

# TeSys Active

## TeSys Tera Motor Management System

### DTM Library 온라인 도움말 가이드

TeSys는 연결성을 갖춘 혁신적인 모터 스타터 솔루션을 제공합니다.

DOCA0275KO-01  
11/2025



## 법률 정보

이 문서에서 제공하는 정보에는 제품/솔루션과 관련된 일반적인 설명, 기술적 특징 및/또는 권장 사항이 포함되어 있습니다.

이 문서는 상세 연구 또는 운영 관련 및 현장 관련 개발 또는 개략적인 계획을 대체하기 위한 것이 아닙니다. 이 문서는 특정 사용자 애플리케이션에 대한 제품/솔루션의 적합성 또는 신뢰성을 판단하기 위해 사용되지 않아야 합니다. 해당 특정 애플리케이션과 관련하여 제품/솔루션에 대한 적절하고 포괄적인 위험 분석, 평가 및 테스트를 직접 수행하거나 자신이 선택한 전문가(통합자, 지정자 등)를 통해 수행하도록 하는 것은 해당 사용자의 의무입니다.

이 문서에서 언급되는 Schneider Electric 브랜드 및 Schneider Electric SE와 그 자회사의 모든 상표는 Schneider Electric SE 또는 그 자회사의 자산입니다. 기타 모든 브랜드는 각 소유자의 상표일 수 있습니다.

이 문서 및 해당 콘텐츠는 관련 저작권법의 보호를 받으며 정보 제공용으로만 제공됩니다. Schneider Electric의 사전 서면 승인 없이는 그 목적을 불문하고 이 문서의 어떠한 부분도 어떤 형태로든 또는 어떤 수단(전자적, 기계적, 복사, 녹음 등)을 통해서든 복제하거나 전송할 수 없습니다.

Schneider Electric은 본 문서 또는 그 콘텐츠를 상업적인 용도로 사용할 수 있는 어떠한 권리나 라이선스도 부여하지 않습니다. 단, 본 가이드를 "있는 그대로" 참고하기 위한 비독점적 및 개인적인 라이선스는 예외로 합니다.

Schneider Electric은 언제든지 통지 없이 이 문서의 내용이나 형식을 변경하거나 업데이트할 수 있는 권리를 보유합니다.

관련 법률에서 허용되는 범위 내에서, **Schneider Electric**과 그 자회사는 이 문서의 정보 내용에서 발견되는 오류나 누락 사항 및 해당 내용의 의도되지 않은 사용 및 잘못된 사용에 대해서 어떠한 책임 또는 배상책임을 지지 않습니다.

# 목차

안전 정보.....	5
문서 정보.....	6
보안 강화 지침 .....	9
소개 .....	11
TeSys 마스터 제품군.....	12
TeSys Tera System .....	13
정의.....	15
TeSys Tera DTM Library 설치를 위한 전제 조건 .....	16
SoMove 설치하기.....	17
TeSys Tera DTM Library 설치하기 .....	19
LTMT Main Unit과 PC 연결 .....	21
문제 해결 .....	22
유저 인터페이스.....	23
TeSys Tera DTM Library 시작하기 .....	24
사용자 인터페이스 설명 .....	33
메뉴 모음 .....	34
툴바 .....	39
상태바 .....	41
탭 영역 .....	43
My Device .....	46
Parameters List .....	50
General Settings .....	54
Starter Function Setting .....	54
Protection Setting .....	55
Communication Setting .....	59
User Map .....	59
Modified Parameters .....	60
설정 .....	61
일반 .....	62
통신 .....	63
보안 .....	65
My Dashboard .....	66
Diagnostics .....	69
Monitoring .....	74
Start Curve.....	78
Custom Logic.....	80
FB Diagram.....	81
Logic Simulator .....	82
사용자 기능 .....	83
펌웨어 업데이트.....	84
HMI 언어 업데이트.....	88
공장 초기화.....	90
핀 관리.....	91
Custom Logic Editor .....	95
Custom Logic Editor 프레젠테이션 .....	96
Custom Logic Editor 사용하기 .....	99
사용자 정의 논리 프로그램의 특징 .....	102

사용자 정의 논리 변수의 정의 .....	103
LTMT Main Unit 변수의 정의.....	104
CALL_EOM 명령 설명 .....	107
사용자 정의 논리 언어 .....	117
사용자 정의 논리 프로그램 작성하기 .....	118
Custom Logic Editor 소개.....	119
Custom Logic Editor 사용자 인터페이스 .....	120
논리 명령 .....	123
논리 명령 .....	128
프로그램 논리 명령 .....	129
Boolean Logic 명령 .....	131
Register Logic 명령 .....	140
Timer Logic 명령 .....	148
Latch logic 명령.....	150
Counter Logic 명령.....	152
Math Logic 명령 .....	154
사용자 정의 논리 프로그램 예 .....	156
타이머 및 곱하기 명령을 확인하는 방법 .....	157
진리표를 생성하는 방법.....	158
기능 블록 다이어그램 언어.....	161
FBD 언어 개요 .....	162
FBD 편집기 소개.....	163
FBD 요소.....	165
Starter Types .....	166
Computation Blocks .....	168
Inputs Blocks .....	171
Function Blocks .....	174
Logic Blocks .....	177
Outputs Blocks .....	177
FBD 언어를 사용하여 프로그래밍하기 .....	180
FBD 블록 삽입하기 .....	181
블록 간 링크 생성하기.....	182
FBD 블록 속성.....	184
FBD 리소스 관리.....	185
FBD 블록 조작하기 .....	186
블록 선택하기 .....	187
개체 삭제 또는 복제하기.....	187
FBD 편집기 표시 옵션 .....	190
표시 옵션 .....	191
작업 영역 모양새 및 그래프 옵션.....	192
프로그램 컴파일, 시뮬레이션 및 전송하기 .....	194
소개.....	195
LTMT Main Unit Logic Simulator .....	196
초기화 및 연결.....	198
LTMT Main Unit과 Custom Logic Editor 간에 논리 파일 전송하기.....	199

# 안전 정보

## 중요 정보

이 설명서를 주의 깊게 읽고, 장치를 설치, 작동, 서비스 또는 유지보수하기 전에 장치에 익숙해지기 위해 장비를 살펴보십시오. 다음의 특정 메시지는 잠재적 위험을 경고하거나 절차를 명확하고 간소화하는 정보를 알려주기 위해 이 문서 전반에 또는 장비에 표시될 수 있습니다.



“위험” 또는 “경고” 라벨에 이 기호가 추가되어 있는 경우 감전의 위험성이 있으며, 지시에 따르지 않는 경우 인적 상해가 발생할 가능성이 있다는 것을 나타냅니다.



안전 경고 기호입니다. 인적 상해의 위험성이 있다는 것을 경고합니다. 이 기호의 뒤에 기재된 안전에 관한 정보에 따라 인적 상해나 사망의 위험에 대해 방지 대책을 마련하십시오.

### ⚠ 위험

위험 이 표시는 지시에 따르지 않으면, 사망 또는 중상을 입을 상황이 되는 것을 나타냅니다.

### ⚠ 경고

경고 이 표시는 지시에 따르지 않으면, 사망 또는 중상을 입을 가능성이 있다는 것을 나타냅니다.

### ⚠ 주의

주의 이 표시는 지시에 따르지 않으면, 경상 또는 중급의 상해를 입을 가능성이 있다는 것을 나타냅니다.

### 주기

이 표시는 지시에 따르지 않으면, 물적 손해를 입을 가능성이 있다는 것을 나타냅니다.

## 주의하십시오

전기 장비는 자격을 갖춘 기사만 설치, 작동, 서비스, 유지관리해야 합니다. 이 자료를 사용하지 않고 일어난 결과에 대해서는 Schneider Electric에서 책임을 지지 않습니다.

전문 인력이란 전기 장비 구축, 작동, 설치에 관한 기술 및 지식을 갖추고 있고, 관련 위험을 인지하고 방지하기 위한 안전 교육을 받은 인력입니다.

## 문서 정보

### 문서 적용범위

이 온라인 도움말 가이드는 TeSys™ Tera DTM 라이브러리를 위한 TeSys Tera Motor Management System.

이 온라인 도움말 가이드에서는 성공적인 시스템 구현의 네 가지 핵심 부분을 설명합니다.

- TeSys island DTM Libray 설치하기 TeSys Tera DTM Library
- 장치 매개 변수 구성
- 장치 상태 모니터링
- TeSys Tera system유지보수 및 업그레이드

이 온라인 도움말 가이드는 다음과 같은 TeSys Tera DTM Library 사용자를 위한 것입니다.

- 설계 엔지니어
- 시스템 통합 사업자
- 시스템 운영자
- 유지보수 엔지니어

### 유효성 정보

이 문서는 다음에 대해 유효합니다.

- SoMove™ 소프트웨어 버전 v2.10.0
- TeSys Tera DTM Library 버전 v2.0.0

이 문서에 설명되어 있는 몇몇 기능의 제공 여부는 TeSys Tera system에서 사용되는 커뮤니케이션 프로토콜 및 이에 설치된 물리적 모듈에 따릅니다.

### 일반 사이버 보안 정보

최근 몇 년 사이에 네트워크화한 기계와 생산 공장의 수가 증가함에 따라 무단 액세스, 데이터 유출, 운영 중단 등 사이버 위협의 잠재성도 그에 상응하여 증가했습니다. 따라서 자산과 시스템을 이러한 위협에서 보호하는 데 도움이 되는 가능한 모든 사이버 보안 조치를 반드시 고려하셔야 합니다.

Schneider Electric 제품을 안전하게 보호하는 데 도움이 되려면 Cybersecurity Best Practices 문서에 설명된 사이버 보안 모범 사례를 시행하시는 것이 가장 좋습니다.

Schneider Electric은 다음과 같은 정보와 지원을 추가로 제공합니다.

- Schneider Electric 보안 뉴스레터 구독.
- 다음이 가능한 Cybersecurity Support Portal 웹 페이지
  - 보안 통지 사항 찾기
  - 취약점 및 사건 보고
- 다음이 가능한 Schneider Electric Cybersecurity and Data Protection Posture 웹 페이지
  - 사이버 보안 태세 액세스
  - 사이버 보안 아카데미에서 사이버 보안 자세히 알아보기
  - Schneider Electric의 사이버 보안 서비스 검색

## 제품 관련 사이버 보안 정보

자세한 내용은 *TeSys Tera Motor Management System Cybersecurity Guide – DOCA0260EN* 항목을 참조하십시오.

## 이 문서의 번역 언어

이 문서는 다음 언어로 제공됩니다.

- 영어
- 중국어
- 프랑스어
- 독일어
- 이탈리아어
- 한국어
- 스페인어

## 관련 문서

문서 제목	설명	참조 번호
TeSys Tera Motor Management System 카탈로그	카탈로그: <ul style="list-style-type: none"> <li>• TeSys Tera system에 대해 설명합니다.</li> <li>• TeSys Tera의 기술적 특성을 포함합니다.</li> </ul>	LVCATENTER
TeSys Tera Motor Management System 사용자 가이드	전체 TeSys Tera system에 대해 소개하는 기본 사용자 가이드입니다. LTMT main units, LTMTCT/LTMTCTV Sensor Module, LTMT expansion modules 및 LTMTCUF control operator unit의 주요 기능을 설명합니다.	DOCA0257EN
TeSys Tera Motor Management System Installation Guide	이 가이드에서는 LTMT main units, LTMTCT/LTMTCTV Sensor Module, LTMT expansion modules 및 LTMTCUF control operator unit의 설치, 시운전 및 유지보수에 대해 설명합니다.	DOCA0356EN
TeSys Tera Motor Management System Modbus RTU Communication Guide	이 가이드는 LTMT main unit의 Modbus 네트워크 프로토콜 통신에 대해 설명합니다.	DOCA0355EN
TeSys Tera Motor Management System PROFIBUS DP 통신 안내서	이 안내서는 LTMT main unit의 PROFIBUS DP 네트워크 프로토콜 통신에 대해 설명합니다.	DOCA0256EN
TeSys Tera Motor Management System EtherNet/IP 통신 안내서	이 안내서는 LTMT main unit의 EtherNet/IP 네트워크 프로토콜 통신에 대해 설명합니다.	DOCA0258EN
TeSys Tera Motor Management System LTMTCUF Control Operator Unit 사용자 가이드	이 가이드에서는 LTMTCUF control operator unit의 설치, 구성 및 사용 방법에 대해 설명합니다.	DOCA0233EN
TeSys Tera Motor Management System DTM Library Software Release Note	이 문서에서는 TeSys Tera DTM Library software에 대한 중요한 정보를 제공하며, 새로운 기능 및 향상된 기능에 대한 요약을 제공합니다.	DOCA0279EN
TeSys Tera Motor Management System Firmware Release Note	이 문서에서는 TeSys Tera system의 펌웨어 버전에 대한 정보와 새로운 기능 및 향상된 기능에 대한 요약을 제공합니다.	DOCA0276EN
TeSys Tera Motor Management System 사이버 보안 가이드	이 가이드는 TeSys Tera Motor Management System에 대한 사이버 보안 측면에 대한 정보를 제공합니다. 이 가이드에서는 운영 기술 네트워크 또는 회사 시리얼 또는 Ethernet 네트워크를 보호하는 방법에 대해 설명합니다.	DOCA0260EN

문서를 온라인으로 찾으시려면 Schneider Electric 다운로드 센터 ([www.se.com/ww/en/download/](http://www.se.com/ww/en/download/))를 방문하십시오.

## 비포용적이거나 몰지각한 용어에 관한 정보

책임감 있고 포용적인 기업으로서 Schneider Electric은 비포용적이거나 몰지각한 용어가 포함된 홍보물과 제품을 지속적으로 업데이트하고 있습니다. 그러나 이 같은 노력에도 불구하고 당사 콘텐츠에는 일부 고객에게 부적절하다고 여겨지는 용어가 들어 있을 수 있습니다.

## 상표

QR 코드는 일본 및 그 외 국가에서 DENSO WAVE INCORPORATED의 등록 상표입니다.

# 보안 강화 지침

## 소개

PC에서 다양한 애플리케이션을 실행하여 제어 환경의 보안을 강화할 수 있습니다. 시스템에는 심층적 방어 접근법에 대한 Schneider Electric의 장치 강화 권장 사항과의 일관성을 위해 재구성을 요구하는 공장 기본 설정이 있습니다.

다음 지침은 Windows 운영 체제에서의 절차를 설명합니다. 이들은 단지 예시로만 제공됩니다. 귀하의 운영 체제 및 애플리케이션이 다른 요구 사항 또는 절차를 가질 수도 있습니다.

## 원격 데스크탑 프로토콜 비활성화하기

Schneider Electric의 심층 방어 접근 방식 권장 사항에는 애플리케이션에서 원격 데스크탑 프로토콜(RDP)을 요구하지 않는 경우 RDP의 비활성화가 포함됩니다.

Windows 11에서 원격 데스크톱 프로토콜(RDP)을 비활성화하려면 **설정 > 시스템 > 원격 데스크톱 > 원격 데스크톱 사용(꺼짐으로 전환)**을 사용합니다.

## 보안 정책 업데이트하기

명령창에 `gpupdate`를 입력하여 시스템에서 PC의 보안 정책을 업데이트합니다. 자세한 내용은 Microsoft에 대한 `gpupdate` 설명서를 참조하십시오.

## 업데이트 관리하기

배포에 앞서, Microsoft의 **Windows 업데이트** 웹 페이지에서 유틸리티를 사용하여 모든 PC 운영 체제를 업데이트하십시오. Windows에서 이 툴에 접속하려면, **시작 > 모든 프로그램 > Windows 업데이트**를 선택하십시오.

## 워크스테이션 보호

엔지니어링 워크스테이션과 관련된 보안 위험을 줄이기 위해 DEP(Data Execution Prevention) 및 ASLR(Address Space Layout Randomization)과 같은 메모리 취약성 보호 설정을 활성화하십시오. 이러한 보안 설정은 Windows 11 운영 체제에서 시스템 취약성 보호 설정을 사용하여 활성화할 수 있습니다. 자세한 내용은 Microsoft 보안 기능 웹페이지를 참조하십시오.

## 안전한 비밀번호 적용

대문자, 소문자, 숫자 및 특수문자 등 필수 요소를 충족하는 강력한 비밀번호를 사용하십시오. 이 기능을 활성화하면 취약한 비밀번호로 인한 위험을 줄여 무단 접근을 방지하는데 도움이 됩니다.

## 기본이 아닌 포트 사용

HTTPS, DWPS, Modbus TCP와 같은 프로토콜의 기본 통신 포트를 변경하면 추가적인 보안 계층이 적용됩니다.

## 허용 IP 목록

IP 허용 목록 기능은 지정된 IP 주소만 허용함으로써 시스템 접근을 제한합니다. 이는 무단 장치의 시스템 연결을 방지하고 신뢰할 수 있는 출처만이 통신할 수 있도록 보장합니다. TeSys Tera system. IP 허용 목록 기능에 접근하려면 **보안 > IP 허용 목록 > 표시**됩니다. 표준 웹 서버에서

# 소개

## 이 파트의 내용

TeSys 마스터 제품군 .....	12
TeSys Tera System.....	13
정의.....	15
TeSys Tera DTM Library 설치를 위한 전제 조건 .....	16
SoMove 설치하기 .....	17
TeSys Tera DTM Library 설치하기 .....	19
LTMT Main Unit과 PC 연결 .....	21
문제 해결.....	22

## TeSys 마스터 제품군

TeSys 글로벌 시장 선도 기업이 제공하는 혁신적인 모터 제어, 모니터링 및 관리 솔루션입니다. TeSys 모든 주요 글로벌 전기 표준을 준수하는 모터 및 전기 부하의 스위칭 및 보호를 위한 연결되고 효율적인 제품 및 솔루션을 제공합니다.

# TeSys Tera System

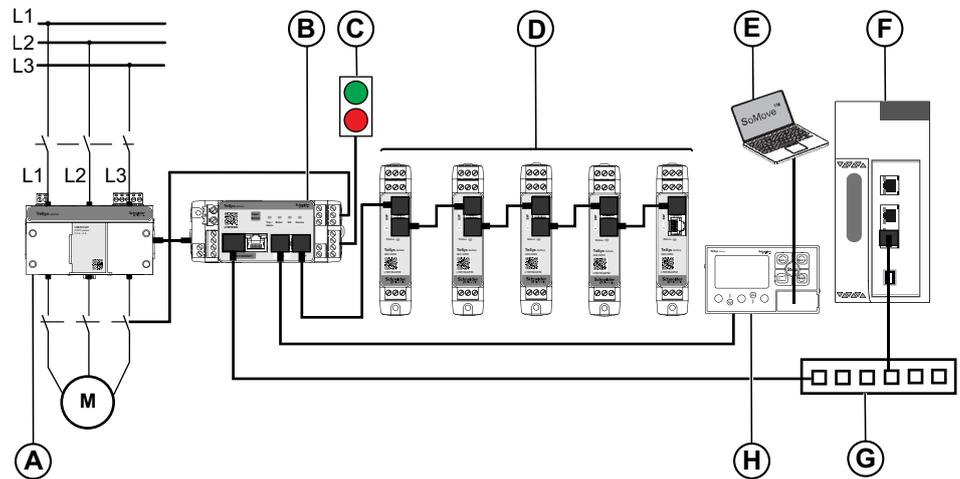
## 개요

그 TeSys Tera Motor Management System (또는 TeSys Tera system)는 TeSys 지능형 릴레이 및 모터 스타터의 액티브 제품군에 속합니다. 이 제품은 TeSys Tera system는 단상 또는 삼상 교류 유도 모터에 대한 완벽한 보호, 계량, 제어 및 모니터링 기능을 제공하기 위해 지능형 모터 제어 센터(iMCC)의 신뢰할 수 있는 구성 요소로 설계되었습니다.

TeSys Tera system은 저압 개폐기 시스템에 설치되며 Fieldbus 네트워크와 모터 피더를 통해 더 높은 수준의 자동화 시스템을 연결합니다.

TeSys Tera system:

- iMCC 피더의 일반 및 고급 모터 보호, 계측, 그리고 모니터링을 독립형 HMI 장치가 있으며 구성하기 쉬운 단일 소형 통신 모듈로 처리합니다.
- 저압 접촉기 제어 모터 스타터 피더를 위한 보호 컨트롤러를 제공합니다.
- 저압 애플리케이션에서 일정한 속도로 작동하는 모터를 위한 유연한 모듈형 모터 관리 시스템을 제공합니다.



- A LTMTCT/LTMTCTV sensor module
- B LTMT main unit
- C 시작/정지 명령
- D LTMT expansion modules
- E PC에서 실행 중인 SoMove FDT 컨테이너 소프트웨어와 TeSys Tera DTM 설치된 상태로 실행 중입니다.
- F 프로그래밍 가능한 로직 컨트롤러(PLC) 또는 분산형 제어 시스템(DCS)
- G Ethernet 스위치
- H LTMTCUF control operator unit

## 기능 특성

TeSys Tera system은 다음을 관리합니다.

- 최대 100A 및 690V 작동 전압의 단상 또는 3상 AC 유도 모터 및 히터(통합 센서 모듈 포함).
- 최대 810A 및 690V 작동 전압의 단상 또는 3상 AC 유도 모터 및 히터(외부 전류 변압기 포함).
- 플랜트 가용성을 높이기 위해 제어 시스템과 모터 피더 사이의 연결
- 설치, 시운전, 운영 및 유지보수에서 상당한 비용 절감
- 애플리케이션 및 프로세스 요구 사항에 따라 모터의 매개 변수를 설정하는 데 사용할 수 있는 수치 마이크로프로세서 장착 컨트롤러

# 정의

## Device Type Manager

Device Type Manager (DTM)는 특정 장치를 위해 FDT container에서 호스팅되는 소프트웨어 모듈입니다.

DTM의 기능은 다음과 같습니다.

- 장치 라이브러리 설치
- 장치 라이브러리 업데이트
- 장치에서 다양한 Fieldbus 스캔
- 장치 전력 및 에너지 모니터링
- 장치 매개 변수의 구성 관리
- 프로젝트 파일 관리
- 장치 매개 변수 단위 사용자 정의
- 문제 해결
- 장치의 펌웨어 업데이트

## SoMove Software 프로젝트 파일

SoMove software 프로젝트 파일은 미리 결정된 장치의 구성 파일로, 오프라인에서 생성하고 나중에 사용할 수 있도록 저장할 수 있습니다.

프로젝트 파일에는 다음과 같은 정보가 들어 있습니다.

- 장치 토폴로지 구성
- 모든 매개 변수 설정

주의:

- 프로젝트 파일에는 사용자 정의 프로그램(사용자 정의 로직 프로그램)이 포함되어 있지 않으므로 수동으로 저장해야 합니다.
- 이 파일은 \*.psx확장명으로 저장됩니다.

프로젝트 생성 방법에 대한 자세한 내용은 **SoMove** 온라인 도움말을 참조하십시오.

# TeSys Tera DTM Library 설치를 위한 전제 조건

## 시스템 요구사항

The TeSys Tera DTM Library 설치할 수 있습니다 Microsoft® Windows® 11 운영 체제에 설치할 수 있습니다.

## 소프트웨어 요구사항

TeSys Tera DTM Library를 PC에 설치하려면 다음 소프트웨어가 필요합니다.

- Microsoft .NET Framework v3.5
- SoMove 소프트웨어 v2.10.0
- TeSys Tera DTM Library v2.0.0

## 하드웨어 요구사항

TeSys Tera DTM Library를 설치하려면 PC가 다음과 같은 하드웨어 요구 사항을 충족해야 합니다.

장비	최소 요구 사항	권장 요구 사항
프로세서	Pentium 4/Core 2 Duo, 2 GHz	인텔®코어™i3
RAM	2 GB	4 GB
디스플레이	해상도: 1024 x 768, 1366 x 768, 1600 x 1900 및 1920 x 1080픽셀	해상도: 1600 x 1200 및 1920 x 1080픽셀
	DPI: 96(100%) 및 120(125%)	DPI: 96(100%) 및 120(125%)
시스템 드라이브의 사용 가능한 하드 디스크 공간	1 GB	2 GB

# SoMove 설치하기

## 개요

SoMove software는 개방형 FDT 또는 DTM 기술을 사용하는 Microsoft Windows® 기반 애플리케이션입니다. SoMove software에는 여러 DTM이 포함되어 있습니다.

## SoMove Software 다운로드하기

SoMove software는 Schneider Electric website에서 다운로드할 수 있습니다.

주의: PC에 소프트웨어를 다운로드, 설치 또는 제거하려면 관리자 권한이 있어야만 합니다.

## SoMove Software 설치하기

다음 단계에서는 SoMove software 설치 방법을 설명합니다.

1. 다운로드한 파일 *SoMove\_FDT*의 압축을 풉니다.
2. 폴더에 이름이 *SoMove\_VX.Y.Z*(여기서 X.Y.Z는 버전 번호)인 .exe 파일과 릴리스 노트가 있습니다.
3. *SoMove\_VX.Y.Z*를 두 번 클릭하여 설치를 시작합니다.
4. **Choose Setup Language(설치 언어 선택)** 대화 상자에서 설치 언어를 선택합니다.
5. **OK(확인)**를 선택합니다.
6. **Welcome to the Installation Wizard for SoMove(SoMove 설치 마법사 시작)** 대화 상자에서 **Next(다음)**를 선택합니다.
7. **Install Shield Wizard(Install Shield 마법사)** 대화 상자가 나타나고 Modbus 드라이버를 설치해야 한다는 메시지가 표시되면 **Install(설치)**를 선택합니다.  
결과: Modbus 드라이버가 자동으로 설치됩니다.
8. **Readme and Release Notes(추가 정보 및 릴리스 노트)** 대화 상자에서 **Next(다음)**를 선택합니다.
9. **License Agreement(라이선스 계약)** 대화 상자에서
  - a. 라이선스 계약을 주의 깊게 읽어보십시오.
  - b. **I accept the terms in the license agreement(라이선스 계약의 약관에 동의)**를 선택합니다.
  - c. **Next(다음)**를 선택합니다.
10. **Customer Information(고객 정보)** 대화 상자에서
  - a. 해당 필드에 다음 정보를 입력합니다.
    - **First name(이름)**
    - **Last name(성)**
    - **Company name(회사 이름)**
  - b. **Next(다음)**를 선택합니다.
11. **Destination Folder(대상 폴더)** 대화 상자에서
  - a. 필요한 경우 **Change(변경)** 옵션을 선택하여 SoMove software 대상 폴더를 수정합니다.
  - b. **Next(다음)**를 선택합니다.

12. **Shortcuts(바로 가기)** 대화 상자에서
  - a. 바탕 화면 및 빠른 실행 표시줄에 바로 가기를 생성하려면 해당 옵션을 선택합니다.
  - b. **Next(다음)**를 선택합니다.
13. **Ready to Install the Program(프로그램 설치 준비 완료)** 대화 상자에서 **Install(설치)**을 선택합니다.

결과: 다음과 같은 SoMove software 구성 요소가 자동으로 설치됩니다.

  - 통신 프로토콜을 포함하는 Modbus 통신 DTM Library
  - 다양한 카탈로그를 포함하는 DTM 라이브러리
  - SoMove software
14. **Installation Wizard Completed(설치 마법사 완료)** 대화 상자에서 **Finish(완료)**를 선택합니다.

결과: SoMove software가 PC에 설치되어 있습니다.

# TeSys Tera DTM Library 설치하기

## 개요

SoMove software에는 TeSys Tera system에 맞는 DTM이 있습니다. SoMove software를 설치한 후 TeSys Tera DTM Library를 설치해야 합니다.

## TeSys Tera DTM Library 다운로드하기

Schneider Electric 웹 사이트([www.se.com](http://www.se.com))에서 TeSys Tera DTM Library를 다운로드할 수 있습니다.

**Search(검색)** 필드에 TeSys Tera DTM Library를 입력하여 다운로드 페이지로 이동합니다.

주의: PC에 소프트웨어를 설치하거나 제거하려면 관리자 권한이 있어야만 합니다.

## TeSys Tera DTM Library 설치하기

다음 단계에서는 TeSys Tera DTM Library의 설치에 대해 설명합니다.

1. 다운로드한 *TeSysTeraDTMLibrary* 파일의 압축을 풉니다.
2. 폴더에 exe 파일 *Schneider\_Electric\_TeSys\_Tera\_DTM\_Library\_V.X.Y.Z*(여기서 X.Y.Z는 버전 번호) 및 릴리스 노트 *TeSysTeraDTMLibrary\_vx.y.z\_ReleaseNotes*(여기서 x.y.z는 버전 번호)가 포함됩니다..
3. *Schneider\_Electric\_TeSys\_Tera\_DTM\_Library\_V.X.Y.Z*을 두 번 클릭하여 설치를 시작합니다.
4. **Choose Setup Language(설치 언어 선택)** 대화 상자에서 설치 언어를 선택하고 **OK(확인)**를 선택합니다.
5. **Welcome to the Installation Wizard for Schneider Electric TeSys Tera DTM Library(Schneider Electric TeSys Tera DTM Library 설치 마법사 시작)** 대화 상자에서 **Next(다음)**를 선택합니다.
6. **Readme and Release Notes(추가 정보 및 릴리스 노트)** 대화 상자에서 **Next(다음)**를 선택합니다.
7. **License Agreement(라이선스 계약)** 대화 상자에서
  - 라이선스 계약을 주의 깊게 읽어보십시오.
  - 라이선스 계약 옵션에서 **I accept the terms in the license agreement(라이선스 계약의 약관에 동의)**를 선택합니다.
  - **Next(다음)**를 선택합니다.
8. **Customer Information(고객 정보)** 대화 상자에서
  - 해당 필드에 다음 정보를 입력합니다.
    - 이름
    - 성
    - 회사 이름
  - **Next(다음)**를 선택합니다.
9. **Destination Folder(대상 폴더)** 대화 상자에서
  - 필요한 경우 **Change(변경)** 옵션을 선택하여 TeSys Tera DTM Library 대상 폴더를 수정합니다.
  - **Next(다음)**를 선택합니다.
10. **Ready to Install the Program(프로그램 설치 준비 완료)** 대화 상자에서 **Install(설치)**를 선택합니다.

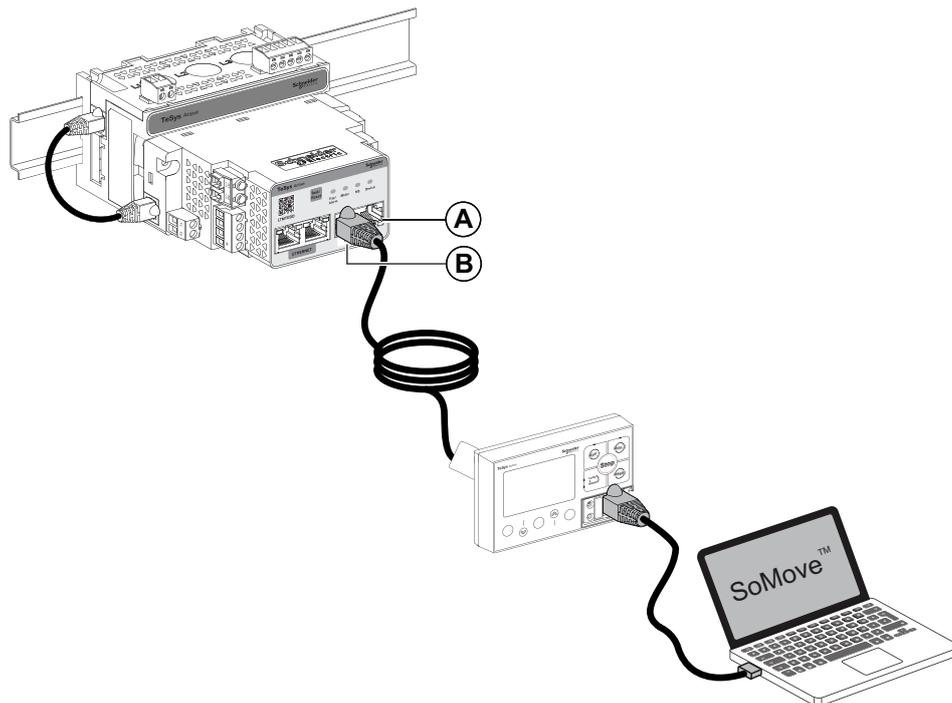
11. **Installation Wizard Completed(설치 마법사 완료)** 대화 상자에서 **Finish(완료)**를 선택합니다.

**결과:** TeSys Tera DTM Library가 PC에 설치되어 있습니다.

# LTMT Main Unit과 PC 연결

다음과 같은 방식으로 LTMT main unit을 SoMove software를 실행 중인 PC에 연결할 수 있습니다.

- 해당 소프트웨어를 실행 중인 PC의 USB 포트에 연결 SoMove 소프트웨어를 실행 중인 PC의 USB 포트를 RJ45 에 LTMTCUF control operator unit USB/RJ45 케이블을 사용하여
- 해당 소프트웨어를 실행 중인 PC의 USB 포트에 연결합니다. SoMove 소프트웨어를 실행 중인 PC의 USB 포트를 HMI USB/ LTMT main unit USB/RJ45 케이블을 사용하여
- 구성을 위해 SoMove software를 실행 중인 PC의 USB 포트를 Fieldbus 포트에 연결합니다.



- A 확장 모듈 포트 (EXP)
- B HMI 포트 (HMI)

TeSys Tera system 및 DTM library간에서 다양한 통신 프로토콜을 구성하고 연결을 설정하는 것에 대한 자세한 내용은 SoMove 온라인 도움말의 Connection(연결) 단원을 참조하십시오.

