Response
01	Read Coil Status	0XXXX	2000 Coils
02	Read Input Status	1XXXX	2000 Coils
03	Read Holding Registers	4XXXX	125 Registers
04	Read Input Registers	3XXXX	125 Registers
05	Force Single Coil	0XXXX	1 Coil
06	Preset Single Register	4XXXX	1 Register
15	Force Multiple Coils	0XXXX	1600 Coils
16	Preset Multiple Registers	4XXXX	100 Registers

MODBUS/TCP 에 대한 보다 자세한 사항은 MODBUS 프로토콜 매뉴얼을 참조하시기 바랍니다.

9.2.2. GLOFA Enet

LG 산전 GLOFA PLC 의 이더넷 프로토콜을 사용하여 통신합니다. 상대기기가 GLOFA Enet 프로토콜을 사용하는 마스터일 경우 사용합니다. ETOS 는 GLOFA Enet 슬레이브로 설정되어, 상대기기의 요청에 응답하는 형식으로 동작합니다. 다음과 같이 설정합니다.

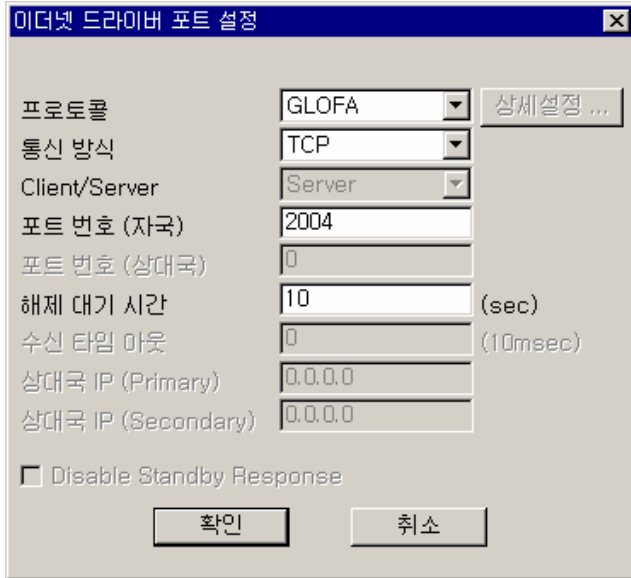


그림 9.2-6 GLOFA Enet 드라이버 포트 설정

표 9.2-4 GLOFA Enet 드라이버 포트 설정 항목

항 목	내 용
포트 형식	GLOFA 선택.
통신 방식	TCP/UDP 중 선택.
포트 번호(자국)	자국의 서버 포트 번호를 지정. 2004(TCP), 2005(UDP)
해제 대기 시간	이 시간 동안 통신을 하지 않으면 연결을 종료. (초 단위로 설정)

9.2.2.1. 프로토콜 상세 설정

상세 설정 항목이 없습니다.

9.2.2.2. 지원 명령

ETOS 의 GLOFA Enet 드라이버가 지원하는 명령은 다음과 같습니다.

명 령	설 명
블록 읽기	바이트 형 직접변수만 지원, 최대 1400바이트 지원
블록 쓰기	
개별 변수 읽기	비트, 바이트, 워드 등 모든 형의 직접변수만 지원합니다. (한번에 최대 16개의 직접 변수를 읽거나 쓸 수 있습니다.)
개별 변수 쓰기	
상대 읽기	

9.2.3. ETOS 전용 드라이버

ETOS의 이더넷 전용 프로토콜입니다. ETOS 간의 통신이 필요할 경우, 한쪽 ETOS는 ETOS 전용 드라이버로 설정하고, 다른쪽 ETOS는 ETOS 프로토콜 마스터로 프로그램하여 통신할 수 있습니다. 프로토콜 자료는 문의 바랍니다.

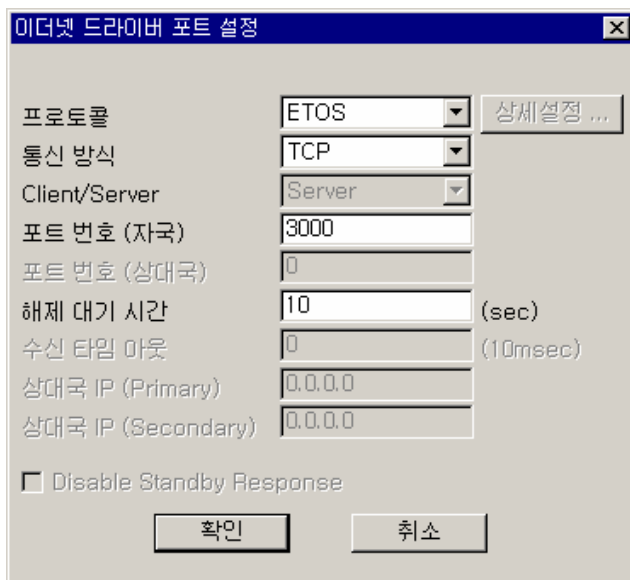


그림 9.2-7 ETOS 이더넷 전용 드라이버 포트 설정

표 9.2-5 ETOS 이더넷 전용 드라이버 포트 설정 항목

항 목	내 용
포트 형식	ETOS
통신 방식	TCP/UDP 중 선택
포트 번호(자국)	1 ~ 65535 범위에서 선택(가급적 1000번 이후를 선택)
해제 대기 시간	설정 시간 동안 통신이 없으면 연결을 종료. 초 단위로 설정

9.2.3.1. 프로토콜 상세 설정

상세 설정 항목이 없습니다.

9.3. 이더넷 채널

이더넷 채널은 ETOS 내장 프로토콜^(*146)을 사용해 마스터로 동작해야 하는 경우나 사용자 정의 프로토콜(User Defined)로 통신을 해야 하는 경우에 사용합니다.

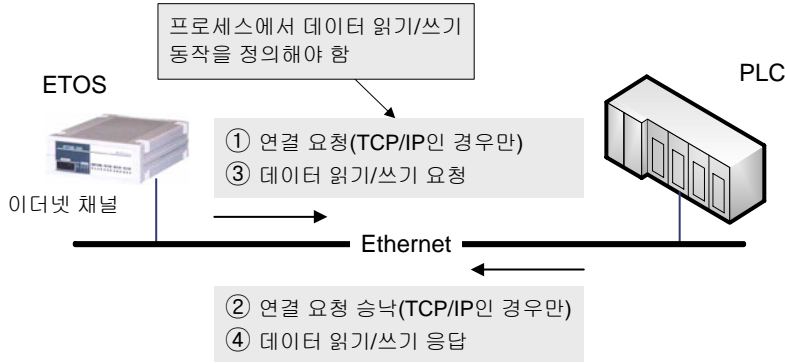


그림 9.3-1 이더넷 채널 사용 예

{이더넷 채널 설정} 대화상자에서 편집하고자 하는 채널 번호에 마우스를 클릭하고 [편집]버튼을 누릅니다. 초기 화면에는 설정된 채널이 없습니다.

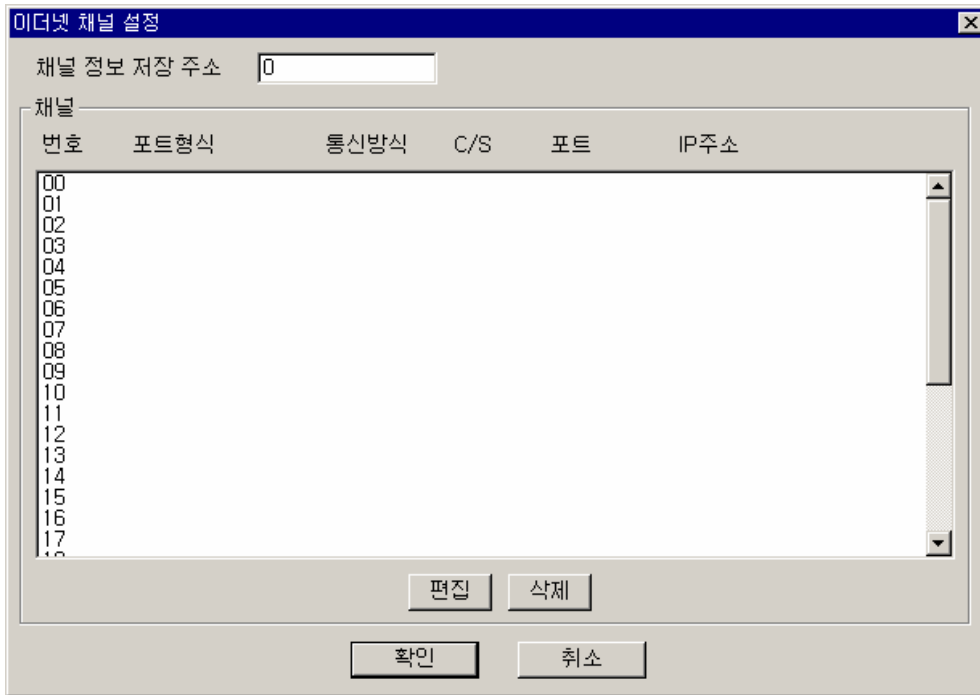


그림 9.3-2 이더넷 채널 설정 대화상자

(*146) OS 버전 1.20 기준으로 GLOFA Enet, MODBUS/TCP 를 지원하고, OS 버전 2.40 부터 MELSEC 지원이 추가되었습니다.

▶ 내장 프로토콜 마스터**포트 파라미터 설정 + 프로세스 정의 필요 (프레임 정의 필요 없음)**

ETOS 가 MELSEC/MODBUS/GLOFA 마스터로 동작해야 하는 경우입니다. ETOS 는 상대국에 정보를 요청하고, 상대국은 수동적으로 응답을 합니다. ETOS 가 정보를 요청하기 위해서 사용자는 ETOS 의 동작을 프로토콜별 전용 아이템을 사용하여, 프로세스에서 정의해 주어야 합니다.

▶ 사용자 정의 프로토콜 마스터/슬레이브**포트 파라미터 설정 + 프레임 정의 + 프로세스 정의 필요**

내장 프로토콜을 제외한 프로토콜로 통신을 해야 하는 경우, 프레임과 프로세스를 정의해서 상대 장비와 통신할 수 있습니다. 프로세스를 작성하는 방법에 따라 마스터/슬레이브 역할이 결정됩니다.

9.3.1. MODBUS/TCP 마스터

MODBUS 마스터로 동작해야 하는 경우, 이더넷 채널의 포트 형식을 MODBUS 로 지정합니다. 통신 방식은 TCP 로 고정되며, Client/Server 역시 Client 로 고정됩니다. 상대국 IP 에는 MODBUS 슬레이브로 동작할 상대 기기의 IP 주소를 입력합니다.

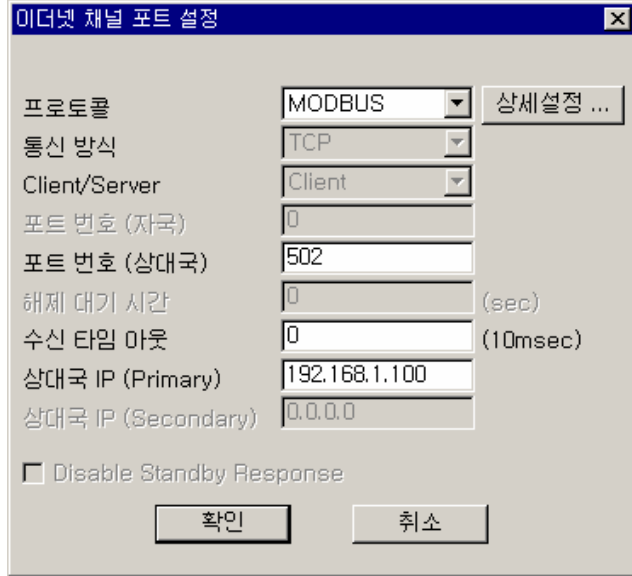


그림 9.3-3 MODBUS 채널 포트 설정

표 9.3-1 MODBUS 채널 포트 설정 항목

항 목	내 용
프로토콜	MODBUS 를 선택합니다.
포트 번호(상대국)	통신 상대국의 서버 포트의 번호를 입력합니다. 보통 502번 포트를 사용합니다.
수신 타임 아웃	IF_MODBUS_READ, IF_MODBUS_WRITE, IF_REQ_RESP 아이템 사용시 응답을 기다리는 최대 시간입니다. 10ms 단위입니다. 예를 들어, 100이라고 설정하면 100 x 10ms = 1000ms(1초)가 됩니다. 값이 0인 경우는, 기본값이 사용되어 500ms(*147) 동안 기다리게 됩니다.
상대국 IP	통신 상대방의 IP 주소를 입력합니다.

* 상대국 IP (Secondary)는 ETOS-1000 에서만 활성화되며, 이중화 모델인 경우만 의미가 있습니다.

(*147) ETOS-50/100A/150 모델인 경우. ETOS-200/1000 인 경우엔 1 초

9.3.1.1. 프로토콜 상세 설정

<프로토콜>을 'MODBUS'로 설정하면, {MODBUS Master} 상세 설정 대화상자가 나타납니다. ({이더넷 채널 포트 설정} 의 [상세설정] 버튼을 눌러 수정할 수 있습니다.)

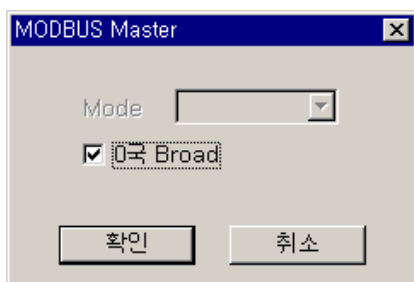


그림 9.3-4 MODBUS Master 상세 설정

<0 국 Broad> 옵션을 선택할 수 있습니다. MODBUS 사양상, 0 번 국번은 브로드캐스트 국번입니다. 그러나 ETOS 는 0 번 국을 브로드캐스트 국번/개별 국번의 두가지로 취급할 수 있습니다. ETOS 기본값은 0 번 국번을 개별 국번으로 취급하는 것입니다. MODBUS 사양과 동일하게 맞추려면 옵션을 체크하시기 바랍니다.

9.3.1.2. MODBUS/TCP 마스터 아이템

마스터 포트에 설정하였으므로, 파라미터 설정 외에 프로세스를 정의해야 합니다. MODBUS 전용의 마스터 아이템은 다음과 같습니다. 시리얼과 동일한 아이템을 사용합니다. 아이템에 대한 설명 및 사용에는 [7.6.37. IF_MODBUS_READ](#) 및 [7.6.38. IF_MODBUS_WRITE](#) 를 참조하시기 바랍니다.

아이템	설 명
IF_MODBUS_READ	상대국의 특정 영역을 읽어 ETOS 메모리에 저장
F_MODBUS_WRITE	ETOS 메모리를 상대국 영역에 쓰기

9.3.1.3. 읽기/쓰기 최대 길이

IF_MODBUS_READ, IF_MODBUS_WRITE 아이템을 사용해서 한 번에 읽거나 쓸 수 있는 데이터의 최대 길이는 다음과 같습니다. 비트의 경우 읽기는 2000 비트까지, 워드는 125 워드, 쓰기의 경우 비트는 1600 비트까지 워드는 100 워드까지 한번에 요청 가능합니다. 그러나, 슬레이브로 동작할 상대기기가 지원하는 범위 안에서 요청해야 합니다.

아이템	데이터 형	Address	최대 길이
IF_MODBUS_READ	Coils(Bits)	0XXXX, 1XXXX	2000 Coils
	Registers(Words)	3XXXX, 4XXXX	125 Registers
IF_MODBUS_WRITE	Coils(Bits)	0XXXX	1600 Coils
	Registers(Words)	4XXXX	100 Registers

9.3.2. GLOFA Enet 마스터

상대기기가 GLOFA Enet 슬레이브이고, 상대기로부터 데이터를 읽어오거나, 데이터를 상대 기기에 쓰고자 하는 경우에 사용합니다. 포트 파라미터 설정은 다음과 같습니다.

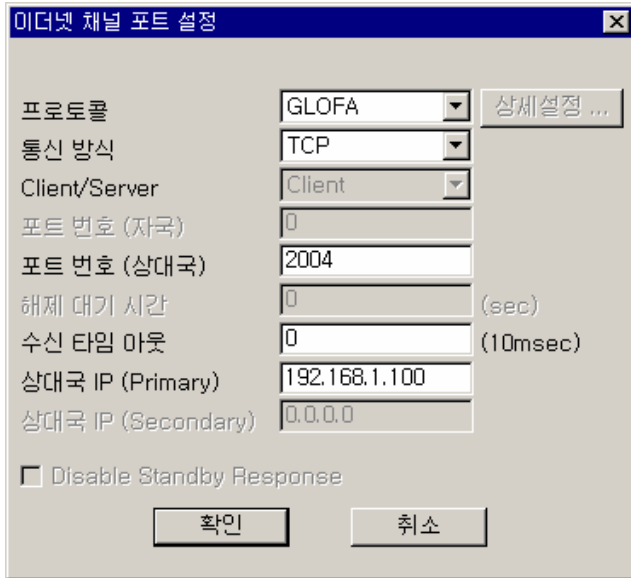


그림 9.3-5 GLOFA 채널 포트 설정

표 9.3-2 GLOFA 채널 포트 설정 항목

항 목	내 용
프로토콜	GLOFA 선택
통신 방식	TCP/UDP 중에서 선택
Client/Server	Client로 고정
포트 번호	상대국 포트, 2004(TCP)/ 2005(UDP)
수신 타임 아웃	IF_GLOFA_READ, IF_GLOFA_WRITE, IF_REQ_RESP 아이템 사용시 응답을 기다리는 최대 시간입니다. 10ms 단위입니다. 예를 들어, 100이라고 설정하면 100 x 10ms = 1000ms(1초)가 됩니다. 값이 0인 경우는, 기본값이 사용되어 500ms ^(*148) 동안 기다리게 됩니다.
상대국 IP	지정된 채널로 ETOS와 통신할 상대국의 IP주소

* 상대국 IP (Secondary)는 ETOS-1000 에서만 활성화되며, 이중화 모델인 경우만 의미가 있습니다.

9.3.2.1. 프로토콜 상세 설정

상세 설정 항목이 없습니다.

(*148) ETOS-50/100A/150 모델인 경우. ETOS-200/1000 인 경우엔 1 초

9.3.2.2. GLOFA Enet 마스터 아이템

마스터 포트에 설정하였으므로, 파라미터 설정 외에 프로세스를 정의해야 합니다. GLOFA 전용의 마스터 아이템은 다음과 같습니다. 시리얼과 동일한 아이템을 사용합니다. 아이템 설명 및 사용 예는 [7.6.31. IF_GLOFA_READ](#) 및 [7.6.32. IF_GLOFA_WRITE](#) 를 참조하시기 바랍니다.

아이템	설 명
IF_GLOFA_READ	상대국의 특정 영역을 읽어 ETOS 메모리에 저장
IF_GLOFA_WRITE	ETOS 메모리를 상대국 영역에 쓰기

9.3.2.3. 읽기/쓰기 최대 길이

IF_GLOFA_READ, IF_GLOFA_WRITE 아이템을 사용해서 한 번에 읽거나 쓸 수 있는 데이터의 최대 길이는 다음과 같이 상대국 주소의 데이터 형에 따라 달라집니다. 상대국 주소에 바이트 형 변수를 지정했을 경우, 최대 1400 바이트까지 읽고, 쓸 수 있으나, 나머지 형에서는 1 개 씩 밖에 읽고, 쓸 수 없습니다. 보다 자세한 사항은 GLOFA Enet 자료를 참조하시기 바랍니다.

아이템	데이터 형	최대 길이
IF_GLOFA_READ IF_GLOFA_WRITE	%MX, %IX, %QX	1
	%MB, %IB, %QB	1400
	%MW, %IW, %QW	1
	%MD, %ID, %QD	1
	%ML, %IL, %QL	1

- * 시리얼과 동일한 아이템을 사용하지만, 시리얼인가 또는 이더넷인가에 따라 송,수신 데이터의 최대 길이가 다르므로 주의를 요합니다. (GLOFA Cnet 사양과 상이하므로 주의하시기 바랍니다.)

9.3.2.4. 프로그램 예제

다음의 예는 1 초 주기로 상대기기의 메모리를 ETOS 메모리로 읽어 오는 프로세스입니다. 상대기기는 GLOFA Slave 로 가정합니다.

NO	아이템	설명
00	IF_TIMER	100
01	REMARK	-----
02	REMARK	<<<< 블록 읽기/쓰기 >>>>
03	REMARK	-----
04	REMARK	<< 100BYTE >>
05	MEM_SET	T:M13000, Size:1400, Type:16진수, Data:FF
06	IF_GLOFA_READ	Media:Ethernet, Port:2, DestAddr:MB11000, Length:100, SelfAddr:M13000, BitPos:0, Status:M1064
07	END_IF	
08	REMARK	-----
09	REMARK	<<<< 개별 변수 읽기/쓰기 >>>>
10	REMARK	-----
11	REMARK	<< BIT >>
12	IF_GLOFA_READ	Media:Ethernet, Port:2, DestAddr:mx128000, Length:1, SelfAddr:M18000, BitPos:0, Status:M1128
13	END_IF	
14	REMARK	-----
15	REMARK	<< WORD >>
16	IF_GLOFA_READ	Media:Ethernet, Port:2, DestAddr:mw8000, Length:1, SelfAddr:M18000, BitPos:0, Status:M1192
17	END_IF	
18	END_IF	

9.3.3. MELSEC 마스터

MELSEC PLC로부터 데이터를 읽어오거나 쓸 경우입니다. 포트 파라미터 설정은 다음과 같습니다.

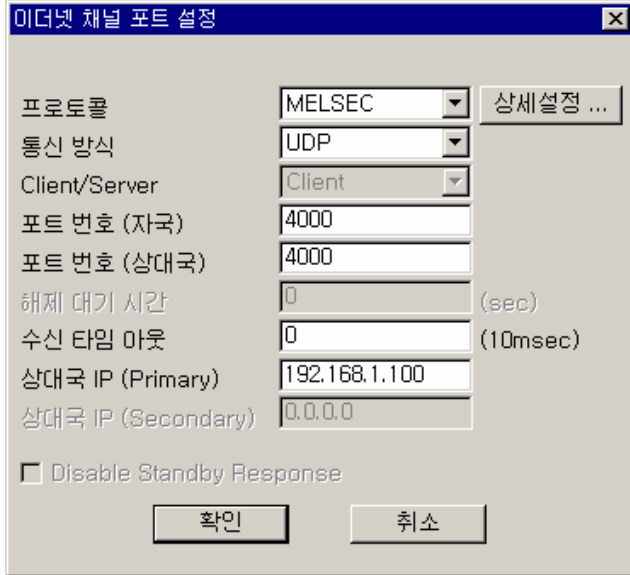


그림 9.3-6 MELSEC 채널 포트 설정

표 9.3-3 MELSEC 채널 포트 설정 항목

항 목	내 용
프로토콜	MELSEC을 선택합니다. 선택과 동시에 {MELSEC Master} 대화상자가 나타납니다. 관련 설정은 다음 페이지의 9.3.3.1 프로토콜 상세 설정 을 참조하시기 바랍니다.
통신 방식	TCP 또는 UDP를 선택할 수 있습니다.
포트번호(자국)	통신방식을 UDP로 설정한 경우만 활성화됩니다.
포트번호(상대국)	통신 상대국의 포트 번호를 입력합니다.
수신 타임 아웃	IF_MELSEC_READ, IF_MELSEC_WRITE, IF_REQ_RESP 아이템 사용시 응답을 기다리는 최대 시간입니다. 10ms 단위입니다. 예를 들어, 100이라고 설정하면 100 x 10ms = 1000ms(1초)가 됩니다. 값이 0인 경우는, 기본값이 사용되어 500ms ^(*149) 동안 기다리게 됩니다.
상대국 IP	통신하고자 하는 상대국의 IP 주소를 입력합니다.

* 상대국 IP (Secondary)는 ETOS-1000에서만 활성화되며, 이중화 모델인 경우만 의미가 있습니다.

(*149) ETOS-50/100A/150 모델인 경우. ETOS-200/1000 인 경우엔 1 초

9.3.3.1. 프로토콜 상세 설정

<프로토콜>을 'MELSEC'으로 선택함과 동시에 다음 그림의 {MELSEC Master} 대화상자가 나타납니다. {이더넷 채널 포트 설정}의 [상세설정] 버튼을 눌러 수정할 수 있습니다.

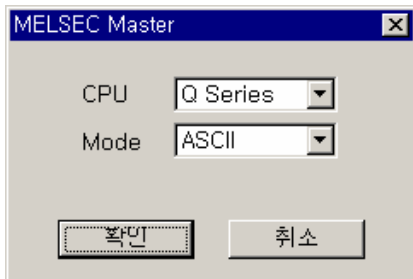


그림 9.3-7 MELSEC Master 상세설정

표 9.3-4 MELSEC Master 상세 설정 항목

항 목	설 명
CPU	PLC의 CPU 타입을 의미합니다. Q 또는 A시리즈를 선택할 수 있으나 현재 Q시리즈만 지원됩니다. 주의하시기 바랍니다.
Mode	통신 데이터의 형식을 의미하며, ASCII 또는 Binary Code를 모두 지원합니다.

9.3.3.2. MELSEC 마스터 아이템

마스터 포트에 설정하였으므로, 파라미터 설정 외에 프로세스를 정의해야 합니다. MELSEC 전용의 마스터 아이템은 다음 표와 같습니다.

아이템에 대한 설명은 [7.6.34. IF_MELSEC_READ](#) 및 [7.6.35. IF_MELSEC_WRITE](#)를 참조하시기 바랍니다.

아이템	설 명
IF_MELSEC_READ	상대 PLC의 특정 디바이스 영역을 읽어 ETOS에 저장
IF_MELSEC_WRITE	ETOS 메모리 영역을 상대 PLC의 특정 디바이스에 쓰기

9.3.4. 사용자 정의 프로토콜 마스터/슬레이브 (User Defined)

ETOS 가 제공하지 않는 프로토콜을 사용하는 기기와 통신하려고 하는 경우, 사용자는 해당 프로토콜의 프레임을 정의하여 상대기와 통신 할 수 있습니다. 포트 파라미터 설정은 다음과 같습니다.

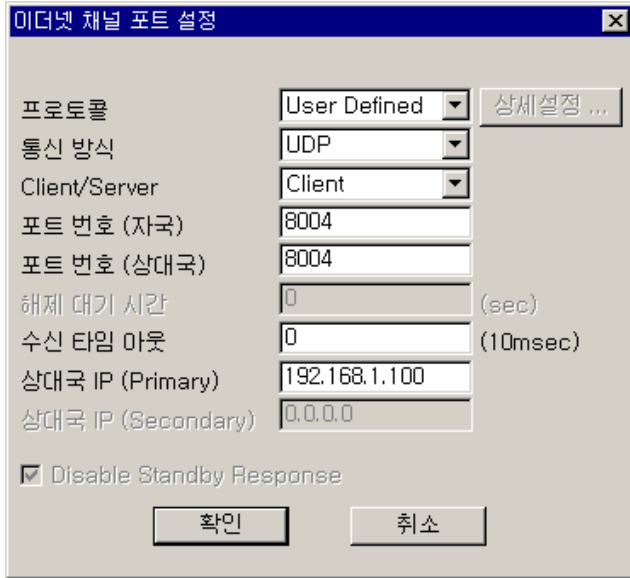


그림 9.3-8 사용자 정의 프로토콜 채널 포트 설정

표 9.3-5 사용자 정의 프로토콜 채널 포트 설정 항목

항 목	내 용
프로토콜	User Defined 선택
통신 방식	TCP/UDP 중에서 선택합니다.
Client/Server	Client/Server 중에서 선택합니다.
포트 번호(자국)	통신에 사용할 로컬 포트 번호를 지정합니다. UDP인 경우만 활성화되며, TCP의 경우는 자동할당합니다.
포트 번호(상대국)	통신 상대국의 포트 번호를 지정합니다.
해제 대기 시간	Server인 경우만 설정합니다. 이 시간 동안 상대방과의 데이터 통신이 없으면, 연결 상태가 비정상적이라고 보고, 연결을 해제합니다. 초 단위
수신 타임 아웃	IF_REQ_RESP 아이템 사용시 응답을 기다리는 최대 시간입니다. 10ms 단위입니다. 예를 들어, 100이라고 설정하면 100 x 10ms = 1000ms(1초)가 됩니다. 값이 0인 경우는, 기본값이 사용되어 500ms ^(*150) 동안 기다리게 됩니다.
상대국 IP	지정된 채널로 ETOS와 통신할 상대국의 IP주소

* 상대국 IP (Secondary)는 ETOS-1000에서만 활성화되며, 이중화 모델인 경우만 의미가 있습니다.

(*150) ETOS-50/100A/150 모델인 경우. ETOS-200/1000 인 경우엔 1 초

▶ Client/Server

설정하고 있는 포트가 상대방 서버 포트에 연결(Connection)을 요청하는 경우는 Client 로 설정합니다. 상대방이 특정 포트를 열어 놓고, 연결을 기다리는 경우에 해당합니다.

반대로, 상대방이 Client 로 ETOS 에 연결을 요청하려는 경우엔, Server 로 설정해서 상대방의 연결 요청을 받아들여야 합니다.

연결 지향 통신인 TCP 의 경우만 설정하며, UDP 는 연결 자체가 없으므로 Client/Server 구분이 없습니다. (Client/Server 설정을 구분하지 않습니다.)

9.3.4.1. 프레임 정의

{프레임 등록} 창에서 마우스 오른쪽 버튼을 눌러 [이더넷 프레임 추가]를 선택합니다. {프레임 등록} 창의 '시리얼 프레임' 또는 'Card #N' 아래쪽에 '이더넷 프레임'이 추가됩니다. '이더넷 프레임'에 마우스를 클릭하고 다시 오른쪽 버튼을 눌러 [그룹 추가]를 선택합니다. 적당한 그룹명을 입력하고, 채널 번호에 위에서 설정한 User Defined 포트의 채널 번호를 입력하고, 프레임 종류에는 송,수신 프레임 중 원하는 프레임을 설정합니다. (*151)

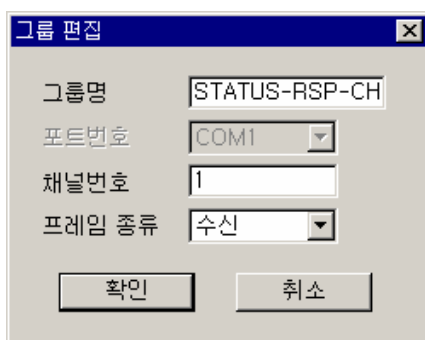


그림 9.3-9 그룹 등록 예

이제 추가된 그룹에 마우스를 클릭하고, 다시 오른쪽 버튼을 눌러 [프레임 추가]를 선택합니다.



그림 9.3-10 프레임 등록

<종류>에는 HEAD, TAIL, BODY 가 있습니다. 사용하려는 프로토콜 자료를 참조하여 각각을 설정합니다. BODY 는 항상 설정해야 하고, 경우에 따라 HEAD, TAIL 은 생략가능합니다.

(*151) 프레임 등록 방법에 대한 자세한 사항은 6.4. 프레임 등록 예제를 참조하시기 바랍니다.

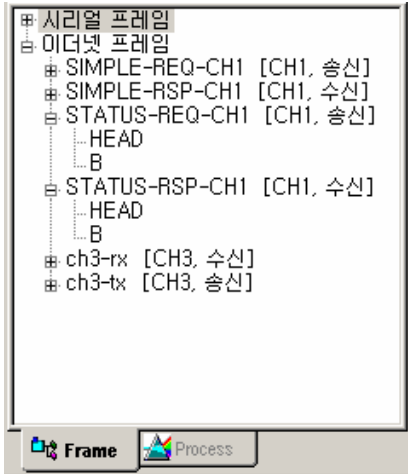


그림 9.3-11 이더넷 프레임 등록 예

9.3.4.2. 사용자 정의 프로토콜 아이템

사용자 정의 프로토콜은 프레임을 정의한 후, 동작을 정의하는 프로세스도 프로그램해야 합니다. 동작을 정의하는 아이템은 다음과 같습니다.

아이템	설 명
IF_REQ_RESP	상대방에 특정 프레임을 송신하고, 응답을 수신합니다. 송,수신 프레임을 모두 정의
FRAME_SND	상대방에 특정 프레임을 송신, 송신 프레임만 정의
IF_FRAME_RCV	상대방으로부터 특정 프레임을 수신, 수신 프레임만 정의

9.3.4.3. 관련 아이템

OS 버전 1.20 이전에는 반드시 사용해야 했으나, 1.20 이후 버전부터는 선택적으로 사용하는 아이템입니다.

아이템	설 명
ENET_OPEN	상대국에 연결 요청을 합니다. 상대국이 연결을 승낙하면 다음 아이템을 수행합니다.
ENET_CLOSE	이더넷 채널 연결을 끊습니다. 이더넷 채널 설정에서 설정한 채널 번호를 입력합니다.

▶ OS 버전 1.20 이후 사용조건

TCP Client 인 경우만 선택적으로 사용하며, TCP Server 나 UDP 인 경우는 사용하지 않습니다. TCP Client 에서 사용하는 경우를 계속해서 설명하겠습니다.

9.3.4.4. TCP Client

다음 그림은 ETOS 가 TCP Client 로 설정된 경우, 데이터 통신의 순서를 나타냅니다. TCP 통신은 Server 에게 연결 요청을 하여, Server 에 의해 연결이 승낙된 경우만 데이터 통신이 가능합니다.

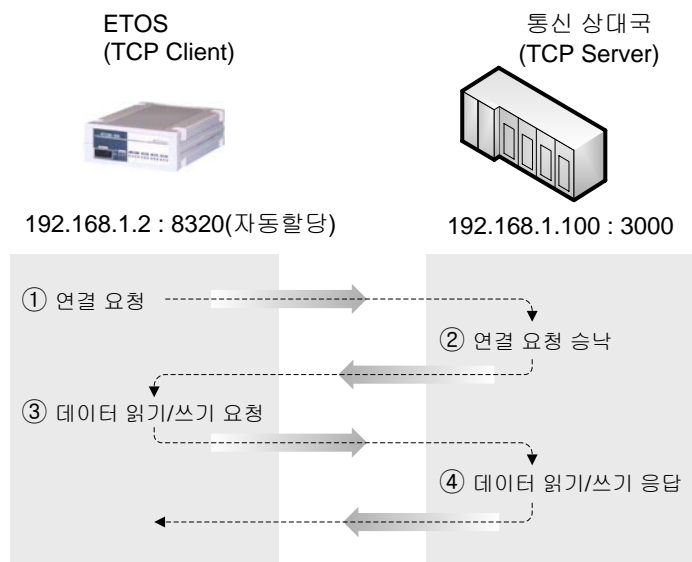


그림 9.3-12 TCP/IP 데이터 통신

▶ ENET_OPEN 을 사용하지 않아도 되는 경우

ETOS 는 프로세스에서 ENET_OPEN 아이템이나, 다음의 아이템을 수행할 때, 연결 요청을 하게 됩니다.

- * FRAME_SND
- * IF_REQ_RESP
- * IF_XXX_READ
- * IF_XXX_WRITE

따라서, 프로세스 안에 위에 열거한 아이템을 사용한다면, ENET_OPEN 아이템을 반드시 사용해야 하는 것이 아님을 알 수 있습니다.

▶ ENET_OPEN 을 반드시 사용해야 하는 경우

프로세스가 다음과 같은 경우에는, ENET_OPEN 아이템을 별도의 IF_TIMER 프로세스에서 반드시 사용해야 합니다.

```
00 IF_FRAME_RCV ...
01     FRAME_SND ...
02 END_IF
```

FRAME_SND 아이템이 수행되면 ETOS 는 연결 요청을 먼저 하고, 상대방에 의해 연결이 승낙되면, 프레임을 송신 합니다. 그러나, IF_FRAME_RCV 가 먼저, 수행되어야 하고, 연결이 맺어지기 전까지는 데이터 통신이 불가능하므로 (그림 9.3-12 참조), IF_FRAME_RCV 는 결코 수행되지 않으리라 예측할 수 있습니다. 따라서, 별도의 IF_TIMER 프로세스에서 ENET_OPEN 아이템을 사용하여 연결을 맺어야 합니다.

9.3.4.5. 프로그램 예제

다음 예는 1 초 주기로 상대기기의 특정 영역을 읽어오는 프로세스입니다. ENET_OPEN, ENET_CLOSE 아이템은 사용하지 않고, 바로 통신 아이템(IF_REQ_RESP, FRAME_SND)을 사용할 수 있습니다.

NO	아이템	설명
00	IF_TIMER	100
01	REMARK	-----
02	REMARK	<< BYTE >>
03	IF_REQ_RESP	Req:SIMPLE-REQ-CH6.%MB16000-R16, Resp:SIMPLE-RSP-CH6.%MB16000-R16
04	ARITHMETIC	Type:U4, 1, +, M1480, M1480
05	ELSE	
06	ARITHMETIC	Type:U4, 1, +, M1488, M1488
07	END_IF	
08	REMARK	-----
09	REMARK	<< WORD >>
10	IF_REQ_RESP	Req:SIMPLE-REQ-CH6.%MW8000-R16, Resp:SIMPLE-RSP-CH6.%MW8000-R16
11	ARITHMETIC	Type:U4, 1, +, M1512, M1512
12	ELSE	
13	ARITHMETIC	Type:U4, 1, +, M1520, M1520
14	END_IF	
15	END_IF	

그림 9.3-13 프로그램 예

9.4. 이더넷 고속링크

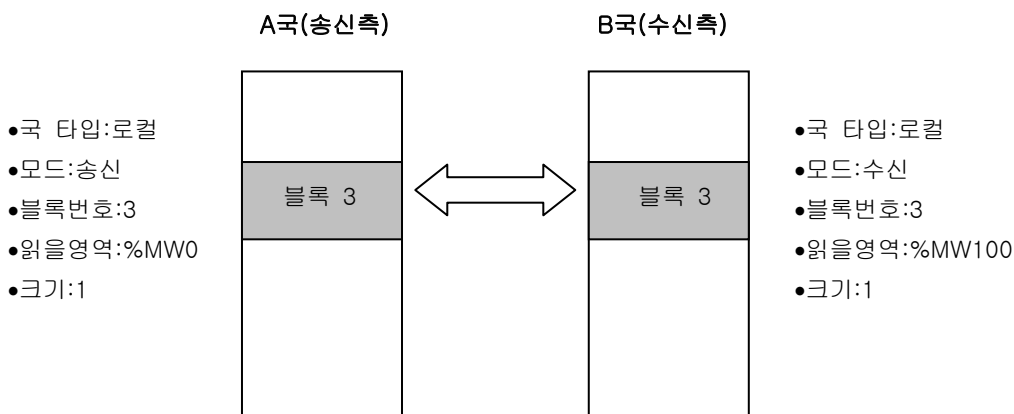
이더넷 고속링크는 ETOS 와 GLOFA PLC 이더넷 통신모듈, 또는 ETOS 와 ETOS 간의 통신 방법으로 특정 시간마다 주기적으로 상대국의 데이터나 정보를 교환할 때 사용합니다.

9.4.1. 데이터 송, 수신 원리

고속링크 통신은 송신시 상대국을 지정하지 않고, 브로드캐스트(Subnet Broadcast)^(*152)로 송신하므로 동일한 송신데이터를 여러국에서 동시에 수신할 수 있습니다. 또, 수신국은 송신국의 국번과 블록 번호를 이용해서 브로드캐스트로 송신되는 데이터 중에서 자기가 원하는 데이터만 선별해서 수신할 수 있습니다.

고속링크의 동작 원리를 예를 들어 설명합니다.

예) A 국은 %MW0 데이터를 송신하고, B 는 A 국의 %MW0 를 수신해서 B 국의 %MW100 에 저장



A 국에서 %MW0 데이터를 읽어서 블록 번호 3 번으로 보낸다고 가정합니다. B 국이 이 데이터를 수신하기 위해서는 국번을 A, 블록번호는 송신 블록 번호인 3 번으로 동일하게 설정해야 하고, 마지막으로 저장 영역으로 %MW100 을 설정하면 됩니다.

(*152) 동일한 Subnet 에 속해있는 상대국이 모두 수신할 수 있습니다.

9.4.2. 이더넷 고속링크 설정

[편집] -> [파라미터 설정]을 선택합니다.

{파라미터 설정} 대화상자의 이더넷 파라미터에서 이더넷 고속링크를 설정할 수 있습니다.

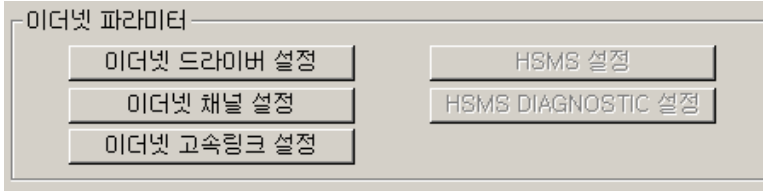


그림 9.4-1 이더넷 파라미터

[이더넷 고속링크 설정]버튼을 누르면 다음과 같은 고속링크 설정 대화상자가 나타납니다.

{이더넷 고속링크 설정} 대화상자에서 최대 64 개까지 고속 링크 파라미터를 설정할 수 있습니다.

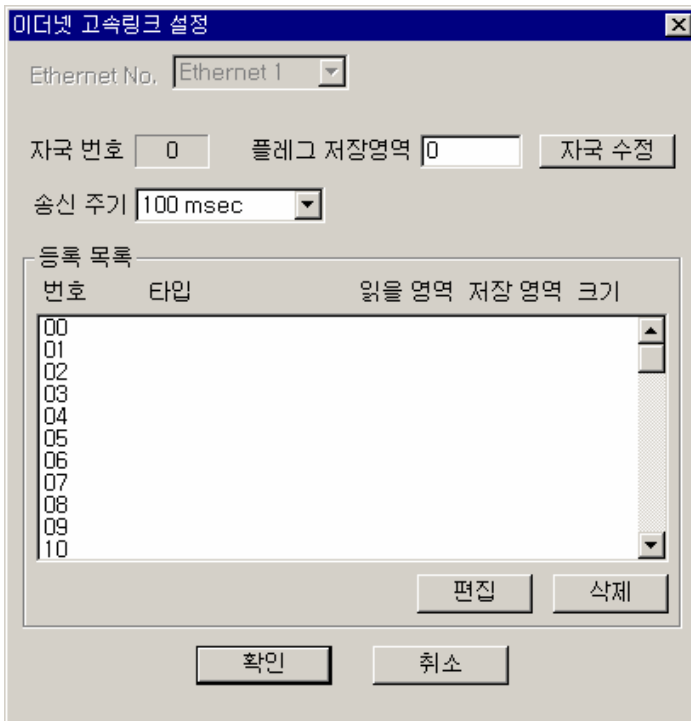
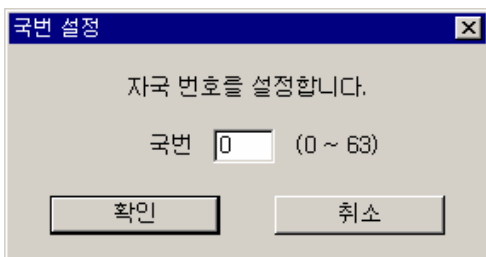


그림 9.4-2 이더넷 고속링크 설정 대화상자

9.4.2.1. 자국 번호

통신국번을 설정하는 항목이며, 여기서 국번이란 ETOS의 고속링크 국번을 의미합니다. [자국 수정] 버튼을 선택하면 {국번 설정} 대화상자에서 국번을 설정할 수 있습니다. 국번 설정 범위는 0~63 국까지이며 동일 네트워크에 동일한 국번이 존재하지 않도록 서로 다른 국번을 배정해야 합니다.



9.4.2.2. 플래그 저장 영역

플래그 저장 영역으로 지정한 번지부터, 고속링크의 동작을 나타내는 RUN-LINK/LINK-TROUBLE 정보를 비트로 저장합니다. 고속링크 플래그는 GLOFA PLC의 이더넷 통신 모듈 고속링크 플래그와 동일한 내용을 저장합니다.

자국 번호 플래그 저장영역 자국 수정

▶ 대표정보 (HS_LINK_INFO)

설정된 고속링크 파라미터에 대한 통신정보를 표시하는 대표 플래그입니다. 다음 예는 고속링크 플래그 영역을 24000 번지로 설정한 경우입니다. (위 그림 참조)

run-link : M24000.0 -> 설정된 파라미터 전체가 정상동작 시 'ON' 됨.
 link-trouble : M24000.1 -> 파라미터중 하나라도 비정상 동작 시 'ON' 됨

▶ 개별 플래그

다음 예는 고속링크 플래그 영역을 24000 으로 설정한 경우로, 사용자 메모리 24002 번지부터 32 바이트(256 비트)의 정보를 표시합니다. 등록된 파라미터 별로 동작상태를 확인할 수 있습니다.

표 9.4-1 고속링크 플래그 비트정보

등록번호 \ 내용	STATE-INFO (24002~24009)	MODE-INFO (24010~24017)	TRX-INFO (24018~24025)	ERRR0-INFO (24026~24033)
0 ~ 7	24002.b0~b7	24010.b0~b7	24018.b0~b7	24026.b0~b7
8 ~ 15	24003.b0~b7	24011.b0~b7	24019.b0~b7	24027.b0~b7
16 ~ 23	24004.b0~b7	24012.b0~b7	24020.b0~b7	24028.b0~b7
24 ~ 31	24005.b0~b7	24013.b0~b7	24021.b0~b7	24029.b0~b7
32 ~ 39	24006.b0~b7	24014.b0~b7	24022.b0~b7	24030.b0~b7
40 ~ 47	24007.b0~b7	24015.b0~b7	24023.b0~b7	24031.b0~b7
48 ~ 55	24008.b0~b7	24016.b0~b7	24024.b0~b7	24032.b0~b7
56 ~ 63	24009.b0~b7	24017.b0~b7	24025.b0~b7	24033.b0~b7

* 개별 플래그의 등록번호는 이더넷 고속링크의 등록번호를 나타내며, 각각의 의미는 GLOFA PLC 이더넷 고속링크 내용과 동일합니다. 고속링크 파라미터 정보는 GLOFA PLC 이더넷 사용설명서를 참조 바랍니다.

9.4.2.3. 링크 항목 설정

각각의 링크 파라미터는 {이더넷 고속링크 설정} 대화상자에서 [편집] 버튼을 눌러 추가할 수 있습니다. {링크 항목 설정} 대화상자에서 송신모드의 <읽을 영역>과, 수신모드의 <저장 영역>은 ETOS의 내부 메모리 영역을 저장하는 것으로, 설정 영역이 워드 단위로 되어 있으므로 ETOS의 실제 주소는 설정한 값의 2 배수 번지를 의미합니다. 예를 들어 수신 파라미터에서 <저장 영역>으로 %MW1000을 설정했으면 워드영역이므로 수신된 데이터는 ETOS의 2000 번지부터 저장됩니다. (ETOS는 바이트 단위) 송신의 경우에는 이와 반대로 읽을 영역의 %MW 영역이 ETOS의 메모리를 나타내며 실제 송신 번지는 설정 주소의 2 배수 번지부터 송신됩니다. 예를 들어 %MW100을 송신 영역으로 설정했으면 ETOS 메모리 200 번지 내용이 송신됩니다.