LTMTCUF control operator unit을 사용하여 PC를 TeSys Tera system에 연결하는 방법에 대한 자세한 내용은 TeSys Tera Motor Management System LTMTCUF Control Operator Unit User Guide – DOCA0233EN를 참조하십시오.

## 문제 해결

문제	원인	해결책
TeSys Tera DTM Library software v2.10.x 이상의 <b>Catalog(카탈로그)</b> 창에 SoMove가 나타나지 않습니다.	DTM을 적용하기 전에 SoMove software가 등록되지 않았습니다.	<p>SoMove software를 등록하십시오.</p> <ol style="list-style-type: none"> <li>1. SoMove 시작 시 다음 메시지가 표시됩니다. <b>x Days left for the demo version to expire.</b>(데모 버전이 만료될 때까지 x일 남았습니다.) <b>Do you want to register now?</b>(지금 등록하시겠습니까?)</li> <li>2. <b>Yes(예)</b>를 선택하여 등록을 확인하고 지침을 따르십시오. 주의: SoMove software 등록은 무료입니다.</li> </ol>
	.dll 파일이 Windows에서 등록되지 않았습니다.	<p>명령 프롬프트 창을 사용하여 TeSys Tera DTM Library에 필요한 .dll 파일을 등록합니다.</p> <ol style="list-style-type: none"> <li>1. 관리자 권한으로 명령 프롬프트 창을 시작하고 다음 명령을 입력합니다. <b>cd C:\Windows\Microsoft.NET\Framework\v2.0.50727</b></li> <li>2. 다음 명령을 입력합니다. <b>RegAsm.exe /codebase "C:\Program Files (x86)\Common Files\Schneider Electric Shared\Schneider Electric TeSys Tera DTM Library\TeSysTerraDtm.Kernel.dll"</b> 결과: 등록이 완료되면 메시지가 표시됩니다.</li> <li>3. SoMove software를 실행합니다. 진행률 표시줄이 있는 <b>Catalog Update(카탈로그 업데이트)</b> 화면이 나타납니다. 결과: 카탈로그 업데이트가 완료되면 <b>Catalog(카탈로그)</b> 창에서 TeSys Tera DTM Library를 사용할 수 있습니다.</li> </ol>

# 유저 인터페이스

## 이 파트의 내용

TeSys Tera DTM Library 시작하기 .....	24
사용자 인터페이스 설명 .....	33
메뉴 모음.....	34
툴바.....	39
상태바.....	41
탭 영역.....	43
My Device.....	46
Parameters List.....	50
설정.....	61
My Dashboard .....	66
Diagnostics.....	69
Monitoring .....	74
Start Curve .....	78
Custom Logic.....	80
FB Diagram .....	81
Logic Simulator .....	82

# TeSys Tera DTM Library 시작하기

## SoMove 시작하기

TeSys Tera DTM Library 인스턴스를 만들려면, 먼저 SoMove software를 실행해야 합니다.

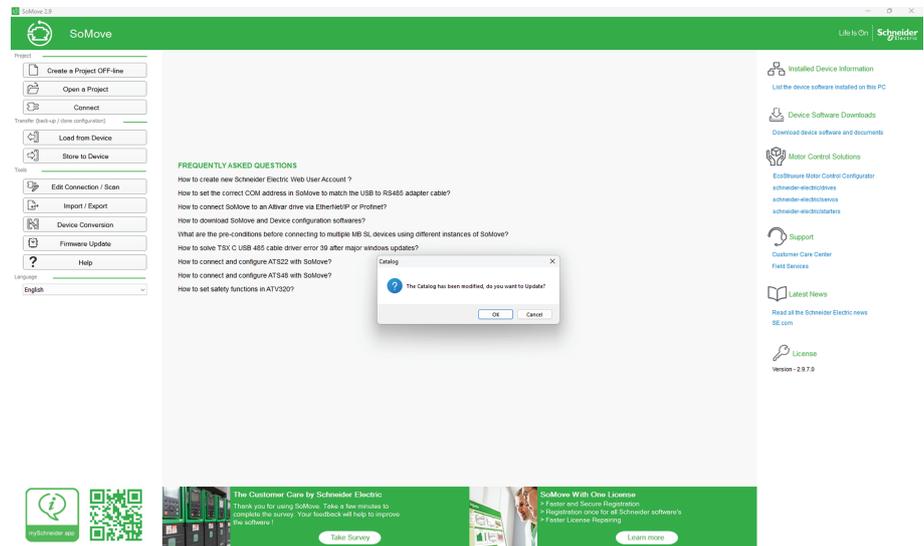
SoMove software를 시작하는 방법에 대한 자세한 내용은 SoMove 온라인 도움말을 참조하십시오.

## TeSys Tera DTM의 인스턴스 생성하기

다음 절차에서는 TeSys Tera DTM Library의 인스턴스를 생성하는 방법에 대해 설명합니다.

1. SoMove software를 여십시오.

결과: 다음 창이 표시됩니다.

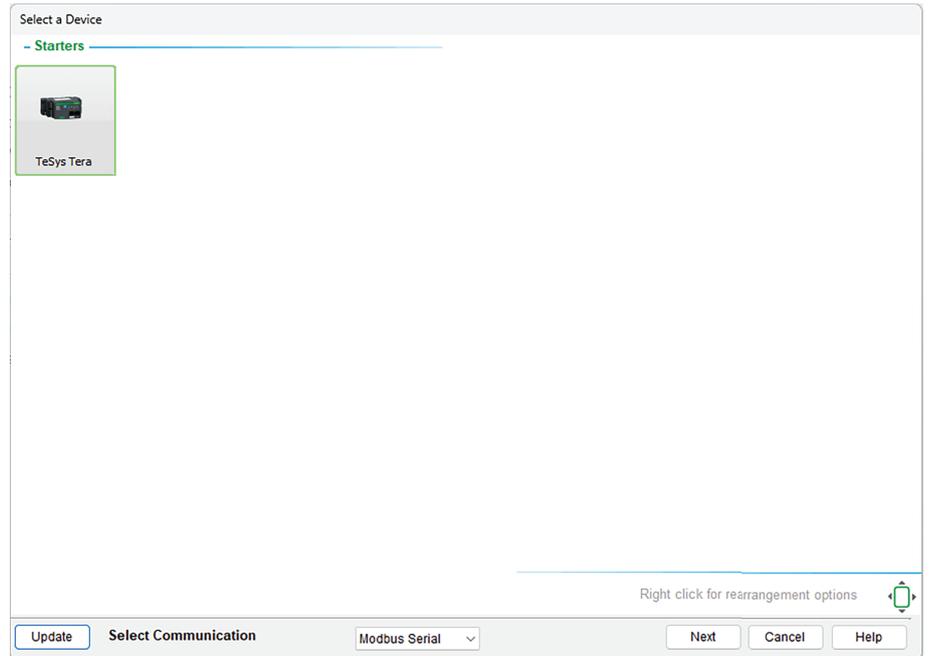


2. **OK(확인)**를 선택하여 카탈로그를 업데이트합니다.

**주의:** TeSys Tera DTM Library를 설치하고 SoMove software를 시작하면 **Catalog(카탈로그)** 창이 처음으로 나타납니다.

3. **Create a Project OFF-line(오프라인 프로젝트 생성)**을 선택합니다.

결과: **Select a Device(장치 선택)** 창이 나타납니다.



4. 선택 **TeSys Tera**을 선택하고, 옆의 드롭다운 목록을 사용하여 필요한 통신을 선택하십시오. **통신 선택**. 다음을 클릭하십시오.

결과: TeSys Tera system 작업 공간이 열립니다.

주의:

- 장치가 나타나지 않으면 문제 해결, 22 페이지항목을 참조하십시오.
- 통신이 선택될 때 **Modbus TCP**, 오직 EtherNet/IP 변형들만이 LTMT main unit 의 변형만 구성할 수 있습니다.

## 네트워크에 장치 연결하기

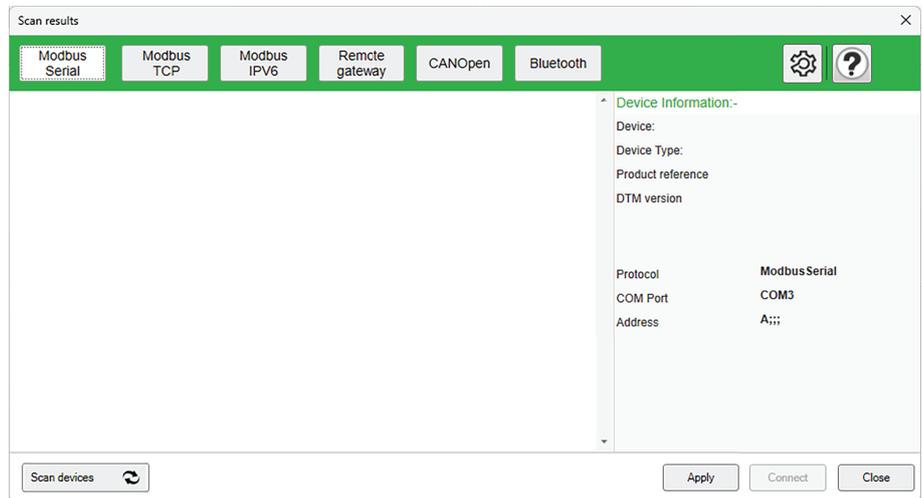
이 행동은 연결된 장치의 구성을 가져옵니다. SoMove software는 세션 동안 이 장치에 연결된 상태를 유지합니다.

## Modbus 직렬

다음 절차는 Modbus 시리얼 장치에 연결하는 방법을 설명합니다. **Modbus 시리얼 통신** 통신을 통해 연결하는 방법을 설명합니다:

1. SoMove software를 여십시오.
2. 다음 방법 중 하나를 사용하여 장치에 연결합니다.
  - 시작 페이지에서 **Edit Connection/Scan(연결 편집/스캔)**을 선택합니다.
  - 도구 모음에서  아이콘을 선택합니다.
  - 메뉴 모음에서 **Communication(통신) > Edit Connection/Scan(연결 편집/스캔)**을 선택합니다.

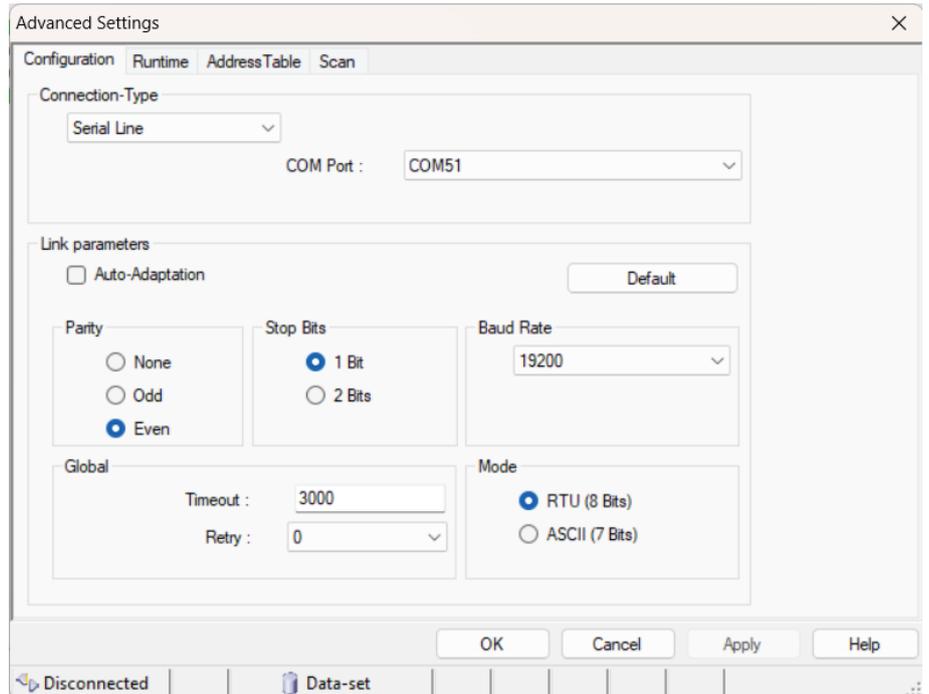
결과: **Scan results(스캔 결과)** 창이 나타납니다.



3. 스캔 결과에서 스캔 결과 대화 상자에서 **Modbus Serial**.

4.  아이콘을 선택합니다.

결과: **Advanced Settings(고급 설정)** 창이 나타납니다.



주의: 장치의 공장 설정은 다음과 같습니다(HMI 또는 Modbus RTU 포트에 해당).

- **Parity(패리티):** 짝수
- **Stop Bits(정지 비트):** 1비트
- **Baud rate(전송 속도):** 19200

5. **Configuration(구성)** 탭에서 Modbus 직렬 연결 유형에 필요한 구성 설정을 입력합니다.

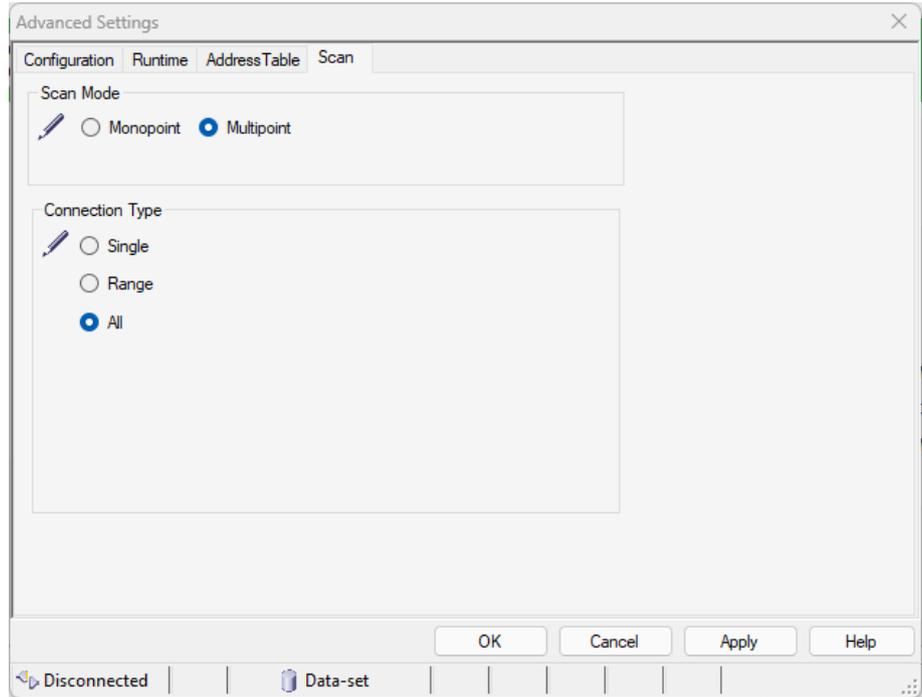
주의: 기본 매개 변수 설정을 사용할 수 없는 경우

**Auto-Adaptation(자동 조정)** 확인란을 선택하여 장치를 검색합니다.

구성 설정에 대한 자세한 내용은 *SoMove* 온라인 도움말의 **Connection(연결)** 단원을 참조하십시오.

6. **Scan(스캔)** 탭을 선택합니다.

결과: **Advanced Settings(고급 설정)** 창에 **Scan(스캔)** 탭이 나타납니다.



주의: Modbus 연결에는 **Multipoint Scan Mode(다지점 스캔 모드)**만 지원됩니다.

7. **Scan Mode(스캔 모드)** 및 **Connection Type(연결 유형)**을 선택합니다.

자세한 내용은 SoMove 온라인 도움말의 Connection(연결) 단원을 참조하십시오.

8. **OK(확인)**를 선택합니다.

결과: 수정 사항이 저장되고 **Advanced Settings(고급 설정)** 창이 닫힙니다. 다음 스캔에서 새로운 값이 적용됩니다.

주의: **Cancel(취소)**을 선택하면, **Advanced Settings(고급 설정)** 창은 어떠한 변경도 없이 닫히며 다음 스캔에서 기본값이 적용됩니다.

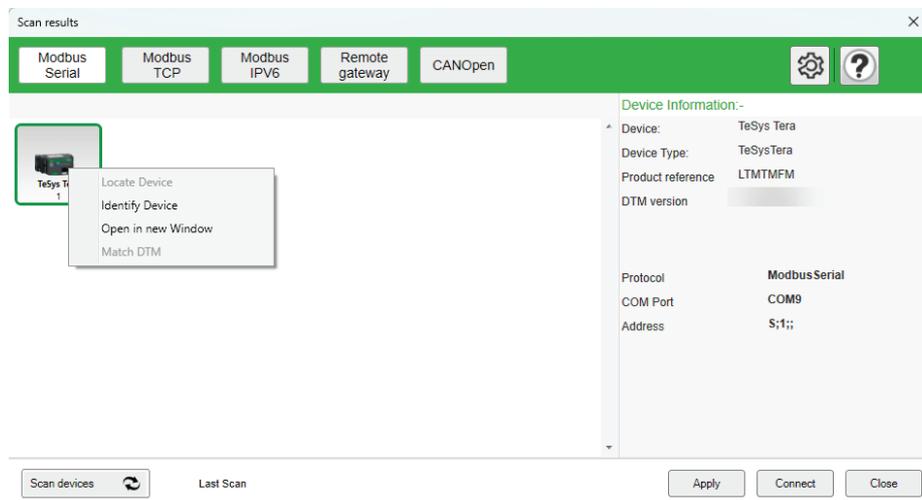
9. **Scan devices(장치 스캔)**를 선택합니다.

결과: 네트워크에서 Modbus 직렬 연결을 사용하는 모든 장치가 표시됩니다.

10. 적합한 TeSys Tera system을 선택하고 **Connect(연결)**를 선택합니다.

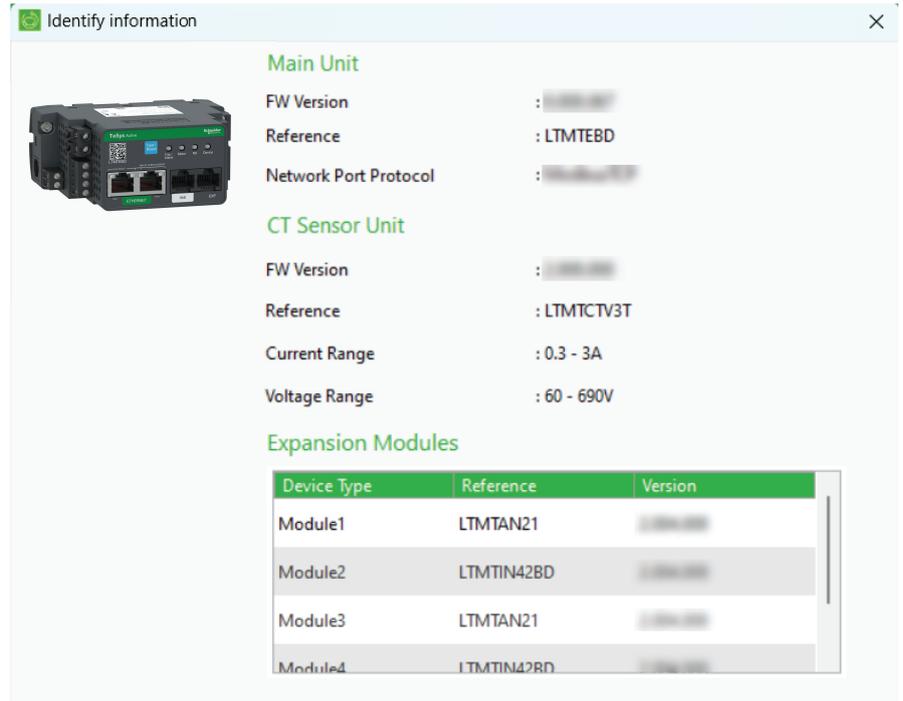
## 11. 장치를 마우스 오른쪽 버튼으로 클릭합니다.

결과: **Scan results(스캔 결과)** 창이 나타납니다.



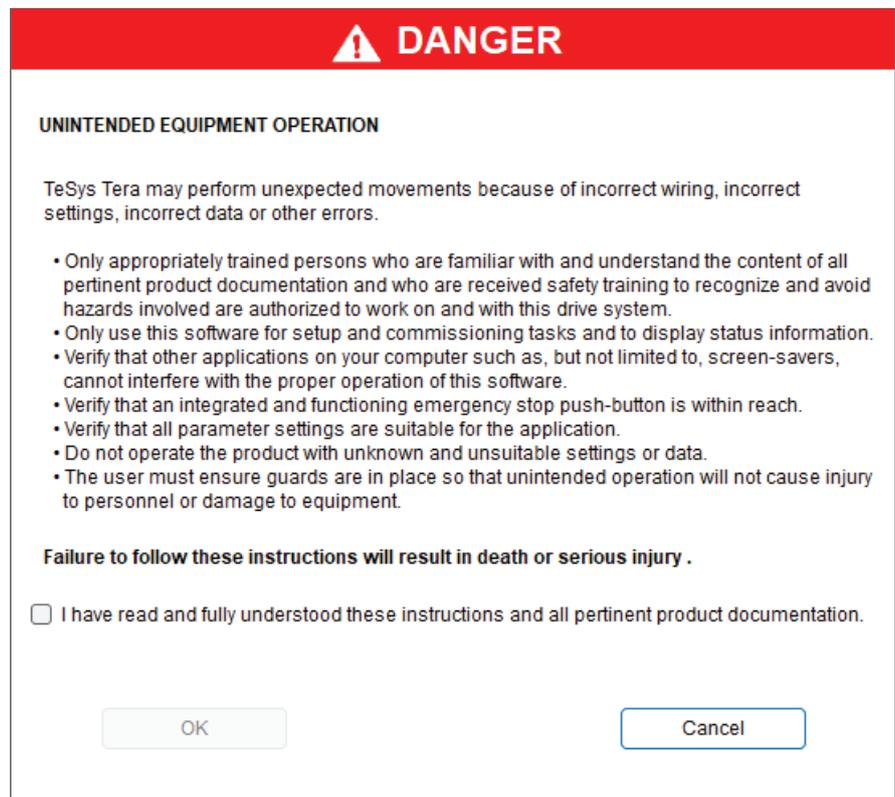
## 12. 선택 장치 식별.

The **정보 확인** 창이 나타납니다. 창에는 연결된 장치의 세부 정보가 표시됩니다.



13. 연결을 위해 식별된 장치를 선택하고 **Connect(연결)**를 선택합니다.

14. 아래와 같이 위험 메시지가 표시됩니다. 안전 메시지에 따라 모든 요구 사항이 충족되는지 확인하십시오. 체크박스를 선택하고 클릭하세요 **확인**.



결과: TeSys Tera system이 PC에 연결되어 있습니다.

**주의:**

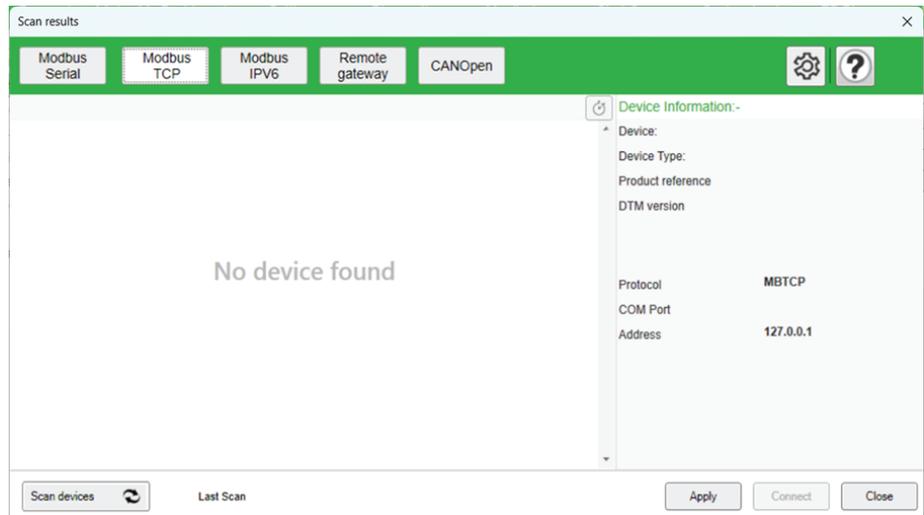
- 장치와 PC 간의 케이블 연결을 확인하십시오.
- 장치가 전원 장치에 연결되었는지를 확인하십시오.
- **Advanced Setting(고급 설정)**을 선택하여 연결 매개 변수가 올바르게 정의되었는지를 확인하십시오.

## 모드버스 TCP

다음 절차는 Modbus TCP를 사용하는 장치에 연결하는 방법을 설명합니다. **Modbus TCP 통신을 통해 통신을 통해 연결하는 방법을 설명합니다:**

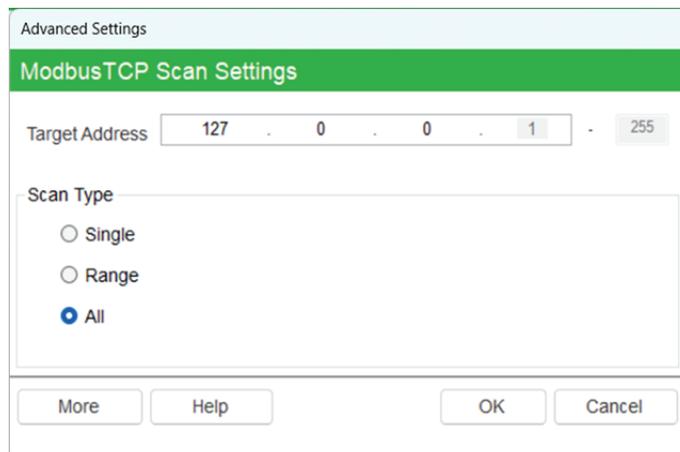
1. SoMove software를 여십시오.
2. 다음 방법 중 하나를 사용하여 장치에 연결합니다.
  - 시작 페이지에서 **Edit Connection/Scan(연결 편집/스캔)**을 선택합니다.
  - 도구 모음에서  아이콘을 선택합니다.
  - 메뉴 모음에서 **Communication(통신) > Edit Connection/Scan(연결 편집/스캔)**을 선택합니다.

**결과: Scan results(스캔 결과)** 창이 나타납니다.

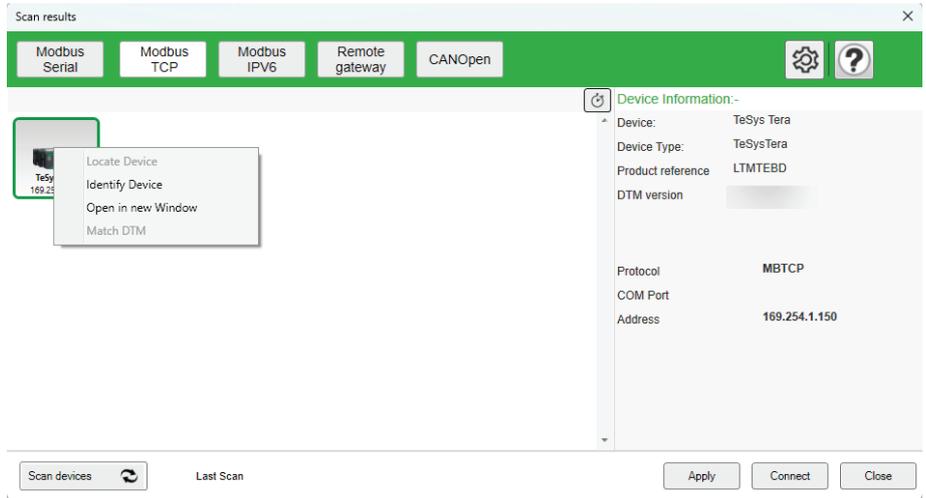


3. 스캔 결과에서 스캔 결과 대화 상자에서 **Modbus TCP**.
4.  아이콘을 선택합니다.

**결과: Advanced Settings(고급 설정)** 창이 나타납니다.

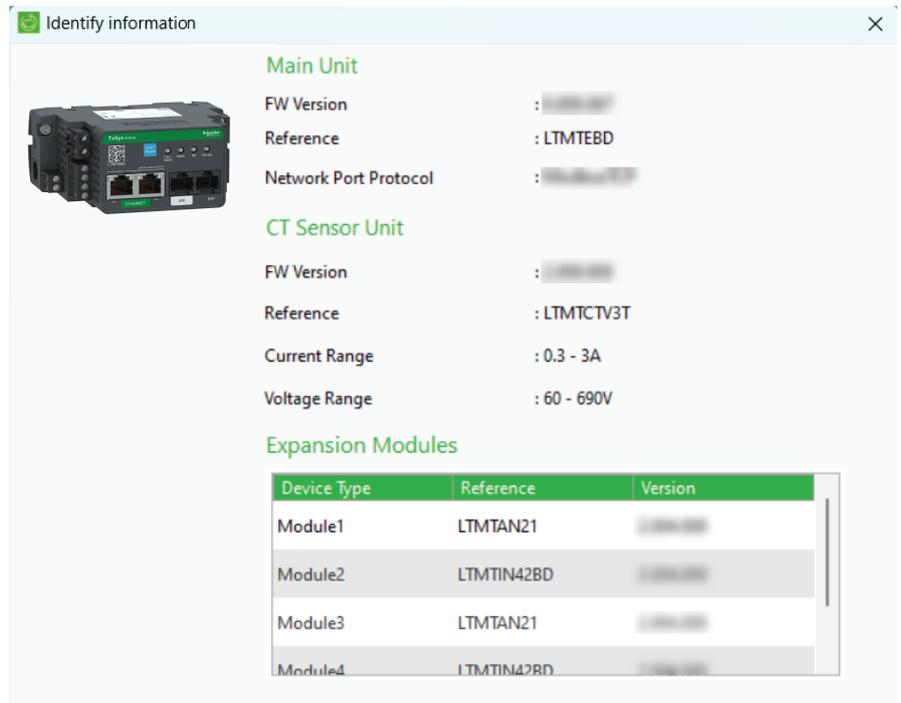


5. 고급 설정에서 **고급 설정** 창에서 **대상 주소** 을 입력하고 필요한 **스캔 유형**. 자세한 내용은 다음을 참조하십시오. *SoMove* 온라인 도움말.
6. **OK(확인)**를 선택합니다.  
**결과:** 수정 사항이 저장되고 **Advanced Settings(고급 설정)** 창이 닫힙니다. 다음 스캔에서 새로운 값이 적용됩니다.  
**주의: Cancel(취소)**을 선택하면, **Advanced Settings(고급 설정)** 창은 어떠한 변경도 없이 닫히며 다음 스캔에서 기본값이 적용됩니다.
7. **Scan devices(장치 스캔)**를 선택합니다.  
**결과:** 네트워크상의 모든 장치를 표시합니다. Modbus TCP 연결이 가능한 모든 장치를 표시합니다.
8. 적합한 TeSys Tera system을 선택하고 **Connect(연결)**를 선택합니다.
9. 장치를 마우스 오른쪽 버튼으로 클릭합니다.



10. 선택 장치 식별.

The **정보 확인** 창이 나타납니다. 창에는 연결된 장치의 세부 정보가 표시됩니다.



11. 연결을 위해 식별된 장치를 선택하고 **Connect(연결)**를 선택합니다.

12. 아래와 같이 위험 메시지가 표시됩니다. 안전 메시지에 따라 모든 요구 사항이 충족되는지 확인하십시오. 체크박스를 선택하고 클릭하세요 **확인**.

⚠ DANGER

**UNINTENDED EQUIPMENT OPERATION**

TeSys Tera may perform unexpected movements because of incorrect wiring, incorrect settings, incorrect data or other errors.

- Only appropriately trained persons who are familiar with and understand the content of all pertinent product documentation and who are received safety training to recognize and avoid hazards involved are authorized to work on and with this drive system.
- Only use this software for setup and commissioning tasks and to display status information.
- Verify that other applications on your computer such as, but not limited to, screen-savers, cannot interfere with the proper operation of this software.
- Verify that an integrated and functioning emergency stop push-button is within reach.
- Verify that all parameter settings are suitable for the application.
- Do not operate the product with unknown and unsuitable settings or data.
- The user must ensure guards are in place so that unintended operation will not cause injury to personnel or damage to equipment.

**Failure to follow these instructions will result in death or serious injury .**

I have read and fully understood these instructions and all pertinent product documentation.

OK

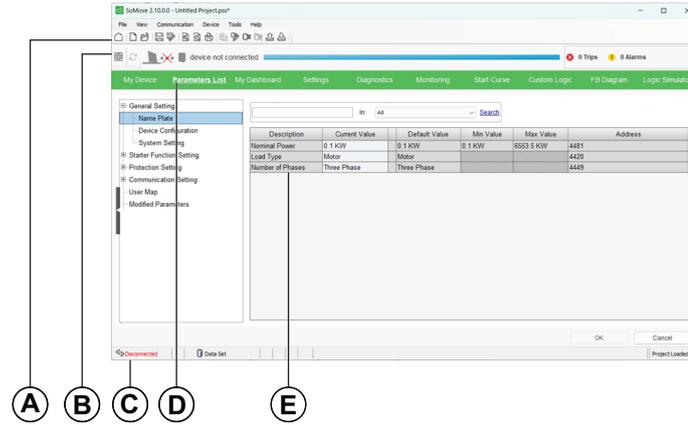
Cancel

**결과:** TeSys Tera system이 PC에 연결되어 있습니다.

**주의:**

- 장치와 PC 간의 케이블 연결을 확인하십시오.
- 장치가 전원 장치에 연결되었는지를 확인하십시오.
- **Advanced Setting(고급 설정)**을 선택하여 연결 매개 변수가 올바르게 정의되었는지를 확인하십시오.

# 사용자 인터페이스 설명



작업 공간은 다음과 같은 영역으로 구성되어 있습니다.

- A      메뉴 모음
- B      툴바
- C      상태바
- D      탭 선택 표시줄
- E      탭 영역(선택한 탭에 따른 내용)

# 메뉴 모음

## 설명

메뉴 모음은 SoMove software의 일부입니다. 작업 공간 상단의 메뉴 모음은 다음과 같습니다.



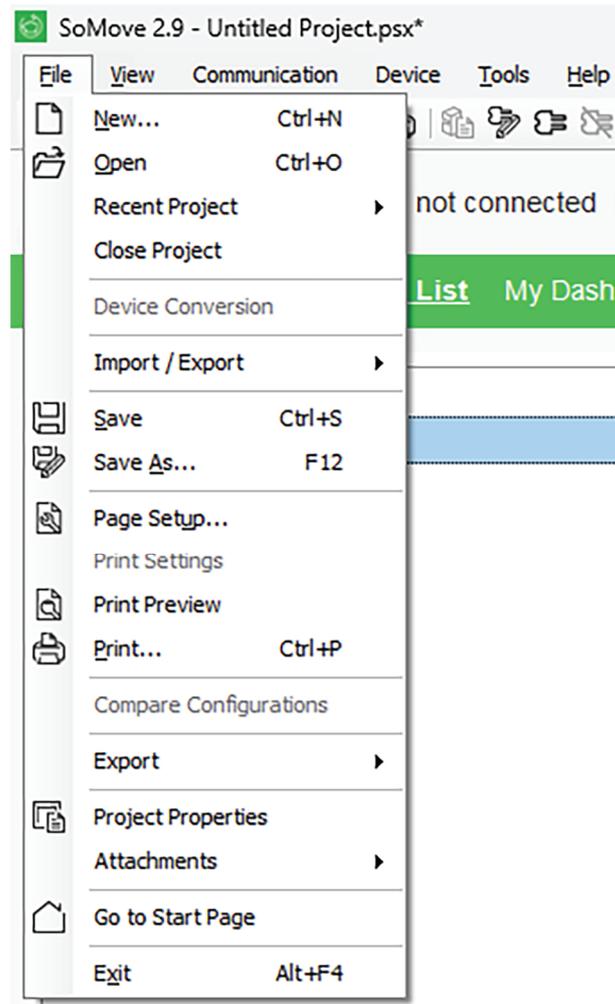
메뉴 바는 메뉴나 아이콘을 통해 사용자 기능에 접근할 수 있게 합니다.

메뉴 모음에서 회색으로 표시된 명령은 사용할 수 없는 명령입니다.

**주의:** SoMove software의 일부 옵션은 TeSys Tera DTM Library에 적용되지 않습니다. 사용할 수 없는 명령의 세부 정보는 설명 테이블에 나열되지 않습니다. SoMove software에서 사용할 수 있는 옵션에 대한 자세한 내용은 **SoMove** 온라인 도움말을 참조하십시오.

## 파일 메뉴

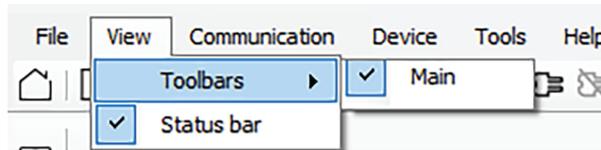
**File(파일)** 메뉴는 메뉴 모음에서 **File(파일)**을 클릭하여 액세스할 수 있습니다.



명령	설명
New...	새 프로젝트를 생성합니다.
Open	PC에 저장된 프로젝트를 엽니다.
Recent Project	최근에 연 프로젝트를 엽니다.
Close Project	저장하라는 메시지를 표시하여 현재 프로젝트를 닫습니다.
Save	기존 프로젝트에 대한 수정 내용을 저장할 수 있습니다.
Save As	열려 있는 프로젝트를 다른 이름 및/또는 새 위치에 저장합니다.
Project Properties	열려 있는 프로젝트의 속성을 표시합니다.
Go To Start Page	시작 페이지를 표시합니다.
Exit	SoMove software를 종료합니다.

## 보기 메뉴

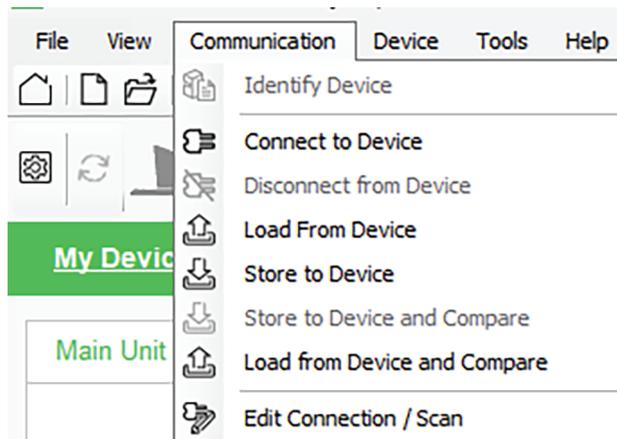
**View(보기)** 메뉴는 메뉴 모음에서 **View(보기)**를 클릭하여 액세스할 수 있습니다.



명령	설명
도구 모음 > 기본	기본 도구 모음을 표시하거나 숨깁니다.
Status bar	상태 표시줄을 표시하거나 숨깁니다.

## 통신 메뉴

**Communication(통신)** 메뉴는 메뉴 모음에서 **Communication(통신)**을 클릭하여 액세스할 수 있습니다.

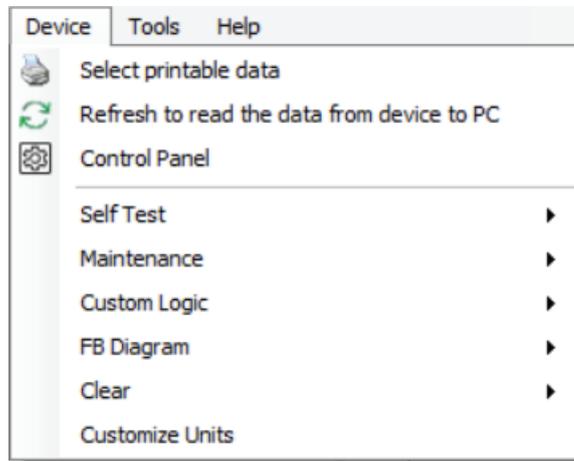


명령	설명
Identify Device	연결된 장치에 대한 다음 정보를 표시합니다. <ul style="list-style-type: none"> <li>• 장치 참조</li> <li>• 제조업체</li> <li>• Firmware version(펌웨어 버전)</li> </ul>
Connect to Device	연결된 장치의 구성을 검색하고 전체 세션 동안 연결된 상태를 유지합니다.

명령	설명
Disconnect from Device	연결된 장치에서 연결을 해제합니다.
Load From Device	연결된 장치의 구성을 가져옵니다. <b>주의:</b> 이 작업이 진행 중일 때는 DTM library 장치와의 연결이 끊어지지 않도록 하여 통신 손실을 방지하십시오.
Store to Device	PC에 저장된 기존 프로젝트의 구성 파일을 연결된 장치로 전송합니다. <b>주의:</b> 이 작업은 구성을 다운로드하는 데 1분 정도 걸립니다.
Load from Device and Compare	연결된 장치에서 구성을 가져와 현재 열려 있는 프로젝트 구성과 비교합니다.
Edit Connection / Scan	연결 매개 변수를 표시하고 연결 테스트를 수행합니다.

## 장치 메뉴

**Device(장치)** 메뉴는 메뉴 모음에서 **Device(장치)**를 클릭하여 액세스할 수 있습니다.

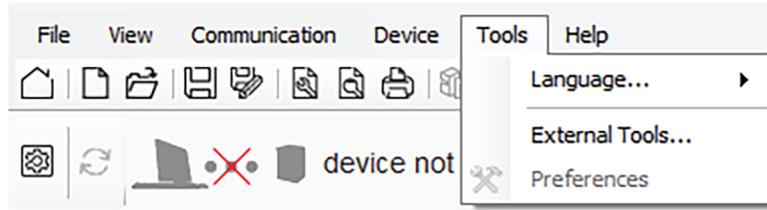


명령	설명
Select printable data	인쇄에 필요한 데이터를 선택할 수 있습니다.
Refresh to read the data from device to PC	매개 변수 목록 탭의 모든 매개 변수를 새로 고칩니다.
Control Panel	제어 패널을 표시하거나 숨깁니다.
자체 테스트 <b>주의:</b> 자가 진단은 온라인 모드에서만 이용 가능합니다.	자체 테스트를 수행할 수 있습니다.
• 자체 테스트(트립과 함께) 명령	트립 기능으로 장치의 진단 테스트를 시작할 수 있습니다.
• 자체 테스트(트립 없음) 명령	트립 없이 장치의 진단 테스트를 시작할 수 있습니다.
유지보수 • 펌웨어 업데이트 • 날짜 및 시간 업데이트 <b>주의:</b> 날짜 및 시간 업데이트 명령은 LTMT main unit가 직렬 포트를 통해 Ethernet 버전입니다.	장치 유지보수 도구에 대한 접근을 제공합니다 펌웨어 업데이트 명령에 액세스합니다. • 날짜와 시간을 설정할 수 있습니다. 다음 옵션을 사용할 수 있습니다. ◦ 시스템 ◦ 사용자 지정 <b>주의:</b> 시스템 시간대가 변경되는 동안 SoMove 실행 중 변경된 경우, SoMove 을 다시 시작하여 업데이트된 PC 시간대를 적용하십시오.

명령	설명
<p>사용자 정의 논리</p> <ul style="list-style-type: none"> <li>• New Custom Logic</li> <li>• Open Custom Logic</li> <li>• Save Custom Logic</li> <li>• SaveAs Custom Logic</li> <li>• Close Custom Logic</li> <li>• Compile Custom Logic</li> </ul>	<p>사용자 정의 논리 언어로 사용자 정의된 프로그램에 대한 Custom Logic Editor에 액세스할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 새 사용자 정의 논리 프로그램 파일을 엽니다.</li> <li>• 기존 사용자 정의 논리 프로그램 파일을 엽니다.</li> <li>• 사용자 정의 로직 프로그램 파일을 *.lf 파일로 저장합니다.</li> <li>• 사용자 정의 로직 프로그램 파일을 다른 위치에 새 파일 이름으로 저장합니다.</li> <li>• 저장하라는 메시지를 표시하여 사용자 정의 논리 프로그램 파일을 닫습니다.</li> <li>• 표준을 기준으로 사용자 정의 논리 프로그램에 작성된 코드의 유효성을 검사합니다.</li> </ul>
<p>FB Diagram</p> <ul style="list-style-type: none"> <li>• New FB Diagram</li> <li>• Open FB Diagram</li> <li>• Save FB Diagram As</li> <li>• Compile FB Diagram to Custom Logic</li> <li>• FBD 편집기</li> <li>• View</li> <li>• Tools</li> </ul>	<p>기능 블록 다이어그램(FBD) 언어로 사용자 정의된 프로그램의 Custom Logic Editor에 액세스할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 새로운 FB 다이어그램을 생성합니다.</li> <li>• 기존 FB 다이어그램 파일을 엽니다.</li> <li>• FB 다이어그램 파일을 *.gef 파일로 저장합니다.</li> <li>• FB Diagram을 컴파일하고 사용자 정의 논리로 변환합니다.</li> <li>• FB 다이어그램을 편집할 수 있습니다.</li> <li>• 사용 가능한 다양한 보기 옵션을 사용하여 FB Diagram을 볼 수 있습니다. 자세한 내용은 FBD 편집기 표시 옵션, 190 페이지 항목을 참조하십시오.</li> <li>• 링크를 변경하거나 기능 블록의 번호를 재지정하는 도구를 표시합니다.</li> </ul>
<p>지우기 &gt; 열적 메모리 지우기</p> <p>주의: 클리어 서멀 메모리는 온라인 모드에서만 이용 가능합니다.</p>	<p>장치 메모리에 저장된 열 데이터를 재설정할 수 있습니다.</p>
<p>Customize Units</p>	<p>다음 표시 단위를 구성할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 온도 (섭씨 또는 화씨)</li> <li>• 전동기 등급 (KW 또는 HP)</li> </ul>

## 도구 메뉴

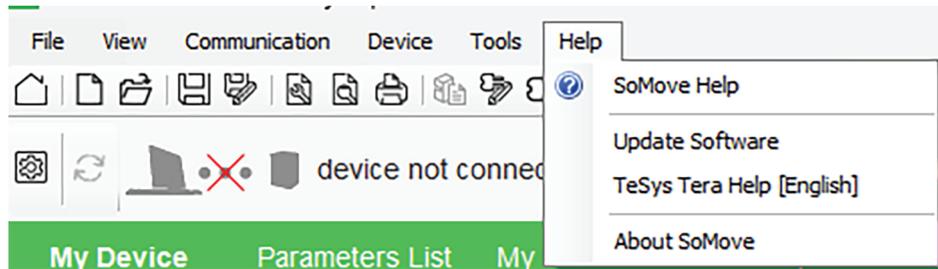
**Tools(도구)** 메뉴는 메뉴 모음에서 **Tools(도구)**를 클릭하여 액세스할 수 있습니다.



도구 메뉴의 명령	설명
언어	디스플레이 언어를 수정합니다. SoMove 소프트웨어의 표시 언어를 변경합니다. 주의: The TeSys Tera DTM Library 영어만 지원합니다.

## 도움말 메뉴

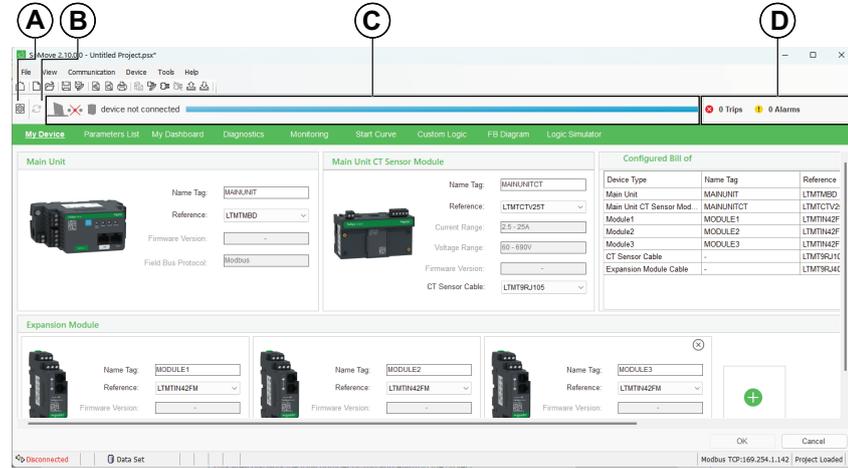
**Help(도움말)** 메뉴는 메뉴 모음에서 **Help(도움말)**를 클릭하여 액세스할 수 있습니다.



명령	설명
SoMove Help	SoMove 온라인 도움말을 표시합니다.
Update Software	SoMove software를 최신 버전으로 업데이트합니다.
TeSys Tera 도움말 [영어]	온라인 도움말을 엽니다. TeSys Tera DTM 온라인 도움말을 엽니다.
About SoMove	SoMove software에 대한 일반 정보를 표시합니다.

# 툴바 설명

메뉴 모음 바로 아래, 작업 공간 상단에 있는 툴바는 TeSys Tera DTM Library 전용입니다.

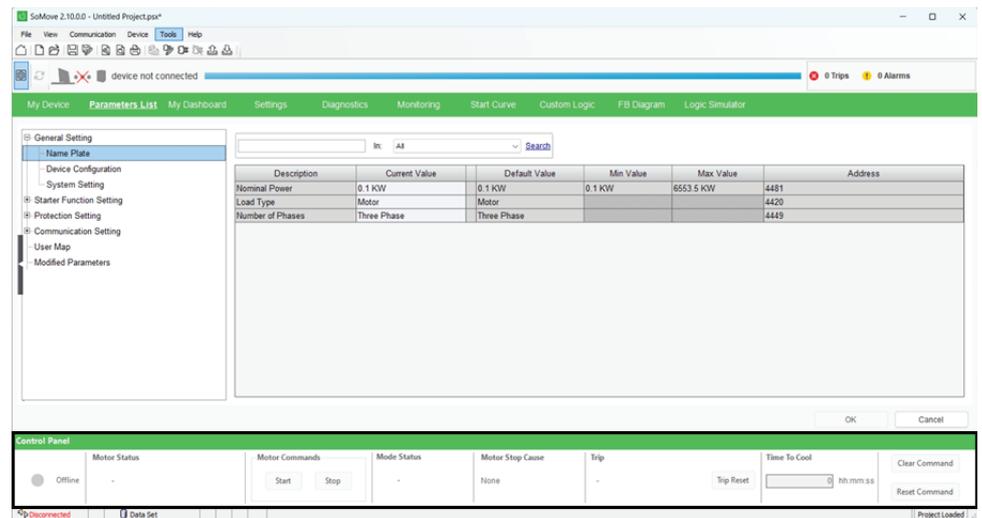


- A 제어 패널
- B 새로 고침
- C 동기화 데이터 영역
- D 여행 및 경보 횟수 표시 영역

## 제어 패널

Control Panel(제어 패널)에 액세스하려면 툴바에서 을 선택합니다.

Control Panel(제어 패널)이 창 하단에 나타납니다.



Control Panel(제어 패널)에 다음 매개 변수가 표시됩니다.

- 장치 상태
- 모터 상태
- 모드 상태

- 모터 정지 원인
- 트립 상태
- 모터가 정지되거나 트립될 때 생각하는 시간

**Control Panel(제어 패널)**에서 다음 작업을 수행할 수 있습니다.

- 모터 명령(시작/정지) 제공
- 트립 초기화
- 지우기 명령
- 초기화 명령

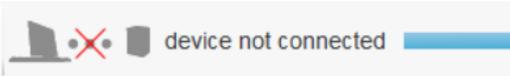
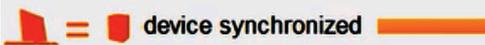
## 새로 고침

틀바에 있는  옵션은 **Parameter List(매개 변수 목록)** 탭 아래의 모든 매개 변수를 새로 고칩니다.

## 동기화 데이터 영역

동기화 데이터 영역에는 LTMT main unit과 PC 간의 데이터 동기화 상태가 표시됩니다.

LTMT main unit이 연결 모드인 경우 표시된 데이터가 자동으로 동기화됩니다.

모드	아이콘	설명
연결 해제		<p>LTMT main unit이 PC와 동기화되지 않습니다.</p> <ul style="list-style-type: none"> <li>• 매개 변수 목록 머리글 및 동기화 데이터 영역은 파란색입니다.</li> <li>• LTMT main unit에서 매개 변수를 실시간으로 읽지 않습니다.</li> <li>• 모든 설정은 구성 모드에서와 같이 수정할 수 있습니다.</li> <li>• <b>Store to Device(장치에 저장)</b> 작업을 수행하여 장치에 설정을 저장해야 합니다. 이 설정을 저장하려면 프로젝트를 저장해야 합니다.</li> </ul>
연결		<p>LTMT main unit이 PC와 동기화됩니다.</p> <ul style="list-style-type: none"> <li>• 매개 변수 목록 머리글 및 동기화 데이터 영역은 주황색입니다.</li> <li>• LTMT main unit에서 표시된 매개 변수를 실시간으로 읽습니다.</li> <li>• 수정된 매개 변수는 확인 시 장치에 기록됩니다.</li> </ul>

## 여행 및 경보 횟수 표시

이 영역은 프로젝트 내 총 트립 및 알람 건수를 표시합니다.

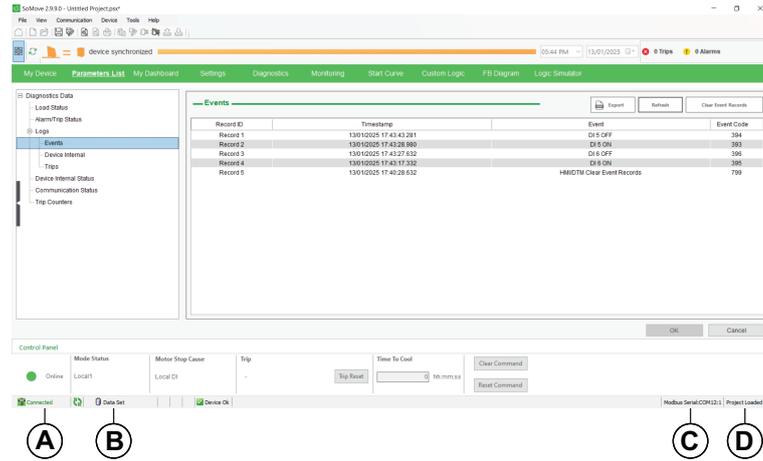


# 상태바

## 목적

작업 공간 하단의 상태 표시줄에는 LTMT main unit의 현재 상태와 SoMove software 관련 정보가 표시됩니다. 상태 표시줄에 대한 자세한 내용은 SoMove 온라인 도움말을 참조하십시오.

## 설명



- A 연결 상태
- B 데이터 출처
- C 특정 메시지 영역
- D 프로젝트 상태

## 연결 상태

The 연결 상태 은 LTMT main unit 와 PC 간의 연결 모드를 나타냅니다:

모드	아이콘	설명
연결 해제		LTMT main unit이 PC에 연결되어 있지 않습니다.
장애		LTMT main unit과 PC 간의 연결이 손실되었습니다.
연결		LTMT main unit이 PC에 연결되어 있습니다.

## 데이터 출처

**Data Source(데이터 출처)**는 데이터의 현재 모드를 나타냅니다. 다음 표에는 데이터 출처 모드가 설명되어 있습니다.

모드	아이콘	설명
데이터 세트		연결 모드에서 데이터 세트의 매개 변수 값을 수정할 수 있습니다.
데이터 세트 또는 장치 잠금		연결 모드에서 장치가 보호됩니다.
장치		장치 데이터 세트가 정보를 포함하고 있습니다.
장치/데이터 세트		모든 매개 변수 값이 두 데이터 출처 모두에 저장됩니다.

## 특정 메시지 영역

장치와 통신하는 데 사용되는 Fieldbus에 따라 특정 메시지 영역에 IP 또는 장치 주소, COM 포트 세부 정보 등이 표시됩니다.

## 프로젝트 상태

SoMove software 프로젝트의 상태는 다음과 같습니다.

- **Project Loaded:** 작업 공간에 프로젝트가 표시됩니다.
- **No Project Open:** 프로젝트 작업 공간이 비어 있습니다.

# 탭 영역

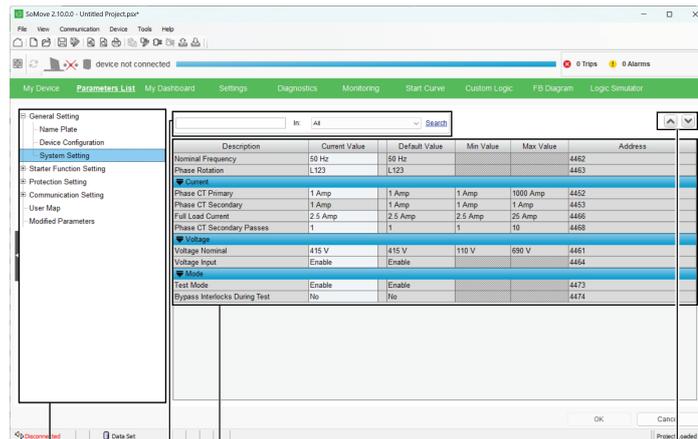
## 개요

아래 표에는 사용 가능한 다양한 탭이 나와 있습니다.

탭 이름	설명
<b>My Device</b>	장치 모듈 및 특성을 표시합니다.
<b>매개변수 목록</b>	모든 장치 모듈의 구성 가능한 매개 변수를 표시합니다.
<b>설정</b> 주의: 설정 설정 탭은 Ethernet의 LTMT main unit에서 내 장치 탭에서	구성 가능한 Modbus TCP 및 EtherNet/IP 설정을 LTMT main unit.
<b>My Dashboard</b>	요구 사항에 따라 사용자 정의할 수 있는 모든 장치 모듈의 구성 가능한 매개 변수와 상태 매개 변수를 표시합니다.
<b>Diagnostic</b>	진단 상태 데이터를 표시합니다.
<b>Monitoring</b>	장치 모듈 모니터링 데이터를 표시합니다.
<b>Start Curve</b>	모터 시작 특성을 그래픽 형식으로 표시합니다.
<b>Custom Logic</b>	사용자 정의 논리 언어로 사용자 정의된 프로그램의 Custom Logic Editor에 액세스할 수 있습니다.
<b>FB Diagram</b>	FBD 언어로 작성된 사용자 정의 프로그램의 사용자 정의 로직 편집기에 대한 접근 권한을 제공합니다.
<b>Logic Simulator</b>	사용자 정의 논리 언어로 사용자 정의 프로그램의 논리 시뮬레이터에 액세스할 수 있습니다.

## 설명

창에는 이러한 탭의 공통 정보가 표시됩니다.



- A
- B
- C
- D

- A 트리 보기
- B 검색 기능
- C 표시 영역
- D 확장 및 축소 버튼

## 트리 보기

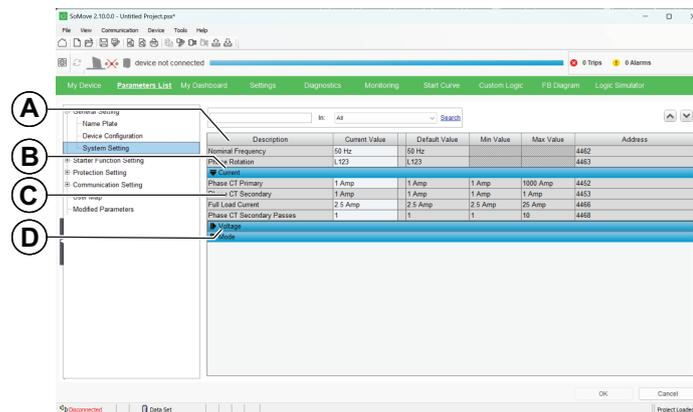
트리 보기는 항목과 하위 항목으로 구분된 매개 변수 그룹으로 구성됩니다. 트리에서 항목 또는 하위 항목을 선택하여 표시 영역에 매개 변수를 표시합니다.

## 검색 기능

다음 단계에서는 표시된 테이블에서 특정 텍스트를 찾는 방법을 설명합니다.

1. 검색 표시줄의 첫 번째 필드에 검색할 문자(단어, 코드 또는 단위의 일부)를 입력합니다.
2. 드롭다운 목록에서 검색할 열을 선택합니다.  
**All(모두)** 옵션을 선택하면 테이블의 모든 열에서 검색이 수행됩니다.
3. **Search(검색)**를 선택합니다.
  - 발견된 첫 번째 일치하는 텍스트가 강조 표시됩니다.
  - 다른 인스턴스를 검색하려면 **Search(검색)**를 선택합니다.
  - 일치하는 텍스트가 발견되지 않으면 검색 필드의 문자 색이 빨간색으로 바뀝니다.

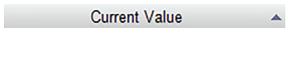
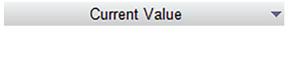
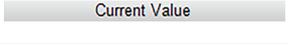
## 그리드 보기의 표시 영역



- A 열 머리글
- B 매개 변수 그룹
- C 매개 변수:
  - 한 행에 하나씩 매개 변수가 표시되며, 여러 필드에 매개 변수 속성이 있습니다.
  - 흰색 셀은 내용을 수정할 수 있으며 회색 셀은 읽기 전용입니다.
- D 축소 또는 확장 아이콘: 매개 변수 그룹을 축소하거나 확장하려면 해당 그룹의 화살표를 클릭합니다.

## 매개 변수 정렬하기

열의 값에 따라 매개 변수를 정렬하려면 머리글을 클릭합니다.

조치	결과	머리글 예
한 번 클릭	<ul style="list-style-type: none"> <li>매개 변수 값이 오름차순으로 정렬됩니다.</li> <li>머리글에 위쪽을 가리키는 화살표가 나타납니다.</li> </ul>	
두 번 클릭	<ul style="list-style-type: none"> <li>매개 변수 값이 내림차순으로 정렬됩니다.</li> <li>머리글에 아래쪽을 가리키는 화살표가 나타납니다.</li> </ul>	
세 번 클릭	<ul style="list-style-type: none"> <li>매개 변수가 초기 순서로 표시됩니다.</li> <li>머리글은 초기 표현에 따라 나타납니다.</li> </ul>	

## 열 순서 수정하기

다음 단계에서는 표시 영역에서 열의 순서를 수정하는 방법을 설명합니다.

1. 열의 머리글을 클릭합니다.
2. 열을 원하는 위치로 끕니다.

## 확장 및 축소 버튼

표시 영역의 오른쪽 상단 모서리에 있는 다음과 같은 버튼을 사용하여 표시 영역의 보기를 수정할 수 있습니다.

버튼	기능	설명
	모두 확장	모든 그룹을 확장하여 모든 매개 변수를 표시합니다.
	모두 축소	표시 영역의 모든 그룹을 축소합니다.

# My Device

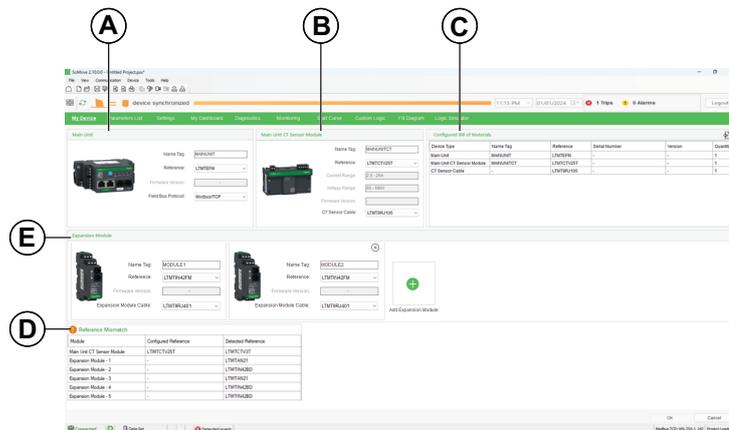
## 개요

**My Device** 탭은 다음 용도로 사용됩니다:

- 연결 해제 모드에서 TeSys Tera system을 구성합니다.
- 연결 모드에서 TeSys Tera system 특성을 표시합니다.

## 설명

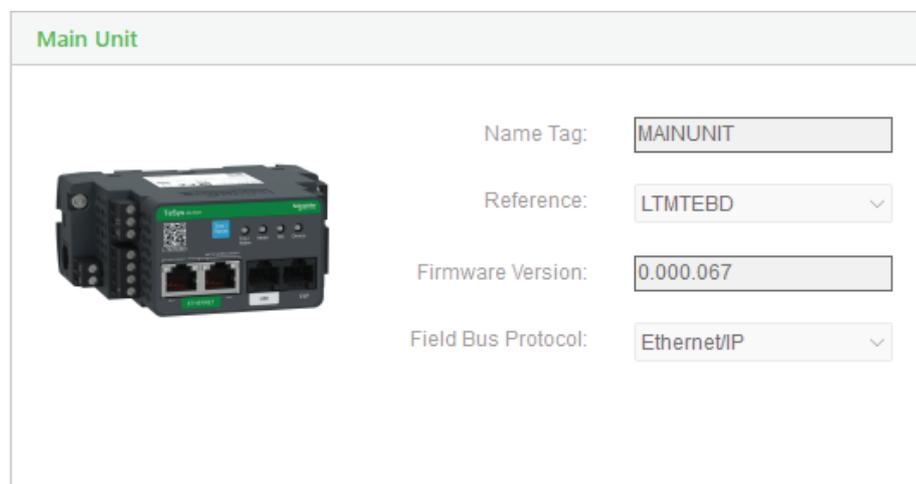
이 그림은 **My Device** 탭의 다양한 섹션을 보여 줍니다.



- A 주 장치
- B 주 장치 CT 센서 모듈
- C 구성된 BOM
- D 참조 불일치
- E 확장 모듈

## 주 장치

The **메인 유닛** 섹션은 LTMT main unit 구성된 TeSys Tera system.



**Main Unit(주 장치)** 섹션에서 다음 특성을 사용할 수 있습니다.

특성	설명
이름 태그	LTMT main unit의 이름을 입력합니다. 주의: <ul style="list-style-type: none"> <li>이름표 이름표 필드는 비워 둘 수 없습니다.</li> <li><b>Name Tag(이름 태그)</b> 필드는 다음 문자를 지원합니다.                             <ul style="list-style-type: none"> <li>알파벳: A부터 Z까지 (대문자)</li> <li>숫자: 1~9</li> <li>기호: _ /, 그리고 -</li> </ul> </li> <li>이름표 이름표 파일은 최대 10자까지 지원합니다.</li> </ul>
레퍼런스	LTMT main unit의 레퍼런스를 선택합니다. 주의: LTMTEFM이 기본값으로 선택됨 LTMT main unit 참조로 선택됩니다.
펌웨어 버전	선택한 LTMT main unit의 펌웨어 버전을 표시합니다. 주의: 펌웨어 버전은 장치가 온라인 상태일 때만 표시됩니다.
필드버스 프로토콜	사용 중인 네트워크 프로토콜을 표시합니다. LTMT main unit. 다음 옵션은 Ethernet 의 변형에 대해 선택할 수 있습니다. LTMT main unit: <ul style="list-style-type: none"> <li>Modbus/TCP</li> <li>EtherNet/IP</li> </ul>

주의: DTM library에 구성된 LTMT main unit 레퍼런스가 실제 장치 구성과 다른 경우 오류 메시지가 표시되며 연결이 설정되지 않습니다.

## Main Unit CT Sensor Module(주 장치 CT 센서 모듈)

The 메인 유닛 CT 센서 모듈 섹션은 TeSys Tera system.