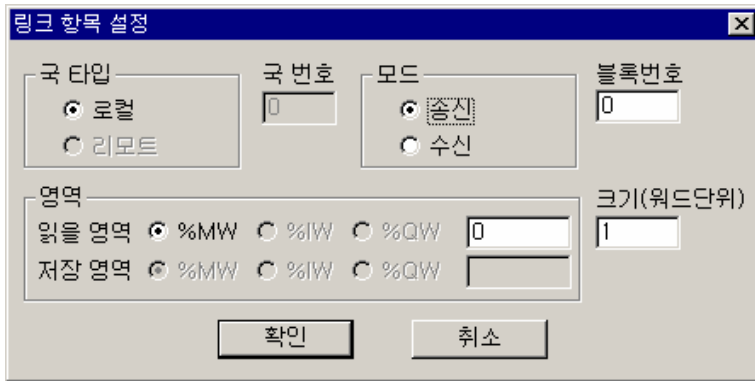


그림 9.4-3 고속링크 항목 설정

표 9.4-2 고속링크 설정 항목

항 목	내 용
국 타입	ETOS의 이더넷 고속링크는 로컬로 고정됩니다.
국 번호	수신국에서 송신국의 국번을 지정합니다. 국번 범위는 0 ~ 63번입니다. 송신 블록의 경우 생략합니다. (수신 모드일때만 활성화됩니다.)
모드	설정 블록의 송, 수신 여부를 설정합니다.
블록번호	송신 블록인 경우 겹치지 않도록 설정합니다. 수신 모드일때는 송신 블록 번호를 참조하여 동일하게 설정합니다.
영역	송, 수신 영역을 워드 단위 주소로 설정합니다.
크기	송, 수신 데이터의 길이를 워드 단위로 입력합니다. 최대 200워드까지 입력할 수 있습니다.

SECS 통신

제 10 장

ETOS 의 SECS 통신 기능에 대해서 설명합니다.

	내용	위치
10.1	개요	358
10.2	SECS1 통신	358
10.3	HSMS 통신	378
10.4	Bridge 통신	381
10.5	진단 시스템 통신	386

10. SECS 통신

10.1. 개요

SECS(SEMI EQUIPMENT COMMUNICATIONS STANDARD)는 반도체 공정 장비와 host 사이의 메시지 교환을 위한 communication interface 표준을 정의하고 있습니다. ETOS 는 SECS-I, HSMS, SECS-II 와 GEM 의 일부를 지원하고 있습니다.

10.2. SECS1 통신

SECS-I 은 Physical Link, Block transfer protocol, Message protocol 로 구성되어 있습니다.

- * Physical Link: RS-232
 - * Block transfer protocol: block 의 전송과 수신
 - * Message protocol: message 를 block 으로 나누거나 block 을 message 로 재 조합, error 추적
- SECS-I 은 Point to point communication 이고, data 는 245byte 이하의 block 형태로 전송합니다. 각 block 은 10byte 를 header 와 data 로 구성되어 있습니다. 각 바이트는 하나의 stop bit, 하나의 시작 bit, 8 개의 bit 로 이루어진 character 로 되어 있습니다. 각 Message 는 request 와 reply 로 짝을 이루어 송수신 됩니다. 양방향 통신이고, 비동기식 통신입니다. 더 자세한 것은 SEMI 의 표준을 참조하시고, 몇 가지 예를 들어 보겠습니다.

10.2.1. 10 초마다 S1F1 을 송신하고, S1F2 를 받기

ETOS 의 COM1 을 SECS-I 으로 설정하고, 10 초마다 S1F1 을 송신하고, S1F2 를 받도록 프로그램 해 보겠습니다. ETOS-150 Dual HSMS 로 예를 들겠습니다.

10.2.1.1. 새 파일의 시작과 파라미터 설정

ETOS-PD V2.3 부터는 프로그램 시작시에 자동으로 새파일을 열도록 되어 있습니다. (이전 버전 사용자는 [파일] -> [ETOS TYPE 변경], [파일] -> [새 파일] 메뉴를 차례로 실행해야 합니다.)

ETOS-PD 를 실행하고, 아래처럼 ETOS 종류를 ETOS-100/150-Dual 로, 드라이버를 SECS 로 선택합니다.

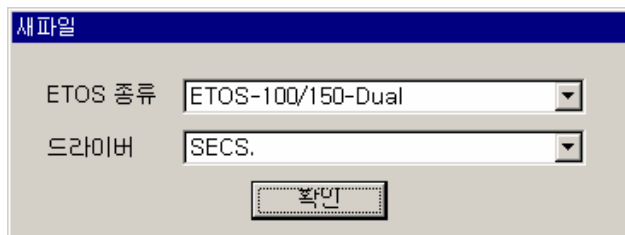


그림 10.2-1 ETOS 종류 및 드라이버 선택

[확인] 버튼을 누르면 바로 아래의 {파라미터 설정} 대화상자가 열립니다.

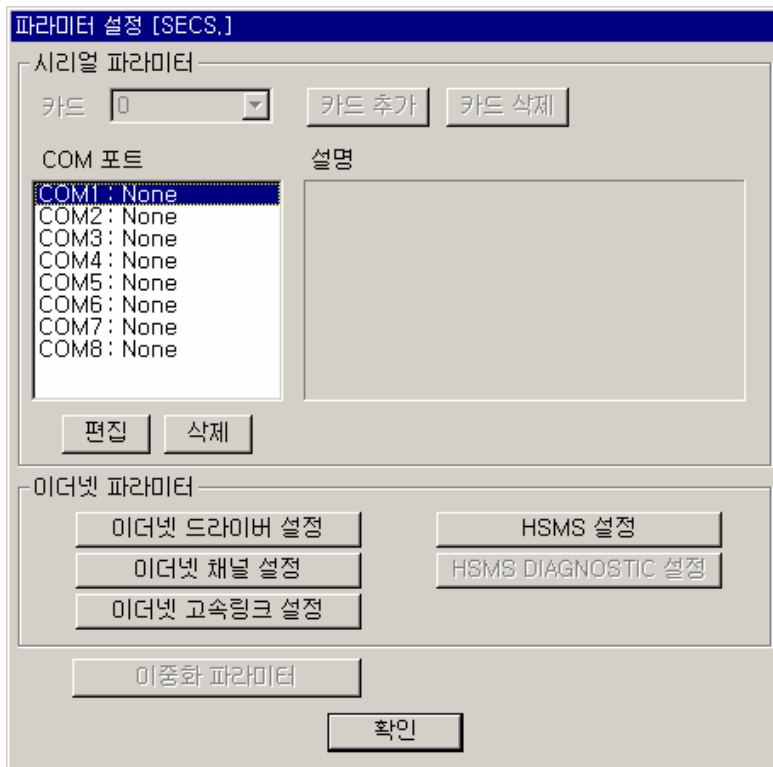


그림 10.2-2 파라미터 설정

COM1 을 SECS-I 으로 설정하려고 합니다. COM 포트의 COM1 을 선택하고 [편집] 버튼을 누르거나, 더블 클릭하십시오. 아래의 {포트 종류 선택} 대화상자에서 SECS-I 을 선택합니다.

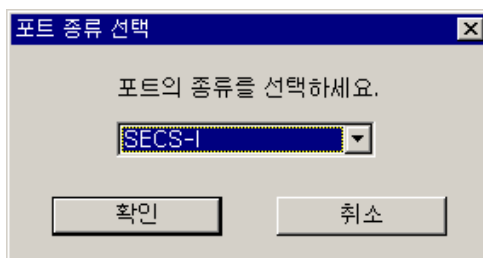


그림 10.2-3 포트 종류 선택

[확인]을 클릭하면 {COM 포트 설정(SECS-I)} 창이 나옵니다. 여기서 SECS-I의 기본 파라미터를 설정합니다. [5.3.1.4. SECS-I 포트](#)를 참조하십시오. 통신 속도는 9600bps, Type은 Equipment로, Device ID는 1로 하겠습니다. 나머지는 모두 초기값을 그대로 사용하겠습니다.

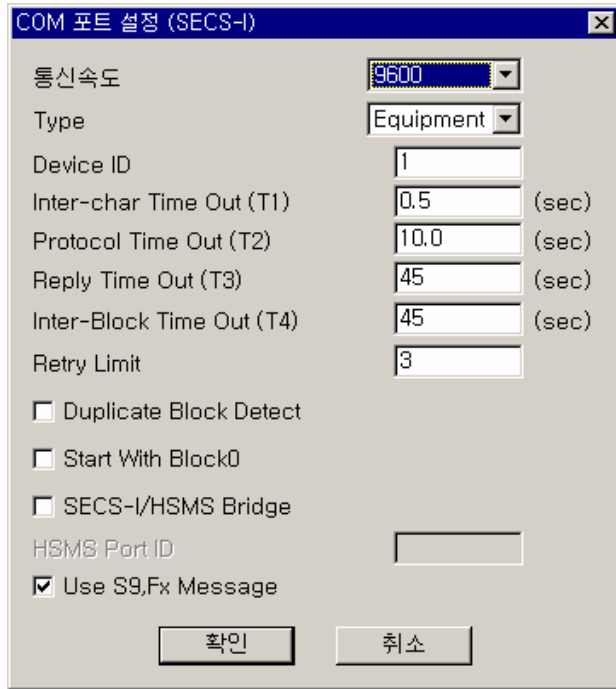


그림 10.2-4 COM 포트 설정 (SECS-I)

[확인]을 클릭하면 아래처럼 COM1이 SECS-I 포트로 설정됩니다.

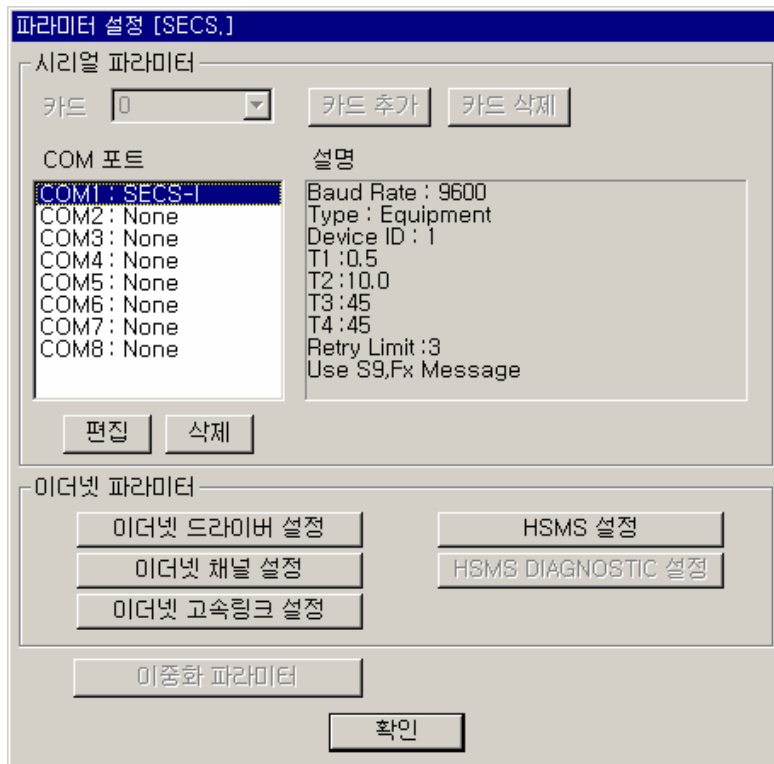


그림 10.2-5 파라미터 설정

10.2.1.2. S1F1, S1F2 메시지의 등록

ETOD-PD의 좌측 창에서 [SECS] 탭을 눌러 {SECS 메시지 등록} 창을 선택합니다. 마우스 오른쪽 버튼으로 클릭한 후, 아래처럼 [메시지 추가] 메뉴를 클릭합니다.

{SECS 메시지}창에서 <Stream>, <Function>번호를 넣고, <이름>은 "S1F1_SND"라고 하겠습니다. 표준 Structure를 그대로 사용할 것이므로 <Structure 파일 참조>를 체크합니다.

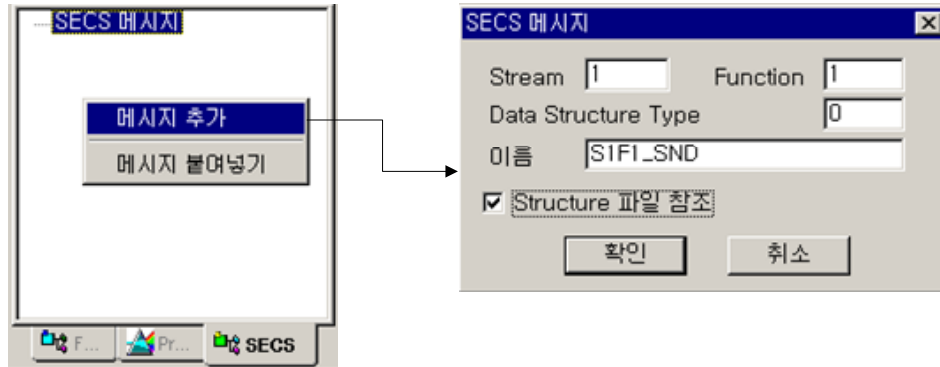


그림 10.2-6 SECS 메시지 등록

확인을 누르면 바로 {SECS-II Message Definition} 창이 나옵니다. 헤더만 있는 메시지 이므로 수정할 것 없이 바로 확인을 누릅니다.

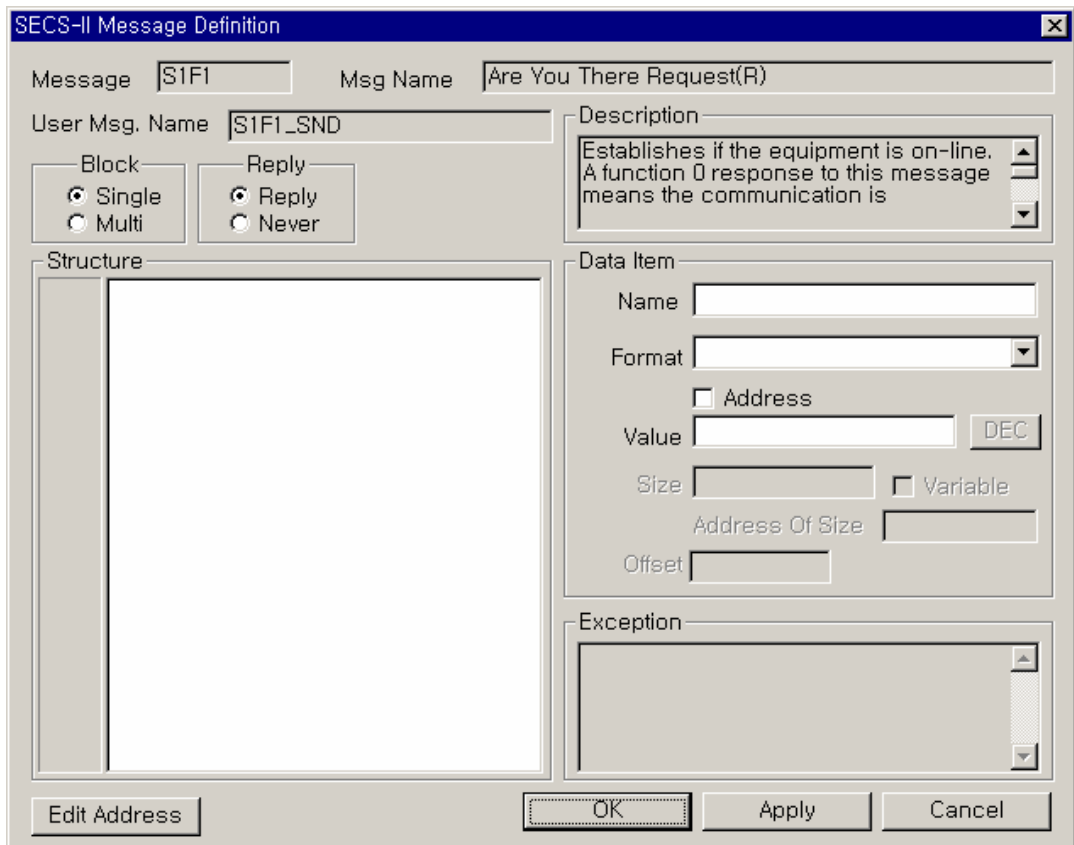


그림 10.2-7 SECS-II Message Definition

마찬가지로 수신할 S1F2 를 등록하겠습니다. 아래와 같이 설정합니다. 역시 표준 Structure 를 그대로 사용할 것이므로 <Structure 파일 참조>를 체크합니다.

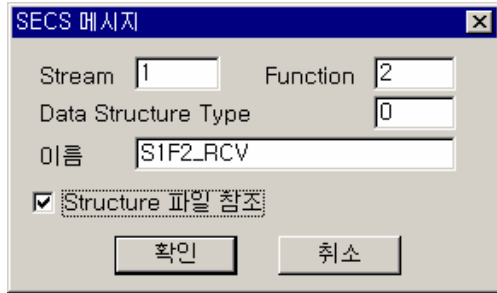


그림 10.2-8 SECS 메시지

첫 번째 LIST 는 수정할 것이 없고, MDLN 은 아래처럼 ASCII 로 최대 10 바이트를 받아 ETOS 의 16 번째에 저장합니다. 16 번째이므로 <Address>를 체크하고, <Value>에 16 을 입력합니다. 크기는 최대 10 바이트이므로 <Size>에 10 을 넣습니다.

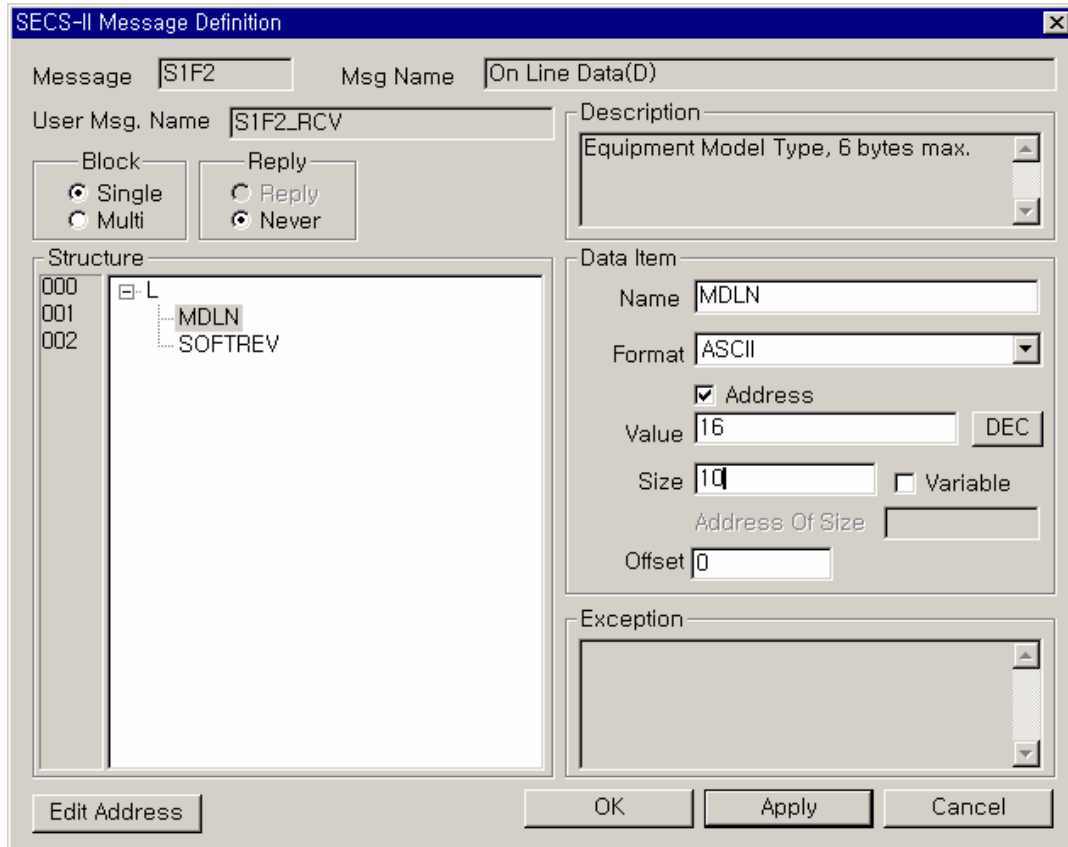


그림 10.2-9 SECS-II Message Definition

<SOFTREV>은 아래처럼 ASCII 로 최대 10 바이트를 받아 ETOS 의 32 번지에 저장합니다.

그림 10.2-10 SECS-II Message Definition

아래처럼 두 개의 메시지가 등록되었습니다.

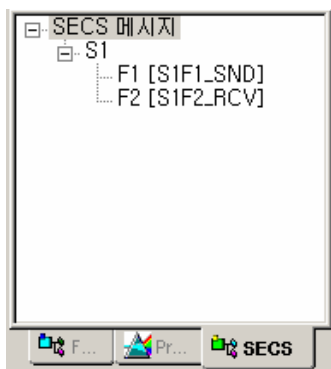


그림 10.2-11 등록된 SECS 메시지

10.2.1.3. 프로세스의 등록

이제 10 초 마다 S1F1 을 전송하고, S1F2 를 수신하도록 프로세스를 등록할 차례입니다. 이번에 [Process] 탭에서 오른쪽 마우스로 클릭, [프로세스 추가]를 선택합니다. 아래의 {프로세스 편집}창에서 이름을 입력하고, [확인]을 클릭합니다.

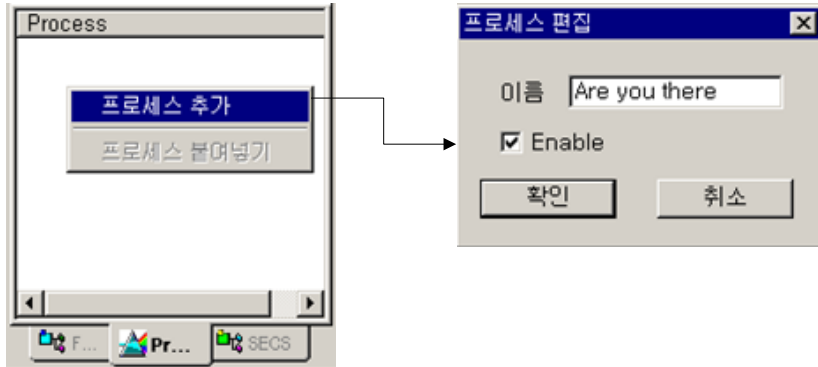


그림 10.2-12 프로세스 등록

아래처럼 프로세스가 하나 등록됩니다. 오른쪽의 {프로세스 편집} 창에서 역시 오른쪽 마우스로 클릭, [아이템 추가]를 선택합니다.

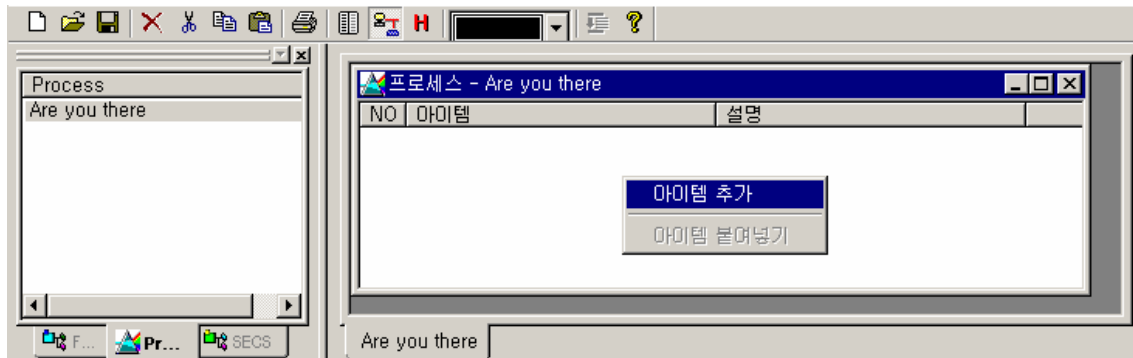


그림 10.2-13 아이템 추가

{아이템 추가}창, <분류>에서 '시작 아이템'을 선택하고, <아이템>에서 'IF_TIMER'를, <주기>엔 "1000" (1000*10mSec = 10Sec)을 입력합니다.

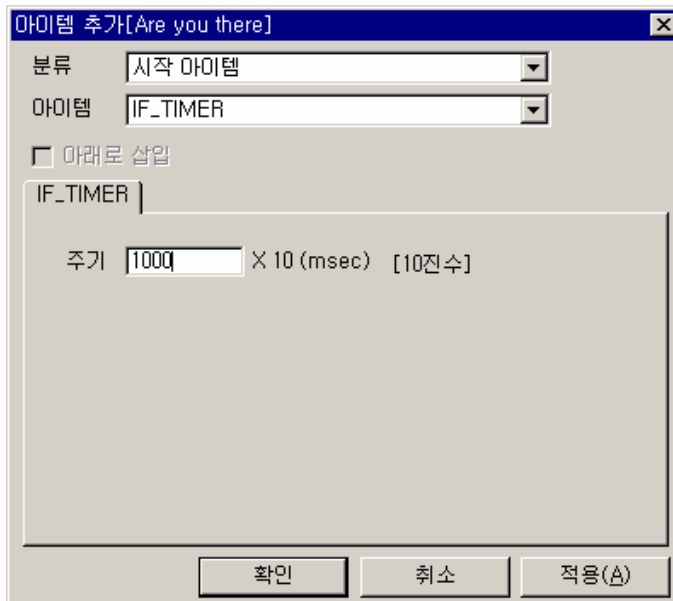


그림 10.2-14 아이템 추가

아래처럼 IF_TIMER 와 END_IF 가 추가됩니다. END_IF 는 자동으로 추가됩니다. IF_TIMER 와 END_IF 사이에 아이템을 추가 하겠습니까. S1F1 을 보내고, S1F2 를 받아야 하므로, IF_SECS_MC_REQ_RESP 아이템을 사용해야 합니다. END_IF 를 선택한 후, 오른쪽 마우스로 클릭 [아이템 삽입]을 선택합니다.

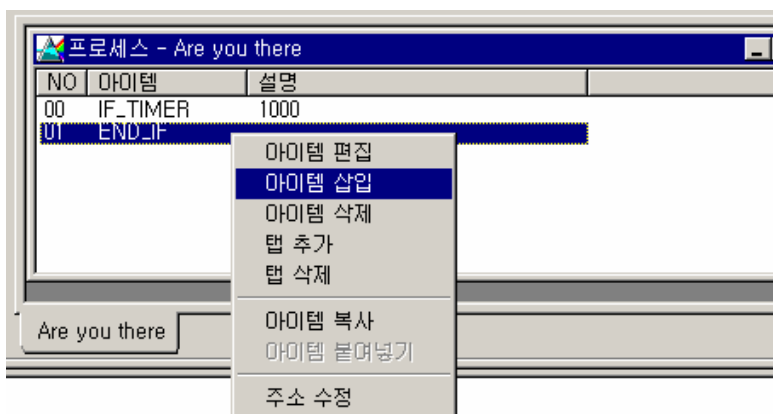


그림 10.2-15 아이템 삽입

{아이템 삽입}창, <분류>에서 'SECS 아이템'을 선택하고, <아이템>에서 IF_SECS_MC_REQ_RESP 를 선택합니다. <요구 메시지>에 "S1F1_SND"를 직접 쳐 넣어도 되고, 아래처럼 좌측의 메시지 정의 창에서 Drag & Drop 으로 <요구 메시지>에 넣어도 됩니다. <응답 메시지>도 비슷하게 "S1F2_RCV"를 넣고, <SECS-I/HSMS>에선 'SECS-I'을 선택하고, <Port>는 COM1 으로 통신할 것이므로, "1"을 넣습니다.

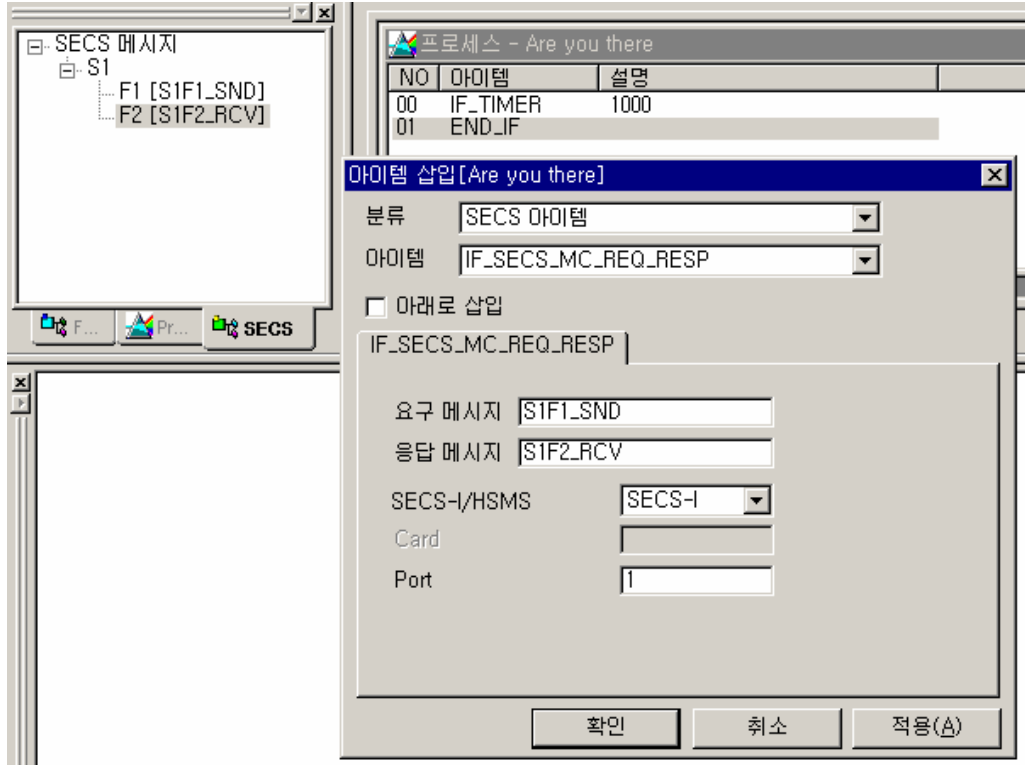


그림 10.2-16 IF_SECS_MC_REQ_RESP

IF_SECS_MC_REQ_RESP 아이템이 추가되었습니다. 역시, IF_XXX 아이템이므로 END_IF 가 자동으로 추가 되었습니다. 키보드에서 [Tab]키를 눌러 아래처럼 들여 쓰기를 하면 보기가 쉽습니다.

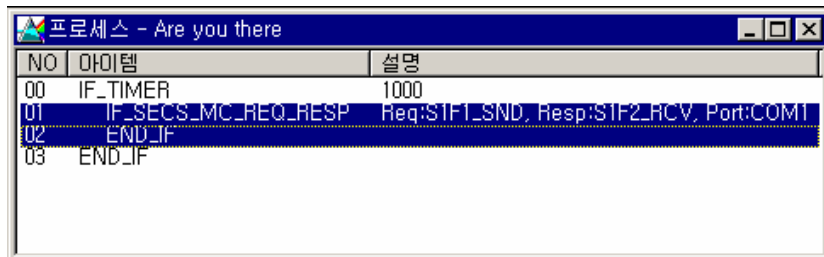


그림 10.2-17 추가된 아이템

10.2.1.4. 다운로드와 테스트

ETOS-PD 의 [온라인][파라미터 쓰기]와 [프로그램 쓰기]를 차례로 수행합니다. 이제 모든 준비가 끝난 것입니다. COM1 에 적절한 호스트가 있어 응답을 준다면, 10 초마다 S1F1 을 보내고, S1F2 를 받아 메모리에 저장 할 것입니다. 메시지의 송수신 상태는 [온라인][프레임 모니터]에서 확인할 수 있습니다. 또한 ETOS 의 동작은 ETOS 의 로그(Log)를 확인하면 이상 유무를 판단 할 수 있습니다. ETOS 의 로그 확인은 별도의 매뉴얼(ETOS 시리즈의 시스템 로그 분석)을 참조하십시오.

10.2.2. S1F1 을 받으면 S1F2 로 응답하기

이번엔 반대로 COM2 포트로 S1F1 을 받으면 S1F2 로 응답하도록 해 보겠습니다.
 앞의 예제에 이어서 계속 하겠습니다.

10.2.2.1. 파라미터 설정

COM2 를 통신 속도는 9600, Type 은 Host 로 하겠습니다. 나머지는 모두 초기값을 그대로 사용하겠습니다.

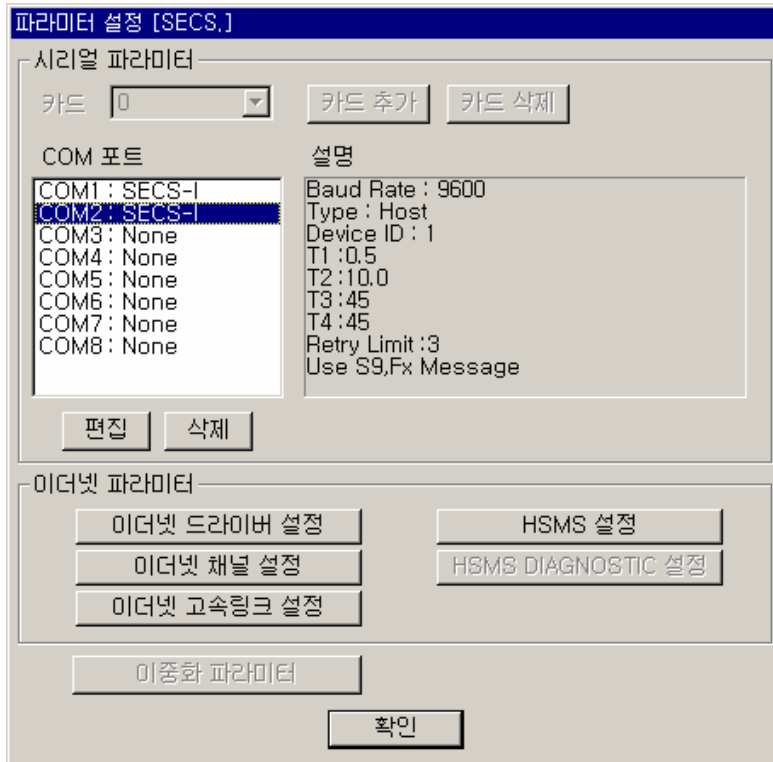


그림 10.2-18 파라미터 설정

10.2.2.2. S1F1, S1F2 메시지의 등록

{SECS 메시지}창에서 <Stream>, <Function>번호를 넣고, <이름>은 “S1F1_RCV’라고 하겠습니까. 표준 Structure 를 그대로 사용할 것이므로 <Structure 파일 참조>를 체크합니다.

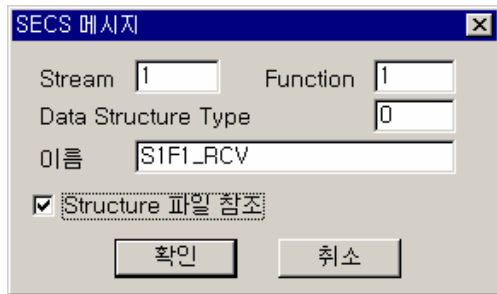


그림 10.2-19 SECS 메시지 추가

[확인]을 누르면 바로 {SECS-II Message Definition}창이 나옵니다. 헤더만 있는 메시지 이므로 수정할 것 없이 바로 [확인]을 누릅니다.

마찬가지로 수신할 S1F2 를 등록하겠습니다. 아래와 같이 설정합니다. 역시 표준 Structure 를 그대로 사용할 것이므로 <Structure 파일 참조>를 체크합니다.

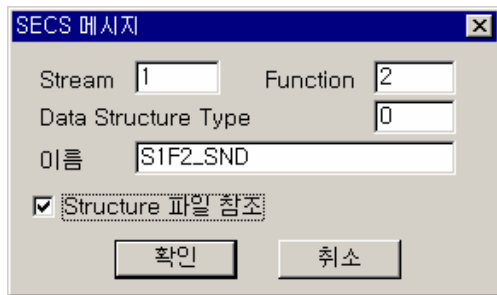


그림 10.2-20 SECS 메시지 추가

첫 번째 LIST 는 수정할 것이 없고, <MDLN>은 아래처럼 ASCII 로 “ETOS”로, <SOFTREV>는 “3.0”으로 입력 하겠습니다.

The screenshot shows the 'SECS-II Message Definition' dialog box. At the top, the 'Message' field contains 'S1F2' and the 'Msg Name' field contains 'On Line Data(D)'. Below this, the 'User Msg. Name' field contains 'S1F2_SND'. There are two groups of radio buttons: 'Block' with 'Single' (selected) and 'Multi', and 'Reply' with 'Reply' and 'Never' (selected). The 'Description' field contains 'Equipment Model Type, 6 bytes max.'. The 'Structure' pane shows a tree view with 'L' expanded, containing 'MDLN' (highlighted) and 'SOFTREV'. The 'Data Item' pane for 'MDLN' shows 'Name' as 'MDLN', 'Format' as 'ASCII', 'Value' as 'ETOS', 'Size' as '4', and 'Offset' as '0'. There are 'Address' and 'Variable' checkboxes, and an 'Address Of Size' field. The 'Exception' pane is empty. At the bottom are 'Edit Address', 'OK', 'Apply', and 'Cancel' buttons.

그림 10.2-21 SECS-II Message Definition

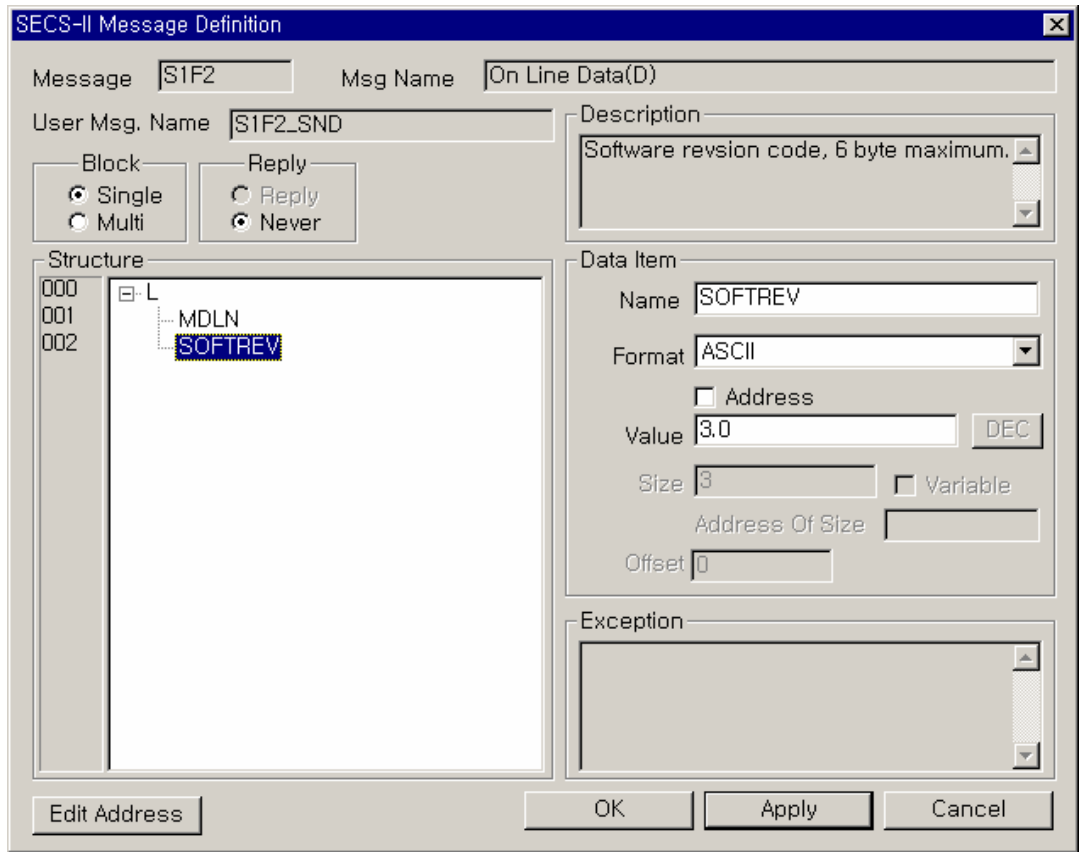


그림 10.2-22 SECS-II Message Definition

아래처럼 두 개의 메시지가 추가 등록되었습니다.

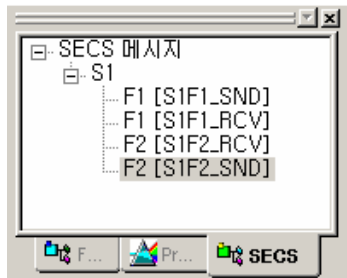


그림 10.2-23 추가된 SECS 메시지

10.2.2.3. 프로세스의 등록

이제 S1F1 을 수신하면, S1F2 를 송신하는 프로세스를 등록할 차례입니다. 이름을 “I am here”로 하겠습니다.

아래의 {프로세스 편집}창에서 이름을 입력하고, [확인]을 클릭합니다.

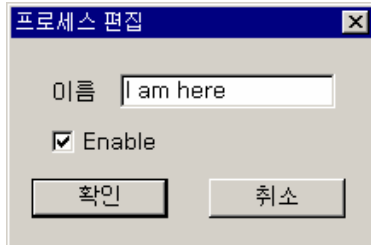


그림 10.2-24 프로세스 편집

아래처럼 IF_SECS_MC_FRAME_RCV 로 시작하겠습니다.

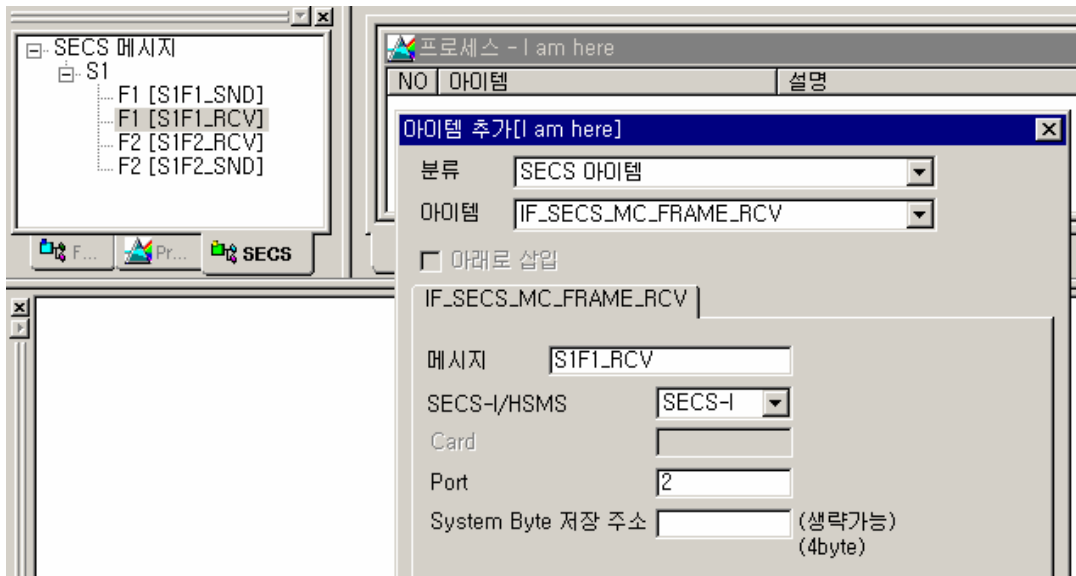


그림 10.2-25 IF_SECS_MC_FRAME_RCV

아래처럼 IF_SECS_MC_FRAME_RCV 와 END_IF 사이에 SECS_MC_FRAME_SND 아이템을 추가합니다.

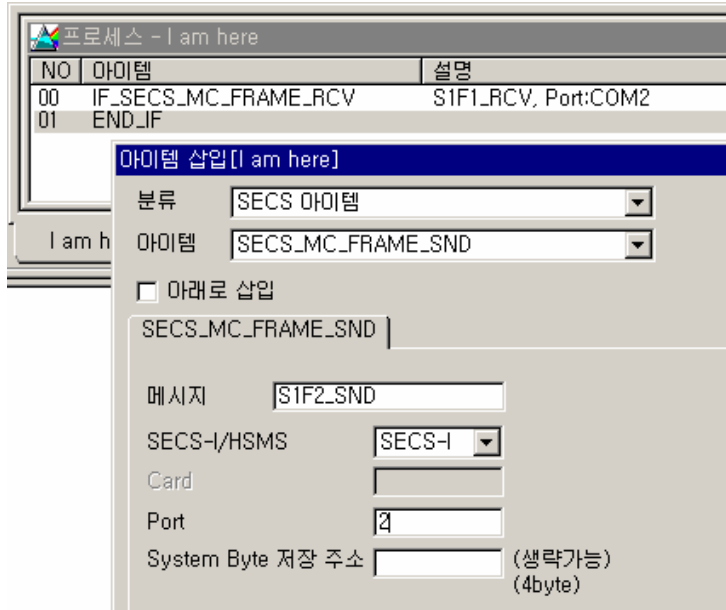


그림 10.2-26 SECS_MC_FRAME_SND

아래처럼 프로세스가 준비되었습니다.

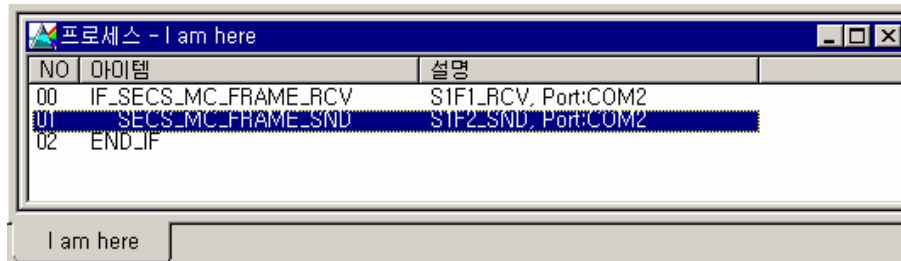


그림 10.2-27 등록된 프로세스

10.2.2.4. 다운로드와 테스트

ETOS-PD 의 [온라인][파라미터 쓰기]와 [프로그램 쓰기]를 차례로 수행합니다. 앞의 예제와 함께 COM1 은 “Equipment”로, COM2 는 “Host”로 동작하게 됩니다. 따라서 두 포트를 서로 연결하면 COM1 과 COM2 가 서로 응답하게 됩니다.