Main Unit CT Sensor Module



Name Tag:

Reference:  ▼

Current Range:

Voltage Range:

Firmware Version:

CT Sensor Cable:  ▼

Main Unit CT Sensor Module(주 장치 CT 센서 모듈) 섹션에서 다음 특성을 사용할 수 있습니다.

특성	설명
이름 태그	<b>Main Unit CT Sensor Module(주 장치 CT 센서 모듈)</b> 의 이름을 입력합니다. <b>주의:</b> <ul style="list-style-type: none"> <li>이름표 이름표 필드는 비워 둘 수 없습니다.</li> <li><b>Name Tag(이름 태그)</b> 필드는 다음 문자를 지원합니다. <ul style="list-style-type: none"> <li>알파벳: A부터 Z까지 (대문자)</li> <li>숫자: 1~9</li> <li>기호: _ , /, 그리고 -</li> </ul> </li> <li>이름표 이름표 파일은 최대 10자까지 지원합니다.</li> </ul>
레퍼런스	<b>Main Unit CT Sensor Module(주 장치 CT 센서 모듈)</b> 의 레퍼런스를 선택합니다.
Current Range(전류 범위)	선택한 센서 모듈의 전류 범위를 표시합니다.
전압 범위	선택한 센서 모듈의 전압 범위를 표시합니다. <b>주의:</b> 전압 범위는 전압 파생 모델에 대해서만 표시됩니다.
펌웨어 버전	선택한 주 장치 CT 센서 모듈의 펌웨어 버전을 표시합니다. <b>주의:</b> 펌웨어 버전은 장치가 온라인 상태일 때만 표시됩니다.
CT Sensor 케이블	사용된 센서 케이블의 유형을 선택합니다.

## Expansion Module

확장 모듈 확장 모듈 섹션은 TeSys Tera system.

**Expansion Module(확장 모듈)** 섹션에서 다음 특성을 사용할 수 있습니다.

특성	설명
이름 태그	<b>LTMT expansion module</b> 의 이름을 입력합니다. <b>주의:</b> <ul style="list-style-type: none"> <li>이름표 이름표 필드는 비워 둘 수 없습니다.</li> <li><b>Name Tag(이름 태그)</b> 필드는 다음 문자를 지원합니다. <ul style="list-style-type: none"> <li>알파벳: A부터 Z까지 (대문자)</li> <li>숫자: 1~9</li> <li>기호: _ , /, 그리고 -</li> </ul> </li> <li>이름표 이름표 파일은 최대 10자까지 지원합니다.</li> </ul>
레퍼런스	<b>LTMT expansion module</b> 의 레퍼런스를 선택합니다.
펌웨어 버전	<b>LTMT expansion module</b> 의 펌웨어 버전을 표시합니다. <b>주의:</b> 펌웨어 버전은 장치가 온라인 상태일 때 표시됩니다.
확장 모듈 케이블	연결에 사용된 케이블의 참조를 선택하십시오. <b>LTMT expansion module</b> .

**Add Expansion Module(확장 모듈 추가)** 카드를 선택하여 최대 5개의 **LTMT expansion modules**을 추가할 수 있습니다.

다음 표에서는 TeSys Tera system에 추가할 수 있는 **LTMT expansion modules**의 최대 수를 보여 줍니다.

LTMT expansion modules 레퍼런스	최대 장치 수
LTMTIN42FM/LTMTIN42BD	5
LTMTAN21	2

## 참조 불일치

참조 불일치 참조 불일치 섹션은 구성된 장치와 식별된 장치 간의 차이를 보여줍니다.

! Reference Mismatch		
Module	Configured Reference	Detected Reference
Main Unit CT Sensor Module	LTMTCTV25T	LTMTCTV3T
Expansion Module - 1	-	LTMTAN21
Expansion Module - 2	-	LTMTIN42BD
Expansion Module - 3	-	LTMTAN21
Expansion Module - 4	-	LTMTIN42BD
Expansion Module - 5	-	LTMTIN42BD

주의: 참조 불일치 참조 불일치 섹션은 온라인 모드에서만 표시됩니다.

## Configured Bill of Materials

Configured Bill of Materials(구성된 BOM)에는 TeSys Tera system에 구성된 모든 장치 및 부속품의 목록이 표시됩니다.

Configured Bill of Materials 					
Device Type	Name Tag	Reference	Serial Number	Version	Quantity
Main Unit	MAINUNIT	LTMTEBD	LTMTEBDEX900003	0.000.067	1
Main Unit CT Sensor Module	MAINUNITCT	LTMTCTV3T	LTMTCTV3TEX900004	2.000.000	1
Module1	MODULE1	LTMTAN21	LTMTAN21DY900107	2.004.000	1
Module2	MODULE2	LTMTIN42BD	LTMTIN42BDDX900042	2.004.000	1
Module3	MODULE3	LTMTAN21	LTMTAN21DY900006	2.004.000	1
Module4	MODULE4	LTMTIN42BD	LTMTIN42BDDX900049	2.004.000	1
Module5	MODULE5	LTMTIN42BD	LTMTIN42BDDX900012	2.004.000	1
CT Sensor Cable	-	LTMT9RJ105	-	-	1
Expansion Module Cable	-	LTMT9RJ401	-	-	5

TeSys Tera system에 설정된 구성을 기반으로 모듈 목록이 표 형식으로 표시됩니다.

주의: Serial Number(일련 번호) 및 Version(버전)은 장치가 온라인 상태일 때 표시됩니다.

특성	설명
장치 유형	연결된 장치의 유형을 표시합니다.
레퍼런스	연결된 장치 또는 부속품의 레퍼런스를 표시합니다.
시리얼 번호	연결된 장치의 일련 번호를 표시합니다. 주의: 일련 번호는 장치가 온라인 상태일 때만 표시됩니다.
버전	연결된 장치의 버전 번호를 표시합니다. 주의: 버전 번호는 장치가 온라인 상태일 때만 표시됩니다.
수량	사용된 장치, CT 센서 케이블 및 확장 모듈 케이블의 수를 표시합니다.

구성된 BOM은  (Export Bill of Materials(BOM 내보내기)) 아이콘을 선택하여 CSV 파일로 내보낼 수 있습니다.

# Parameters List

## 이 장의 내용

General Settings .....	54
Starter Function Setting .....	54
Protection Setting .....	55
Communication Setting .....	59
User Map .....	59
Modified Parameters .....	60

# 개요

**Parameters List** 탭은 TeSys Tera system의 매개 변수 설정을 정의하는 데 사용됩니다.

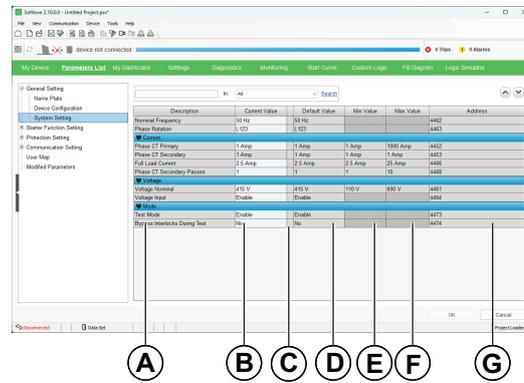
연결 모드와 연결 해제 모드 모두, 매개 변수 값은 **Current Value(현재 값)** 필드에서 수정할 수 있습니다.

연결 모드에서는 **OK(확인)**를 선택하여 설정을 장치에 업데이트할 수 있습니다.

연결 해제 모드에서는 **Store to Device(장치에 저장)** 작업으로 구성 설정을 장치에 다운로드할 수 있습니다.

**주의:** 매개변수 목록의 값들 매개변수 목록 탭의 값은 초기 로딩 중에 한 번만 업데이트됩니다. 이후에 업데이트된 값을 보려면 온라인 모드의 **Parameter List(매개변수 목록)** 탭에서 새로 고침 작업을 수행합니다.

# 설명



- A 설명: 매개 변수의 이름 또는 설명
- B 현재 값: 매개 변수 값
- C 렌치 표시: 현재 값이 해당 기본값과 다른 경우 나타납니다.
- D 기본값: 매개 변수의 기본값
- E 최소 값: 매개 변수의 최소값
- F 최대 값: 매개 변수의 최대값
- G 주소: 매개 변수의 레지스터 주소

다음 범주는 **Parameters Lists(매개 변수 목록)** 탭의 트리 보기에서 사용할 수 있습니다.

- General Setting
- Starter Function Setting
- Protection Setting
- Communication Setting
- User Map
- Modified Parameters

## 숫자 값 설정하기

매개 변수의 숫자 값은 다음 두 가지 방법으로 설정할 수 있습니다.

- 숫자 값을 직접 입력합니다.
- 스피ن 버튼을 사용하여 값을 선택합니다..

다음 단계에서는 직접 입력으로 숫자 값을 설정하는 방법을 설명합니다.

1. 트리 보기에서 필요한 섹션으로 이동합니다.
2. 목록에서 매개 변수를 선택합니다.
3. **Current Value(현재 값)** 필드에 매개 변수 값을 입력합니다.
4. **OK(확인)**를 선택하여 연결 모드에서 장치에 값을 쓰거나 연결 해제 모드에서 DTM 오프라인 구성에 값을 씁니다.

다음 단계에서는 스피ن 버튼을 사용하여 숫자 값을 설정하는 방법을 설명합니다.

1. 트리 보기에서 필요한 섹션으로 이동합니다.
2. 목록에서 매개 변수를 선택합니다.
3. **Current Value(현재 값)** 필드에서 스피ن 버튼을 사용하여 매개 변수 값을 설정합니다.
4. **OK(확인)**를 선택하여 연결 모드에서 장치에 값을 쓰거나 연결 해제 모드에서 DTM 오프라인 구성에 값을 씁니다.

## 문자열 편집하기

다음 단계에서는 문자열 매개 변수를 설정하는 방법을 설명합니다.

1. 트리 보기에서 필요한 섹션으로 이동합니다.
2. 목록에서 매개 변수를 선택합니다.
3. **Current Value(현재 값)** 필드에 문자열을 입력합니다.
4. **OK(확인)**를 선택하여 연결 모드에서 장치에 값을 쓰거나 연결 해제 모드에서 DTM 오프라인 구성에 값을 씁니다.

## 목록에서 값 선택하기

다음 단계에서는 목록에서 값을 선택하는 방법을 설명합니다.

1. 트리 보기에서 필요한 섹션으로 이동합니다.
2. 목록에서 매개 변수를 선택합니다.
3. **Current Value(현재 값)** 필드에서 드롭다운 목록을 사용하여 매개 변수 값을 설정합니다.
4. **OK(확인)**를 선택하여 연결 모드에서 장치에 값을 쓰거나 연결 해제 모드에서 DTM 오프라인 구성에 값을 씁니다.

탭에 대한 전체 설명은 탭 영역 설명, 43 페이지단원을 참조하십시오.

**주의:** 매개 변수 목록에서 기본값을 편집하고 **OK(확인)**를 선택하면 아래와 같은 경고 메시지가 나타납니다. 필요한 조치가 충족되었는지 확인하고 **OK(확인)**를 선택합니다.

## ⚠ 경고

### 장비의 의도하지 않은 동작

- 장치 구성을 변경하면 단락이 발생하거나 부하의 전원 공급 장치가 켜질 수 있습니다.
- 장치 구성에 따라 적절한 배선 및 구성이 이루어졌는지 확인하십시오.
- CTVT 센서 및 스타터를 교체할 때 3상 전원 공급이 차단되었는지 확인하십시오.

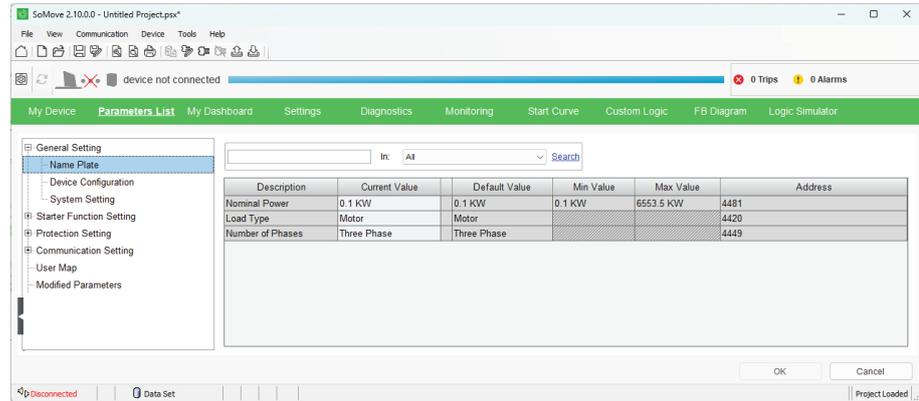
이러한 지침을 따르지 않을 경우 심각한 부상 또는 사망으로 이어지거나 장비가 손상될 수 있습니다.

## 중요 정보

- 구성에 사용할 수 있는 매개 변수는 TeSys Tera system 구성에 따라 결정됩니다.
- 입력한 매개 변수 값이 유효하지 않으면 오류 메시지가 표시됩니다. 메시지를 읽고 값을 적절하게 설정하십시오.
- 모터가 실행 상태인 경우 **Store to Device(장치에 저장)** 작업을 수행할 수 없습니다.
- 기본적으로 **Communication Loss(통신 손실)** 보호 기능은 비활성화됩니다. 이 기능이 활성화되고 장치가 SoMove 통신 포트를 통해 Modbus RTU software에 연결된 경우 DTM을 분리하려고 하면 장치에서 통신 손실 트립이 트리거됩니다.
- 모터가 실행 상태이면 **Parameters List(매개 변수 목록)** 아래 모든 설정의 구성은 **User Map(사용자 맵)**을 제외하고 비활성화됩니다. 매개 변수를 구성하려면 모터를 중지한 다음 설정을 구성합니다.
- 픽업 레벨 이전에 경보가 작동하려면 **경보 레벨 매개변수** 설정이 **트리거 레벨** 설정 값보다 작거나 같아야 합니다.
- 시스템에서 모터의 시작 및 정지를 단일 계전기로 구성할 때 DI 시작 **Validation Time(유효성 검사 시간)**은 DI 정지 **Validation Time(유효성 검사 시간)**보다 커야 합니다.
- 다음 항목 아래의 매개변수를 구성하려면 **필드버스 프로토콜 설정**, **Profibus 설정** 및 **HMI 통신** 섹션의 매개변수를 구성하려면 장치를 DTM에서 분리해야 합니다. 장치에 저장 작업을 수행하여 변경된 설정을 장치에 기록한 후 제공된 새 매개 변수 값을 사용하여 **Scan Devices(장치 스캔)**를 수행하면 장치에 연결할 수 있습니다.
- 만약 **메인 유닛 온도 값이 없음**, **온도 보호** 설정을 구성할 수 없습니다.
- LTMTCUF control operator unit이 HMI 포트에 연결된 경우 HMI 통신 설정은 다음과 같이 구성해야 합니다.
  - **Node Address(노드 주소):** 1
  - **Baud rate(전송 속도):** 19200bps
  - **Parity(패리티):** 짝수
  - **Byte Format(바이트 형식):** 빅 엔디안
- **Starter Type(스타터 유형)**이 가역으로 구성되어 있고 DI가 **Forced Start(강제 시작)**로 선택된 경우 모터 방향 변경 중에 **Interlocking Time(인터로킹 시간)**이 고려되지 않습니다.
- 개별 매개변수에 대한 자세한 내용은 다음을 참조하십시오. *TeSys Tera Motor Management System User Guide – DOCA0257EN*.
- 값은 **기본값** 열의 값은 구성 변경과 무관하게 항상 동일하게 유지됩니다.

## General Settings

**General Settings(일반 설정)** 창에는 TeSys Tera system의 일반 설정이 포함되어 있습니다.

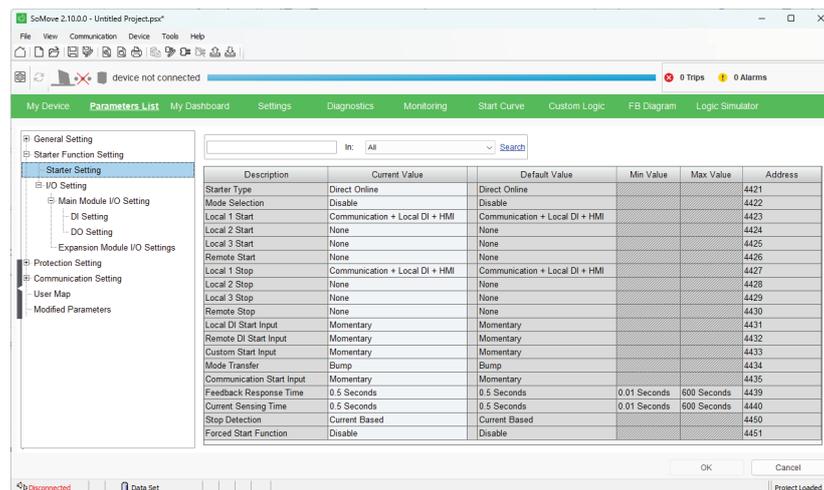


다음과 같은 하위 섹션을 사용할 수 있습니다.

- **Name Plate(이름 플레이트):** 이름 플레이트 매개 변수가 포함되어 있습니다.  
주의: 변경하는 부하 유형 변경 또는 상 수 을 변경하고 설정을 저장하면 장치가 재부팅됩니다.
- **Device Configuration(장치 구성):** LTMT main unit 온도 설정이 포함되어 있습니다.
- **System Setting(시스템 설정):** 전류, 전압, 위상 회전, 공칭 주파수 및 모드 매개 변수를 포함합니다.

## Starter Function Setting

**Starter Function Setting(스타터 기능 설정)** 창에는 모터 스타터 작동 방식을 제어하는 구성 가능한 매개 변수와 옵션이 포함되어 있습니다.



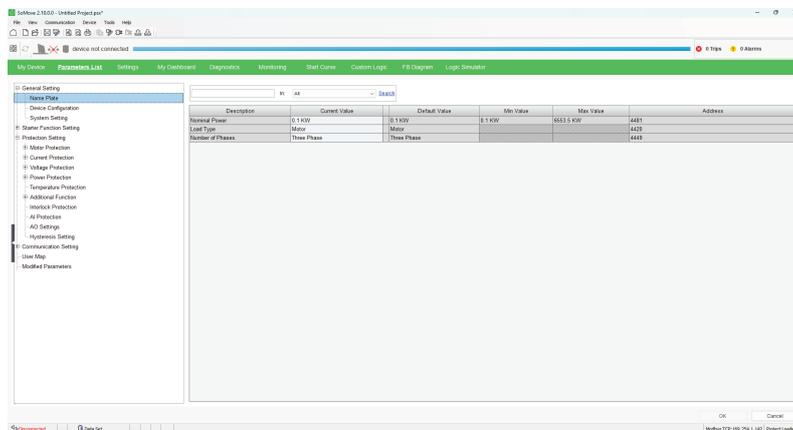
매개 변수는 다음과 같은 하위 섹션으로 나뉩니다.

- **Starter Setting(스타터 설정):** 모터 스타터와 관련된 조정 가능한 매개 변수가 포함되어 있습니다.  
주의: 다음 설정에서 매개변수 수정 스타터 설정 설정에서 매개변수를 수정하고 변경 사항을 저장하면 장치가 재부팅됩니다.
- **I/O Setting(I/O 설정):** LTMT main unit 및 LTMT expansion modules의 디지털 입력 및 출력 설정이 포함되어 있습니다..

# Protection Setting

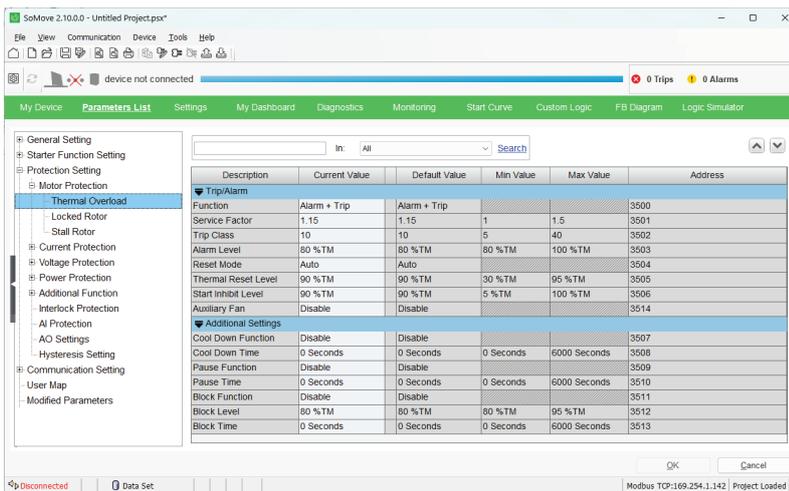
**Protection Setting(보호 설정)** 창에는 TeSys Tera system을 보호하기 위한 매개 변수가 포함되어 있습니다.

**주의:** 사용 가능한 매개변수 **보호 설정** 에서 선택한 값에 따라 달라집니다. **로드 유형 및 상수.**



다음과 같은 하위 섹션을 사용할 수 있습니다.

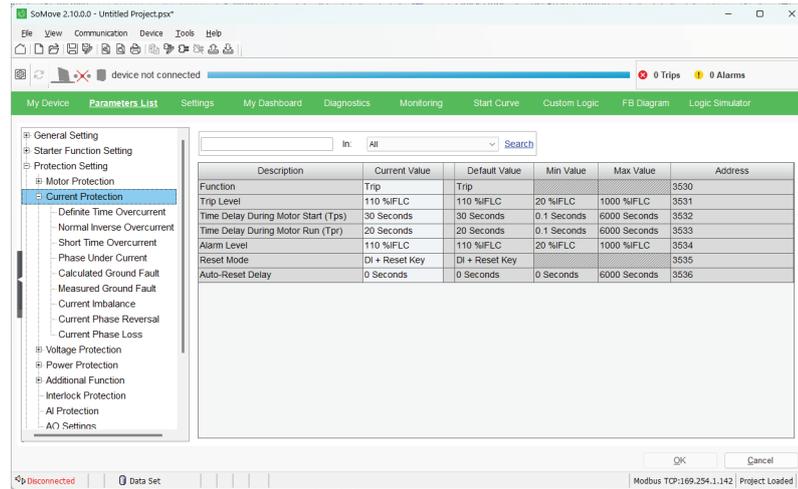
- **Motor Protection(모터 보호):** 구성 가능한 모터 보호 매개 변수가 포함되어 있습니다.



모터 보호 매개 변수는 다음과 같은 하위 섹션으로 분류되어 있습니다.

- **Thermal Overload(열 과부하)**
- **Locked Rotor(잠긴 로터)**
- **Stall Rotor(정지 로터)**

- **Current Protection(전류 보호):** 구성 가능한 전류 보호 매개 변수가 포함되어 있습니다.

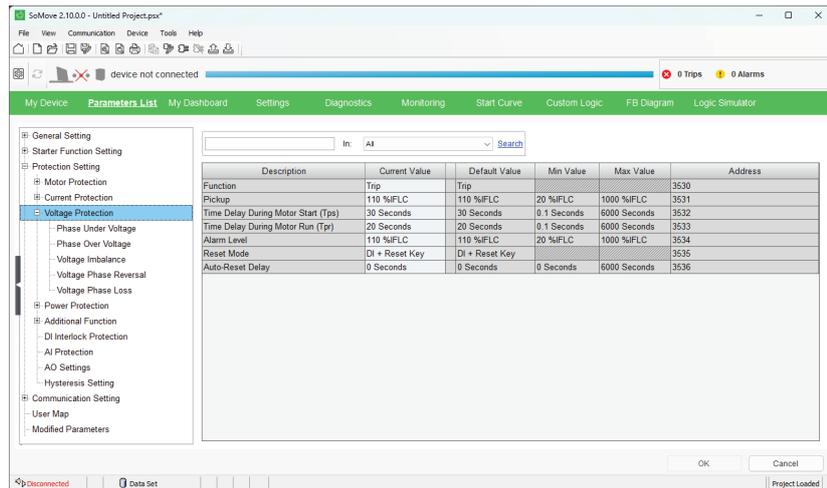


전류 보호 매개 변수는 다음과 같은 하위 섹션으로 분류되어 있습니다.

- **Definite Time Overcurrent(확정 시간 과전류)**
- **Normal Inverse Overcurrent(정상 역과전류)**
- **Short Time Overcurrent(단기 과전류)**
- **Phase Under Current(위상 부족 전류)**
- **Calculated Ground Fault(계산된 접지 오류)**
- **Measured Ground Fault(측정된 접지 오류)**
- **Current Imbalance(전류 불균형)**
- **Current Phase Reversal(전류 위상 반전)**
- **Current Phase Loss(전류 위상 손실)**

주의: 다음 항목 아래에서 이용 가능한 하위 항목 전류 보호 은 센서 모듈의 유형과 상 수 값에 따라 달라집니다.

- **Voltage Protection(전압 보호):** 구성 가능한 전압 보호 매개 변수가 포함되어 있습니다.



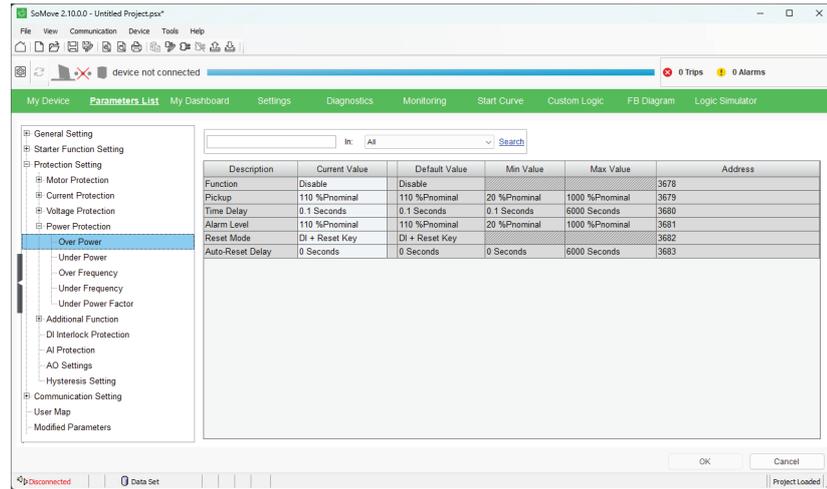
전압 보호 매개 변수는 다음과 같은 하위 섹션으로 분류되어 있습니다.

- **Phase Under Voltage(위상 부족 전압)**
- **Phase Over Voltage(위상 과전압)**
- **Voltage Imbalance(전압 불균형)**
- **Voltage Phase Reversal(전압 위상 반전)**
- **Voltage Phase Loss(전압 위상 손실)**

주의:

- 전압 보호 설정은 다음에 대해서만 가능합니다. LTMCTV sensor modules.
- 다음 항목 아래에서 이용 가능한 하위 항목 전압 보호의 값에 따라 달라집니다. 상수 값에 따라 달라집니다.

- **Power Protection(전력 보호):** 구성 가능한 전력 보호 매개 변수가 포함되어 있습니다.



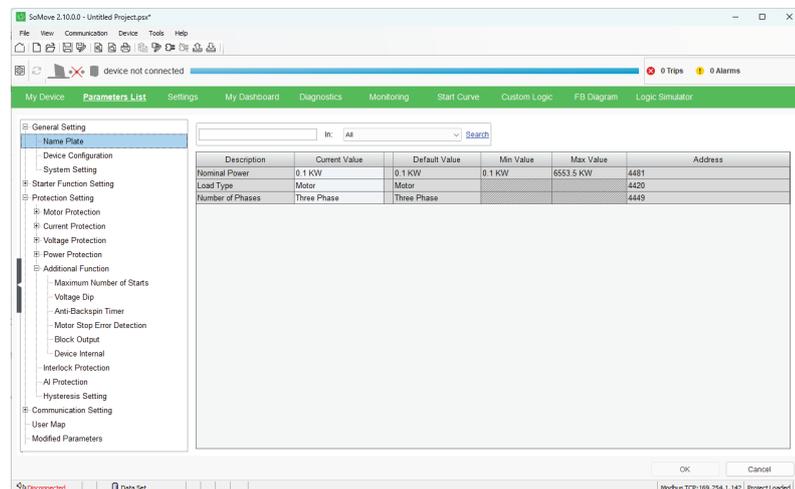
전력 보호 매개 변수는 다음과 같은 하위 섹션으로 분류되어 있습니다.

- **Over Power(과전력)**
- **Under Power(저전력)**
- **Over Frequency(과주파수)**
- **Under Frequency(부족 주파수)**
- **Under Power Factor(저역률)**

주의:

- 전원 보호 설정은 다음에 대해서만 가능합니다. LTMTCTV sensor modules.

- **Additional Function(추가 기능):** 구성 가능한 추가 기능 매개 변수가 포함되어 있습니다.



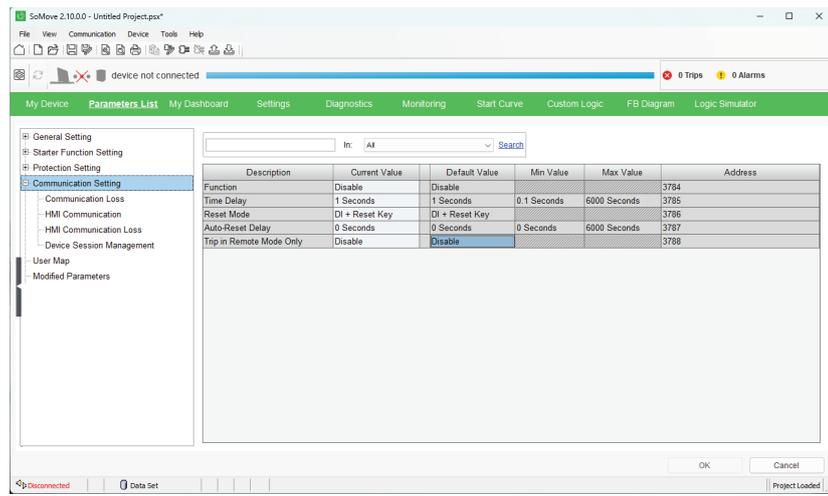
이러한 매개 변수는 다음과 같은 범주로 분류되어 있습니다.

- **Maximum Number of Starts(최대 시작 횟수)**
- **Voltage Dip(전압 강하)**
- **Anti-Backspin Timer(백스핀 방지 타이머)**
- **모터 정지 오류 감지**
- **Block Output(블록 출력)**
- **Device Internal(장치 내부)**
- **DI Interlock Protection(DI 인터록 보호):** 구성 가능한 디지털 입력 인터록 보호 매개 변수가 포함되어 있습니다.

- **AI Protection(AI 보호):** 구성 가능한 아날로그 입력 보호 매개 변수가 포함되어 있습니다. The **AI 보호** 매개 변수는 LTMTAN21 expansion module 에서 **내 장치** 탭에 각 장치당 두 개의 아날로그 입력을 구성할 수 있습니다.
- **AO 설정:** 구성 가능한 아날로그 출력 보호 매개 변수가 포함되어 있습니다. The **AO 설정** 매개 변수는 LTMTAN21 expansion module 에 **내 장치** 탭에 각 장치당 하나의 아날로그 출력을 구성할 수 있습니다.
- **Hysteresis Setting(히스테리시스 설정):** 구성 가능한 히스테리시스 매개 변수가 포함되어 있습니다.

## Communication Setting

**Communication Setting(통신 설정)** 창에는 통신 매개 변수가 포함되어 있습니다.

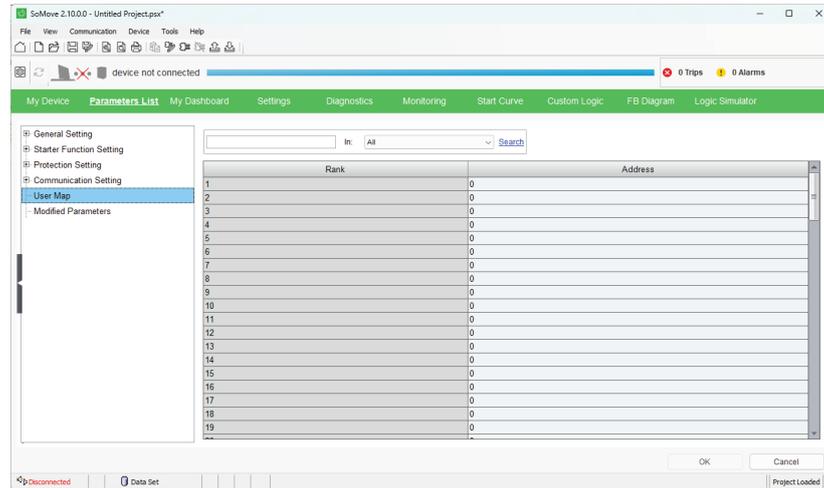


매개 변수는 다음과 같은 하위 섹션으로 나뉩니다.

- **필드버스 프로토콜 설정 또는 프로피버스 설정:** 이 창에는 TeSys Tera system의 네트워크 설정에 대한 매개 변수가 포함되어 있습니다.  
 주의: 필드버스 프로토콜 설정 또는 프로피버스 설정 다음과 같은 경우 사용할 수 있습니다. 필드버스 프로토콜 으로 선택된 경우 Modbus 또는 PROFIBUS 로 각각 선택된 경우에 사용할 수 있습니다.
- **Communication Loss(통신 손실):** 이 창에는 통신 손실 설정을 구성하는 매개 변수가 포함되어 있습니다.
- **HMI Communication(HMI 통신):** 이 창에는 LTMTCUF control operator unit의 네트워크 설정에 대한 매개 변수가 포함되어 있습니다.
- **HMI Communication Loss(HMI 통신 손실):** 이 창에는 LTMTCUF control operator unit의 통신 손실 설정을 구성하는 매개 변수가 포함되어 있습니다.
- **Device Session Management(장치 세션 관리):** 이 창에는 장치 세션 시간 초과를 구성하는 매개 변수가 포함되어 있습니다.