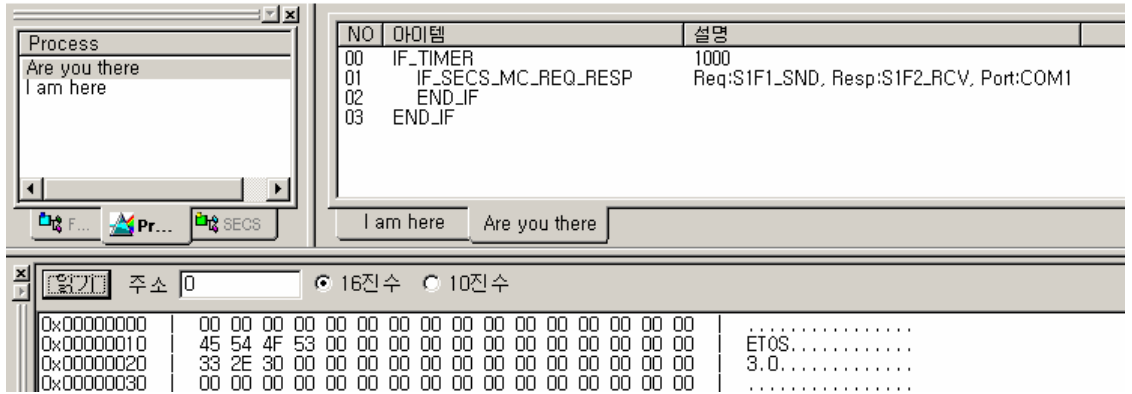


그림 10.2-28 ETOS 메모리

다음은 COM1 포트를 메시지 모니터 한 화면입니다.

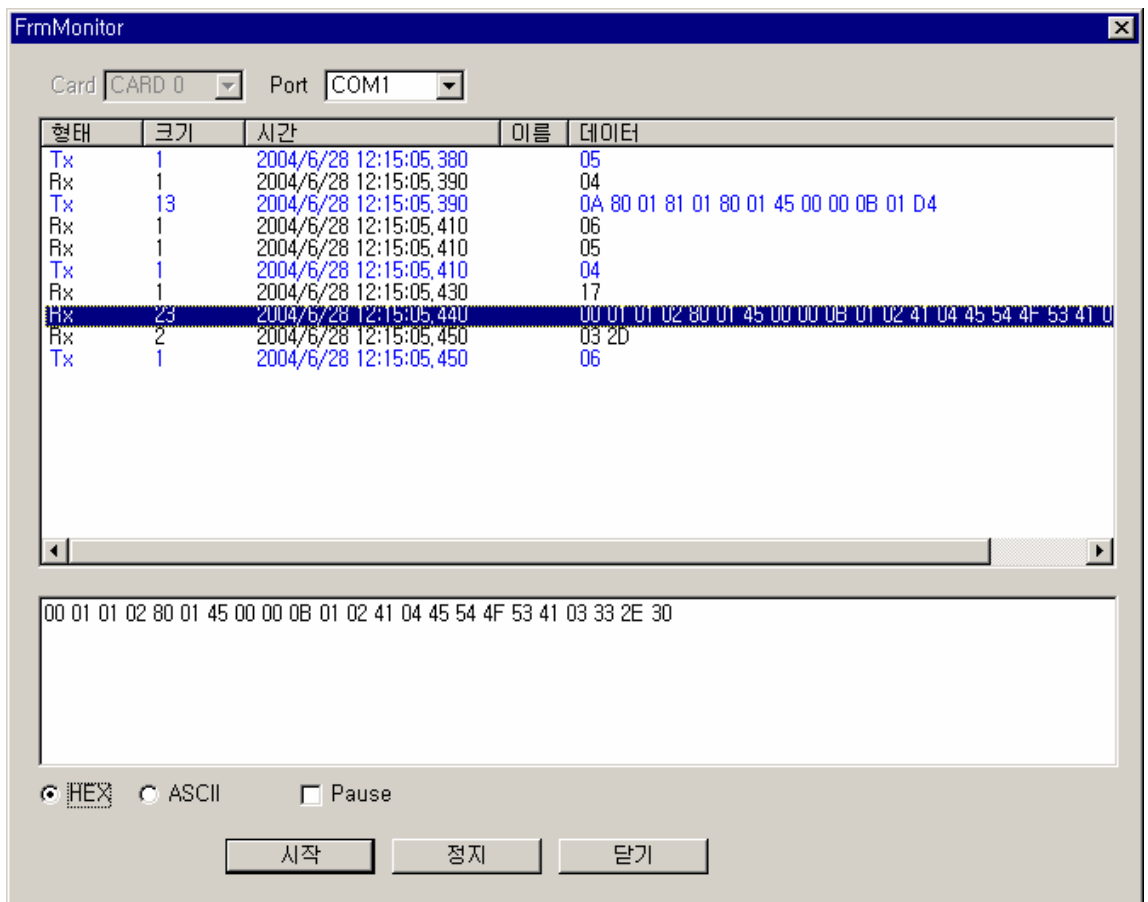


그림 10.2-29 프레임 모니터

10.2.3. S1F1 을 받는 프로세스와 S1F2 로 응답하는 프로세스 분리하기

이번에는 S1F1 을 받는 프로세스와 S1F2 로 응답하는 프로세스를 분리해 보겠습니다. 어떤 Request 를 받으면, 다른 채널로 데이터를 수집해야 한다든지, 연산을 많이 해야 한다든지 하는 경우, 하나의 프로세스가 너무 길어 질 수 있습니다. 따라서 프로세스를 분리할 수 있습니다. 파라미터와 메시지는 앞의 예제와 동일하고, COM2 의 “I am here” 프로세스를 둘로 나눌 것입니다.

10.2.3.1. 프로세스의 수정

먼저 “I am here” 프로세스를 복사해서, “S1F1_RCV_PRS”와 “S1F2_SND_PRS”로 이름을 할당합니다. 아래처럼 “I am here” 프로세스를 복사해서 붙여 넣습니다.

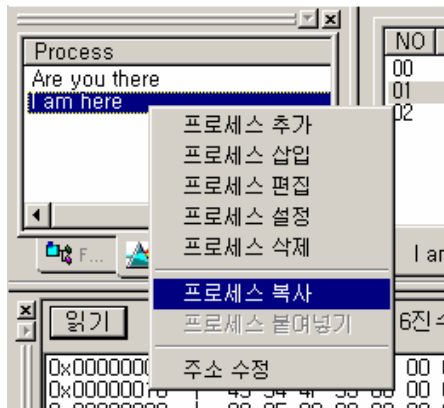


그림 10.2-30 프로세스 복사

이 때 “붙여 넣기”에서 이름 변경을 요구합니다. “S1F1_RCV_PRS”와 “S1F2_SND_PRS” 변경합니다.

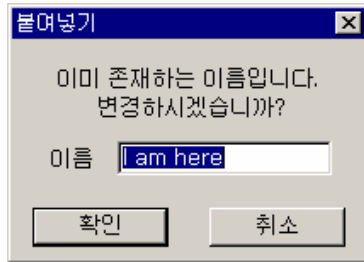


그림 10.2-31 프로세스 이름 변경

사용하지 않는 “I am here” 프로세스를 Disable 시키겠습니다. 아래처럼 [프로세스 편집]을 선택한 후

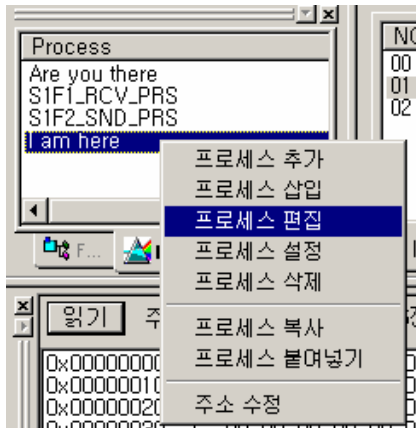


그림 10.2-32 프로세스 편집

<Enable> 체크를 해제합니다.

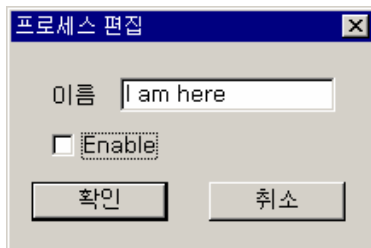


그림 10.2-33 프로세스 Disable

그림처럼 “S1F1_RCV_PR”를 더블 클릭해서, 오른쪽 {프로세스 편집} 창으로 띄우고, 필요없는 SECS_MC_FRAME_SND 아이템을 삭제합니다.

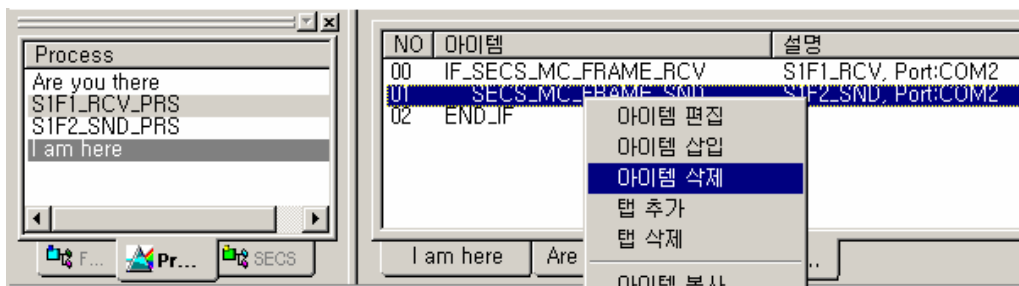


그림 10.2-34 아이템 삭제

IF_SECS_MC_FRAME_RCV 아이টে을 더블 클릭해서 <System Byte 저장 주소>에 0 번지를 할당합니다. 이제 S1F1 을 수신하면, 해당 시스템 바이트는 0 번지에 저장됩니다.

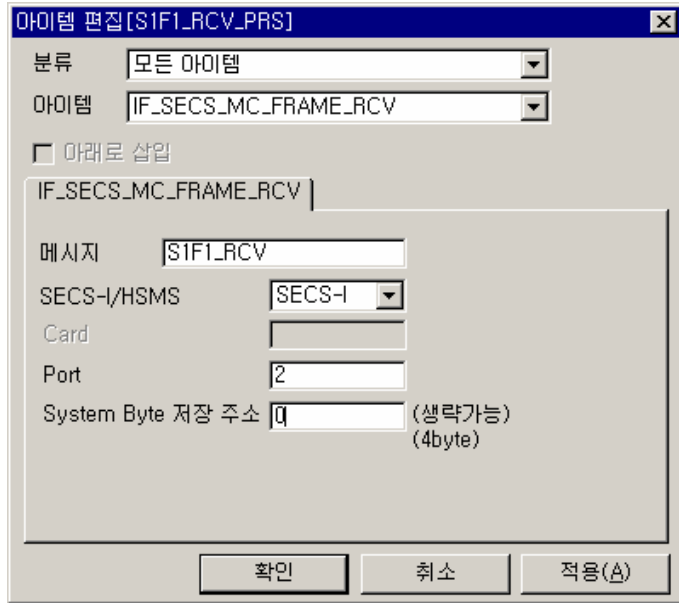


그림 10.2-35 IF_SECS_MC_FRAME_RCV 편집

“S1F2_SND_PRS” 에서 S1F2 를 보내기 전에 S1F1 이 여러 개 도착하게 되면 System Byte 가 겹쳐 써져 먼저 온 것은 없어지게 됩니다. 이를 막기 위해선 수신 프로세스가 송신이 끝날 때까지 종료되지 않도록 해줘야 합니다.

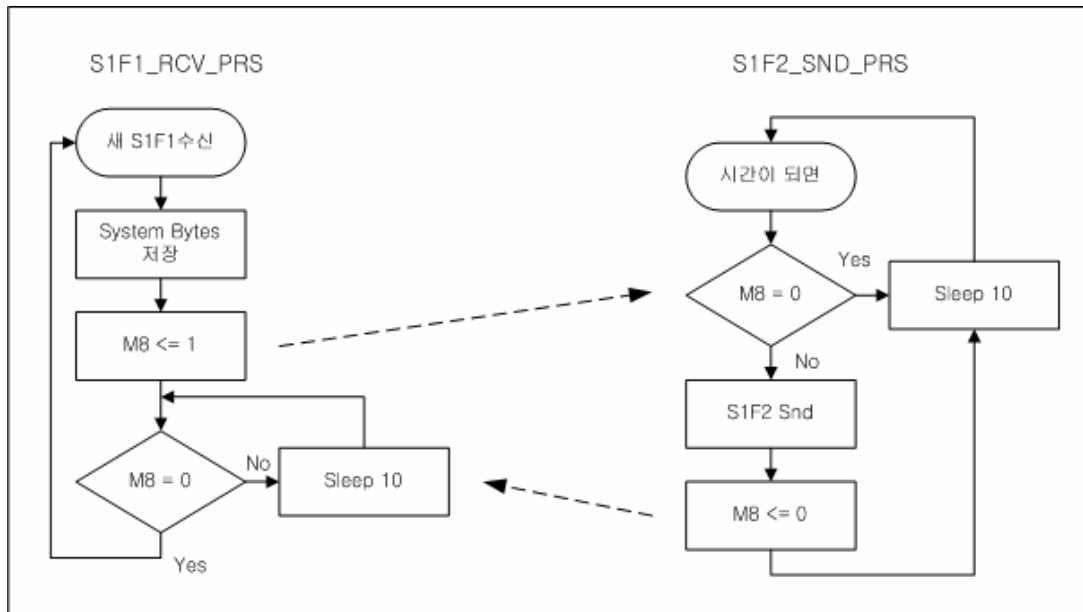


그림 10.2-36 순서도

따라서 아래처럼 아이템을 추가합니다.

NO	아이템	설명
00	IF_SECS_MC_FRAME_RCV	S1F1_RCV, Port:COM2, SysByteAddr:M0
01	REMARK	메모리에 도착했다고 쓰고
02	MEM_SET	T:M8, Size:1, Type:16진수, Data:01
03	LABEL	wait_snd
04	REMARK	송신할 때 까지 기다린다.
05	IF_NUM_CMP	Type:S1, 0, <>, M8
06	DELAY	10
07	JUMP	wait_snd
08	END_IF	
09	END_IF	

I am here Are you there S1F1_RCV...

그림 10.2-37 S1F1_RCV_PRS

앞의 순서도를 참조해서 “S1F2_SND_PRS”도 아래처럼 프로그램 합니다.

SECS_MC_FRAME_SND의 <System Byte 저장 주소>에 0 번지를 입력, 수신에서 저장한 것을 그대로 사용하게 됩니다.

NO	아이템	설명
00	IF_TIMER	10
01	REMARK	프레임이 도착했으면
02	IF_NUM_CMP	Type:S1, 1, =, M8
03	SECS_MC_FRAME_SND	S1F2_SND, Port:COM2, SysByteAddr:M0
04	REMARK	다 보냈다고 메모리에 쓰고
05	MEM_SET	T:M8, Size:1, Type:16진수, Data:00
06	END_IF	
07	END_IF	

I am here Are you there S1F1_RCV... S1F2_SND...

그림 10.2-38 S1F2_SND_PRS

10.2.3.2. 다운로드와 테스트

ETOS-PD의 [온라인][파라미터 쓰기]와 [프로그램 쓰기]를 차례로 수행합니다. 앞의 예제와 동일하게 COM1은 “Equipment”로, COM2는 “Host”로 동작하게 됩니다. 따라서 두 포트를 서로 연결하면 COM1과 COM2가 서로 응답하게 됩니다.

The screenshot shows the ETOS software interface. On the left, a 'Process' window lists 'Are you there', 'S1F1_RCV_PRS', 'S1F2_SND_PRS', and 'I am here'. On the right, a table shows the program logic for S1F2_SND_PRS, with item 03 highlighted. Below, a memory dump window shows hexadecimal addresses and their corresponding data values.

NO	아이템	설명
00	IF_TIMER	10
01	REMARK	프레임이 도착했으면
02	IF_NUM_CMP	Type:S1, 1, =, M8
03	SECS_MC_FRAME_SND	S1F2_SND, Port:COM2, SysByteAddr:M0
04	REMARK	다 보냈다고 메모리에 쓰고
05	MEM_SET	T:M8, Size:1, Type:16진수, Data:00
06	END_IF	
07	END_IF	

I am here Are you there S1F1_RCV... S1F2_SND...

주소	16진수	10진수
0x00000000	45 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00	E.....
0x00000010	45 54 4F 53 00 00 00 00 00 00 00 00 00 00 00 00	ETOS.....
0x00000020	33 2E 30 00 00 00 00 00 00 00 00 00 00 00 00 00	3.0.....

그림 10.2-39 ETOS 메모리

10.3. HSMS 통신

10.3.1. 10 초마다 S1F1 을 송신하고, S1F2 를 받기

앞의 예제를 HSMS 로 변경해 보겠습니다.

10.3.1.1. 파라미터의 설정

HSMS 포트를 추가하겠습니다.

ETOS-PD 의 [편집] -> [파라미터 설정]을 클릭합니다.

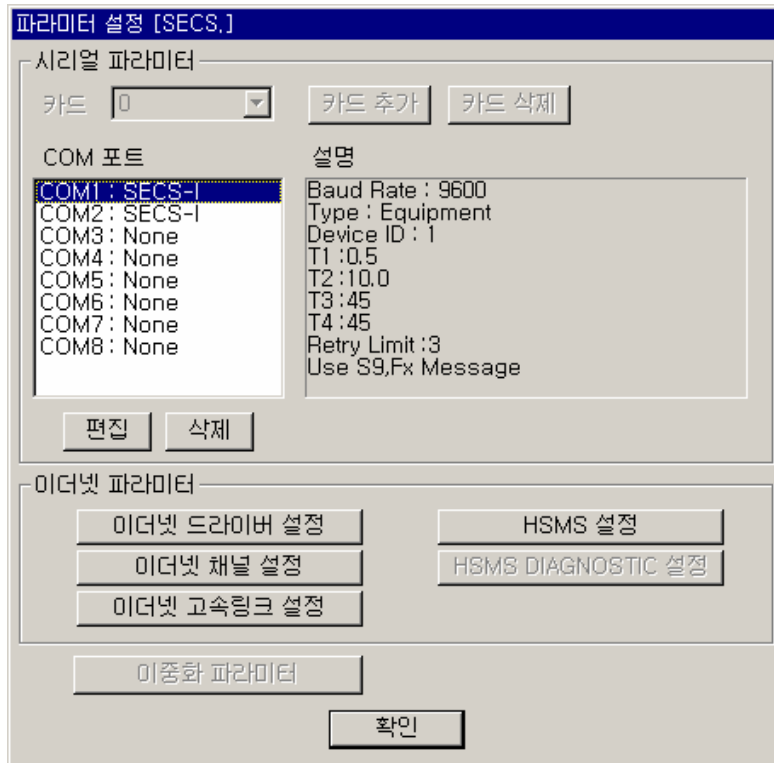


그림 10.3-1 시스템 설정

이더넷 파라미터에서 [HSMS 설정]을 선택합니다.

[HSMS 설정]에서 추가 버튼을 클릭합니다.

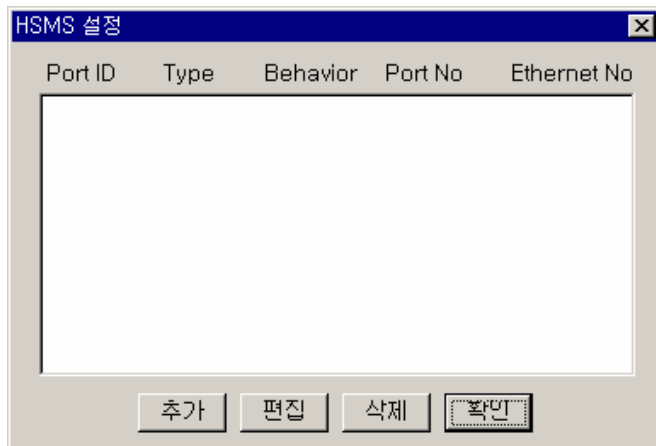


그림 10.3-2 HSMS 설정

{HSMS Port 설정}을 아래와 같이 합니다. 모두 default 값입니다. 자주 변경하게 되는 부분은 <Ethernet No.>, <Type>, <Device ID>, <Behavior>, <Passive Entity IP Address>, <Passive Entity Port No.>등입니다.

Ethernet No.	Ethernet 1
HSMS Port ID	0
Type	Equipment
Device ID	1
Protocol	HSMS-SS
Behavior	Passive
Passive Entity IP Address	0.0.0.0
Passive Entity Port No.	5000
T3 (Reply Time Out)	45 (sec)
T5 (Connect Separation Timeout)	10 (sec)
T6 (Control Transaction Timeout)	5 (sec)
T7 (NOT SELECTED Timeout)	10 (sec)
T8 (Network Intercharcter Timeout)	5 (sec)
CE (Connection Establishment Timer)	10 (sec)
LT (LinkTest Msg. Tx. Timer)	15 (sec)
Retry Limit	3
Connection Number	1
<input checked="" type="checkbox"/> Use S9,Fx Message	

그림 10.3-3 HSMS Port 설정

[확인]을 클릭하면 아래처럼 HSMS 포트 0 번이 추가됩니다.

Port ID	Type	Behavior	Port No	Ethernet No
00	Equip.	Passive	5000	1

그림 10.3-4 HSMS 설정

10.3.1.2. S1F1, S1F2 메시지의 등록

메시지의 등록은 앞의 예제와 동일합니다.

10.3.1.3. 프로세스의 등록

역시 대동소이합니다. 앞의 프로세스를 수정해 보겠습니다. 아래처럼 “Are you there”의 두 번째 아이템인 IF_SECS_MC_REQ_RESP 를 더블 클릭해서 <SECS-I/HSMS>를 ‘HSMS’로, <Port>를 “0”으로 변경합니다.

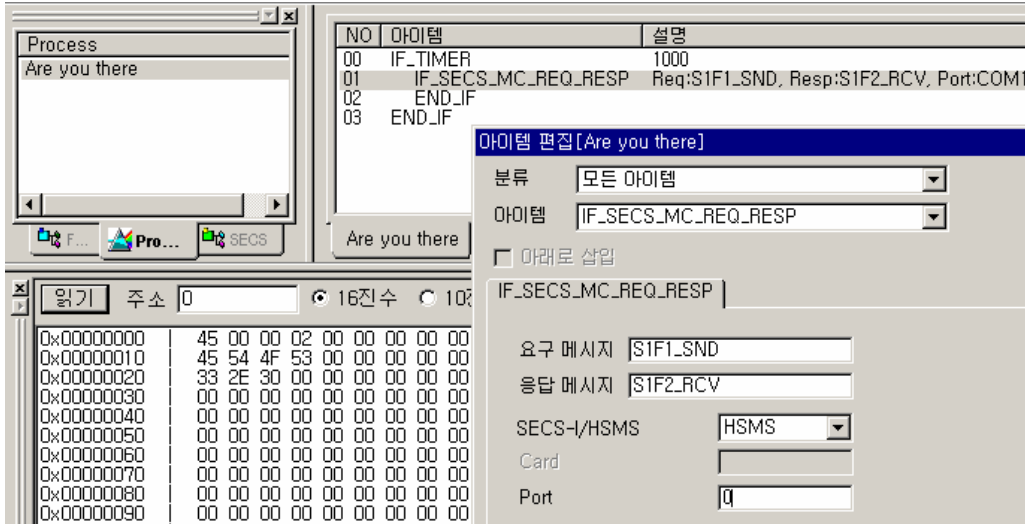


그림 10.3-5 프로세스의 수정

10.3.1.4. 다운로드와 테스트

ETOS-PD 의 [온라인][파라미터 쓰기]와 [프로그램 쓰기]를 차례로 수행합니다. 이 프로그램을 테스트하려면 Active HSMS Host 가 있어야 합니다. 별도의 호스트나, 시뮬레이터, 또는 2 대의 ETOS 를 이용해서 테스트 할 수 있을 것입니다.

10.4. Bridge 통신

10.4.1. 시스템의 구성

아래의 그림은 일반적인 SECS-I 과 HSMS 간의 Bridge 장비를 나타냅니다. ETOS 장비를 통해서 상위 호스트는 SECS-I 만 지원되는 장비를 HSMS 가 지원되는 장비로 인식하고, 핸들링 할 수 있게 됩니다.

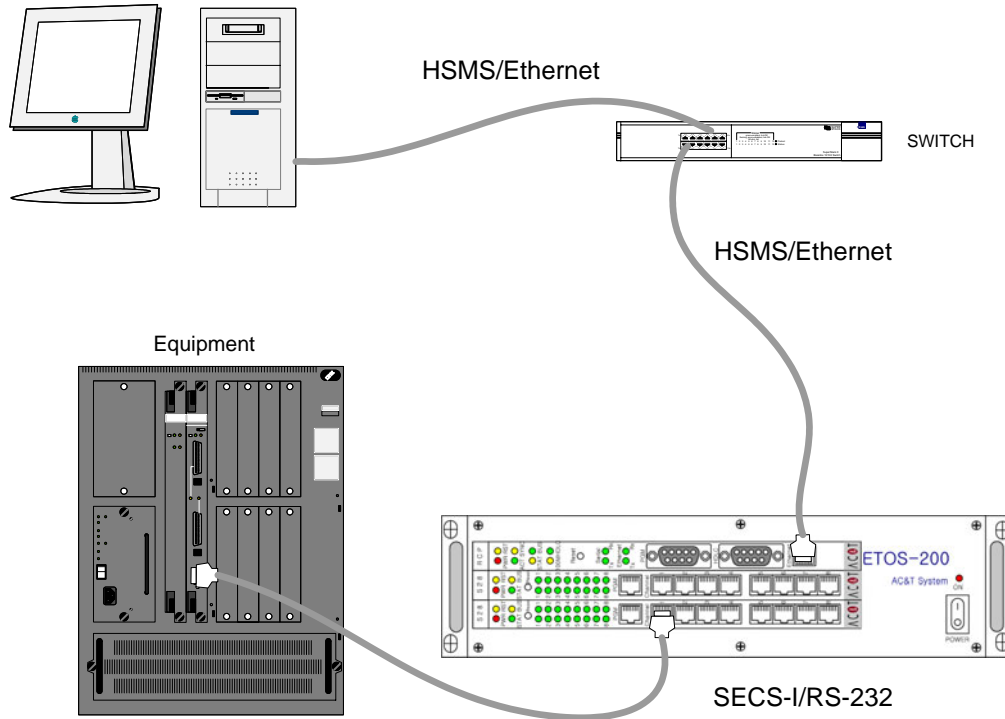


그림 10.4-1 Bridge 통신

조금 더 개념적으론 아래처럼 표현할 수 있습니다.

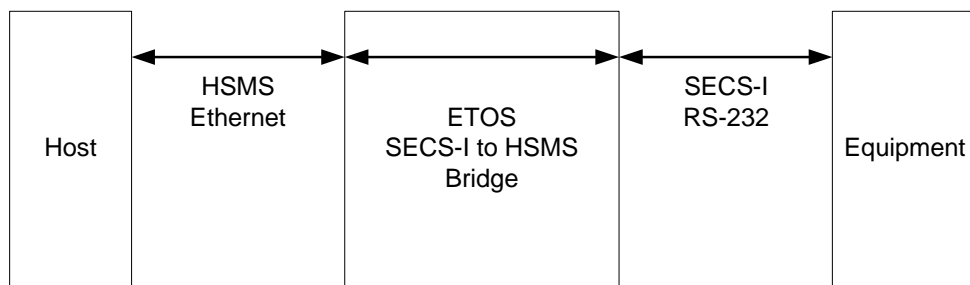


그림 10.4-2 Bridge 통신의 개념

ETOS 는 가운데에서 Ethernet 과 RS-232 의 전기적 물리적인 Medium 을 바꿔줄 뿐 만 아니라, SECS-I 의 Multi-Block 을 조립하거나, 거꾸로 만들고, Open Transaction Message 와 Interleaving Message 에 대응하도록 설계되어 있습니다.

ETOS-200 의 경우, 한 장비에서 HSMS 16 채널, SECS-I 16 채널을 동시에 Bridging 할 수 있습니다.

10.4.2. ETOS-PD 를 이용한 파라미터의 설정

Bridge 통신은 메시지와 프로세스의 정의 없이, 파라미터 만으로 통신을 합니다.

ETOS-PD V2.3 부터는 프로그램 시작시에 자동으로 새파일을 열도록 되어 있습니다. (이전 버전 사용자는 [파일] -> [ETOS TYPE 변경], [파일] -> [새 파일] 메뉴를 차례로 실행해야 합니다.)

ETOS-PD 를 실행하고, 아래처럼 <ETOS 종류>를 'ETOS-200'으로, <드라이버>를 'SECS'로 선택합니다.

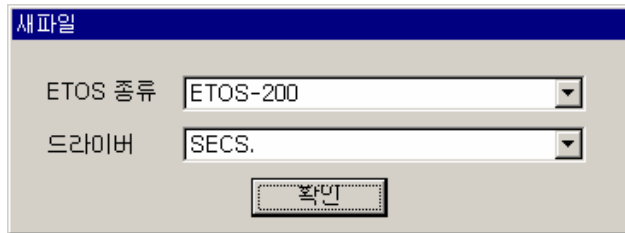


그림 10.4-3 ETOS 종류 및 드라이버 선택

연이어 나오는 {파라미터 설정(SECS)}에서 각 파라미터를 설정합니다.

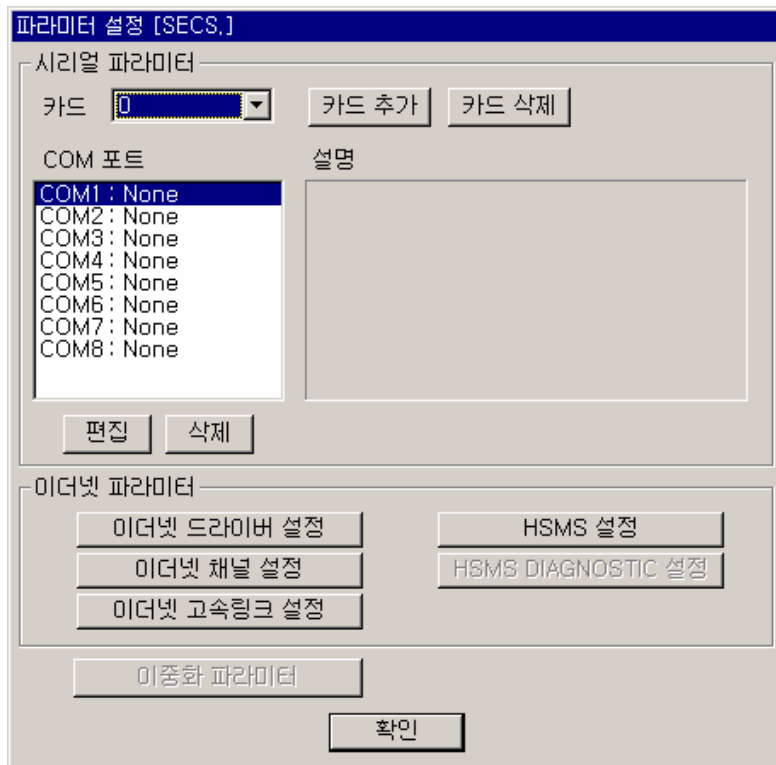


그림 10.4-4 파라미터 설정

먼저 이더넷 파라미터의 [HSMS 설정]을 클릭하여, HSMS Port 를 추가 하십시오.
{HSMS 설정} 창에서 [추가]를 누릅니다.

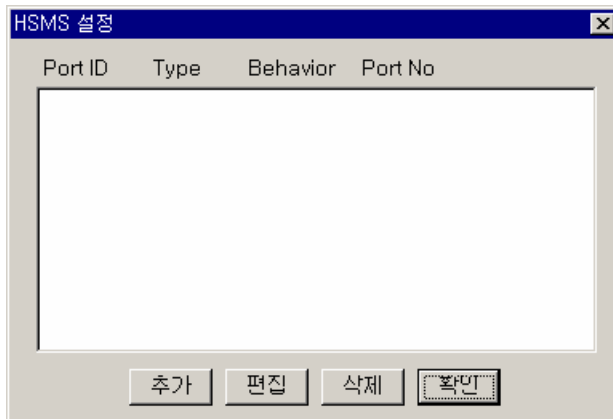


그림 10.4-5 HSMS 설정

{HSMS Port 설정} 창에서

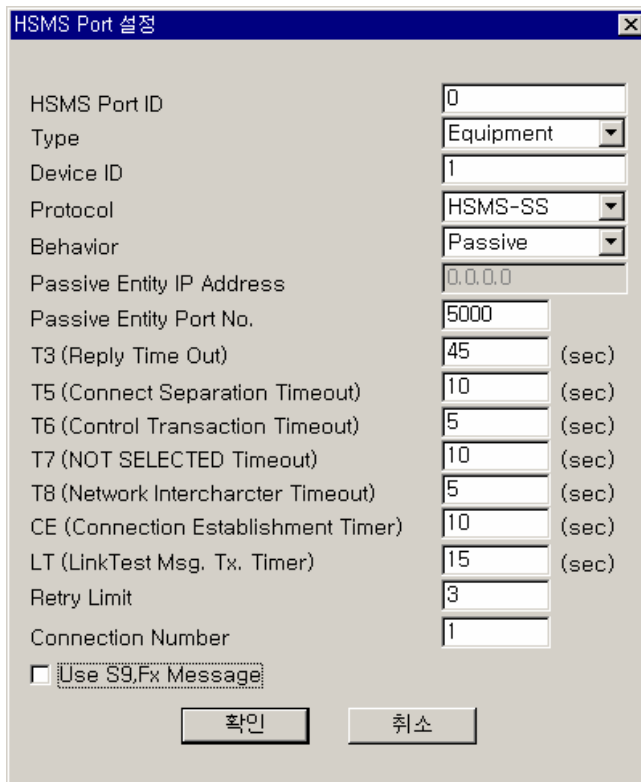


그림 10.4-6 HSMS Port 설정

<Type>, <Behavior>, <Passive Entity IP Address>, <Passive Entity Port No. >, <Timeout>과 <Retry Limit>를 상황에 맞게 설정하십시오. HSMS Port ID 는 0 번부터 자동으로 부여되는데, 중복되지 않게 수정이 가능하고, 이것을 이용해서 SECS-I 과 연결됩니다. <Device ID>, <T3>, <Connection Number>, <Use S9Fx Message>항목은 Bridge 에서 사용되지 않는 항목입니다. Bridge 로 사용되는 경우는 ETOS 가 S9Fx 를 발생시키지 않는 것이 좋습니다. 맨 아래의 <Use S9Fx Message>는 체크하지 않습니다.

[확인]을 눌러 {파라미터 설정}까지 빠져 나옵니다.

이번엔 SECS-I 을 설정합니다.

{파라미터 설정}창의 좌측 COM 포트에서 COM1: NONE 을 선택한 후, [편집]을 클릭합니다. 또 는 더블 클릭합니다.

{포트 종류 선택} 창에서 'SECS-I'을 선택합니다.

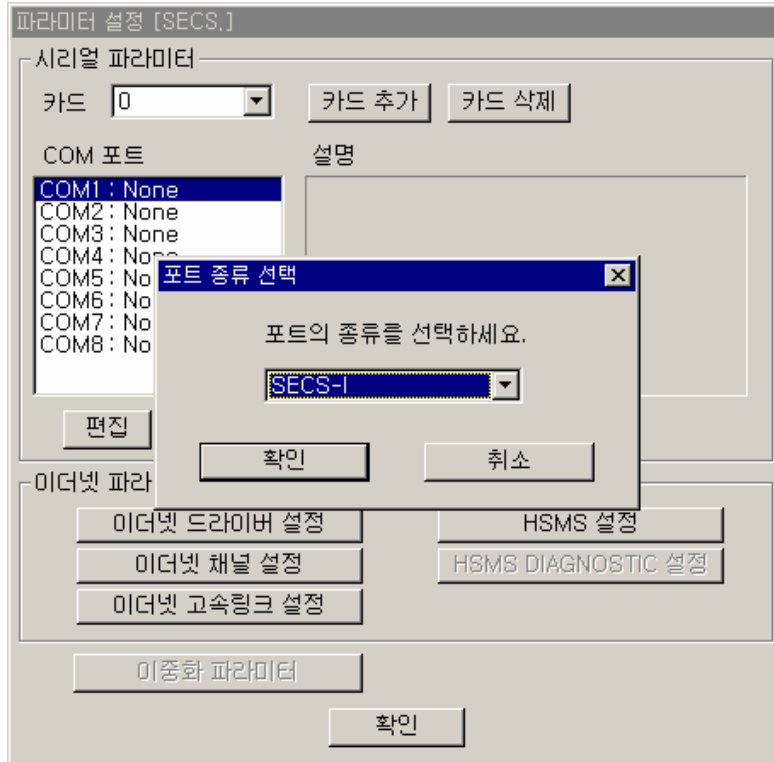


그림 10.4-7 SECS-I 포트 추가

{COM 포트 설정(SECS-I)} 창에서 <통신속도>, <Timeout>, <Retry Limit>등을 설정하고, <SECS-I/HSMS Bridge>의 체크박스를 선택하고, <HSMS PORT ID>에 이 SECS-I 포트와 연결할 HSMS 포트 ID 를 설정합니다.

이 예제에선 COM1 과 HSMS Port ID 0 번이 서로 연결되었습니다. 이제 ETOS 에 다운로드 하면 됩니다.

그림 10.4-8 SECS-I 설정

[온라인]/[파라미터 쓰기]를 선택하십시오. 아직 PC 에 저장하지 않았다면 적당한 이름을 주고, 저장한 후에 다운로드 됩니다.

10.5. 진단 시스템 통신

ETOS 시리즈 중 일부는 반도체 장비와 호스트간의 통신을 모니터하고, 진단하기 위한 시스템으로 활용할 수 있습니다.

10.5.1. SECS1 을 HSMS 로 모니터와 진단하기

10.5.1.1. 기본 개념

물리적으로 아래의 그림처럼 장비와 호스트는 SECS-I으로, 진단 장비와는 HSMS 로 연결됩니다.

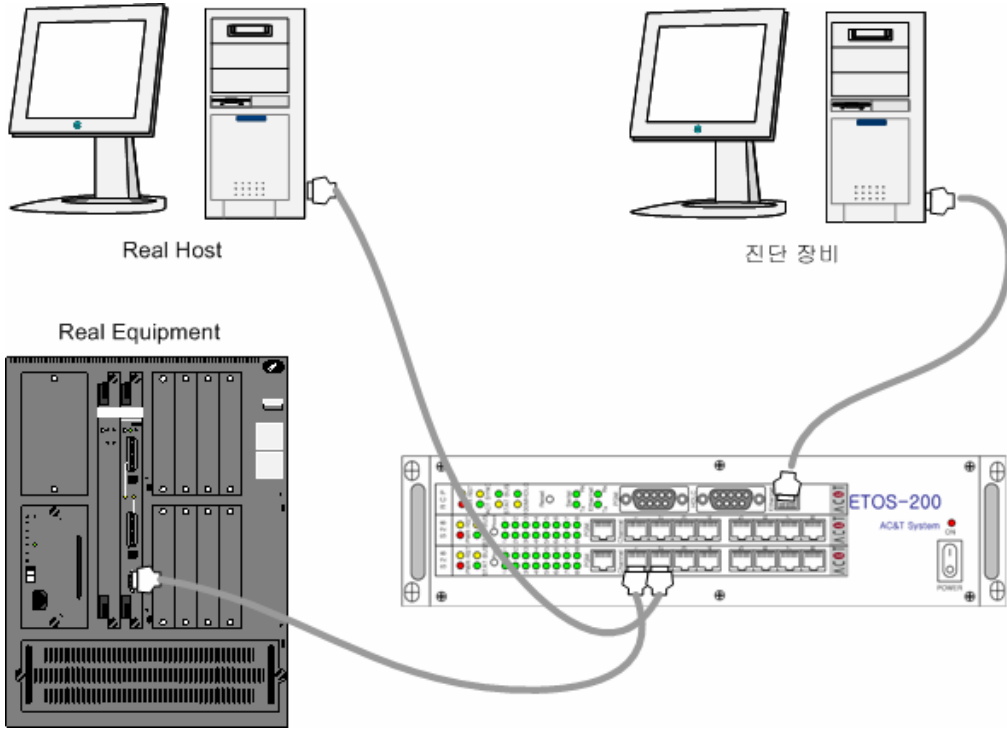


그림 10.5-1 SECS-I 시스템의 진단

논리적으로 아래의 그림처럼 구성됩니다.

진단 장비가 가동 전에는 1번 Path를 통해서 호스트와 장비가 SECS-I으로 통신합니다. 진단 장비가 연결되면 1번 Path를 끊고 2,3번 Path를 통해서 메시지를 전달합니다. 이 때 1번 Path를 통해서 통신하는 경우를 정상 상태로 정의하고, 2,3번 Path를 통해서 통신하는 경우를 진단 상태라고 하겠습니다.

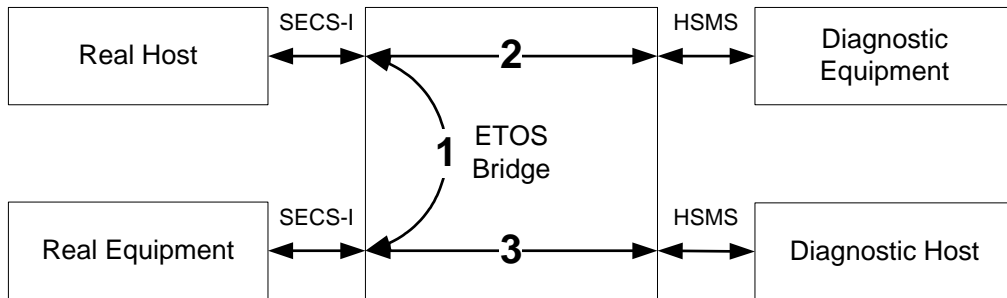


그림 10.5-2 SECS-I 시스템의 진단 개념

10.5.1.2. 상태의 천이

정상 상태에서 진단 상태로의 천이는 Diagnostic Host 나 Diagnostic Equipment 가 연결되어서 Select 상태가 된 후, Diagnostic Host 나 Diagnostic Equipment 에서 S9Fx 나 Link Test 를 제외한 메시지가 나오면 3 번 Path 와 2 번 Path 가 연결됩니다.

반대로 진단 상태에서 정상 상태로의 천이는 Diagnostic Host 나 Diagnostic Equipment 의 연결이 끊어지거나, 진단 시스템이 2 번 연속으로 T3 동안 Reply 하지 않으면, 해당 채널이 끊어지고, 정상 상태로 돌아갑니다. 예를 들면, Diagnostic Host 가 2 번 연속해서 T3 동안 Wait Bit 가 있는 Primary 메시지에 대해 응답하지 않으면 3 번 Path 가 끊어지고, 정상 상태로 돌아갑니다.

10.5.1.3. ETOS-PD의 설정

ETOS-PD 의 설정은 Bridge 통신의 [10.4.2. ETOS-PD 를 이용한 파라미터의 설정](#)과 동일합니다. 다만 이러한 채널을 두 개씩 쌍으로 설정해야 하고, 반드시 COM1 과 COM2, COM3 와 COM4... 는 쌍을 이뤄야 합니다. HSMS 쪽의 호스트, 장비, Active, Passive 설정은 무관합니다.

10.5.2. HSMS 를 HSMS 로 모니터와 진단하기

10.5.2.1. 기본 개념

물리적으로 아래의 그림처럼 허브나 스위치를 중심으로 모두 동일하게 수평적으로 연결되겠지만

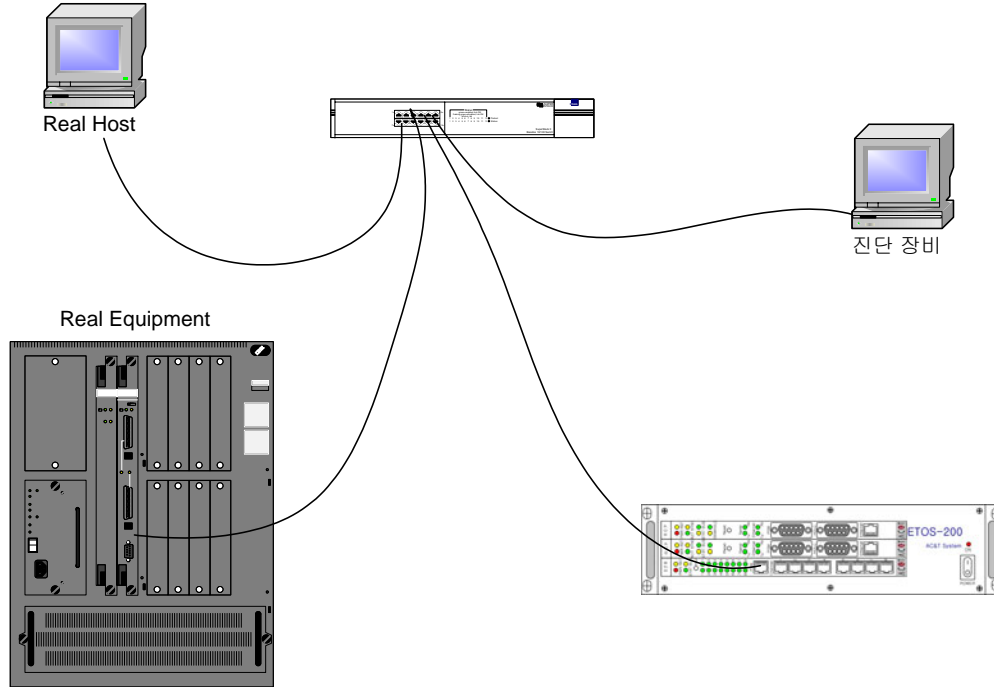


그림 10.5-3 HSMS 시스템의 진단

논리적으로 아래처럼 구성됩니다.

진단 장비가 가동 전에는 1번 Path를 통해서 호스트와 장비가 통신합니다. 진단 장비가 연결되면 1번 Path를 끊고 2,3번 Path를 통해서 메시지를 전달합니다. 이 때 1번 Path를 통해서 통신하는 경우를 정상 상태로 정의하고, 2,3번 Path를 통해서 통신하는 경우를 진단 상태라고 하겠습니다.

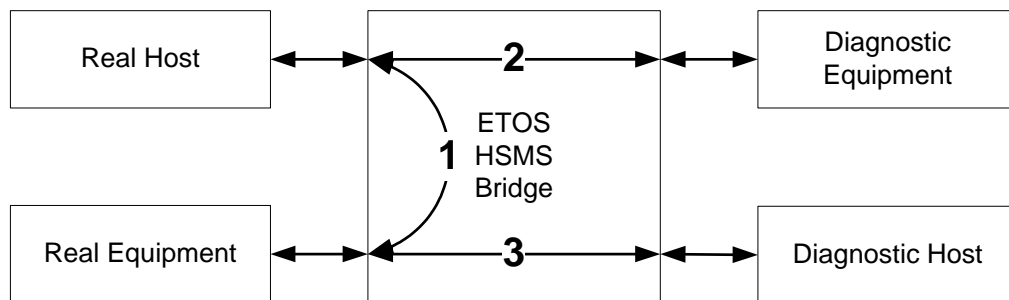


그림 10.5-4 HSMS 시스템의 진단 개념

10.5.2.2. 상태의 천이

정상 상태에서 진단 상태로의 천이는 사용자의 설정에 따라

- ▶ **Diagnostic Host Priority**
- ▶ **Diagnostic Equipment Priority**
- ▶ **And**
- ▶ **Or**

4 가지가 있습니다.