## User Map

**User Map(사용자 맵)** 변수는 읽기 및 쓰기 레지스터를 최적화하도록 설계되었습니다.

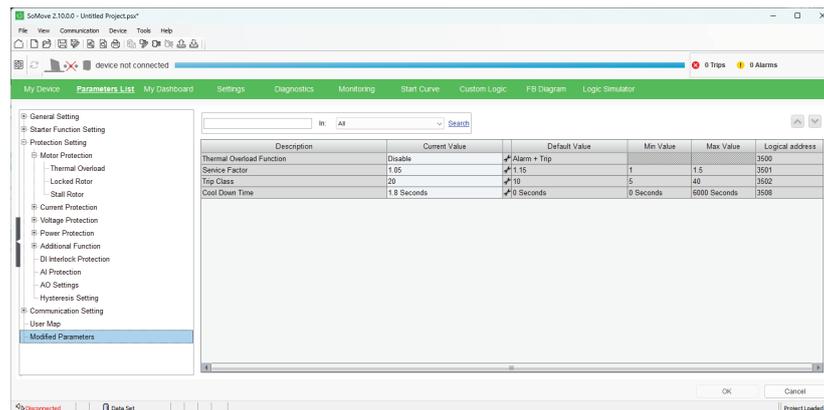


다음 절차에서는 사용자 맵 주소를 설정하는 방법에 대해 설명합니다.

- 트리 보기에서 **User Map(사용자 맵)**을 선택합니다.
  - 주소는 1에서 100까지 순위가 매겨집니다.
  - 주소는 네 개의 그룹으로 나뉩니다.
- 테이블에 주소 값을 입력합니다.
  - 입력한 주소는 10진수 형식이어야 하며 0에서 9249 사이여야 합니다.
  - 사용자 맵에서 주소를 제거하려면 주소를 0으로 입력합니다.
- Enter 키를 눌러 새 주소를 확인합니다.
  - 주소가 수락되면 사용자 맵에 주소가 추가됩니다.
  - 주소가 수락되지 않으면 이전에 수락된 주소가 사용자 맵에 유지됩니다.

## Modified Parameters

**Modified Parameters(수정된 매개 변수)** 창에는 값이 기본 설정에서 변경된 매개 변수 목록이 표시됩니다.



# 설정

## 이 장의 내용

일반.....	62
예신.....	63
보안.....	65

## 개요

설정 탭에서는 의 설정을 구성할 수 Ethernet있습니다LTMT main unit.

설정값은 연결 모와 연결 해제 모드 모두에서 수정할 수 있습니다.

연결 모드에서는 **OK(확인)**를 선택하여 설정을 장치에 업데이트할 수 있습니다.

연결 해제 모드에서는 **Store to Device(장치에 저장)** 작업으로 구성 설정을 장치에 다운로드할 수 있습니다.

주의:

- 설정 **설정** 탭은 통신이 **Modbus TCP** 으로 선택되었거나 Ethernet 변형이 LTMT main unit 버전이 선택된 경우에만 사용할 수 있습니다.
- 설정 탭의 매개변수는 의 변형이 직렬 인터페이스를 Ethernet 통해 연결된 LTMT main unit 경우 읽기 전용으로 표시됩니다.
- 설정 **탭의** 값은 한 번만 업데이트됩니다. 온라인 모드에서 업데이트된 값을 보려면 아이콘을 사용하여 새로 고침을  수행합니다.

## 설명

설정 탭의 트리 보기에는 다음 **카테고리**가 제공됩니다:

- 일반
- 통신
- 보안

## 일반

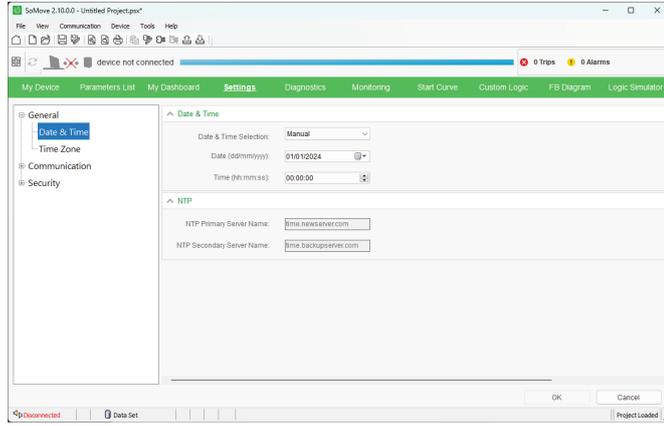
일반 창에는 의 다음 일반 설정이 포함되어 있습니다LTMT main unit:

- 날짜 및 시간
- 시간대

## 날짜 및 시간

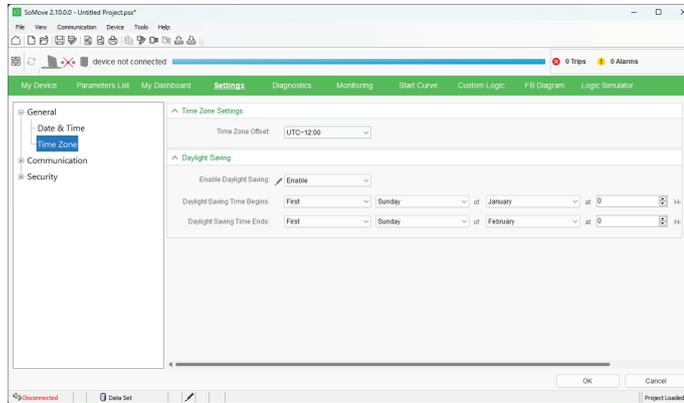
The **날짜 및 시간** 창을 통해 날짜 및 시간 설정을 구성할 수 있습니다. LTMT main unit. 다음 옵션은 **날짜 및 시간 선택** 드롭다운 목록에서 사용할 수 있습니다:

- **매뉴얼** – 날짜와 시간을 수동으로 설정하려면 **날짜 (dd/mm/yyyy)** 및 **시간 (hh:mm:ss)** 필드를 각각 사용하여 날짜와 시간을 수동으로 설정합니다.
- **NTP** – 날짜와 시간은 **NTP 기본 서버 이름** 및 **NTP 보조 서버 이름** 필드에 설정된 네트워크 시간 프로토콜(NTP) 서버에서 가져옵니다.
- **필드 버스 프로토콜** – 날짜와 시간은 구성된 필드 버스 프로토콜에서 획득됩니다.



## 시간대

The **시간대** 창을 통해 시간대와 일광 절약 시간 설정을 설정할 수 있습니다. LTMT main unit.



시간대를 설정하려면 **시간대 오프셋** 드롭다운 목록에서 원하는 시간대를 선택하십시오.

일광 절약 시간제를 적용하려면 **활성화** 을 선택하십시오. **일광 절약 시간 사용** 드롭다운 목록에서 필요한 설정을 다음에 입력하십시오 **일광 절약 시간제 시작** 및 **일광 절약 시간 종료** 필드에 필요한 설정을 입력하십시오.

## 통신

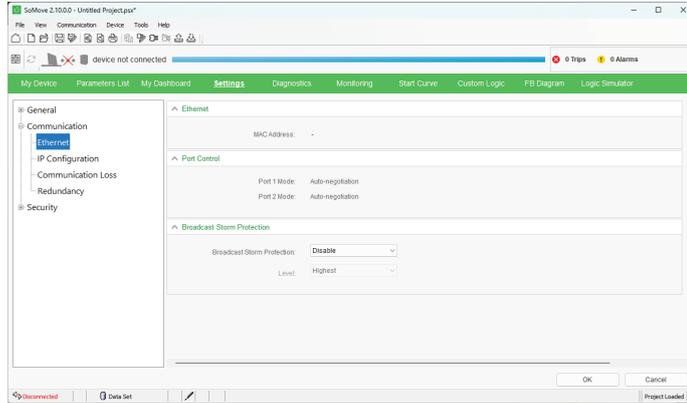
의 **통신** 창은 통신 설정을 표시하고 구성할 수 있게 합니다. LTMT main unit. 다음과 같은 하위 섹션이 제공됩니다:

- 이더넷
- IP 구성
- **Communication Loss(통신 손실)**

## 이더넷

이더넷 **이더넷** 창에는 다음과 같은 정보가 표시됩니다:

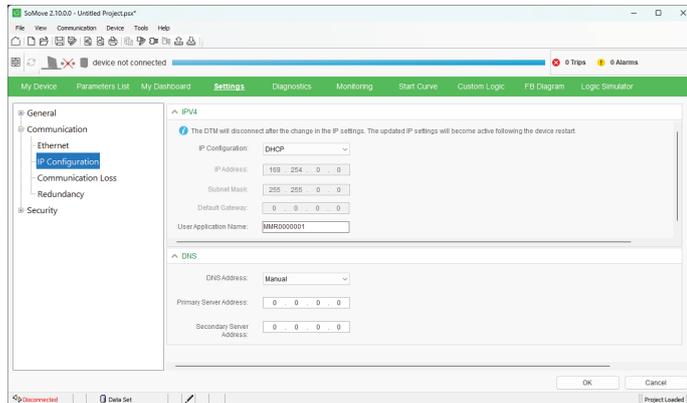
- **MAC** 주소
- **포트 1** 모드
- **포트 2** 모드



폭풍 보호 방송은 다음을 사용하여 활성화할 수 있습니다. 방송 폭풍 보호 드롭다운 목록을 사용하여 활성화할 수 있습니다. 폭풍 보호 방송의 수준을 선택하려면 레벨 드롭다운 목록을 사용하여 선택하십시오.

## IP 구성

IP 구성 IP 구성 창을 통해 IPv4 및 DNS 설정을 구성할 수 있습니다. LTMT main unit.



IP 구성 IP 구성 다음과 같이 설정할 수 있습니다:

- **DHCP – IP 주소, 서브넷 마스크, 그리고 기본 게이트웨이**는 동적 호스트 구성 프로토콜(DHCP) 서버로부터 자동으로 할당됩니다.
- **고정 (정적 IP) – 고정 IP 주소를 설정할 수 있습니다 IP 주소, 서브넷 마스크, 그리고 기본 게이트웨이** 설정을 필요에 따라 설정할 수 있습니다.

DNS 주소 DNS 주소 설정은 다음과 같이 설정할 수 있습니다:

- **자동 – 기본 서버 주소 및 보조 서버 주소**는 DHCP 서버로부터 자동으로 할당됩니다.
- **매뉴얼 – 다음을 설정할 수 있습니다. 기본 서버 주소 및 보조 서버 주소.**

**주의:** When IP 구성 이 고정(정적 IP)으로 설정된 경우, DNS 주소는 수동 모드에서만 설정할 수 있습니다.

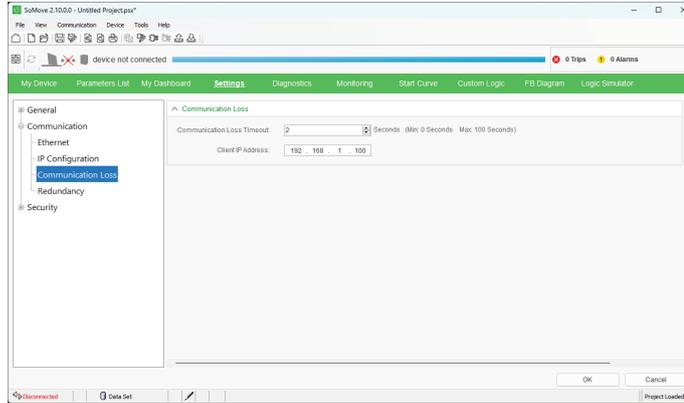
사용자 애플리케이션 이름 사용자 애플리케이션 이름 필드를 통해 애플리케이션에 대한 사용자 지정 이름을 지정할 수 있습니다.

**주의:** 영숫자만 지원됩니다.

## 통신 손실

의 통신 손실 창에서는 다음 설정을 구성할 수 있습니다:

- **커뮤니케이션 장애 일시정지**
- **클라이언트 IP 주소**

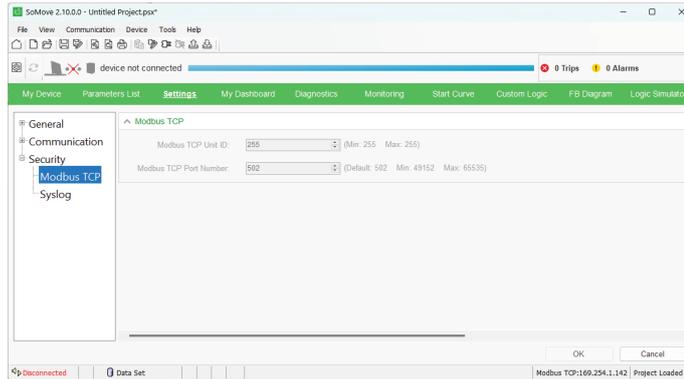


## 보안

The 보안 창을 통해 Modbus TCP 보안 설정 및 시스템 로그를 다운로드할 수 있습니다. LTMT main unit.

## 모드버스 TCP

The Modbus TCP 창은 Modbus TCP 장치 ID 를 표시하며 Modbus TCP 포트 번호.



## Syslog

시스템 로그 정보를 CSV 파일로 내보내려면 수출 옵션을 선택하여 CSV 파일로 내보낼 수 있습니다.

# My Dashboard

## 개요

**My Dashboard** 탭을 사용하면 요구 사항에 따라 구성 또는 모니터링할 데이터를 사용자 정의할 수 있습니다. 이 탭은 다음과 같은 용도로 사용됩니다.

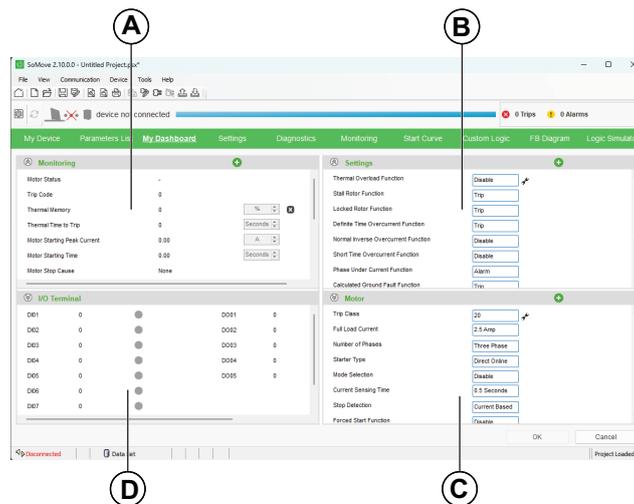
- 스타터 관련 매개 변수의 실시간 값을 표시합니다.
- 입력/출력 단자의 정보를 표시합니다.
- 스타터 및 모터 관련 매개 변수를 구성합니다.
- 보호 기능 매개 변수를 구성합니다.

**주의: My Dashboard(내 대시보드) 설정은 Save Project(프로젝트 저장) 옵션을 사용하여 저장할 수 있습니다. 프로젝트를 저장하고 저장된 프로젝트를 열면 설정이 유지됩니다.**

## 설명

작업 공간은 다음과 같은 두 영역으로 구성되어 있습니다.

- **표시 영역:** 분석을 위해 스타터 매개 변수의 실시간 값과 입력/출력 단자의 상태를 표시합니다.
- **설정 영역:** 스타터 및 모터 매개 변수를 설정합니다.



- A           모니터링
- B           설정
- C           모터
- D           I/O 단자

## 모니터링

**Monitoring(모니터링) 영역을 사용하면 필요한 매개 변수를 모니터링할 수 있습니다.**

## 설정

**Settings(설정)** 영역을 사용하면 요구 사항에 따라 보호 기능 매개 변수 값을 모니터링하고 변경할 수 있습니다.

설정 영역 아래에 있는 매개 변수에는 드롭다운 목록을 사용하여 필요한 값을 할당할 수 있습니다.

## 모터

**Motor(모터)** 영역을 사용하면 요구 사항에 따라 모터 매개 변수 값을 모니터링하고 변경할 수 있습니다.

## I/O 단자

**I/O Terminals(I/O 단자)** 영역을 사용하면 I/O 신호의 상태를 모니터링할 수 있습니다.

다음 표는 입력/출력의 상태를 설명합니다. LTMT main unit 및 LTMT expansion modules 의 입력/출력 상태를 설명합니다.

상태 입력/출력	색상 상태 상자	설명 텍스트
활성	녹색	IO 단자가 할당되고 활성 상태임을 나타냅니다.
비활성	회색	IO 단자가 할당되고 비활성 상태임을 나타냅니다.

## 매개 변수 추가하기

다음 절차에서는 매개 변수를 추가하는 방법을 보여 줍니다.

1. 매개 변수를 추가해야 하는 영역에서 **+** 아이콘을 선택합니다.

**결과:** **Select Parameter(매개 변수 선택)** 대화 상자가 나타납니다.

2. **All Parameters(모든 매개 변수)** 열에서 필요한 매개 변수를 선택하고 **오른쪽 화살표**를 선택합니다. 검색 표시줄을 사용하여 **All Parameters(모든 매개 변수)** 열에서 사용할 수 있는 매개 변수를 찾을 수 있습니다.

**결과:** 매개 변수가 **Selected Parameters(선택한 매개 변수)** 열로 이동됩니다.

3. 매개 변수 목록의 순서를 변경하려면 **Selected Parameters(선택한 매개 변수)** 열에서 필요한 매개 변수를 선택한 다음 **위쪽/아래쪽 화살표**를 선택합니다.
4. **OK(확인)**를 선택합니다.

추가된 매개 변수는 관련 표시 영역에 표시됩니다.

## 매개 변수 삭제하기

영역에서 매개 변수를 제거하려면 매개 변수 앞에 있는 **×** 아이콘을 선택합니다.

다음 절차에서는 여러 매개 변수를 제거하는 방법을 보여 줍니다.

1. 매개 변수를 제거해야 하는 영역에서 **+** 아이콘을 선택합니다.

**결과:** **Select Parameter(매개 변수 선택)** 대화 상자가 나타납니다.

2. **Selected Parameters(선택한 매개 변수)** 열에서 필요한 매개 변수를 선택하고 **왼쪽 화살표**를 선택합니다.

**결과:** 매개 변수가 **All Parameters(모든 매개 변수)** 열로 이동됩니다.

3. **OK(확인)**를 선택합니다.

매개 변수가 관련 표시 영역에서 제거됩니다.

**주의:** 매개 변수 목록은 **Settings(설정)** 및 **Motor(모터)** 영역에서만 구성할 수 있습니다.

# Diagnostics

## 개요

진단 진단 탭은 모터 상태, 경보 또는 트립 상태, 로그, 장치 내부 상태, 통신 상태 및 트립 카운터 정보를 표시합니다. TeSys Tera system.

이 탭의 데이터는 연결 모드에서만 의미가 있습니다.

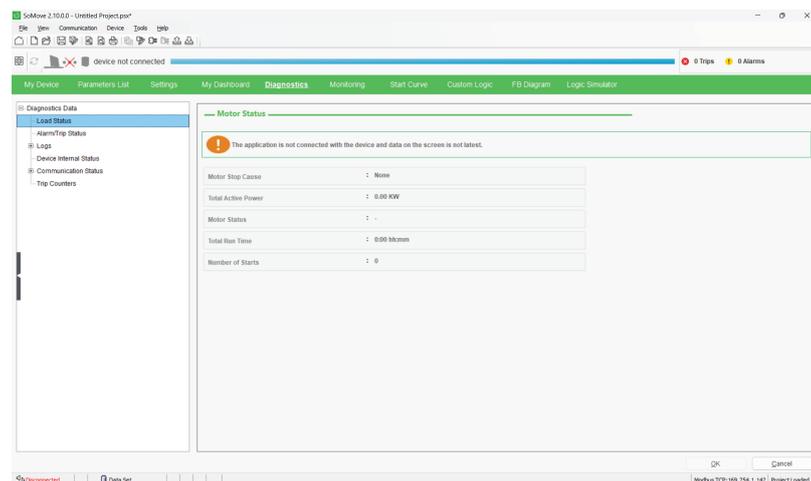
진단 데이터 진단 탭의 데이터는 동적이며, **Logs** 노드를 제외하고 1초마다 새로 고쳐집니다.

## 설명

**Diagnostics(진단)** 탭에는 다음 매개 변수가 표시됩니다.

- **Motor Status(모터 상태)** 또는 **Load Status(부하 상태)**
- **Alarm or Trip Status(경보 또는 트립 상태)**
- TeSys Tera system의 로그에는 다음이 포함됩니다.
  - **Events(이벤트)**
  - **Device Internal(장치 내부)**
  - **Trips(트립)**
- **Device Internal Status(장치 내부 상태)**
- **Communication Status(통신 상태)**
- **Trip Counters(트립 카운터)**

## 모터 또는 부하 상태



**Motor Status(모터 상태)** 또는 **Load Status(부하 상태)** 창에는 구성된 모터 또는 히터의 매개 변수가 각각 표시됩니다.

다음 모터 매개 변수가 **Motor Status(모터 상태)** 또는 **Load Status(부하 상태)** 창 아래에 표시됩니다.

- **Motor Stop Cause(모터 정지 원인):** 모터 작동이 중지된 이유입니다.
- **Total Active Power(총 유효 전력):** 모터 또는 히터 작동 중에 소비된 총 전력입니다.

- **Motor Status(모터 상태)** 또는 **Load Status(부하 상태)**: 모터 또는 히터의 현재 상태입니다.
- **Total Run Hour(총 실행 시간)**: 모터 또는 히터의 총 작동 시간입니다.
- **Number of Starts(시작 횟수)**: 모터 또는 히터의 총 시작 횟수입니다.

## 경보 또는 트립 상태

**Alarm or Trip Status(경보 또는 트립 상태)** 창에는 경보 또는 트립과 관련된 모터 매개 변수의 상태가 표시됩니다.

검색 상자에 키워드를 추가하고 **In(위치)** 드롭다운 목록에서 필요한 옵션을 선택한 다음 **Search(검색)**를 선택하여 필요한 매개 변수를 검색할 수 있습니다.

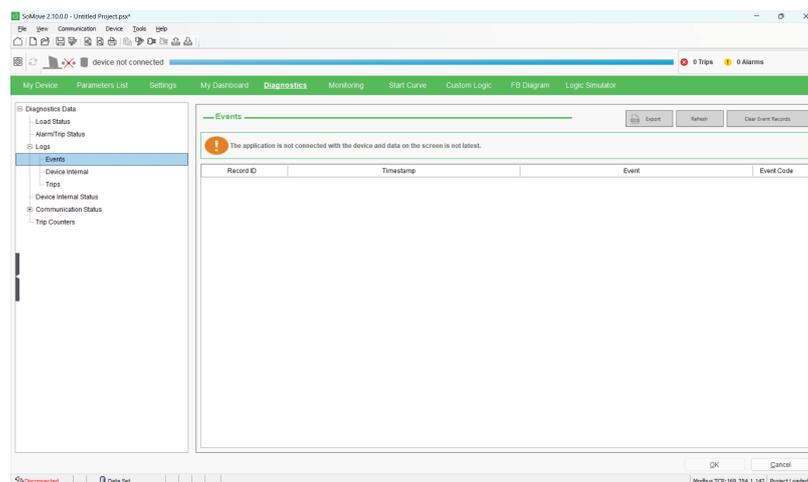
표시기 색상		상태
회색 (비활성 상태)		여행이나 알람 없음
주황색		경보
빨간색		트립

## 로그

**Logs(로그)** 창에는 다음 범주에 대한 로그가 표시됩니다.

- **Events(이벤트)**
- **Device Internal(장치 내부)**
- **Trips(트립)**

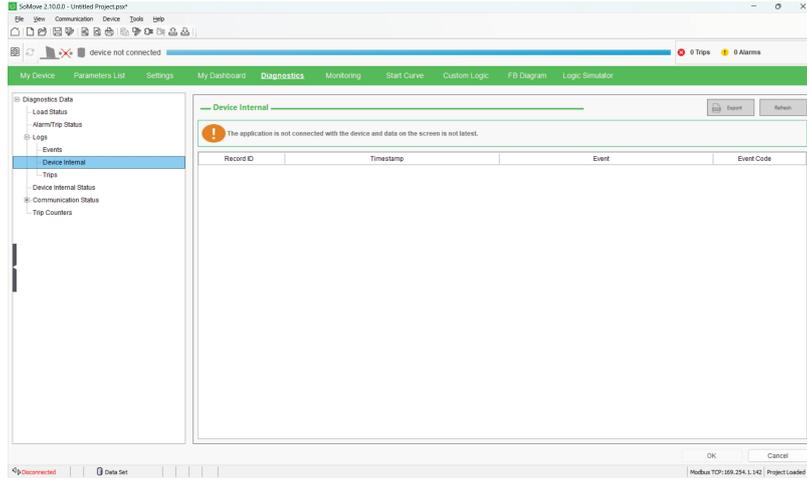
## 이벤트



**Events(이벤트)** 창에는 TeSys Tera system에 대한 이벤트 레코드의 목록이 표시됩니다.

최대 100개의 이벤트 로그가 기록 ID, 타임스탬프, 이벤트 설명 및 이벤트 코드와 함께 창에 표시됩니다. 기록 ID 1은 가장 최근에 기록된 이벤트이며, 기록 ID 100은 가장 오래된 기록된 이벤트입니다. 창에 100개의 이벤트 로그가 있고 새 이벤트가 발생하는 경우 가장 오래된 이벤트가 목록에서 제거되고 새 이벤트가 목록 상단에 추가됩니다.

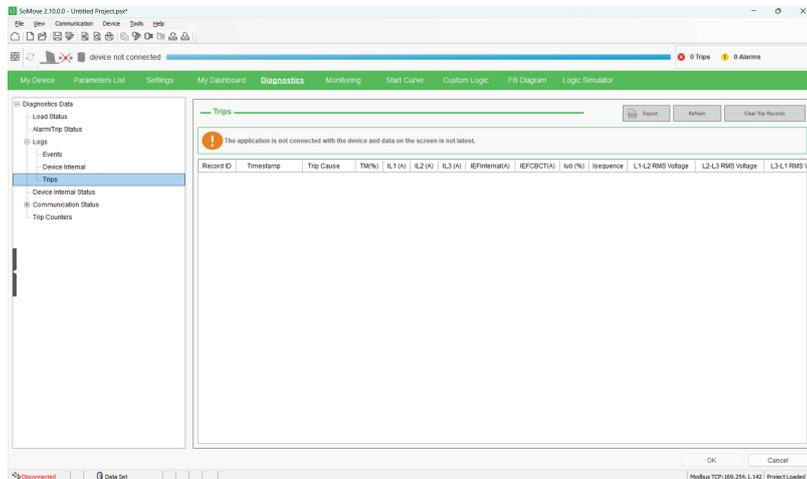
# 장치 내부



**Device Internal(장치 내부)** 창에는 TeSys Tera system의 내부 계전기 이벤트 목록이 목록이 표시됩니다.

최대 20개의 내부 릴레이 로그가 기록 ID, 타임스탬프, 이벤트 설명 및 이벤트 코드와 함께 창에 표시됩니다. 기록 ID 1은 가장 최근에 감지된 오류 기록이며, 기록 ID 20은 가장 오래된 감지 오류 기록입니다. 윈도우에 오류가 20개 있는 경우 새 오류가 감지되면, 가장 오래된 오류가 목록에서 제거되고 새로 감지된 오류가 목록 상단에 추가됩니다.

# 트립



**Trips(트립)** 창에는 TeSys Tera system의 트립 레코드 목록이 표시됩니다.

최대 20개의 트립 로그가 창에 표시됩니다. 각 로그에는 기록 ID, 타임스탬프, 트립 원인, 트립 발생 시 중요 설정값, 트립 감지 시 기록된 측정값, 트립 코드가 포함됩니다. 기록 ID 1은 가장 최근에 기록된 여행이며, 기록 ID 20은 가장 오래된 여행 기록입니다. 창에 20개의 트립 레코드가 있고 새로운 트립이 감지되는 경우 가장 오래된 트립 레코드가 목록에서 제거되고 새 트립 레코드가 목록 상단에 추가됩니다.

# 내보내기

The 내보내기 함수는 레코드 목록을 CSV 형식으로 내보냅니다.

레코드 목록을 내보내려면 **Export(내보내기)**를 선택하고 대상 폴더를 선택한 다음 **Save(저장)**를 선택합니다.

**주의:** 내보내기 옵션은 하나 이상의 레코드 데이터가 있는 경우에만 사용할 수 있습니다.

## 새로 고침

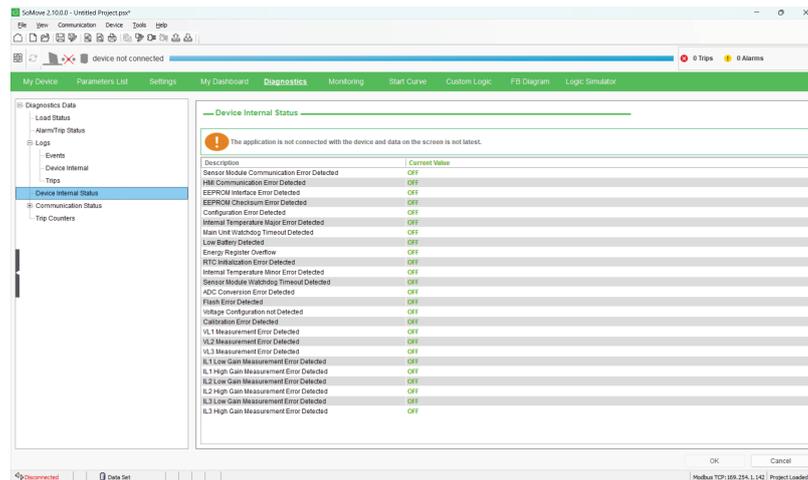
**Refresh(새로 고침)** 기능은 레코드 목록을 업데이트하고 최신 목록을 표시합니다.

Data Record(데이터 레코드) 노드는 연결 모드에서 **Refresh(새로 고침)**를 선택하여 업데이트할 수 있습니다.

## 이벤트 로그 지우기 또는 트립 로그 지우기

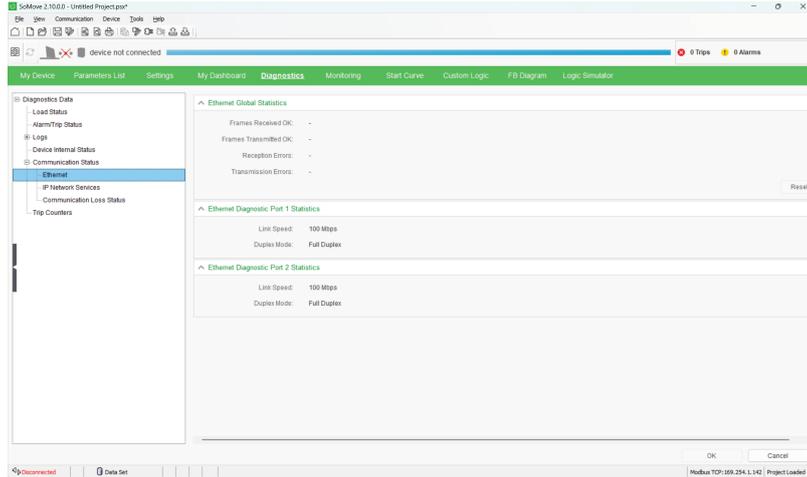
**Clear Event Logs(이벤트 로그 지우기)** 또는 **Clear Trip Logs(트립 로그 지우기)**는 이벤트 또는 트립 목록을 지웁니다. 각각 해당 창에서 **Clear Event Logs(이벤트 로그 지우기)** 또는 **Clear Trip Logs(트립 로그 지우기)**를 클릭합니다.

## 장치 내부 상태



**Device Internal Status(장치 내부 상태)** 창에는 TeSys Tera system에 있는 다양한 모듈의 통신 및 작동 상태가 표시됩니다.

# 통신 상태

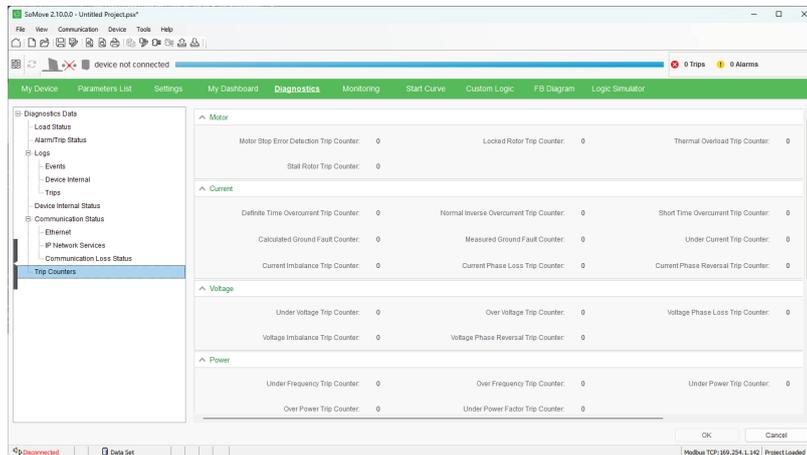


**Communication Status(통신 상태)** 창에는 TeSys Tera system의 통신 매개 변수 상태가 표시됩니다.

변종이 Ethernet 의 변형이 LTMT main unit 에 사용될 때 TeSys Tera system다음과 같은 하위 섹션을 사용할 수 있습니다:

- 이더넷 – 다음을 표시합니다. 이더넷 글로벌 통계, 이더넷 진단 포트 1 통계, 그리고 이더넷 진단 포트 2 통계 를 표시합니다. LTMT main unit.
- IP 네트워크 서비스 – 표시합니다 Modbus TCP 포트 상태를 표시합니다.
- 통신 손실 상태 – 통신 손실 상태를 표시합니다. LTMT main unit.