Diagnostic Host Priority 는 Diagnostic Host 가 연결되어서 Select 상태가 된 후, Diagnostic Host 에서 S9Fx 나 Link Test 를 제외한 메시지가 나오면 3 번 Path 와 2 번 Path 가 연결됩니다.(이 때 2 번 Path, Diagnostic Equipment 의 상태는 무시됨을 주의해야 합니다.)

Diagnostic Equipment Priority 는 Diagnostic Equipment 가 연결되어서 Select 상태가 된 후, Diagnostic Equipment 에서 S9Fx 나 Link Test 를 제외한 메시지가 나오면 3 번 Path 와 2 번 Path 가 연결됩니다.(이 때 2 번 Path, Diagnostic Host 의 상태는 무시됨을 주의해야 합니다.)

And 는 Host/Equipment 모두 연결된 후 메시지가 나와야 합니다.

Or 는 Host/Equipment 상관없이 어느 한 쪽만 되면 2,3 번 Path 가 열립니다.

반대로 진단 상태에서 정상 상태로의 천이는 위의 조건이 깨지면 정상상태로 돌아 갑니다. 또한 진단 시스템이 2 번 연속으로 T3 동안 Reply 하지 않으면, 해당 채널이 끊어 집니다. 예를 들면, Diagnostic Host 가 2 번 연속해서 T3 동안 Wait Bit 가 있는 Primary 메시지에 대해 응답하지 않으면 3 번 Path 가 끊어 집니다. 이 후는 사용자의 설정에 따라 위의 Rule 을 적용해서 상태가 결정됩니다. 즉 Diagnostic Host Priority 나 AND 로 설정했으면, 정상 상태로 돌아 갑니다. 그러나 Diagnostic Equipment Priority 나 Or 로 설정했으면, 2 번 채널의 상태에 따라 결정됩니다.

10.5.2.3. ETOS-PD의 설정

ETOS-PD의 {파라미터 설정}창에서 [HSMS Diagnostic 설정]을 선택합니다. (*153)

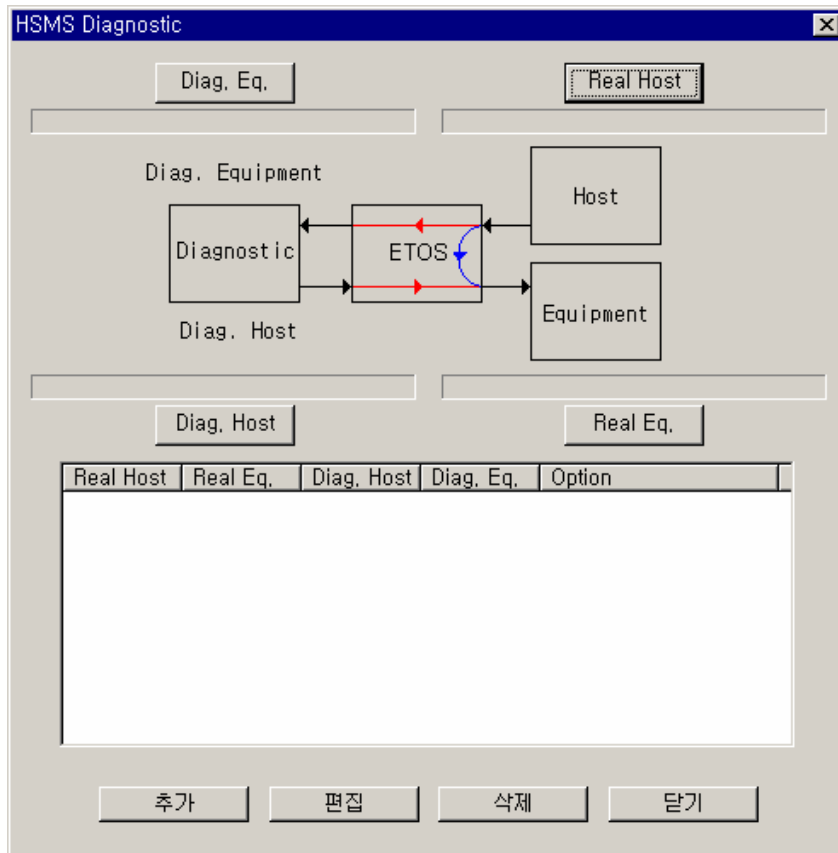


그림 10.5-5 HSMS Diagnostic

[추가]버튼을 누르고



그림 10.5-6 HSMS Diagnostic Info

이미 HSMS Port를 설정했으면 HSMS Port의 ID를 써주거나, 오른쪽의 [...]버튼을 눌러서 여기서 HSMS Port를 설정할 수 있습니다.

하단의 <Connection Option>을 선택하시고, 다운로드하면 됩니다.

(*153) [옵션] -> [편집 옵션]에서 HSMS Diagnostic 편집을 체크해야 [HSMS Diagnostic 설정] 버튼이 활성화됩니다.

10.5.2.4. 상태 모니터

ETOS-PD 의 [온라인] -> [HSMS Diagnostic Status 읽기]에서 아래 그림처럼 연결 상태와 현재 상태를 확인할 수 있습니다.

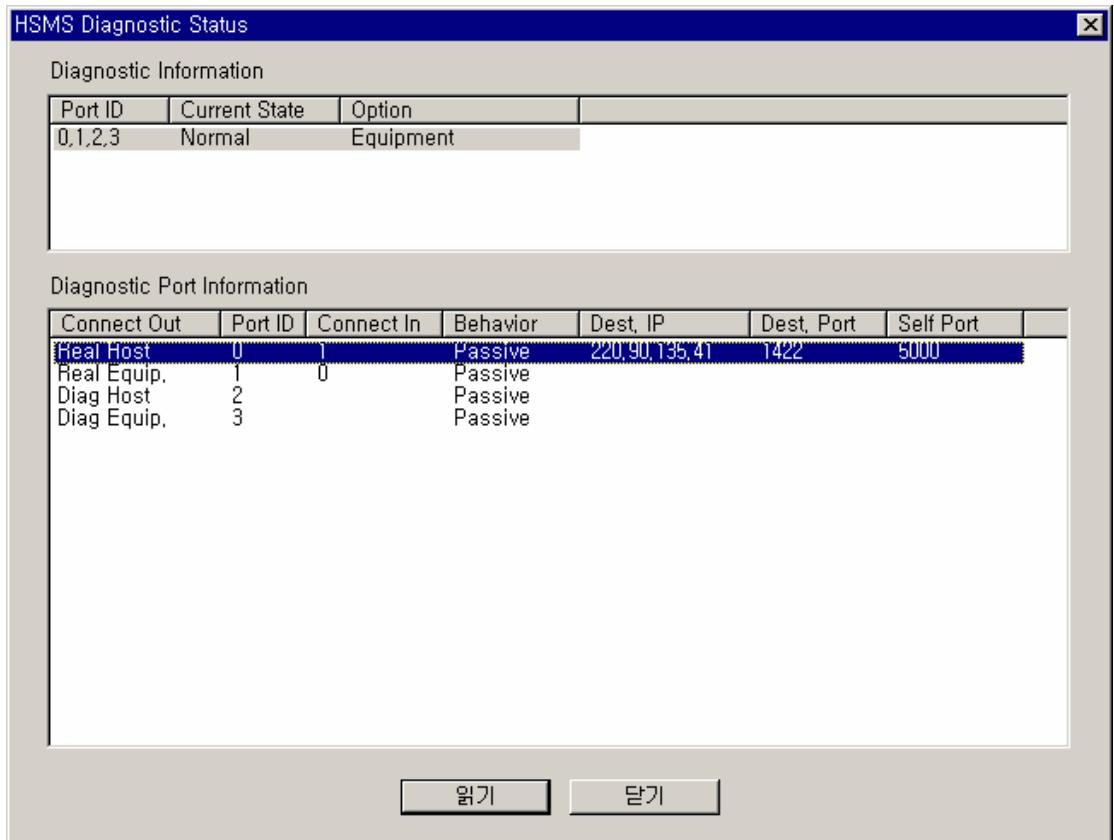


그림 10.5-7 HSMS Diagnostic Status

이중화

제 11 장

ETOS의 이중화 기능에 대해서 설명합니다.

	내용	위치
11.1	개요	394
11.2	H/W 구성	394
11.3	CPU 이중화	398
11.4	네트워크 이중화	401
11.3	전원 이중화	401

11. 이중화

11.1. 개요

이중화 시스템은 오류를 감지하고, 고장을 극복하기 위해 동일 기능을 하는 모듈을 복수로 장비하고있는 시스템을 말합니다.

ETOS 이중화 모델은 CPU 모듈 고장시, 여분의 대기 CPU 모듈이 그 역할을 대신하는 수동적 이중화 (Passive redundancy or Standby redundancy) 시스템입니다.

ETOS 이중화 모델은 ETOS-1000 만 지원합니다.

ETOS-1000 이중화 모델은 CPU 모듈, 네트워크 및 전원 이중화를 지원합니다.

11.2. H/W 구성

11.2.1. 시스템 구성

ETOS-1000 이중화 모델은 CPU 모듈 및 전원을 이중화하고 있습니다.

다음 그림은 ETOS-1000 이중화 모델의 전면부 모습입니다.

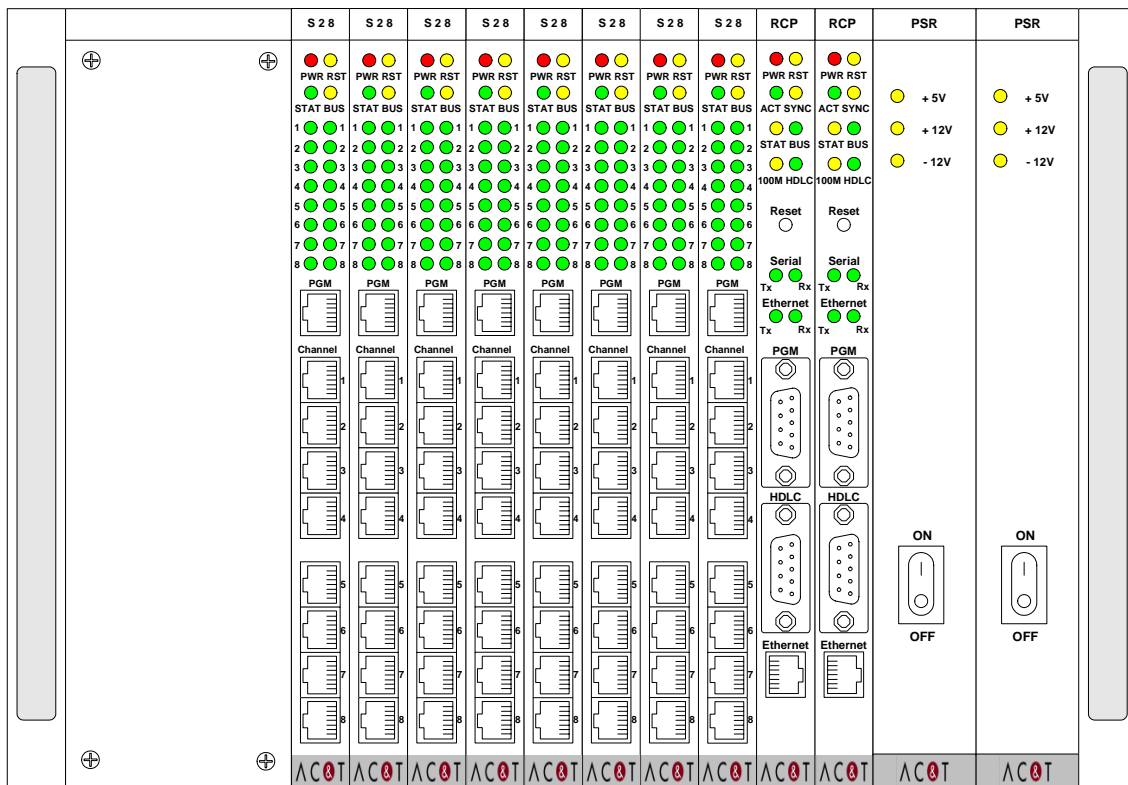


그림 11.2-1 ETOS-1000 이중화 모델 정면도

▶ 이중화 전원

전원은 단독형 모듈이 아닌 이중화 전용 전원을 사용합니다. 이중화 전원의 형명은 ETOS-1000-PSR 입니다.

▶ 이중화 CPU (모듈) (이하 CPU)

CPU는 단독형 모듈(SCP)이 아닌 이중화 전용 모듈(RCP)을 사용합니다. 이중화 전용 CPU의 형명은 ETOS-1000-RCP입니다.

출고시 설정되어 있는 CPU의 위치를 변경하면 정상 동작하지 않습니다. 그러나, CPU간의 위치 변경은 가능합니다.

다음 그림은 ETOS-1000 이중화 모델의 후면부 모습입니다.

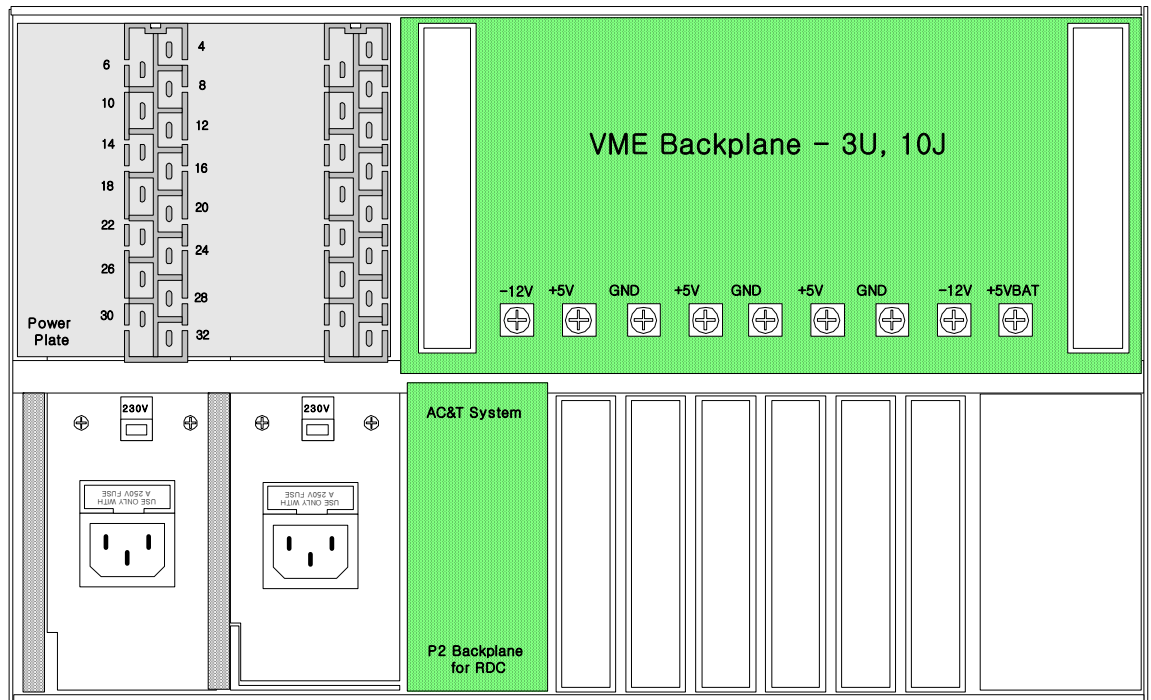


그림 11.2-2 ETOS-1000 이중화 모델의 후면 기구 구성도

이중화 용으로 다음과 같은 모듈이 후면부에 장착됩니다.

▶ Power Plate

전원 이중화를 위한 모듈입니다. 전원 모듈이 고장날 경우, 고장난 모듈의 교체를 용이하게 합니다.

▶ P2 Backplane for RDC

동작 모드 체크 및 정상 동작 여부 확인 및 공유 메모리 송,수신 경로로 사용됩니다.

CPU 두 장이 모두 이 슬롯 위치에 장착되어야 이중화 시스템이 정상 동작합니다.

11.2.2. DIP Switch 설정

▶ **DIP S/W1 (RDC Mode Setup)**

이중화 CPU(RCP)가 정상 동작하기 위한 DIP Switch 설정은 다음과 같습니다.

표 11.2-1 DIP S/W1 설정 방법

모드	단독 모드 CPU (SCP)	이중화 마스터 CPU (RCP)	이중화 슬레이브 CPU (RCP)
3 번 Pin	OFF	ON	ON
4 번 Pin	OFF	ON	OFF

이중화 CPU 의 경우, 3 번 핀은 항상 ON 이어야 하고, 4 번 핀은 마스터(Master) CPU 는 ON, 슬레이브(Slave) CPU 는 OFF 이어야 합니다.

4 번 핀이 모두 OFF 이거나, 모두 ON 이면 정상동작하지 않습니다.

11.2.3. LED 정의

ETOS-1000 이 이중화 모드로 동작하는 경우, CPU의 LED 동작이 단독 모드로 동작할 때와 다릅니다. 이중화 모드로 동작하는 경우 LED의 용도는 다음과 같습니다.

표 11.2-2 이중화 CPU(RCP) 모듈의 LED 동작

전면 표기	LED 색상	동작 상태	
PWR	Red	Power On 시 점등	
RST	Yellow	Reset 시 점등	
ACT	Green	Active 동작 시 점멸	
SYNC	Yellow	사용자 메모리 Sync 시 점등	
STAT ^(*154)	Red	Error 발생시 점등, Standby 상태로 동작 시 점멸	
BUS ^(*155)	Green	VME Bus를 이용한 시리얼 통신 모듈 Access 시 점등	
100M	Red	100M 이더넷 통신 시 점등	
HDLC	Green	Active CPU와 Standby CPU 간의 Alive Check	
Serial	Tx	Green	Monitor Port 송신 시 점등
	Rx	Green	Monitor Port 수신 시 점등
이더넷	Tx	Green	이더넷 송신 시 점등
	Rx	Green	이더넷 수신 시 점등

다음 LED는 Active CPU와 Standby CPU에서 그 기능을 달리하는 경우입니다. 주의하시기 바랍니다.

▶ ACT

- * 이중화 모델인 경우, 한 쪽 CPU의 ACT LED만 점멸합니다.
- * ACT LED가 주기적으로 점멸하는 CPU가 Active CPU입니다.
- * ACT의 점멸 주기가 다소 불규칙적일 수 있으나, 이상은 아닙니다.
- * 두 CPU의 ACT LED가 모두 점멸하지 않는 경우는 정상 상태가 아닙니다. 확인 후, 문의바랍니다.

▶ STAT

- * 초기 모드를 결정하기 전까지 양 쪽 CPU의 STAT LED가 점멸합니다.
- * 초기 모드가 결정되면 한 쪽 CPU만이 STAT LED를 점멸합니다. 주기적으로 점멸하는 쪽이 Standby CPU입니다.

▶ BUS

- * Active CPU에서만 점등합니다.
- * 두 CPU 모듈에서 BUS LED가 점등되면 시스템에 문제가 있는 경우로써 확인이 필요합니다.

(*154) Active CPU에서는 Error 상태 표시로 사용되며, Standby CPU에서는 정상 동작 상태 표시로 사용됩니다.

(*155) BUS LED는 Active CPU에서만 점등됩니다

11.3. CPU 이중화

ETOS-1000 이중화 모델은 CPU 고장시, 여분의 대기 CPU 가 그 역할을 대신하는 수동적 이중화 (Passive redundancy or Standby redundancy) 시스템입니다. 대기 CPU 가 동작 중인 CPU 를 대신하기 위해서는 동작중인 CPU 의 고장 상황을 감지할 수 있어야 합니다. 두 CPU 은 주기적으로 Watchdog 신호를 주고 받아, 상대방의 고장 상황을 감시합니다. 고장 상황을 감지한 CPU 는 고장인 상대방 CPU 을 리셋하고, 자신은 동작(Active) 모드가 됩니다.

▶ Active CPU (동작 모드 CPU)

시리얼 통신 모듈과 정보를 교환하고, 사용자가 작성한 프로그램을 수행하는 CPU 입니다. 실질적으로 시스템을 관장하고 있는 주된 CPU 입니다.

▶ Standby CPU (대기 모드 CPU)

여분의 CPU 로 Active CPU 의 상태를 감시하고 있다가, 고장이 발생하면 그 역할을 대신합니다. 시리얼 통신 모듈과 정보를 교환할 수 없으며, 사용자 프로그램 또한 실행되지 않습니다. 다만, Active CPU 와 사용자 메모리를 동일하게 유지합니다. 따라서, 어느 CPU 의 사용자 메모리를 읽더라도 결과는 동일합니다.

대기 CPU 에서는 사용자가 작성한 프로세스가 수행되지는 않지만, 이더넷 드라이버는 정상 동작합니다. (이더넷 드라이버를 이용한 사용자 메모리 읽기 및 쓰기가 모두 가능합니다) 따라서, 이더넷 드라이버를 이용하는 상대방 기기^(*156)는 CPU 모듈이 Active 인지 또는 Standby 인지와 관계없이 항상 데이터 송,수신을 할 수 있습니다.

두 장의 CPU 중 어느 CPU 가 Active 가 될지는 초기 설정과 CPU 의 동작 상태에 따라 결정됩니다.

▶ 주의

이더넷 드라이버를 이용한 사용자 메모리 읽기 및 쓰기는 CPU 의 동작/대기 모드에 관계없이 어느 CPU 로 부터도 가능합니다. 그러나 파라미터/프로그램 쓰기 등의 ETOS-PD 온라인 명령은 반드시 Active CPU 를 통해서만 가능합니다.

(*156) 주로 MMI (Man Machine Interface) 장비

11.3.1.1. 초기 동작 모드 설정

전원을 켜는 경우 어느 CPU가 Active CPU가 될지는 DIP Switch로 결정하게 됩니다. 전원을 켤 때, Active가 되는 CPU를 마스터 CPU, Standby가 되는 CPU를 슬레이브 CPU이라고 합니다.

네트워크 이중화 시스템을 구성할 경우, 마스터 CPU에는 주 네트워크(Primary Network)를 연결하고, 슬레이브 CPU에는 부 네트워크(Secondary Network)를 연결합니다.

출하시 전원케이블 가까이 있는 CPU가 마스터로 설정되어 있습니다. 마스터/슬레이브 설정은 [12.2.2. DIP Switch 설정](#)을 참조하시기 바랍니다.

11.3.1.2. 시스템 정보

ETOS-1000 이중화 CPU의 현재 동작 상태를 메모리 특정 영역에 표시할 수 있습니다. ETOS-PD 화면에서 [편집] -> [파라미터 설정]을 선택합니다. {파라미터 설정} 대화상자의 [이중화 파라미터] 버튼을 누르면 다음 그림과 같은 {이중화 파라미터} 대화상자가 나타납니다.

<이중화 정보 저장>을 체크하고, 주소를 입력하면, 해당 주소에 CPU의 현재 동작 모드를 기록합니다.

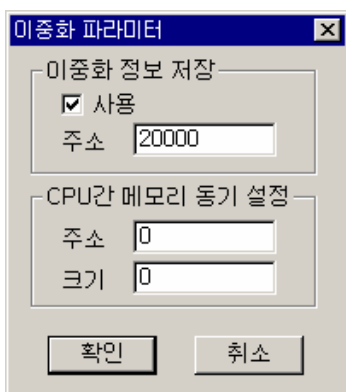


그림 11.3-1 이중화 파라미터 설정 예

총 2 바이트를 사용하며 0xAA, 0x55, 0x77 중 한 값을 가지고 각 의미는 다음과 같습니다.

- * 55H : Active 로 동작 중.
- * AAH : Standby 로 동작 중.
- * 77H : Undefined 모드로 동작 중. 모드를 결정하는 과정 중임을 나타냅니다.

다음은 <이중화 정보 저장> 주소를 "20000"으로 설정한 경우의 예입니다.