# 트립 카운터



**Trip Counters(트립 카운터)** 창에는 TeSys Tera system내의 범주당 트립 수가 표시됩니다.

# Monitoring

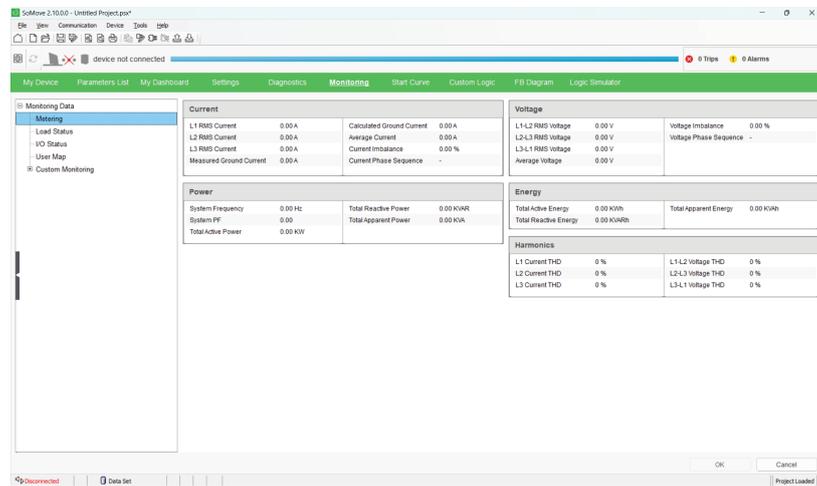
## 개요

**Monitoring(모니터링)** 탭에는 TeSys Tera system과 관련된 모니터링 정보가 표시됩니다.

다음 범주에 대한 모니터링 정보가 표시됩니다.

- 계량
- 로드 상태
- 입출력 상태
- 사용자 지도
- 사용자 지정 모니터링

## Metering



**Metering(계측)** 창에는 TeSys Tera system의 계측 데이터가 표시됩니다.

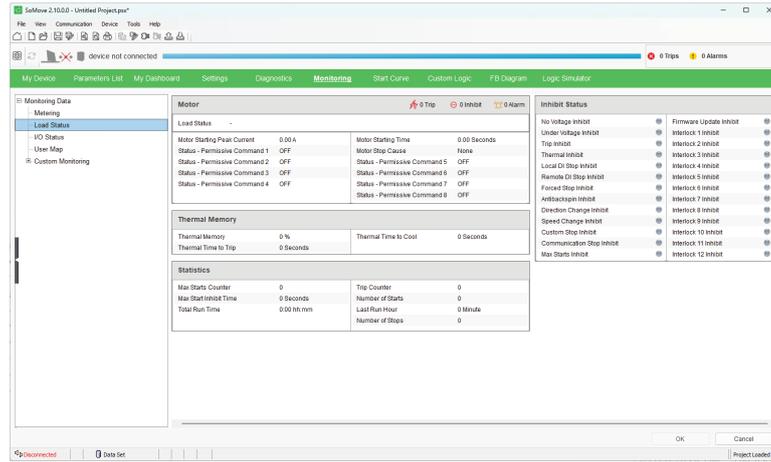
계측 신호는 다음과 같은 범주로 나뉩니다.

- 현재
- 전압
- 힘
- 에너지
- 온도
- 고조파

**주의:** 미터링에 설정된 매개변수 **Metering** 창에 채워지는 매개변수는 다음에서 수행된 구성에 따라 달라집니다:

- 내 기기 탭
- 일반 설정 > 기기 구성
- 일반 설정 > 명판 > 상 수

# Load Status



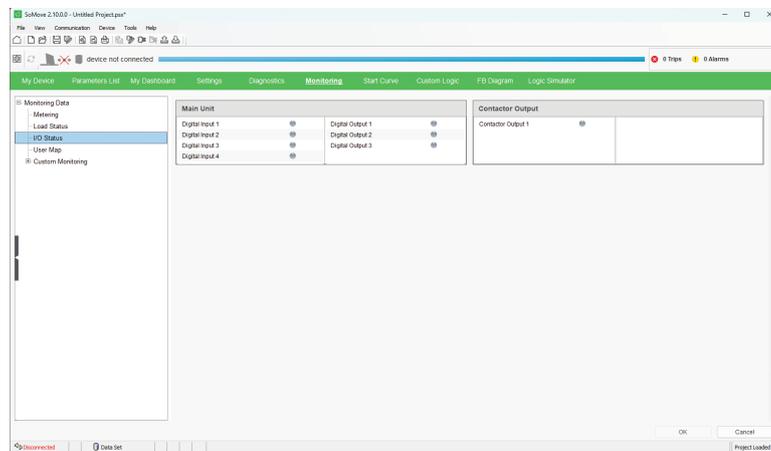
**Load Status(부하 상태)** 창에는 TeSys Tera system의 부하 상태가 표시됩니다.

Load Status(부하 상태) 페이지의 매개 변수는 다음과 같은 범주로 나뉩니다.

- 모터
- 열 기억
- 통계
- 억제 상태

로드 상태 로드 상태 창에는 또한 모터.

# I/O Status

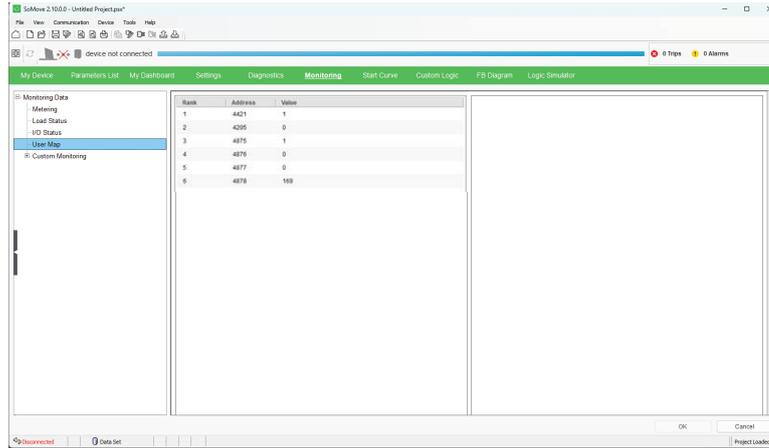


I/O 상태 I/O 상태 창은 아날로그 및 디지털 입력과 출력의 상태를 표시합니다. TeSys Tera system.

I/O 신호는 다음과 같은 범주로 나뉩니다.

- 본체
- 접촉기 출력
- Expansion Module

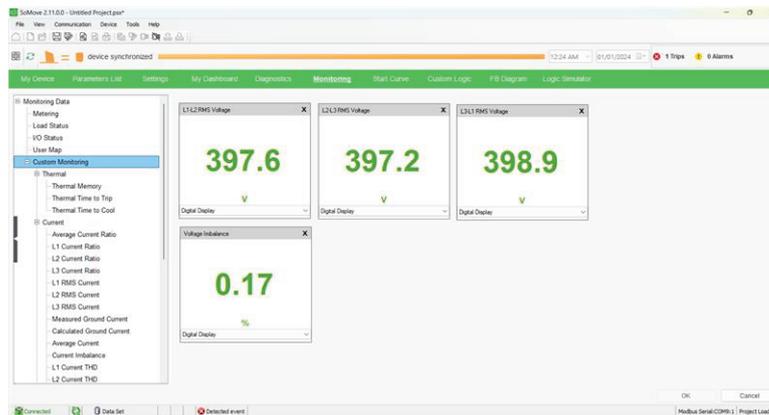
## User Map



**User Map(사용자 맵)** 창에는 **Parameters(매개 변수) > User Map(사용자 맵)** 섹션에 구성된 매개 변수의 라이브 값이 표시됩니다.

최대 100개의 매개 변수가 표시됩니다.

## Custom Monitoring



**Custom Monitoring(사용자 정의 모니터링)** 창에서는 위젯을 사용하여 개인화된 모니터링 창을 생성할 수 있습니다.

신호는 다음과 같은 범주로 나뉩니다.

- 열
- 현재
- 전압
- 힘
- 모터

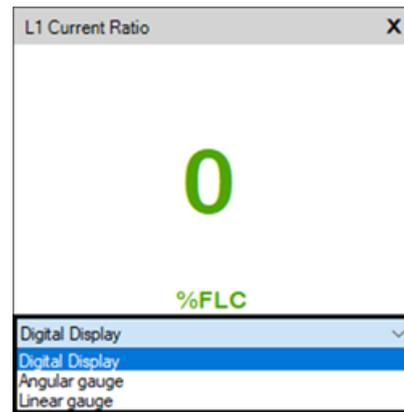
## 사용자 정의 모니터링 창 사용자 정의하기

다음 절차에서는 위젯을 추가하는 방법에 대해 설명합니다.

1. 사용 가능한 신호 범주에서 제공되는 모니터링 대상 신호를 선택합니다.
2. **Custom Monitoring(사용자 정의 모니터링)** 창의 표시 영역을 클릭합니다.

**결과:** 선택한 신호에 대한 위젯이 추가됩니다.

드롭다운 목록에서 옵션을 선택하여 위젯 표시 옵션을 수정할 수 있습니다.



표시 영역에서 위젯을 제거하려면 **X** 아이콘을 선택합니다.

표시 영역에서 다음 작업을 수행할 수 있습니다.

- 모든 위젯을 제거하려면 표시 영역을 마우스 오른쪽 버튼으로 클릭하고 **Clear All** (모두 지우기)을 선택합니다.



- 추가된 위젯을 구성하려면 표시 영역을 마우스 오른쪽 버튼으로 클릭하고 **Arrange All**(모두 정렬)을 선택합니다.

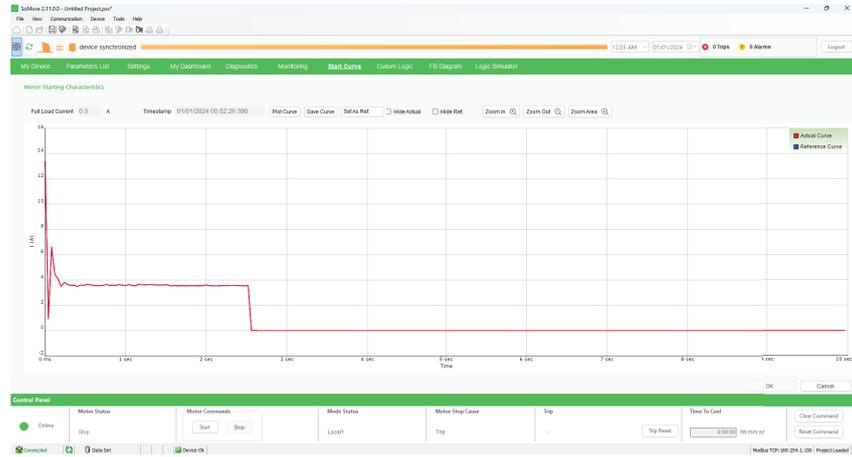


# Start Curve

## 개요

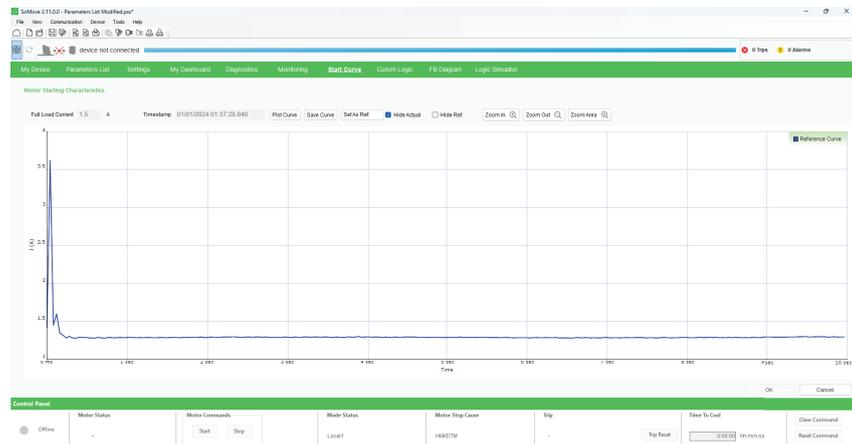
**Start Curve(시작 곡선)** 탭에는 모터 시작 시간 동안 샘플 시간 간격으로 모터 전류(I)의 그래프가 표시됩니다. **시작 곡선(Start Curve)** 탭은 곡선에서 모터 시작 특성을 분석하는 데 사용됩니다.

공장 초기화를 수행하면 시작 곡선 데이터가 장치에서 지워집니다.



## 레퍼런스 곡선

**Set as Ref.(레퍼런스로 설정)**를 클릭하여 도표화된 곡선을 레퍼런스 곡선으로 설정할 수 있습니다. 레퍼런스 곡선은 새 곡선을 레퍼런스로 설정할 때까지 동일하게 유지됩니다. 파란색 그래프 선은 기존 곡선을 나타냅니다.



## 확대/축소 표시 옵션

다음 옵션을 사용할 수 있습니다.

- **확대/축소:** 한 번에 더 많은 프로그램을 보려면.
- **확대:** 특정 프로그램에 대해 상세히 집중하다.
- **확대 영역:** 프로그램에 대한 맞춤형 보기를 가지다.

## 주요 버튼

- **전부하 전류:** 전체 부하 전류 값을 암페어로 표시합니다.
- **타임스탬프:** 모터 시작 시간을 표시합니다.
- **줄거리:** 최신 값을 읽고 모터 시작 특성 곡선을 새로 고칩니다.
- **저장:** 모터 시작 특성 곡선의 스크린 샷을 캡처하여 PC에 저장할 수 있습니다.
- **참조로 설정:** 실제 곡선을 레퍼런스 곡선으로 설정합니다.

# Custom Logic

## 개요

**Custom Logic(사용자 정의 논리)** 탭에는 새 사용자 정의 논리 프로그램 파일을 작성하거나 기존 사용자 정의 논리 프로그램 파일을 열고 편집할 수 있는 Custom Logic Editor가 표시됩니다.

**Custom Logic(사용자 정의 논리)** 탭에 대한 자세한 내용은 Custom Logic Editor, 95 페이지 항목을 참조하십시오.

# FB Diagram

## 개요

**FB Diagram(FB 다이어그램)** 탭은 기능 블록 다이어그램 편집기를 표시하는 데 사용되며, 이를 통해 새 기능 블록 다이어그램 파일을 생성하거나 기존 기능 블록 다이어그램 파일을 열고 편집할 수 있습니다.

**FB Diagram(FB 다이어그램)** 탭에 대한 자세한 내용은 FBD 편집기 소개, 163 페이지 항목을 참조하십시오.

# Logic Simulator

## 개요

**Logic Simulator(논리 시뮬레이터)** 탭을 사용하여 사용자 정의 논리 프로그램의 기능을 테스트할 수 있는 LTMT main unit **Logic Simulator(논리 시뮬레이터)**를 표시합니다.

Logic Simulator에 대한 자세한 내용은 LTMT Main Unit Logic Simulator, 196 페이지 항목을 참조하십시오.

# 사용자 기능

## 이 파트의 내용

펌웨어 업데이트 .....	84
HMI 언어 업데이트 .....	88
공장 초기화 .....	90
핀 관리 .....	91

# 펌웨어 업데이트

## 개요

해당 펌웨어의 TeSys Tera system 및 HMI 장치의 펌웨어는 다음을 사용하여 업데이트할 수 있습니다. TeSys Tera DTM.

펌웨어는 다음 모드에서 업데이트할 수 있습니다:

- TeSys Tera system – 온라인 모드
- HMI 기기 – 오프라인 모드

**주의:** 펌웨어 업데이트는 SoMove 소프트웨어 홈페이지에서는 펌웨어 업데이트를 수행할 수 없습니다.

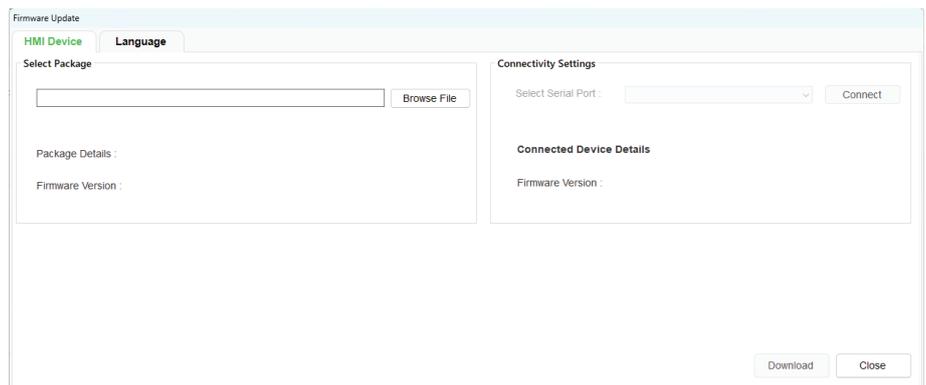
## HMI 장치 펌웨어 업데이트

해당 장치의 펌웨어는 HMI 장치는 오프라인 모드에서 DTM을 사용하여 업데이트할 수 있습니다.

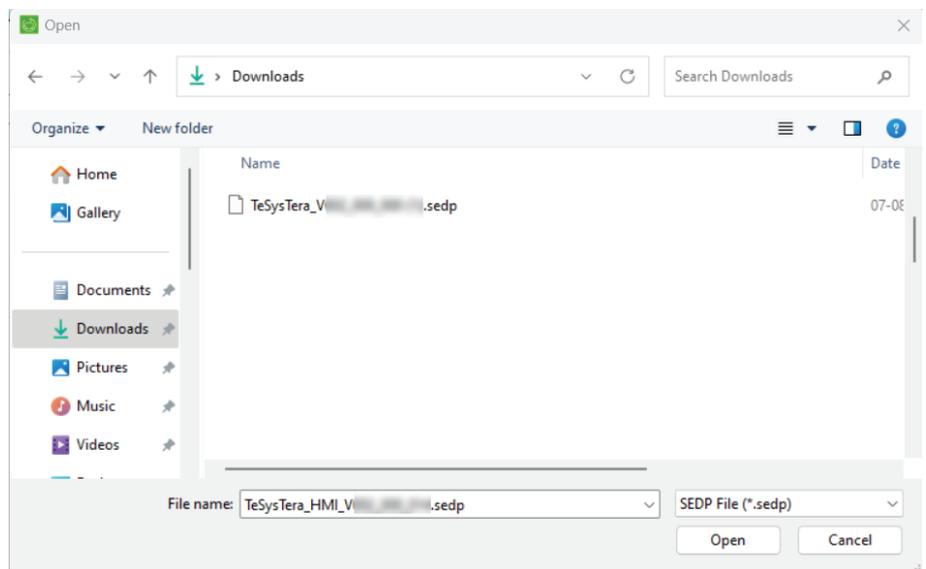
다음 단계를 수행하여 HMI 장치:

1. 메뉴 막대에서 선택하십시오 **장치 > 유지 관리 > 펌웨어 업데이트**.

**결과: Firmware Update(펌웨어 업데이트) 창이 나타납니다.**



2. 선택 **파일 찾아보기** 아래와 같이 PC에서 펌웨어 패키지 파일을 찾아 선택하십시오:



3. **Open(열기)**을 선택합니다.

4. 시리얼 포트를 선택하십시오. 직렬 포트 선택 드롭다운 목록을 사용하여 포트를 선택하고 연결.
5. 다음 지시사항이 포함된 팝업 LTMTCUF의 "정지" 및 "재설정" 버튼을 누른 상태에서 LTMT에 연결하여 LTMTCUF의 전원을 동시에 켭니다. 그런 다음 확인을 누르세요 표시됩니다.  
표시된 장치에 수행할 지침을 따르십시오. HMI 표시된 대로 기기에서 수행해야 할 지침을 따르십시오.
6. 위 절차를 수행한 후, 확인 을 선택하십시오.
7. 선택 다운로드.  
주의: 이 작업은 펌웨어를 업데이트하는 데 4~12분이 걸립니다.
8. 장치를 재시작하십시오. HMI 장치를 재시작하십시오.  
결과: 해당 장치의 펌웨어가 업데이트됩니다. HMI 장치의 펌웨어가 업데이트됩니다.

## TeSys Tera 시스템 펌웨어 업데이트

TeSys Tera system의 펌웨어를 다음 소스 중 하나를 사용하여 업데이트할 수 있습니다.

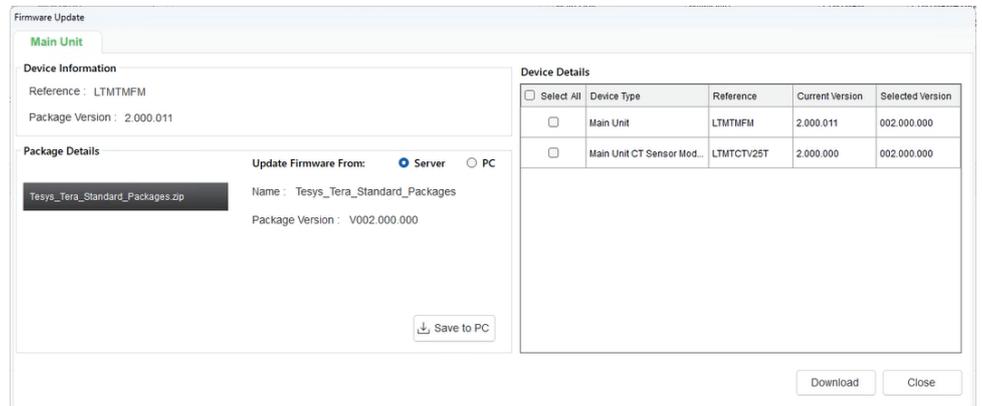
- 서버
- PC

주의: PC가 HMI 포트에 연결되어 있지 않은지 확인하십시오. LTMTCUF control operator unit.

메뉴 막대에서 선택하십시오 장치 > 유지 관리 > 펌웨어 업데이트.

주의: 해당 펌웨어 업데이트는 TeSys Tera system 온라인 모드에서 수행할 수 있습니다.

결과: Firmware Update(펌웨어 업데이트) 창이 나타납니다.



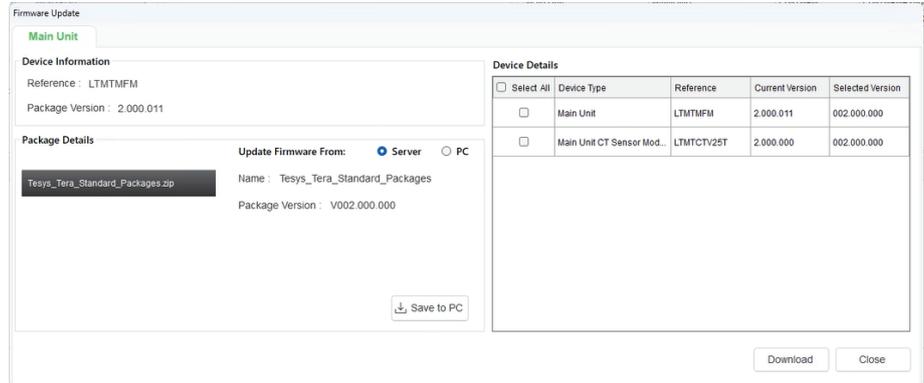
## 서버에서 펌웨어 업데이트하기

서버에서 펌웨어를 업데이트하려면 다음 단계를 수행하십시오.

1. 펌웨어 업데이트 펌웨어 업데이트 창에서 서버 을 선택하십시오. 옵션에서 옵션에서

- 다음에서 펌웨어 패키지를 선택하십시오. **펌웨어 패키지 목록** 에서 펌웨어 패키지를 선택하고 **다운로드**를 선택하십시오..

**주의:** 해당 펌웨어 업데이트는 Ethernet 버전에 대한 펌웨어 업데이트는 TeSys Tera system 전체 패키지 업로드 방식으로 수행됩니다. 개별 모듈 업데이트를 수행할 수 없습니다.



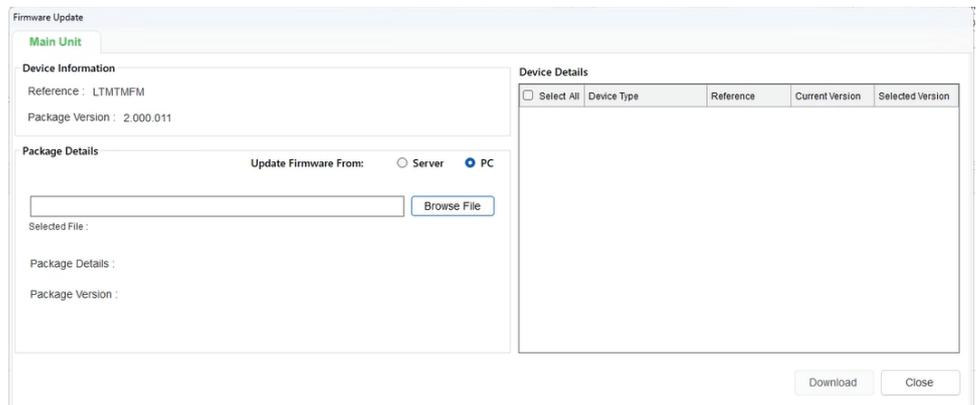
**주의:** 펌웨어 업데이트 속도를 높이기 위해 장치에서 지원하는 가장 높은 전송 속도(115200)를 선택하십시오.

**결과:** **Firmware Update(펌웨어 업데이트)** 창 하단에 업데이트에 성공했다는 메시지가 표시됩니다.

## PC에 펌웨어 다운로드하기

선택  아이콘을 선택하여 펌웨어 패키지를 다운로드하고 **PC**.

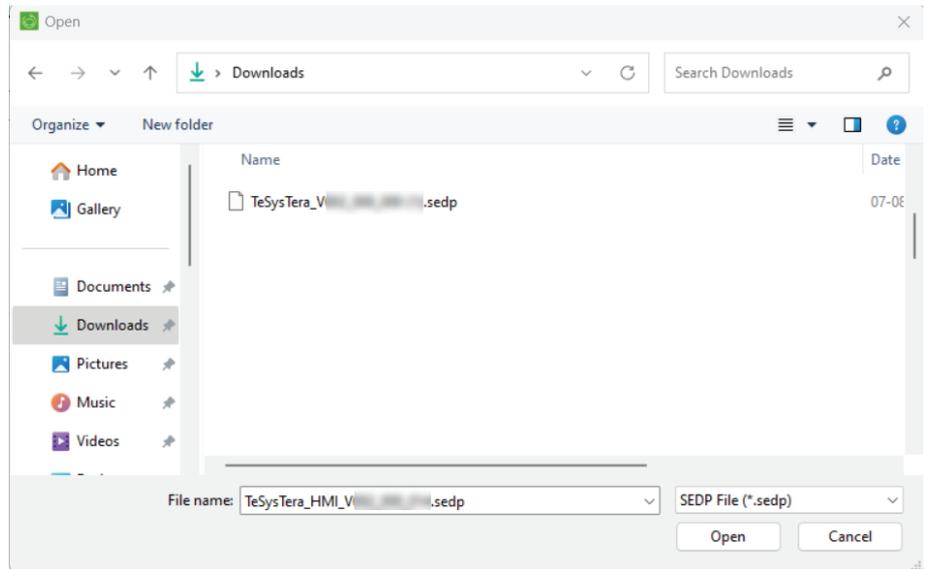
## PC에서 펌웨어 업데이트하기



펌웨어를 업데이트하려면 **PC**에서 펌웨어를 업데이트하려면 다음 단계를 수행하십시오:

- 펌웨어 업데이트 **펌웨어 업데이트** 창에서 **PC** 을 선택하십시오.

2. 선택 **파일 찾아보기** 아래와 같이 PC에서 펌웨어 패키지 파일을 찾아 선택하십시오:



3. **Open(열기)**을 선택합니다.

**결과: Firmware Update(펌웨어 업데이트)** 창 하단에 업데이트에 성공했다는 메시지가 표시됩니다.

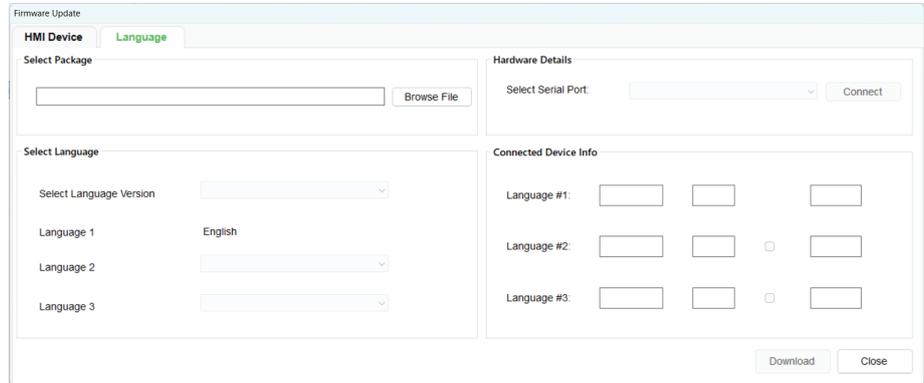
4. 업데이트 실패 시 **Firmware Update Failed(펌웨어 업데이트 실패)** 대화 상자가 나타납니다. 위 절차를 다시 수행하여 펌웨어를 업데이트하십시오.

## HMI 언어 업데이트

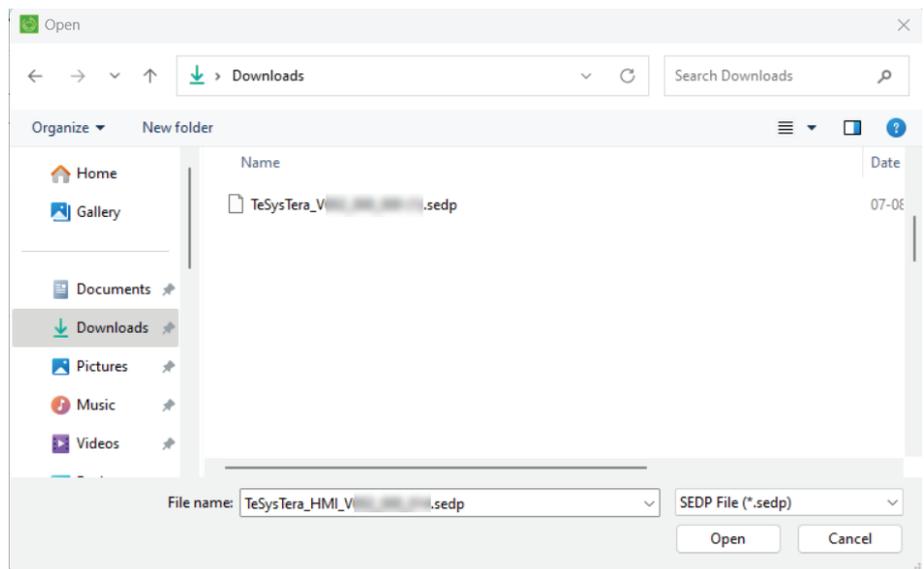
언어HMI업데이트는 를 사용하여 수행할 수 있습니다 TeSys Tera DTM.

다음 절차는 언어를 업데이트하는 방법을 설명합니다. HMI:

1. 메뉴 막대에서 선택 **장치 > 유지 관리 > 펌웨어 업데이트**.
2. 다음으로 이동하십시오 **언어** 탭으로 이동하세요.



3. 클릭 **파일 찾아보기** 을 클릭하여 아래와 같이 PC에서 펌웨어 패키지 파일을 찾아 선택하십시오:



4. 열기를 클릭합니다.

5. In the 언어 선택 섹션에서, 각 드롭다운 목록을 사용하여 언어 버전과 사용 언어를 선택하십시오.

**주의:** 지원되는 언어는 다음과 같습니다.

- 체코어
- 브라질 포르투갈어
- 덴마크어
- 독일어
- 영어
- 스페인어
- 페르시아어
- 핀란드어
- 프랑스어
- 히브리어
- 인도네시아어
- 이탈리아어
- 한국어
- 네덜란드어
- 포르투갈어
- 러시아어
- 중국어

6. 시리얼 포트를 선택하십시오. 직렬 포트 선택 드롭다운 목록을 사용하여 포트를 선택하고 연결.

7. 현재 언어 세부 정보는 HMI 연결된 장치 정보 연결된 장치 정보.

**주의:** 언어는 체크박스를 사용하여 활성화하거나 비활성화할 수 있습니다.

8. 선택 다운로드.

9. 장치를 재시작하십시오. HMI 장치를 재시작하십시오.

**결과:** 해당 장치의 언어가 HMI 장치의 언어가 업데이트됩니다.

# 공장 초기화

## 개요

**Factory Reset(공장 초기화)** 기능을 사용하면 TeSys Tera system 매개 변수를 해당 기본값으로 초기화할 수 있습니다.

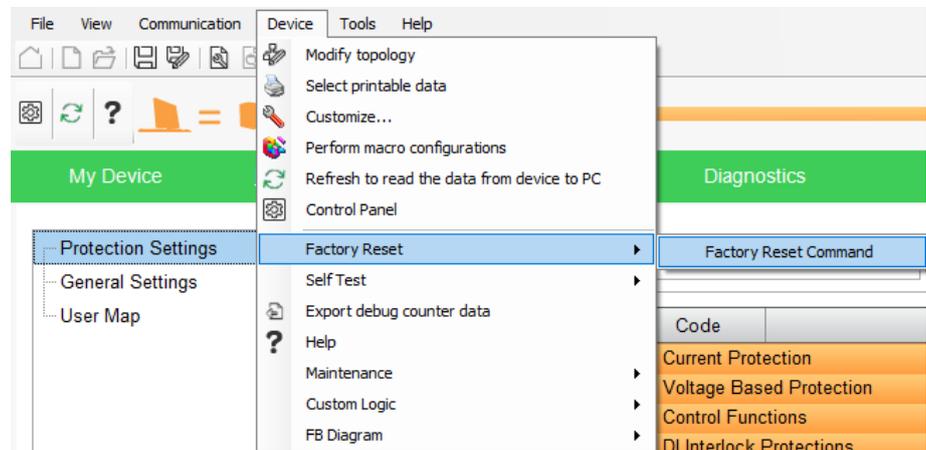
**주의:** 직렬 인터페이스(HMI 포트)를 통해 초기화 기능이 TeSys Tera DTM 가 연결된 경우, Ethernet 변형에 대해 를 통한 공장 LTMT main unit 지원됩니다.