이중화 모드가 결정되고 정상 동작하는 경우, 다음 그림과 같이 사용자가 설정한 메모리를 읽어 시스템의 동작 모드 정보를 얻을 수 있습니다.

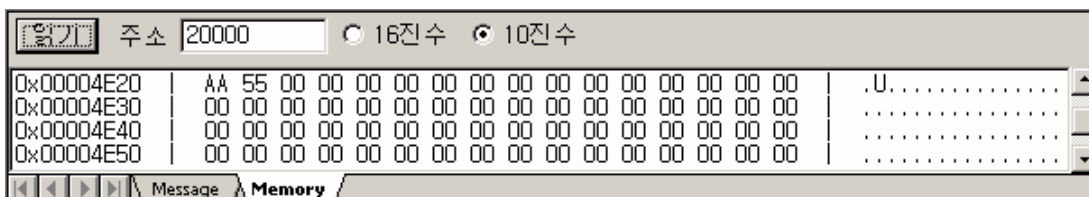


그림 11.3-2 이중화 시스템 정보 저장 예

이때 각 메모리의 값에 대한 의미는 다음과 같습니다.

표 11.3-1 시스템 정보의 의미

주소	내용
20000	마스터 CPU의 현재 동작 모드를 기록합니다.
20001	슬레이브 CPU의 현재 동작 모드를 기록합니다.

지정한 앞 번지에는 마스터 CPU의 현재 모드가 기록되고, 그 다음 번지에는 슬레이브 CPU의 현재 모드가 기록됩니다. 앞 그림의 예와 같이 그 값이 AAH 55H 라면 마스터 CPU는 Standby로 동작 중이며, 슬레이브 CPU는 Active로 동작 중임을 나타냅니다.

11.3.1.3. 사용자 메모리 동기화

Active CPU와 Standby CPU는 사용자 메모리를 동일하게 유지합니다. 기본값은 전 영역을 동일하게 유지하는 것입니다. 그러나, 시스템의 부하를 줄이기 위해서 때로는, 동기를 유지하는 백업 메모리 영역을 제한해야 하는 경우가 있을 수 있습니다. 이때에는, 프로그램의 연산 결과 등, 동기를 유지해야 하는 메모리는 앞 쪽으로 배치하고, 연산의 중간값 등 메모리 동기가 필요치 않은 값 등은 메모리 뒤 쪽에 배치한 후, 동기 영역으로 사용자 메모리 앞 부분만 지정하는 방법을 사용할 수 있습니다.

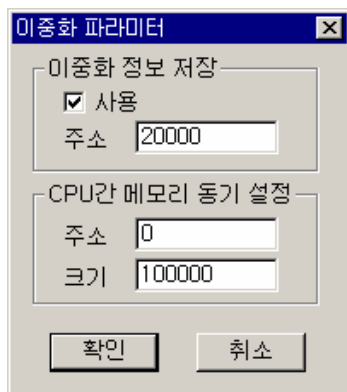


그림 11.3-3 동기 영역 설정 예

위의 그림은 동기 영역으로 0번지부터 100000바이트를 지정한 경우입니다. 99999번지까지의 메모리는 양쪽 CPU가 동일하게 유지하며, 이후 메모리는 동기화하지 않습니다. 따라서, 동기가 불필요한 연산의 중간값 등을 이 번지 이후에 할당하면, 메모리 동기의 빈도를 줄여 시스템의 성능 향상에 도움을 줄 수 있습니다.

11.4. 네트워크 이중화

ETOS-1000 이중화 모델은 네트워크 이중화를 지원합니다. MMI 이중화를 포함한 시스템 구성은 다음 그림과 같습니다. (*157) 이중화 네트워크는 주(Primary) 네트워크와 보조(Secondary) 네트워크로 구성되며 주 네트워크는 ETOS-1000의 마스터 CPU와 보조 네트워크는 ETOS-1000의 슬레이브 CPU와 연결됩니다. MMI Workstation도 같은 방식으로 연결합니다.

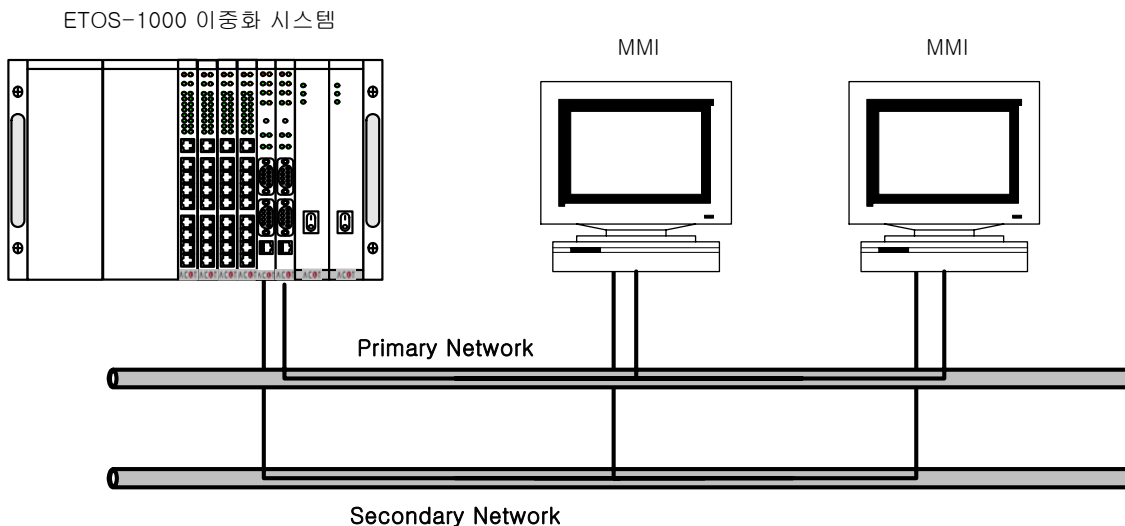


그림 11.4-1. ETOS-1000 네트워크 이중화 시스템 구성도

이더넷 드라이버는 CPU의 동작 모드와 관계없이 연결이 가능하며, 사용자 메모리 읽기/쓰기 또한 가능합니다. 따라서, MMI는 CPU의 고장이나 네트워크 자체의 이상 상황에서 보조 네트워크를 통해 통신을 지속할 수 있습니다.

또, 이더넷 드라이버 아이템인 IF_MEM_WRITE_RCV로 시작하는 프로세스는 Active CPU에서 실행되지만, 메모리 쓰기가 Standby CPU에서 발생해도 동일하게 수행됩니다. 즉, IF_MEM_WRITE_RCV 아이템이 감시하는 쓰기 영역은 Active CPU뿐만 아니라 Standby CPU의 메모리도 포함합니다. 따라서, MMI가 어느 CPU에 연결해서 메모리를 읽거나 써도 그 내용은 Active CPU의 프로세스 수행에 반영되며, 결론적으로 MMI는 현재 연결한 CPU의 모드를 구분할 필요가 없어집니다.

11.5. 전원 이중화

ETOS-1000 이중화 모델은 두 개의 전원이 장착됩니다. 운전 중 전원에 문제가 발생하더라도, 나머지 하나의 전원만으로 정상동작할 수 있으며, 문제가 발생한 전원을 운전 상황에서 교체할 수 있습니다.

정전 상황을 대비하기 위해서는 공급 전원의 이중화 또한 필요합니다. 두 전원을 각기 다른 전원 공급 장치에 연결하면 정전 상황에 대비할 수 있습니다.

(*157) 이중화 네트워크는 Primary Network와 Secondary Network이 각각의 HUB(Switch)를 통해 분리되어야 합니다.

부록

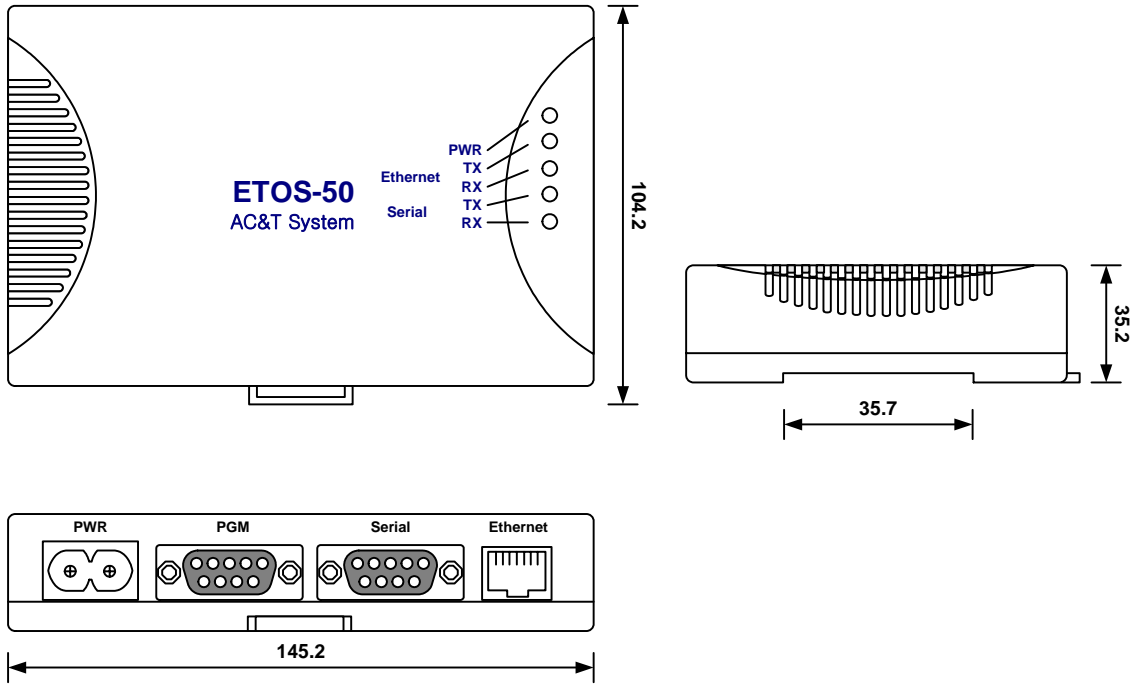
	내용	위치
A	제품 외형도	404
B	에러 코드	408

부록

A. 외형도

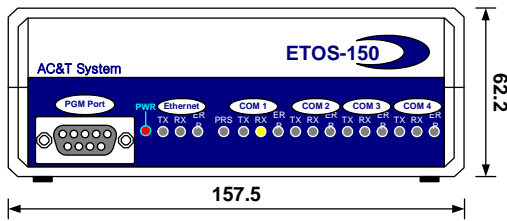
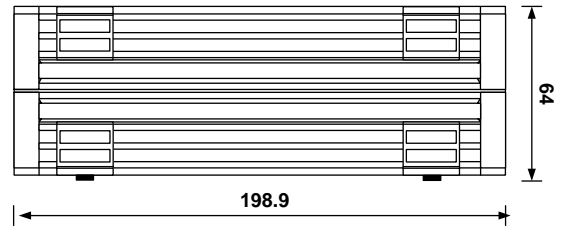
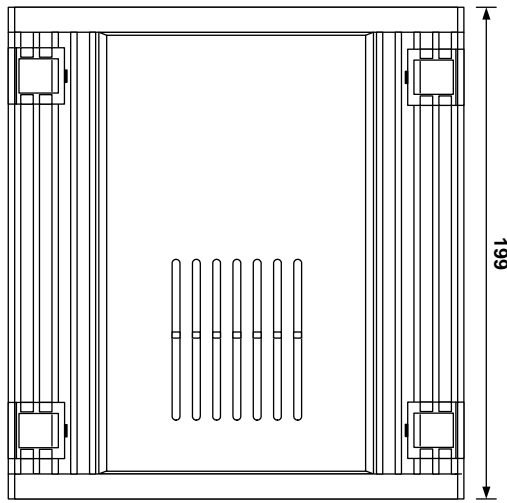
A.1. ETOS-50

치수단위 : mm



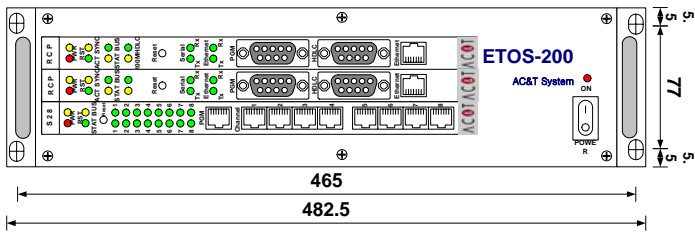
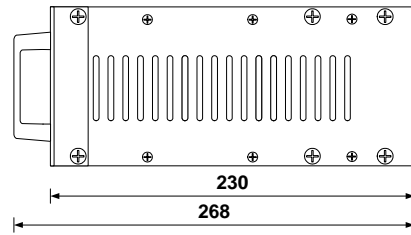
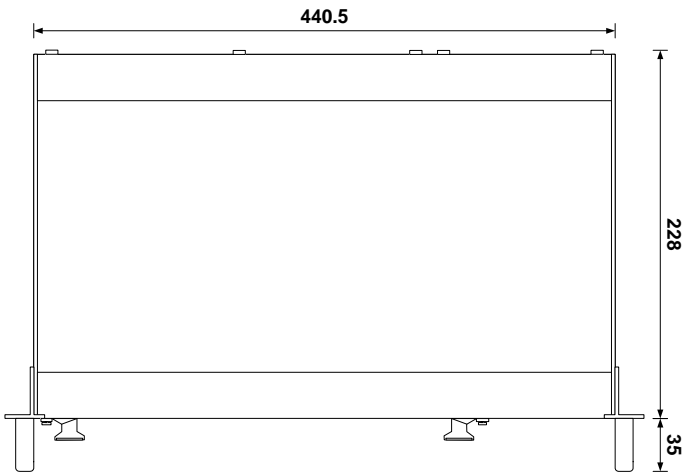
A.2. ETOS-100A/150

치수단위 : mm



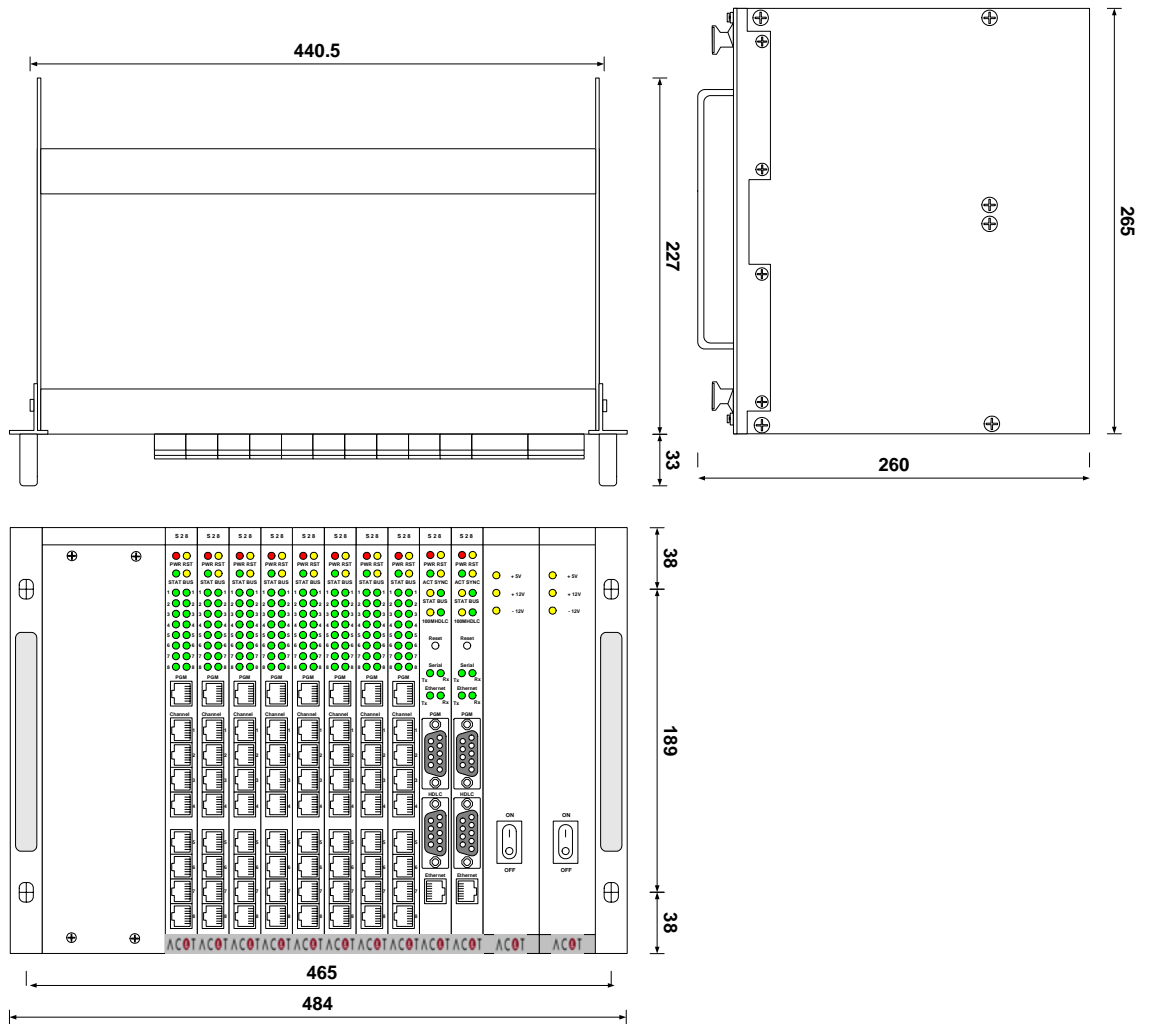
A.3. ETOS-200

치수단위 : mm



A.4. ETOS-1000

치수단위 : mm



B. 에러코드

B.1. MELSEC 에러 코드

- ▶ IF_MELSEC_READ
- ▶ IF_MELSEC_WRITE

MELSEC Q 시리즈 PLC 와 통신할 때, 상대 PLC 가 리턴하는 에러 코드입니다. 에러 코드를 확인하기 위해서는 IF_MELSEC_READ/IF_MELSEC_WRITE 아이템의 <상태값 주소>를 설정해야 합니다.

대표적인 에러 코드 일람으로, 누락된 에러 코드는 MELSEC-Q 매뉴얼을 참조하시기 바랍니다.

에러 코드	내용	대책
C051H ~ C054H	읽기/쓰기 점수가 허용 범위를 초과한다.	읽기/쓰기 점수를 수정한다.
C056H	최대 어드레스를 초과하는 읽기/쓰기 요구	선두 어드레스 또는 읽기/쓰기 점수를 수정한다.
C05AH, C05BH	지정 디바이스에 대해 읽기/쓰기를 할 수 없다.	읽기/쓰기하는 디바이스를 수정한다.
C071H	Q/QACPU 이외로 읽기/쓰기할 디바이스 점수가 너무 많다.	읽기/쓰기 디바이스 점수를 수정한다.

B.2. GLOFA 에러 코드

- ▶ IF_GLOFA_READ
- ▶ IF_GLOFA_WRITE

GLOFA PLC 와 통신할 때, 상대 PLC 가 리턴하는 에러 코드입니다. 에러 코드를 확인하기 위해서는 IF_GLOFA_READ/IF_GLOFA_WRITE 아이템의 <상태값 주소>를 설정해야 합니다. 대표적인 에러 코드 일람으로, 누락된 에러 코드는 GLOFA 매뉴얼을 참조하시기 바랍니다.

▶ 시리얼 에러 코드

에러코드	내용	대책
0001	PLC 와의 인터페이스가 불가능	전원 On/Off
0011	ASCII 데이터 값을 숫자로 변환할 때 발생하는 에러	변수이름 및 데이터에 대/소문자, ('%','_','.'), 숫자 이외의 문자가 사용되었는지 체크하고 수정 후 다시 실행
1132	잘못된 디바이스 메모리 지정	디바이스 타입 검사
1232	실행 데이터 개수의 크기가 120 바이트를 초과	데이터 길이 수정
1332	변수들간의 데이터 타입 불일치	데이터 타입을 동일하게 함
1432	데이터 값이 숫자가 아님.	데이터 값을 검사 함
2432	실제 변수와 데이터 타입 불일치	PLC 프로그램의 변수와 데이터 타입을 동일하게 함
7132 2232	* P,M,L,K,D,T,C,F,S 영역 초과. * 요구 포맷이 맞지 않는 경우.	포맷을 검사하고 수정 후, 다시 실행
무응답	* 국번 에러 * BCC 에러 * 주명령/명령어 타입 에러 * 헤더 및 테일 문자 이상 * 케이블 이상 * 동작 모드 이상. * 통신속도/스톱/데이터 /패리티 비트 이상. * PLC 이상.	각 발생가능한 에러내용에 대해 확인후 조치

▶ 이더넷 에러 코드

에러코드	내용	대책
108	최대 송신 개수 초과(아스키 데이터 개수 = hexa 데이터 개수 * 2 이므로 아스키 데이터 개수가 1400 바이트를 초과하면 안됩니다.)	데이터 길이 수정
117	전용 서비스에서 프레임 헤더의 선두 부분이 잘못된 경우 ('LGIS_GLOFA')	
118	전용 서비스에서 프레임 헤더의 Length 필드가 잘못된 경우	
119	전용 서비스에서 프레임 헤더의 Checksum 필드가 잘못된 경우	
120	전용 서비스에서 명령어가 잘못된 경우	

B.3. MODBUS 에러 코드

- ▶ IF_MODBUS_READ
- ▶ IF_MODBUS_WRITE

MODBUS 슬레이브 기기와 통신할 때, 상대기기가 리턴하는 에러 코드입니다. 에러 코드를 확인하기 위해서는 IF_MODBUS_READ/IF_MODBUS_WRITE 아이템의 <상태값 주소>를 설정해야 합니다.

대표적인 에러 코드 일람으로, 누락된 에러 코드는 MODBUS 매뉴얼을 참조하시기 바랍니다.

에러코드	이름	내용
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the slave. If a Poll Program Complete command was issued, this code indicates that no program function preceded it.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the slave.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for the slave.
04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the slave was attempting to perform the requested action.
05	ACKNOWLEDGE	The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the master. The master can next issue a Poll Program Complete message to determine if processing is completed.
06	SLAVE DEVICE BUSY	The slave is engaged in processing a long-duration program command. The master should retransmit the message later when the slave is free.

B.4. ETOS 에러 코드

- ▶ IF_GLOFA_READ
- ▶ IF_GLOFA_WRITE
- ▶ IF_MELSEC_READ
- ▶ IF_MELSEC_WRITE
- ▶ IF_MODBUS_READ
- ▶ IF_MODBUS_WRITE

위의 아이템 사용시, 아이템 설정 등에 문제가 있어서 ETOS 가 자체적으로 리턴하는 에러 코드입니다. (통신 상대방이 리턴하는 에러 코드가 아닙니다.)

각 아이템의 <상태값 주소>가 설정되어 있어야 합니다.

에러코드	내용	대책
F001H	국번 설정이 잘못됐다.	국번을 수정한다.
F003H	주소 설정이 잘못됐다.	자국 주소를 수정하거나, 읽기/쓰기 길이를 수정한다.
F100H	포트 파라미터 설정이 비어 있다.	포트 파라미터를 설정한다.
FF22H	상대방 프레임의 Checksum 오류	통신 상태를 점검한다.
FF23H	아이템 설정 항목 오류	아이템의 설정 항목을 점검한다.
FF24H	국번 설정 오류	아이템에 설정한 국번을 수정한다.
FF31H	모뎀 에러	모뎀 상태와 연결을 점검한다.
FF32H	TCP/IP 연결이 끊어졌다.	이더넷 상태를 점검한다.
FF80H ~ FFFFH	ETOS 시스템 에러	문의 바랍니다.