## 공장 초기화 명령 수행하기

공장 초기화 명령을 수행하려면 다음 단계를 수행하십시오.

**주의:** 공장 초기화 기능은 연결 모드에서만 작동합니다.

- 아래 그림과 같이 **Device(장치) > Factory Reset(공장 초기화) > Factory Reset Command(공장 초기화 명령)**를 선택합니다.



- 명령을 선택한 후 확인 팝업 메시지가 나타납니다.

TeSys Tera



Are you sure to send this command?

Yes

No

**주의:** 이 명령을 실행한 후 TeSys Tera DTM Library에서 TeSys Tera system 연결이 해제됩니다.

**Factory Reset(공장 초기화)** 명령이 실행되면 TeSys Tera system의 구성이 공장 설정으로 복구하고 TeSys Tera system이 재시작됩니다.

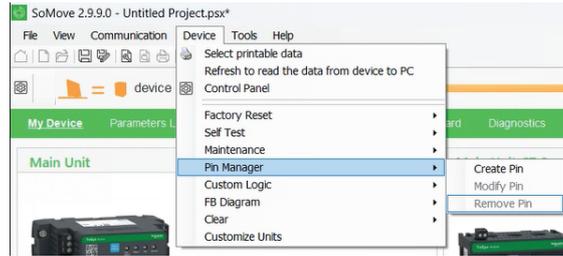
# 핀 관리

**Pin Manager(핀 관리자)**를 사용하면 SoMove software에서 핀을 설정하여 장치를 보호할 수 있습니다.

**Pin Manager(핀 관리자)**는 **Device(장치) > Pin Manager(핀 관리자)**로 이동하여 액세스할 수 있습니다.

주의:

- **Pin Manager(핀 관리자)**는 SoMove software에 장치가 연결된 경우에만 사용할 수 있습니다.
- **핀 관리자** 기능은 Ethernet 변형에 대해 지원됩니다. LTMT main unit 시리얼 인터페이스(HMI포트)로 연결된 경우



**Pin Manager(핀 관리자)**에서 다음 작업을 수행할 수 있습니다.

- Create Pin(핀 생성)
- PIN 수정
- Remove Pin(핀 제거)

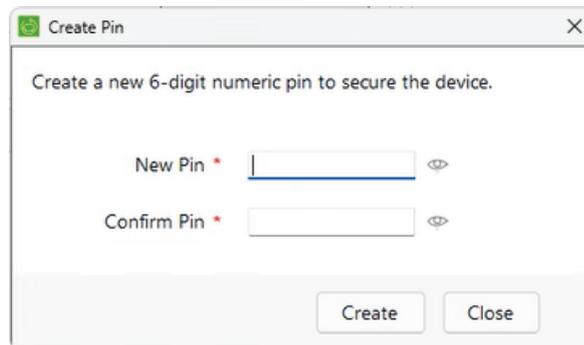
# 핀 생성

**Create Pin(핀 생성)** 옵션을 사용하면 새 핀을 생성할 수 있습니다. **Create Pin(핀 생성)** 옵션은 핀이 설정되어 있지 않은 경우에만 사용할 수 있습니다.

다음 절차에서는 핀을 생성하는 방법을 보여 줍니다.

1. **Device(장치) > Pin Manager(핀 관리자) > Create Pin(핀 생성)**으로 이동한 다음.

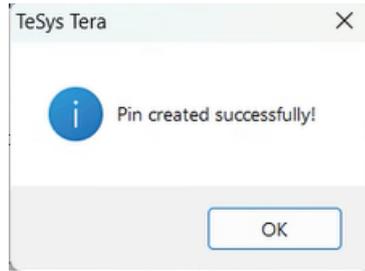
결과: **Create Pin(핀 생성)** 대화 상자가 나타납니다.



2. **New Pin(새 핀)** 및 **Confirm Pin(Pin 확인)** 필드에 새로운 6자리 숫자 핀을 입력합니다.

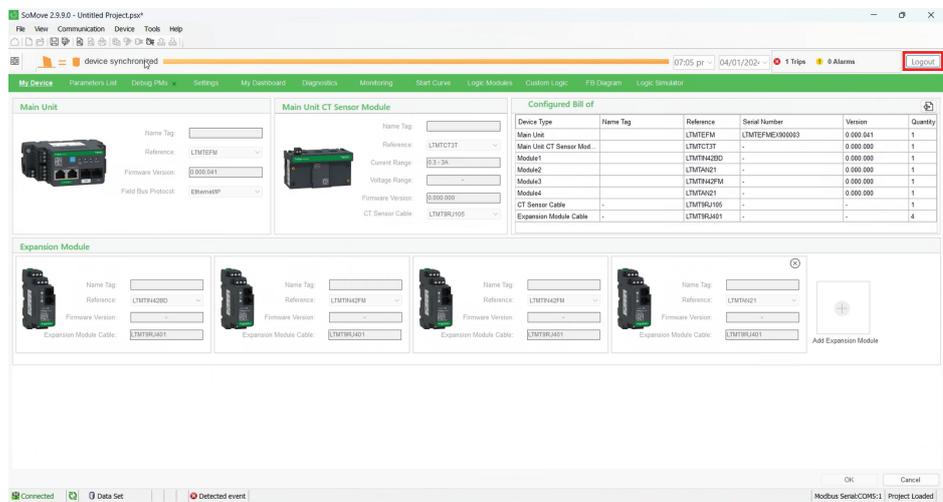
3. **Create(생성)**를 선택합니다.

결과: 핀이 생성됩니다.



**Parameter List(매개 변수 목록) > Device Session Management(장치 세션 관리)** 탭에서 **Device Session Timeout(장치 세션 시간 초과)** 매개 변수를 사용하여 세션 시간을 설정할 수 있습니다. 각 세션에 대해 최소 5분에서 최대 60분까지 설정할 수 있습니다.

**Logout(로그아웃)**을 선택하여 SoMove software에서 장치를 분리합니다.



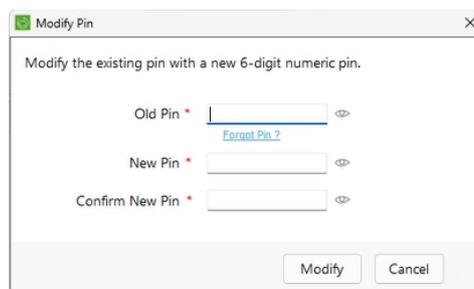
## 핀 수정

**Modify Pin(핀 수정)** 옵션을 사용하면 핀을 수정할 수 있습니다. **Modify Pin(핀 수정)** 옵션은 핀을 설정한 후에만 사용할 수 있습니다.

다음 절차에서는 핀을 수정하는 방법을 보여 줍니다.

1. **Device(장치) > Pin Manager(핀 관리자) > Modify Pin(핀 수정)**으로 이동합니다.

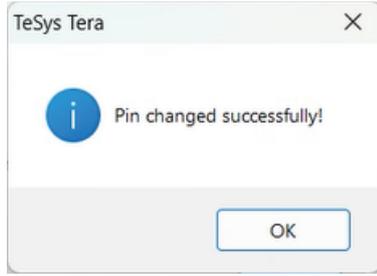
결과: **Modify Pin(핀 수정)** 대화 상자가 나타납니다.



2. **Old Pin(이전 핀), New Pin(새 핀) 및 Confirm New Pin(새 Pin 확인)** 각각에 대한 값을 입력합니다.

3. **Modify(수정)**를 선택합니다.

결과: 핀이 수정됩니다.



## 핀 삭제

**Remove Pin(핀 제거)** 옵션을 사용하면 기존 핀을 삭제할 수 있습니다. **Remove Pin(핀 제거)** 옵션은 핀을 설정한 후에만 사용할 수 있습니다.

다음 절차에서는 핀을 제거하는 방법을 보여 줍니다.

1. **Device(장치) > Pin Manager(핀 관리자) > Remove Pin(핀 제거)**으로 이동합니다.

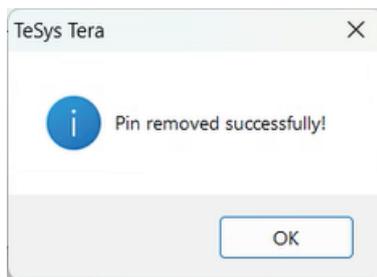
결과: **Remove Pin(핀 제거)** 대화 상자가 나타납니다.



2. **Pin(핀)** 필드에 기존 핀을 입력합니다.

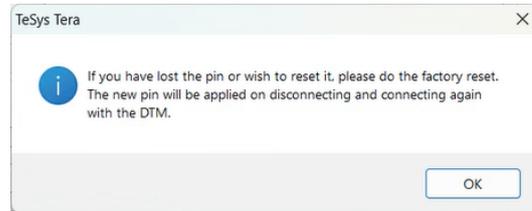
3. **Remove(제거)**를 선택합니다.

결과: 핀이 제거됩니다.



## 핀을 잊음

핀을 잊은 경우 장치에서 사용할 수 있는 **Reset(초기화)** 버튼을 사용하여 공장 초기화를 수행해야 합니다.



**주의:** 공장 초기화를 수행하면 TeSys Tera DTM에서 장치 연결을 끊습니다.

# Custom Logic Editor

## 이 파트의 내용

Custom Logic Editor 프레젠테이션.....	96
Custom Logic Editor 사용하기.....	99
사용자 정의 논리 프로그램의 특징.....	102
사용자 정의 논리 변수의 정의.....	103
LTMT Main Unit 변수의 정의.....	104
CALL_EOM 명령 설명.....	107

# Custom Logic Editor 프레젠테이션

## 개요

Custom Logic Editor를 사용하면 LTMT main unit의 사전 정의 제어 프로그램을 사용자 정의할 수 있습니다. Custom Logic Editor는 SoMove가 있는 TeSys Tera DTM Library software에서만 사용할 수 있는 강력한 프로그래밍 도구입니다. LTMT main unit을 위한 사용자 정의 제어 프로그램을 생성하는 것은 사용자 정의 논리 프로그래밍 언어 중 하나로 일련의 지침(논리 명령)을 작성하는 과정입니다.

## Custom Logic Editor의 목적

Custom Logic Editor의 주요 목적은 스타터 유형에 대해 사전 정의 논리를 수정하거나 추가 논리 기능을 추가할 수 있는 유연성을 제공하는 것입니다.

## 논리 ID

각 LTMT main unit 작동 모드(또는 모터 스타터 유형)마다 사전 정의 제어 프로그램이 하나 있습니다. 이러한 작동 모드(또는 모터 스타터 유형) 프로그램은 고유한 논리 ID로 식별됩니다. 사전 정의 작동 모드 프로그램의 논리 ID는 12부터 15까지의 숫자이다. 사전 정의 작동 모드 프로그램을 사용자 정의할 때, 사용자 정의 프로그램의 논리 ID는 사전 정의 프로그램의 논리 ID에 256을 더한 값과 같아야 합니다.

이 표에서는 작동 모드에 따른 논리 ID를 제공합니다.

작동 모드/스타터 유형	사전 정의 프로그램의 논리 ID	사용자 정의 프로그램의 논리 ID
예약됨	0~11	-
과부하	12	268
Direct Online	13	269
Reverse Direct Online	14	270
Star Delta	15	271
사용자 정의 프로그램	-	256~267, 279~511

## 사용자 정의 프로그램

사용자 정의 프로그램은 개별 애플리케이션 요구 사항을 충족하기 위해 특정 기능을 갖춘 LTMT main unit의 사전 정의 프로그램입니다.

사전 정의 프로그램 중 하나로 구성된 경우 LTMT main unit은 LTMT main unit의 펌웨어와 PCode를 모두 사용하여 제어 기능을 관리합니다.

**주의:** 의사 코드(Pseudo Code)라고도 하는 PCode는 오류가 없는 16진수 명령문 집합입니다.

사용자 정의 프로그램으로 구성된 경우 LTMT main unit은 LTMT main unit에 의해 제어되는 기능을 유지합니다. 이러한 기능에는 상위 사전 정의 프로그램에 내재된 다음과 같은 특성이 포함됩니다.

- 주소 704(0x02C0)에 등록하기 위해 기록할 수 있는 항목에 대한 제한 사항(네트워크 명령 레지스터)

- 프레젠테이션 모드에서 작동 상태 표시(예: 정방향 또는 역방향, 저속 또는 고속)
- 메뉴를 통해 설정할 수 있는 대체 모드에 대한 제한 사항
- 스타-델타 기동 시기의 특정 동작
- 메뉴를 통해 설정할 수 있는 전환 타이머에 대한 제한 사항

## 사전 정의 프로그램 구조

SoMove software에는 TeSys Tera DTM Library와 함께 사용할 수 있는 11개의 사전 정의 프로그램이 있으며, 각 작동 모드(또는 모터 스타터 유형)마다 하나의 사전 정의 프로그램이 있습니다.

사전 정의 프로그램은 다음 부분을 차례로 수행합니다.

- 논리 ID를 사용하여 프로그램의 논리 식별
- 입력 관리
- 작동 모드 실행
- 출력 업데이트

작동 모드의 실행은 내장되어 있으며 `CALL_EOM` 함수로 호출됩니다.

따라서 작동 모드 실행을 수정하지 않고 사용자 정의 프로그램의 입력 및 출력 관리를 사용자 정의할 수 있습니다.

## Custom Logic Editor 프로그래밍 언어 및 도구

Custom Logic Editor는 두 가지 프로그래밍 언어 및 도구를 제공합니다.

- 사용자 정의 논리 언어: 리스트 명령어 언어는 사용자 정의 로직 에디터 프로그래밍 도구를 통해 편집 가능한가?
- FBD: 객체 지향 프로그래밍 언어는 FBD 편집기 프로그래밍 도구를 통해 편집할 수 있습니다.

각 프로그래밍 방법은 프로그래밍 목표를 충족합니다. 그러나 Custom Logic Editor를 사용하면 원하는 프로그래밍 방법의 스타일을 선택할 수 있습니다.

## Logic Commands

사용자 정의 논리 언어와 FBD 언어는 모두 다음 유형의 명령을 구현합니다.

- 프로그램 논리 명령
- Boolean Logic 명령
- Register Logic 명령
- Timer Logic 명령
- Counter Logic 명령
- Latch logic 명령
- Math Logic 명령

## Custom Logic Editor

다음 그림은 TeSys Tera DTM Library에 통합된 Custom Logic Editor를 보여 줍니다.



# Custom Logic Editor 사용하기

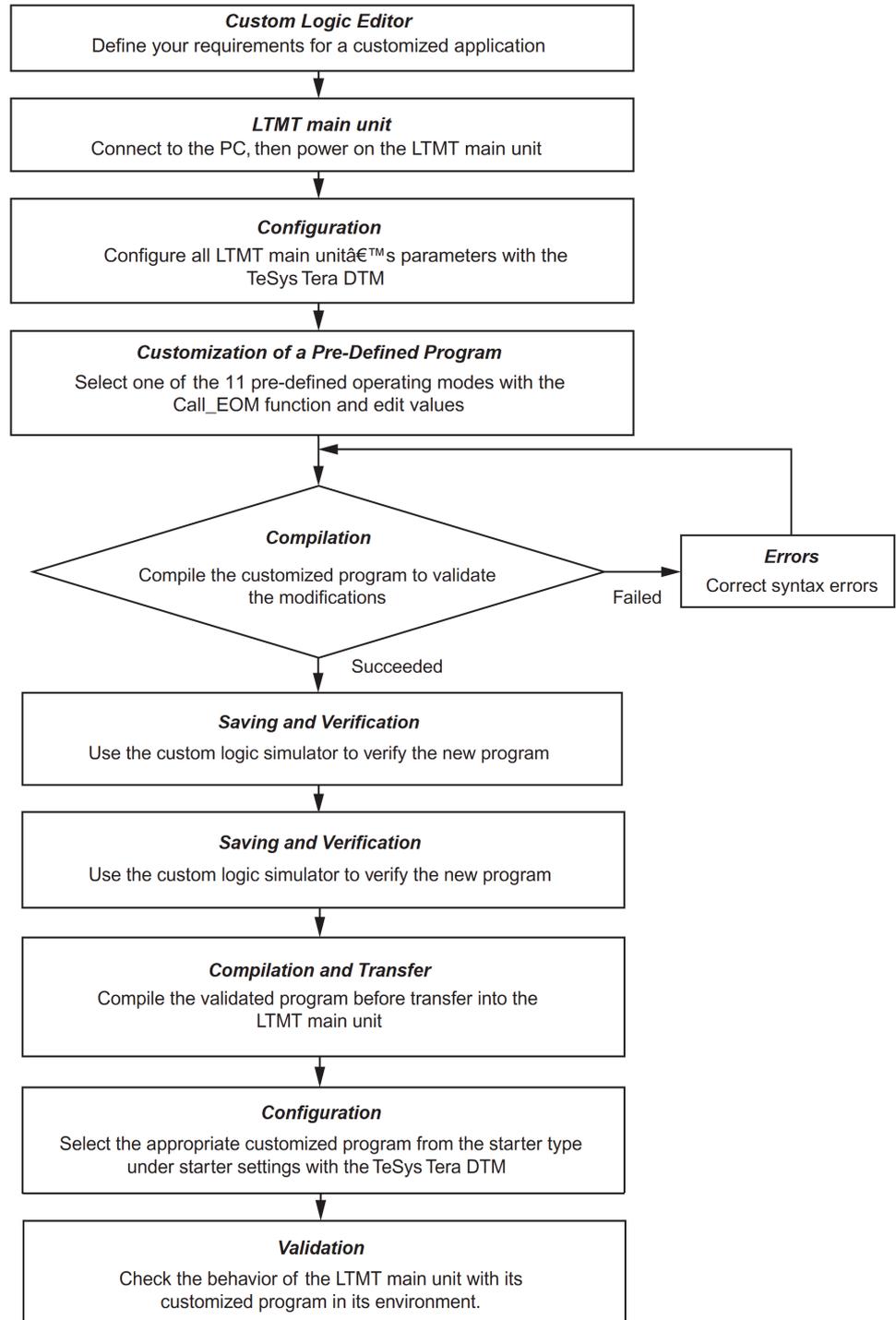
## 개요

Custom Logic Editor를 사용하면 요구 사항에 맞는 고유한 사용자 정의 프로그램을 만들고 검증할 수 있습니다. LTMT main unit 펌웨어는 이러한 사용자 정의 프로그램을 로드하고 사용자가 작성한 명령문을 실행합니다.

## 작업 흐름 다이어그램

다음 다이어그램에서는 사용자 정의 프로그램을 만들고 수정하는 동안 수행하는 모든 작업을 보여 줍니다.

**참고:** 정의된 순서는 예를 들기 위한 것입니다. 사용하는 순서는 사용자의 작업 방법에 따라 달라집니다.



## 사용자 정의 논리의 사용자 정의 방법

사전 정의 프로그램 파일을 사용자 정의하려면 다음 단계를 따르십시오.

1. 애플리케이션 요구 사항에 맞는 작동 모드를 정의합니다.
2. Custom Logic Editor에서 사전 정의 작동 모드 프로그램 파일(\*.rtf)을 엽니다.

3. 사용자 정의 논리에 따라 사전 정의 프로그램을 편집하고 다음 세 가지 방법 중 하나를 사용하여 프로그램을 사용자 정의합니다.
  - 사전 정의된 작동 모드는 애플리케이션 요구 사항에 부합합니다: 오직 CALL\_EOM 함수만 사용하십시오.
  - 사전 정의된 작동 모드는 애플리케이션 요구 사항에 부합하지만, 추가 기능이 필요합니다: 사용하십시오 CALL\_EOM 함수를 사용하고 CALL\_EOM 지시사항 뒤에 추가 지시사항을 추가하십시오.
  - 사전 정의된 작동 모드가 귀하의 애플리케이션 요구 사항과 일치하지 않습니다: 새 프로그램을 처음부터 시작하세요(권장하지 않음).
4. 필요한 경우 사용자 정의 프로그램의 입력을 편집합니다.
5. 필요한 경우 사용자 정의 프로그램의 출력을 편집합니다.
6. CALL\_EOM 및 제어 모드에 따라 Custom Logic Editor 프레젠테이션, 96 페이지를 업데이트합니다.
7. 사용자 정의 프로그램을 컴파일합니다.

## FBD의 사용자 정의 방법

1. 빈 FBD 프로그램 페이지를 엽니다.
2. 사용자 정의 프로그램의 입력 관리를 생성합니다.
3. 다음 세 가지 방법 중 하나를 사용하여 작동 모드 실행을 생성합니다.
  - 다음 운영 모드 중 하나가 귀하의 애플리케이션 요구 사항에 부합합니다: 오직 CALL\_EOM 함수만 사용하십시오.
  - 운영 모드 중 하나가 애플리케이션 요구 사항과 일치하지만, 추가 기능이 필요합니다: 사용하십시오 CALL\_EOM 함수를 사용하고 CALL\_EOM 지시사항 뒤에 추가 지시사항을 추가하십시오.
  - 사용 중인 운영 모드 중 어느 것도 귀하의 애플리케이션 요구 사항과 일치하지 않습니다: 새 프로그램을 처음부터 생성합니다(권장하지 않음).
4. 사용자 정의 프로그램의 출력 관리를 만듭니다.
5. CALL\_EOM 및 제어 모드에 따라 Custom Logic Editor 프레젠테이션, 96 페이지를 업데이트합니다.
6. FBD를 사용자 정의 논리로 컴파일합니다.

# 사용자 정의 논리 프로그램의 특징

## 소개

LTMT main unit에서 송수신되는 데이터는 16비트 레지스터 형태입니다. 레지스터는 16비트 레지스터 주소(0-65,535)로 순서가 지정되고 참조됩니다.

사용자 정의 프로그램은 다음과 같은 세 가지 유형의 레지스터 값을 수정할 수 있습니다.

- LTMT 레지스터
- 임시 레지스터
- 비휘발성 레지스터

## 논리 메모리 특성

사용자 정의 프로그램의 명령 목록은 LTMT main unit의 내부 비휘발성 메모리 영역에 저장됩니다.

이 논리 메모리의 형식은 다음 표에 설명되어 있습니다.

메모리 위치	항목	범위	설명
0	논리 프로그램 크기(n)	0~8,191 0은 로드된 사용자 정의 프로그램이 없다는 의미입니다.	16비트 워드
1	논리 체크섬	0~65,535	오프셋 2-n+2의 프로그램 메모리 합계
2	논리 ID	256~511, 참조	LTMT main unit내부에서 사용자 정의 프로그램의 식별자
3	논리 명령/인수 1	논리 명령에 따라 다름	논리 기능의 1워드
4	논리 명령/인수 2		
5	논리 명령/인수 3		
-	-	-	-
n+2	논리 명령/인수 n	-	논리 기능의 1워드

## 논리 메모리 제한

프로그램 크기는 논리 명령 수에 따라 다릅니다. 텍스트 편집기에서는 명령과 해당 인수가 한 줄을 차지하지만, 메모리에서는 인수가 있는 수만큼 메모리 위치를 차지합니다.

예를 들어, 명령 타이머 0.1980은 네 개의 메모리 위치를 사용합니다.

# 사용자 정의 논리 변수의 정의

## 소개

Custom Logic Editor를 사용하면 제어 프로그램에서 LTMT main unit의 다음 동작을 제어하는 명령을 구현할 수 있습니다.

- 임시 레지스터 읽기 또는 쓰기
- 비휘발성 레지스터 읽기 또는 쓰기
- LTMT main unit 레지스터 읽기 또는 쓰기(LTMT Main Unit 변수의 정의, 104 페이지 참조)

LTMT main unit은 사용자 정의 논리 메모리 공간의 주소를 나타내는 정수로 각 사용자 정의 논리 레지스터를 정의합니다. 이 정수의 값은 주소 0에서 시작하며 최대 주소의 값은 LTMT main unit에서 레지스터에 사용할 수 있는 메모리 위치 수보다 1 작습니다.

## 임시 레지스터

LTMT main unit은 논리 명령에서 액세스할 수 있는 임시 메모리에 레지스터를 제공합니다. 이러한 레지스터는 임시 또는 휘발성 메모리에 존재하기 때문에 컨트롤러에서 전원 주기가 발생하면 값 설정이 유지되지 않습니다.

0에서 299 범위의 주소에서 300개의 임시 레지스터를 사용할 수 있습니다.

사용할 수 있는 임시 레지스터의 수는 사용자 정의 논리 임시 공간의 주소 1204 (0x04B4)인 LTMT main unit 레지스터에서 읽을 수 있습니다.

## 비휘발성 레지스터

LTMT main unit은 논리 명령에서 사용할 수 있도록 비휘발성 메모리에 레지스터를 제공합니다. 이러한 레지스터는 비휘발성 메모리에 존재하기 때문에 컨트롤러에서 전원 주기가 발생해도 값 설정이 유지됩니다.

0에서 63 범위의 주소에서 64개의 비휘발성 레지스터를 사용할 수 있습니다.

사용할 수 있는 비휘발성 레지스터의 수는 사용자 정의 논리 비휘발성 공간의 주소 1205(0x04B5)인 LTMT main unit 레지스터에서 읽을 수 있습니다.

## LTMT Main Unit 변수의 정의

### LTMT Main Unit 레지스터

LTMT main unit 메모리에는 주소 범위가 0에서 9249(0x0000에서 0x2421)인 9250개의 레지스터가 포함되어 있습니다.

각 레지스터는 16비트 워드이며, 다음 상태 중 하나입니다.

- 읽기 전용 - 값을 편집할 수 없습니다.
- 읽기/쓰기 - 값을 편집할 수 있습니다.

레지스터 번호는 레지스터 주소 + 1과 같습니다.

Custom Logic Editor는 레지스터 주소만 사용합니다.

사용자 정의 논리 명령을 사용하여 LTMT main unit의 읽기/쓰기 레지스터 값을 변경할 수 있습니다.

### 레지스터에 액세스하기

Custom Logic Editor를 사용하면 LTMT main unit TeSys Tera 통신 가이드에 정의되어 있는 모든 레지스터에 액세스할 수 있습니다, 7 페이지.

### 사용자 정의 논리 레지스터

주소 1200~1205(0x04B0~0x04B5)의 레지스터와 주소 1291(0x050B)의 레지스터는 TeSys Tera DTM Library에서 LTMT main unit 레지스터에 액세스하기 위해 사용됩니다.

이러한 레지스터는 통신 포트에서 액세스할 수 있는 사용자 정의 논리 레지스터이기도 합니다. 이러한 읽기 전용 레지스터에 대해서는 다음 단원에서 설명합니다.

다음 표는 이러한 레지스터를 보여 줍니다.

레지스터 주소	정의	범위(값)
1201(0x04B1)	사용자 정의 논리 버전	0~65,535
1202(0x04B2)	사용자 정의 논리 메모리 공간	
1203(0x04B3)	사용된 사용자 정의 논리 메모리	
1204(0x04B4)	사용자 정의 논리 임시 공간	
1205(0x04B5)	사용자 정의 논리 비휘발성 공간	
1291(0x050B)	사용자 정의 논리 DO 입력 정보	

### 주소 1201(0x04B1)의 레지스터

주소 1201(0x04B1)의 레지스터는 사용자 정의 논리 기능 버전을 나타냅니다. 버전 번호는 LTMT main unit에서 지원하는 특정 논리 명령 그룹을 식별합니다.

## 주소 1202(0x04B2)의 레지스터

주소 1202(0x04B2)의 레지스터는 사용 가능한 논리 메모리 공간을 정의합니다. 논리 명령을 저장하는 데 사용할 수 있는 비휘발성 LTMT main unit 논리 메모리 워드(16비트)의 수입니다.

## 주소 1203(0x04B3)의 레지스터

주소 1203(0x04B3)의 레지스터는 사용된 논리 메모리를 정의합니다. 이는 현재 LTMT main unit에 저장되어, 논리 명령에 사용되는 비휘발성 LTMT main unit 논리 메모리 워드(16비트)의 수입니다.

## 주소 1204(0x04B4)의 레지스터

주소 1204(0x04B4)의 레지스터는 LTMT main unit에 의해 제공되는 임시 레지스터의 수를 정의합니다.

## 주소 1205(0x04B5)의 레지스터

주소 1205(0x04B5)의 레지스터는 LTMT main unit에 의해 제공되는 비휘발성 레지스터의 수를 정의합니다.

## 주소 1291(0x050B)의 레지스터

주소 1291(0x050B)의 레지스터는 사용자 정의 논리 DO 입력 정보 레지스터입니다. 이를 통해 사용자 정의 프로그램에서 I/O 할당을 구성할 수 있습니다.

다음 표에서는 이 레지스터의 각 비트를 설명합니다.

레지스터 주소	비트 번호	설명
1291(0x050B)	0	사용자 정의 논리 디지털 출력 1(DO1) 입력 정보
	1	사용자 정의 논리 디지털 출력 2(DO2) 입력 정보
	2	사용자 정의 논리 디지털 출력 3(DO3) 입력 정보
	3	사용자 정의 논리 디지털 출력 4(DO4) 입력 정보
	4	사용자 정의 논리 디지털 출력 5(DO5) 입력 정보
	5	사용자 정의 논리 디지털 출력 6(DO6) 입력 정보
	6	사용자 정의 논리 디지털 출력 7(DO7) 입력 정보
	7	사용자 정의 논리 디지털 출력 8(DO8) 입력 정보
	8	사용자 정의 논리 디지털 출력 9(DO9) 입력 정보
	9	사용자 정의 논리 디지털 출력 10(DO10) 입력 정보
	10	사용자 정의 논리 디지털 출력 11(DO11) 입력 정보
	11	사용자 정의 논리 디지털 출력 12(DO12) 입력 정보
	12	사용자 정의 논리 디지털 출력 13(DO13) 입력 정보
	13	예약됨
14	예약됨	

레지스터 주소	비트 번호	설명
	15	예약됨

## 1301~1399(0x0515~0x0577) 주소의 레지스터

1301~1399(0x0515~0x0577) 주소의 레지스터는 논리 기능을 위한 범용 레지스터입니다. 이러한 레지스터는 외부 소스(예: PLC)와 사용자 정의 논리 애플리케이션 간에 정보를 교환하는 데 사용됩니다.

이러한 휘발성 레지스터는 읽기 또는 쓰기가 가능하며, 사용자 정의 논리 기능이나 통신 포트를 통해 편집할 수 있습니다.

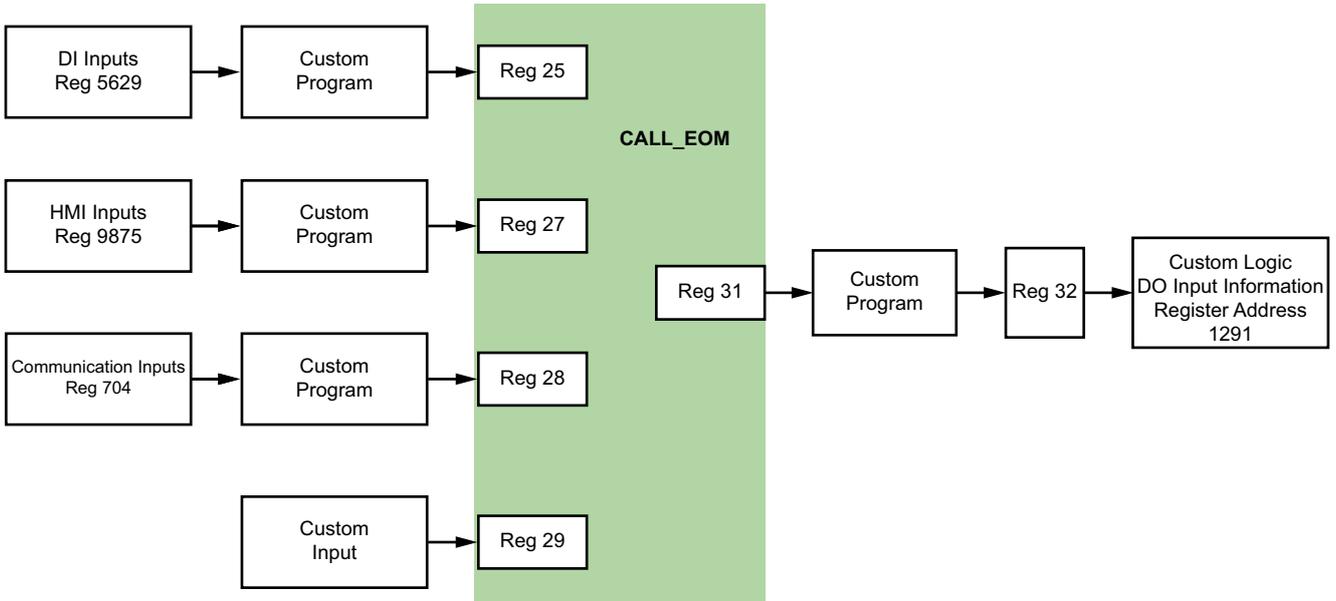
# CALL\_EOM 명령 설명

## 개요

CALL\_EOM 기능을 사용하면 작동 모드를 실행할 수 있습니다.

이를 위해 이 기능은 0에서 61까지의 주소에 있는 임시 레지스터를 사용합니다.

CALL\_EOM 기능을 중심으로 사용자 정의 프로그램을 구축하려면 애플리케이션과 LTMT main unit의 다양한 레지스터를 사용하는 방법을 이해해야 합니다.



- 25에서 29까지의 주소에 있는 임시 레지스터는 CALL\_EOM 기능의 입력 레지스터입니다. 사용자 정의할 때 레지스터에는 개별 비트를 할당해야 합니다.
- 주소 31의 임시 레지스터는 CALL\_EOM 기능의 출력 레지스터입니다. 이 값은 작동 모드를 실행한 후에 제공됩니다.
- 주소 32의 임시 레지스터는 주소 1291(0x050B)의 사용자 정의 논리 DO 입력 정보 레지스터를 일회성으로 설정하는 데 사용되는 임시 레지스터입니다. CALL\_EOM 출력의 사용자 정의는 주소 32의 임시 레지스터를 사용하여 수행해야 합니다.

## 디지털 입력 레지스터

레지스터 주소	레지스터 설명	비트	비트 설명
5629(0x15FD)	디지털 입력 레지스터	0	로컬-시작> DI
		1	로컬-정지 DI
		2	로컬-시작>> DI
		3	모드 선택 1
		4	로컬-시작< DI
		5	로컬-시작<< DI
		6	리모트-시작> DI
		7	리모트-정지 DI
		8	리모트-시작>> DI
		9	모드 선택 2
		10	리모트-시작< DI
		11	리모트-시작<< DI
		12	실행 DI
		13	속도 변경 DI
		14	예약됨
15	예약됨		

## HMI 입력 레지스터

레지스터 주소	레지스터 설명	비트	비트 설명
9875(0x2693)	HMI 명령 레지스터 1	0	모터 정방향 실행/고속 정방향 실행 명령
		1	모터 역방향 실행/고속 역방향 실행 명령
		2	로컬/리모트 모드 선택 1
		3	트립 초기화 명령
		4	로컬/리모트 모드 선택 2
		5	자체 테스트(트립 없음) 명령
		6	모터 저속 정방향 명령
		7	모터 저속 역방향 명령
		8	금지 초기화 명령
		9	시작 횟수 초기화 명령
		10	정지 횟수 초기화 명령
		11	에너지 해제 명령
		12	예약됨
		13	논리 테스트 명령
		14	실행 시간 초기화 명령
15	자체 테스트(트립 포함) 명령		
9876(0x2694)	HMI 명령 레지스터 2	0	예약됨
		1	예약됨
		2	열 용량 수준 해제 명령

		3	예약됨
		4	예약됨
		5	트립 카운터 해제 명령
		6	공장 초기화 명령
		7	소프트 스타터 초기화 명령
		8~12	예약됨
		13	레퍼런스 시작 곡선 저장 명령
		14	트립 레코드 해제 명령
		15	이벤트 레코드 해제 명령

## 통신 입력 레지스터

레지스터 주소	레지스터 설명	비트	비트 설명
704(0x02C0)	통신 명령 레지스터 1	0	모터 정방향 실행/고속 정방향 실행 명령
		1	모터 역방향 실행/고속 역방향 실행 명령
		2	로컬/리모트 모드 선택 1
		3	트립 초기화 명령
		4	로컬/리모트 모드 선택 2
		5	자체 테스트(트립 없음) 명령
		6	모터 저속 정방향 명령
		7	모터 저속 역방향 명령
		8	금지 초기화 명령
		9	시작 횟수 초기화 명령
		10	정지 횟수 초기화 명령
		11	에너지 해제 명령
		12	모터 정지 명령
		13	논리 테스트 명령
		14	실행 시간 초기화 명령
15	자체 테스트(트립 포함) 명령		
705(0x02C1)	통신 명령 레지스터 2	0	예약됨
		1	예약됨
		2	열 용량 수준 해제 명령
		3	예약됨
		4	예약됨
		5	트립 카운터 해제 명령
		6	공장 초기화 명령
		7	소프트 스타터 초기화 명령
		8~12	예약됨
		13	레퍼런스 시작 곡선 저장 명령
		14	트립 레코드 해제 명령
		15	이벤트 레코드 해제 명령

## CALL\_EOM 12 설명

CALL\_EOM 인수가 12이면 이 기능이 과부하 작동 모드를 실행합니다.

사용자 정의 프로그램에서 사용할 논리 ID는 다음과 같습니다.

- 과부하 작동 모드용 LOGID\_ID 268

임시 레지스터는 다음과 같이 사용됩니다.

레지스터	레지스터 설명	비트	비트 설명
입력 레지스터 25	DI 입력 사본	0-15	사용되지 않음
입력 레지스터 27	다음의 사본 HMI 입력값	0-15	사용되지 않음
입력 레지스터 28	통신 입력 사본	0-15	사용되지 않음
출력 레지스터 31	LTMT main unit의 출력에 할당할 CALL_EOM 명령문의 출력	0	사용자 정의 로직 "Run1 Cde" 정보
		1-15	사용되지 않음
출력 레지스터 32	LTMT main unit의 출력에 할당할 CALL_EOM 명령문의 출력	0	사용자 정의 논리 DO1 입력 정보
		1-15	사용되지 않음

## CALL\_EOM 13 설명

CALL\_EOM 인수가 13이면 이 기능이 Direct Online 작동 모드를 실행합니다.

사용자 정의 프로그램에서 사용할 논리 ID는 다음과 같습니다.

- Direct Online 작동 모드용 LOGID\_ID 269

임시 레지스터는 다음과 같이 사용됩니다.

레지스터	레지스터 설명	비트	비트 설명
입력 레지스터 25	DI 입력 사본	0	로컬-시작> DI
		1	로컬-정지 DI
		2	사용되지 않음
		3	DI 모드 선택 1
		4-5	사용되지 않음
		6	리모트-시작> DI
		7	리모트-정지 DI
		8	사용되지 않음
		9	DI 모드 선택 2
		10-11	사용되지 않음
		12	실행 DI
		13-15	사용되지 않음
입력 레지스터 27	다음의 사본 HMI 입력값	0	HMI_START >
		1	사용되지 않음
		2	HMI_STOP
		3-4	사용되지 않음
		5	HMI 모드 선택 1
		6	HMI 모드 선택 2
		7-15	사용되지 않음
입력 레지스터 28	통신 입력 사본	0	통신 시작 >
		1-3	사용되지 않음
		4	리모트 모드 선택 1
		5	리모트 모드 선택 2
		6	통신 정지
		7-15	사용되지 않음
출력 레지스터 31	LTMT main unit의 출력에 할당할 CALL_EOM 명령문의 출력	0	사용자 정의 로직 "Run1 Cde" 정보
		1-15	사용되지 않음
출력 레지스터 32	LTMT main unit의 출력에 할당할 CALL_EOM 명령문의 출력	0	사용자 정의 논리 DO1 입력 정보
		1~15	사용되지 않음

## CALL\_EOM 14 설명

CALL\_EOM 인수가 14이면 이 기능이 Reverse Direct Online 작동 모드를 실행합니다.

사용자 정의 프로그램에서 사용할 논리 ID는 다음과 같습니다.

- Reverse Direct Online 작동 모드용 LOGID\_ID 270

임시 레지스터는 다음과 같이 사용됩니다.

레지스터	레지스터 설명	비트	비트 설명
입력 레지스터 25	DI 입력 사본	0	로컬-시작> DI
		1	로컬-정지 DI
		2	사용되지 않음
		3	DI 모드 선택 1
		4	로컬-시작< DI
		5	사용되지 않음
		6	리모트-시작> DI
		7	리모트-정지 DI
		8	사용되지 않음
		9	DI 모드 선택 2
		10	리모트-시작< DI
		11	사용되지 않음
		12	실행 DI
		13-15	사용되지 않음
입력 레지스터 27	다음의 사본 HMI 입력값	0	HMI_START >
		1	사용되지 않음
		2	HMI_STOP
		3	HMI_START <
		4	사용되지 않음
		5	HMI 모드 선택 1
		6	HMI 모드 선택 2
		7-15	사용되지 않음
입력 레지스터 28	통신 입력 사본	0	통신 시작 >
		1	사용되지 않음
		2	통신 시작 <
		3	사용되지 않음
		4	리모트 모드 선택 1
		5	리모트 모드 선택 2
		6	통신 정지
		7-15	사용되지 않음
출력 레지스터 31	LTMT main unit의 출력에 할당할 CALL_EOM 명령문의 출력	0	사용자 정의 로직 "Run1 Cde" 정보
		1	사용자 정의 로직 "Run2 Cde" 정보
		2-15	사용되지 않음
출력 레지스터 32	LTMT main unit의 출력에 할당할 CALL_EOM 명령문의 출력	0	사용자 정의 논리 DO1 입력 정보
		1	사용자 정의 논리 DO2 입력 정보
		2-15	사용되지 않음



## CALL\_EOM 15 설명

CALL\_EOM 인수가 15이면 이 기능이 Star Delta 작동 모드를 실행합니다.

사용자 정의 프로그램에서 사용할 논리 ID는 다음과 같습니다.

- Star Delta 작동 모드용 LOGID\_ID 271

임시 레지스터는 다음과 같이 사용됩니다.

레지스터	레지스터 설명	비트	비트 설명
입력 레지스터 25	DI 입력 사본	0	로컬-시작> DI
		1	로컬-정지 DI
		2	사용되지 않음
		3	모드 선택 1
		4-5	사용되지 않음
		6	리모트-시작> DI
		7	리모트-정지 DI
		8	사용되지 않음
		9	모드 선택 2
		10-11	사용되지 않음
		12	실행 DI
		13-15	사용되지 않음
		입력 레지스터 27	다음의 사본 HMI 입력값
1	사용되지 않음		
2	HMI_STOP		
3-4	사용되지 않음		
5	모드 선택 1		
6	모드 선택 2		
7-15	사용되지 않음		
입력 레지스터 28	통신 입력 사본		
		1-3	사용되지 않음
		4	리모트 모드 선택 1
		5	리모트 모드 선택 2
		6	통신 정지
		7-15	사용되지 않음
		출력 레지스터 31	LTMT main unit의 출력에 할당할 CALL_EOM 명령문의 출력
1	사용자 정의 논리 실행 2 명령 정보		
2	사용자 정의 논리 실행 3 명령 정보		
3-15	사용되지 않음		
출력 레지스터 32	LTMT main unit의 출력에 할당할 CALL_EOM 명령문의 출력		
		1	사용자 정의 논리 DO2 입력 정보
		2	사용자 정의 논리 DO3 입력 정보
		3-15	사용되지 않음

## 프로그램 예

```

LOGIC_ID 13
//
// DI Input
LOAD_BIT      5629 0
SET_TMP_BIT   25 0           //Local-START> DI
LOAD_BIT      5629 1
SET_TMP_BIT   25 1           //Local-STOP DI
LOAD_BIT      5629 2
SET_TMP_BIT   25 2           //Local-START>> DI
LOAD_BIT      5629 3
SET_TMP_BIT   25 3           //DI Mode Selection 1
LOAD_BIT      5629 4
SET_TMP_BIT   25 4           //Local-START< DI
LOAD_BIT      5629 5
SET_TMP_BIT   25 5           //Local-START<< DI
LOAD_BIT      5629 6
SET_TMP_BIT   25 6           //Remote-START> DI
LOAD_BIT      5629 7
SET_TMP_BIT   25 7           //Remote-STOP DI
LOAD_BIT      5629 8
SET_TMP_BIT   25 8           //Remote-START>> DI
LOAD_BIT      5629 9
SET_TMP_BIT   25 9           //DI Mode Selection 2
LOAD_BIT      5629 10
SET_TMP_BIT   25 10          //Remote-START< DI
LOAD_BIT      5629 11
SET_TMP_BIT   25 11          //Remote-START<< DI
LOAD_BIT      5629 12
SET_TMP_BIT   25 12          //Run DI
LOAD_BIT      5629 13
SET_TMP_BIT   25 13          //Speed Change
LOAD_BIT      5629 14
SET_TMP_BIT   25 14          //Not Used
LOAD_BIT      5629 15
SET_TMP_BIT   25 15          //Not Used
//
// HMI Input
LOAD_BIT      9875 0
SET_TMP_BIT   27 0           //HMI_START >
LOAD_BIT      9875 6
SET_TMP_BIT   27 1           //HMI_START >>
LOAD_BIT      9875 12

```

```

SET_TMP_BIT 27 2 //HMI_STOP,
LOAD_BIT 9875 1
SET_TMP_BIT 27 3 //HMI_START <
LOAD_BIT 9875 7
SET_TMP_BIT 27 4 //HMI_START <<
LOAD_BIT 9875 2
SET_TMP_BIT 27 5 //HMI Mode Selection 1
LOAD_BIT 9875 4
SET_TMP_BIT 27 6 //HMI Mode Selection 2
//
//
// Remote Input
LOAD_BIT 704 0
SET_TMP_BIT 28 0 //COMM Start >
LOAD_BIT 704 6
SET_TMP_BIT 28 1 //COMM Start >>
LOAD_BIT 704 1
SET_TMP_BIT 28 2 //COMM Start <
LOAD_BIT 704 7
SET_TMP_BIT 28 3 //COMM Start <<
LOAD_BIT 704 2
SET_TMP_BIT 28 4 //Remote Mode Selection 1
LOAD_BIT 704 4
SET_TMP_BIT 28 5 //Remote Mode Selection 2
//
//End customer Zone
//Call Command
//Output
//=====
// Customer Zone: Custom application
// Add specific code for Custom Logic function here"
//
CALL_EOM 13
//=====
// Customer Zone: Outputs management"
//
// Output
LOAD_TMP_BIT 31 0 //CL "Run1 Cde" information
SET_TMP_BIT 32 0 //CL D01 Input information
LOAD_TMP_BIT 31 1 //CL "Run2 Cde" information
SET_TMP_BIT 32 1 //CL D02 Input information
LOAD_TMP_BIT 31 2 //CL "Run3 Cde" information

SET_TMP_BIT 32 2 //CL D03 Input information
LOAD_TMP_BIT 31 3 //CL "Run4 Cde" information
SET_TMP_BIT 32 3 //CL D04 Input information
LOAD_TMP_BIT 31 4 //CL "Run5 Cde" information
SET_TMP_BIT 32 4 //CL D05 Input information
//
//
//
// End Customer Zone
//=====
// Schneider Zone (Do not modify)
//
// CL output in 1291
LOAD_K_BIT 1
SET_NOT_TMP_BIT 0 3
LOAD_TMP_REG 32 //Get image of 1291
ON_SET_REG 1291 0 //Put it into 1291
//

```

# 사용자 정의 논리 언어

## 이 파트의 내용

사용자 정의 논리 프로그램 작성하기.....	118
논리 명령.....	128
사용자 정의 논리 프로그램 예.....	156

# 사용자 정의 논리 프로그램 작성하기

## 이 장의 내용

Custom Logic Editor 소개 .....	119
Custom Logic Editor 사용자 인터페이스 .....	120
논리 명령.....	123

# Custom Logic Editor 소개

## 개요

Custom Logic Editor는 SoMove를 포함하는 TeSys Tera DTM Library software의 기능입니다. Custom Logic Editor를 사용하면 기존 프로그램 파일을 보거나, 그래픽 기반 프로그래밍 언어가 아닌 명령문 기반 텍스트 언어를 사용하여 프로그램 파일을 작성할 수 있습니다.

## 사용자 정의 논리 프로그램 편집하기

사용자 정의 프로그램을 작성하는 가장 쉬운 방법은 작동 모드(또는 모터 스타터 유형) 중 하나의 사전 정의 프로그램으로 시작하는 것입니다. Custom Logic Editor 설치에는 각 모터 스타터 유형에 대해 하나씩, 11개의 사전 정의 프로그램 파일이 포함되어 있습니다.

각 프로그램 파일은 설명적인 이름(예: DIRECT\_ONLINE)과 \*.rtf.

## Custom Logic Editor 사용자 인터페이스

Custom Logic Editor를 열려면 **Device(장치) > Custom Logic(사용자 정의 논리) > New Custom Logic(새 사용자 정의 논리)**을 선택합니다.

Custom Logic Editor는 TeSys Tera DTM Library가 연결 모드인지와 관계없이 사용할 수 있습니다. 그러나 TeSys Tera DTM Library와 장치 간의 프로그램 전송은 연결 모드에서만 작동합니다.

## Custom Logic Editor 사용자 인터페이스

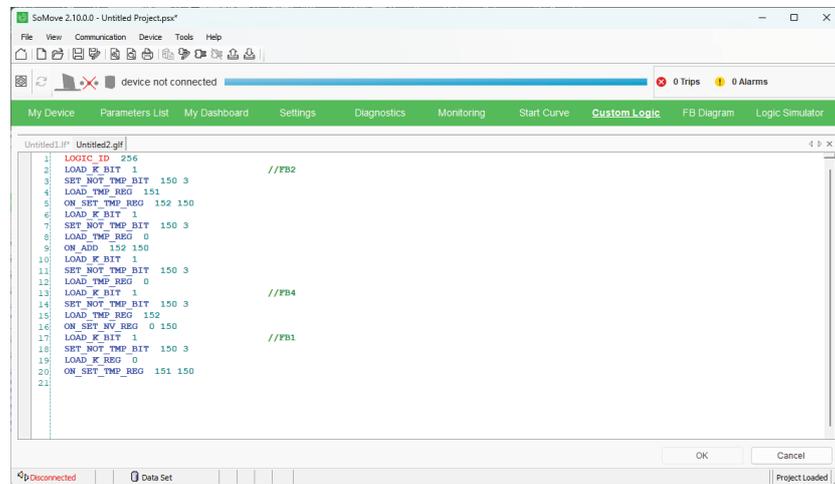
### 소개

이 언어로 작성된 프로그램은 LMTT main unit에 의해 순차적으로 실행되는 일련의 명령문으로 구성됩니다. 각 목록 명령문은 단일 프로그램 라인으로 표시되며 다음 네 가지 구성 요소로 구성됩니다.

- 라인 번호
- 논리 명령(니모닉)
- 인수
- 주석

### 사용자 정의 논리 프로그램의 예

다음 창은 Custom Logic Editor로 작성한 프로그램의 예입니다.



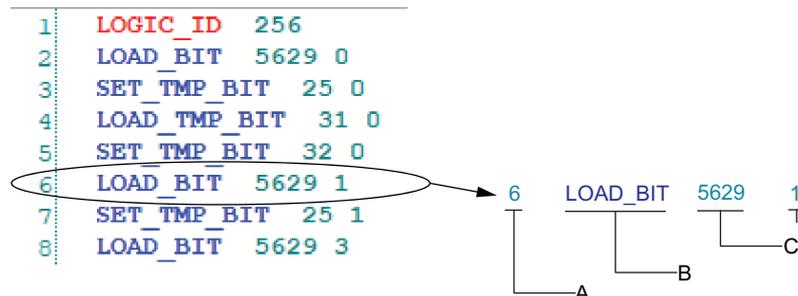
### 여러 프로그램 편집하기

여러 사용자 정의 프로그램을 동시에 작성하거나 수정할 수 있습니다. 사용자 정의 프로그램 간에서 전환하려면 파일 이름을 클릭합니다.

예를 들어 위의 텍스트 보기에서 편집할 프로그램에 따라 **DIRECT\_ONLINE.rtf** 또는 **Untitled.lf**를 선택합니다.

### 명령문 요소

다음 그림은 사용자 정의 논리 프로그램의 샘플입니다.



A	라인 번호
B	논리 명령
C	인수

## 라인 번호

라인 번호는 추가 정보입니다.

- 편집기에서만 정의됩니다.
- 사용자 정의 논리 기능 자체에는 중요하지 않습니다.

## 논리 명령

논리 명령은 하나 이상의 인수를 사용하여 수행될 연산을 식별하는 명령문입니다. 이 예에서 `LOAD_BIT` 명령은 인수의 값을 1비트 누산기라고 하는 내부 레지스터에 로드합니다.

두 가지 유형의 명령이 있습니다.

- **설정 명령**  
작업을 수행하는 데 필요한 조건을 설정하거나 테스트합니다(예: `LOAD` 및 `AND` 명령).
- **작업 명령**  
LTMT main unit에게 설정 명령문의 정보를 기반으로 작업을 수행하도록 지시합니다(예: `COMP` 같은 할당 명령).

**주의:** 대문자나 소문자로 논리 명령을 입력하면 자동으로 인식되어 파란색으로 표시됩니다.

## 인수

인수는 값(레지스터 주소, 비트 번호 또는 상수)을 나타내는 숫자이며, LTMT main unit은 명령문에서 이러한 인수를 조작할 수 있습니다. 논리 명령은 논리 명령 유형에 따라 0~3개의 인수를 가질 수 있습니다.

예를 들어, 샘플 프로그램에서 `LOAD_BIT 5629 1` 명령문은 논리 명령 `LOAD_BIT`와 2개의 인수 `5629` 및 `1`을 포함합니다.

이 명령문은 LTMT main unit에게 주소 `5629(0x15FD)`인 레지스터의 비트 1 값을 누산기에 로드하도록 지시합니다.

사용자 정의 프로그램은 명령과 인수를 포함하는 명령문을 사용하여 다음을 수행할 수 있습니다.

- 디지털 입력 상태를 읽습니다.
- 디지털 출력의 상태를 읽거나 씁니다.
- 타이머 및 카운터와 같은 기본 논리 기능을 활성화합니다.
- 산술, 논리, 비교 및 수치 연산을 수행합니다.
- LTMT main unit 레지스터 또는 해당 레지스터의 개별 비트를 읽거나 씁니다.

**주의:** 인수를 입력하면 자동으로 인식되고 인수에 할당된 색상으로 표시됩니다.

## 설명

Custom Logic Editor에서 프로그램의 다음 위치에 주석을 추가할 수 있습니다.

- 각 라인에서 인수 뒤의 끝 위치
- 전체 라인

**주의:**

- //를 입력하면 Custom Logic Editor가 자동으로 뒤에 있는 텍스트를 주석으로 인식하고 녹색으로 표시합니다.
- LTMT main unit에서 주석을 검색할 수 없습니다.

## 구문

Custom Logic Editor에서 다음과 같이 명령문을 작성할 수 있습니다.

- 인수 사이에 공백, 쉼표 또는 점 사용
- 대문자 또는 소문자

## 구문 검사

입력하는 동안 텍스트 편집기는 명령문 구문을 확인합니다.

- 올바른 명령문은 굵은 파란색 문자로 표시됩니다.
- 잘못된 명령문은 검정색으로 계속 표시되며 컴파일 전에 수정해야 합니다.

## 키보드 명령

키보드 명령과 단축키는 Windows 운영 체제와 동일합니다. 예를 들어, 문자나 라인을 삭제하려면 **DEL** 또는 **DELETE** 키를 누르고, 다음 라인으로 이동하려면 **ENTER** 키를 누릅니다.

## 저장 중

편집하거나 작성한 프로그램을 저장하려면 **Device(장치) > Custom Logic(사용자 정의 논리)**을 선택한 다음 **Save Custom Logic(사용자 정의 논리 저장)** 또는 **SaveAs Custom Logic(다른 이름으로 사용자 정의 논리 저장)**을 선택합니다.

**주의:** 이 파일은 \*.If 확장명으로 저장됩니다.

# 논리 명령

## 개요

LTMT main unit 프로젝트 파일은 일련의 논리 명령으로 구성됩니다. 각 논리 명령은 명령 자체와 최대 3개의 인수로 구성됩니다.

각 논리 명령은 1비트 부울 누산기(값 0 또는 1) 또는 16비트 부호 없는 누산기(값 범위 0~65,535)에 링크로 연결된 연산을 수행합니다.

Custom Logic Editor는 다음과 같은 종류의 논리 명령을 제공합니다.

- Boolean
- Register
- Timers
- Latch
- Counters
- Math

## 상승 에지 감지 메커니즘

일부 논리 명령은 1비트 누산기의 상승 에지에서 작동합니다.

비트의 상승 에지는 현재 상태가 1이고 이전 상태가 0일 때 감지됩니다. 비트의 이전 상태는 전용 이력 비트에 저장됩니다.

**주의:** 이 이력 비트가 수정되면 상승 에지 감지가 방해받을 수 있습니다.

## Boolean Logic 명령

부울 명령은 단순한 부울(On 또는 Off) 값을 평가하고 제어합니다. 부울 명령에는 다음이 포함됩니다.

명령	인수 1	인수 2	인수 3	설명
LOAD_K_BIT	상수 값(0 또는 1)	-	-	1비트 누산기에 상수 값을 로드합니다.
LOAD_BIT	레지스터 주소	레지스터 비트 번호(0~15)	-	인수 1에서 식별된 주소에서 레지스터 비트를 로드하고 인수 2에서 식별된 비트를 1비트 누산기로 로드합니다.
LOAD_TMP_BIT	임시 레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기에 임시 레지스터 비트를 로드합니다.
LOAD_NV_BIT	비휘발성 레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기에 비휘발성 레지스터 비트를 로드합니다.
LOAD_NOT_BIT	레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기에 레지스터 비트의 반전된 부울 값을 로드합니다.
LOAD_NOT_TMP_BIT	임시 레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기에 임시 레지스터 비트의 반전된 부울 값을 로드합니다.
LOAD_NOT_NV_BIT	비휘발성 레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기에 비휘발성 레지스터 비트의 반전된 부울 값을 로드합니다.
AND_BIT	레지스터 주소	레지스터 비트 번호(0~15)	-	레지스터 비트 값과 1비트 누산기 내용 간의 논리 AND 링크 결과를 로드합니다. 결과는 1비트 누산기에 저장됩니다.
AND_TMP_BIT	임시 레지스터 주소	레지스터 비트 번호(0~15)	-	임시 레지스터 비트 값과 1비트 누산기 내용 간의 논리 AND 링크 결과를 로드합니다. 결과는 1비트 누산기에 저장됩니다.

명령	인수 1	인수 2	인수 3	설명
AND_NV_BIT	비휘발성 레지스터 주소	레지스터 비트 번호(0~15)	-	비휘발성 레지스터 비트 값과 1비트 누산기 내용 간의 논리 AND 링크 결과를 로드합니다. 결과는 1비트 누산기에 저장됩니다.
AND_NOT_BIT	레지스터 주소	레지스터 비트 번호(0~15)	-	반전된 레지스터 비트와 1비트 누산기의 논리 AND 결과를 로드합니다. 결과는 1비트 누산기에 저장됩니다.
AND_NOT_TMP_BIT	임시 레지스터 주소	레지스터 비트 번호(0~15)	-	반전된 임시 레지스터 비트와 1비트 누산기의 논리 AND 결과를 로드합니다. 결과는 1비트 누산기에 저장됩니다.
AND_NOT_NV_BIT	비휘발성 레지스터 주소	레지스터 비트 번호(0~15)	-	반전된 비휘발성 레지스터 비트와 1비트 누산기의 논리 AND 결과를 로드합니다. 결과는 1비트 누산기에 저장됩니다.
OR_BIT	레지스터 주소	레지스터 비트 번호(0~15)	-	레지스터 비트 값과 1비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 1비트 누산기에 저장됩니다.
OR_TMP_BIT	임시 레지스터 주소	레지스터 비트 번호(0~15)	-	임시 레지스터 비트 값과 1비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 1비트 누산기에 저장됩니다.
OR_NV_BIT	비휘발성 레지스터 주소	레지스터 비트 번호(0~15)	-	비휘발성 레지스터 비트 값과 1비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 1비트 누산기에 저장됩니다.
OR_NOT_BIT	레지스터 주소	레지스터 비트 번호(0~15)	-	반전된 레지스터 비트와 1비트 누산기의 논리 OR 링크를 만듭니다. 결과는 1비트 누산기에 저장됩니다.
OR_NOT_TMP_BIT	임시 레지스터 주소	레지스터 비트 번호(0~15)	-	반전된 임시 레지스터 비트와 1비트 누산기의 논리 OR 링크를 만듭니다. 결과는 1비트 누산기에 저장됩니다.
OR_NOT_NV_BIT	비휘발성 레지스터 주소	레지스터 비트 번호(0~15)	-	반전된 비휘발성 레지스터 비트와 1비트 누산기의 논리 OR 링크를 만듭니다. 결과는 1비트 누산기에 저장됩니다.
SET_BIT	레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기의 값을 레지스터 비트에 설정합니다.
SET_TMP_BIT	임시 레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기의 값을 임시 레지스터 비트에 설정합니다.
SET_NV_BIT	비휘발성 레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기의 값을 비휘발성 레지스터 비트에 설정합니다.
SET_NOT_BIT	레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기의 반전된 값을 레지스터 비트에 설정합니다.
SET_NOT_TMP_BIT	임시 레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기의 반전된 값을 임시 레지스터 비트에 설정합니다.
SET_NOT_NV_BIT	비휘발성 레지스터 주소	레지스터 비트 번호(0~15)	-	1비트 누산기의 반전된 값을 비휘발성 레지스터 비트에 설정합니다.
- 논리 명령에 적용되지 않는 인수입니다.				

## Register Logic 명령

레지스터 명령은 16비트 값을 평가하고 제어합니다. 레지스터 명령에는 다음이 포함됩니다.

명령	인수 1	인수 2	인수 3	설명
LOAD_K_REG	상수 값 (0~65,535)	-	-	16비트 누산기에 상수 값을 로드합니다.
LOAD_REG	레지스터 주소	-	-	16비트 누산기에 레지스터 사본을 로드합니다.
LOAD_TMP_REG	임시 레지스터 주소	-	-	16비트 누산기에 임시 레지스터 사본을 로드합니다.

명령	인수 1	인수 2	인수 3	설명
LOAD_NV_REG	비휘발성 레지스터 주소	-	-	16비트 누산기에 비휘발성 레지스터 사본을 로드합니다.
COMP_K_REG	상수 값 (0~65,535)	임시 레지스터 주소	-	인수 1의 내용과 16비트 누산기 내용을 비교하고 상태 인수 2비트를 다음과 같이 설정합니다.  16비트 누산기 < 인수 1의 내용:BIT 1 ON  16비트 누산기 = 인수 1의 내용:BIT 2 ON  16비트 누산기 > 인수 1의 내용:BIT 3 ON
COMP_REG	레지스터 주소	임시 레지스터 주소	-	인수 1에 의해 정의된 레지스터의 내용과 16비트 누산기 내용을 비교하고 상태 인수 2비트를 다음과 같이 설정합니다.  16비트 누산기 < 인수 1에 의해 정의된 레지스터의 내용:BIT 1 ON  16비트 누산기 = 인수 1에 의해 정의된 레지스터의 내용:BIT 2 ON  16비트 누산기 > 인수 1에 의해 정의된 레지스터의 내용:BIT 3 ON
COMP_TMP_REG	임시 레지스터 주소	임시 레지스터 주소	-	인수 1에 의해 정의된 레지스터의 내용과 16비트 누산기 내용을 비교하고 상태 인수 2비트를 다음과 같이 설정합니다.  16비트 누산기 < 인수 1에 의해 정의된 레지스터의 내용:BIT 1 ON  16비트 누산기 = 인수 1에 의해 정의된 레지스터의 내용:BIT 2 ON  16비트 누산기 > 인수 1에 의해 정의된 레지스터의 내용:BIT 3 ON
COMP_NV_REG	비휘발성 레지스터 주소	임시 레지스터 주소	-	인수 1에 의해 정의된 레지스터의 내용과 16비트 누산기 내용을 비교하고 상태 인수 2비트를 다음과 같이 설정합니다.  16비트 누산기 < 인수 1에 의해 정의된 레지스터의 내용:BIT 1 ON  16비트 누산기 = 인수 1에 의해 정의된 레지스터의 내용:BIT 2 ON  16비트 누산기 > 인수 1에 의해 정의된 레지스터의 내용:BIT 3 ON
AND_K	상수 값(0 또는 1)	-	-	상수 값과 16비트 누산기 내용 간의 논리 AND 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
AND_REG	레지스터 주소	-	-	레지스터 값과 16비트 누산기 내용 간의 논리 AND 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
AND_TMP_REG	임시 레지스터 주소	-	-	임시 레지스터 값과 16비트 누산기 내용 간의 논리 AND 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
AND_NV_REG	비휘발성 레지스터 주소	-	-	비휘발성 레지스터 값과 16비트 누산기 내용 간의 논리 AND 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
OR_K	상수 값(0 또는 1)	-	-	상수 값과 16비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
OR_REG	레지스터 주소	-	-	레지스터 값과 16비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
OR_TMP_REG	임시 레지스터 주소	-	-	임시 레지스터 값과 16비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
OR_NV_REG	비휘발성 레지스터 주소	-	-	비휘발성 레지스터 값과 16비트 누산기 내용 간의 배타적 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

명령	인수 1	인수 2	인수 3	설명
XOR_K	상수 값 (0~65,535)	-	-	상수 값과 16비트 누산기 내용 간의 배타적 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
XOR_REG	레지스터 주소	-	-	레지스터 값과 16비트 누산기 내용 간의 배타적 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
XOR_TMP_REG	임시 레지스터 주소	-	-	임시 레지스터 값과 16비트 누산기 내용 간의 배타적 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
XOR_NV_REG	비휘발성 레지스터 주소	-	-	비휘발성 레지스터 값과 16비트 누산기 내용 간의 배타적 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.
ON_SET_REG	레지스터 주소	임시 레지스터 주소	-	1비트 누산기의 상승 에지에서 16비트 누산기 내용을 인수 1에 의해 정의된 레지스터에 저장합니다.
ON_SET_TMP_REG	임시 레지스터 주소	임시 레지스터 주소	-	1비트 누산기의 상승 에지에서 16비트 누산기 내용을 인수 1에 의해 정의된 임시 레지스터에 저장합니다.
ON_SET_NV_REG	비휘발성 레지스터 주소	임시 레지스터 주소	-	1비트 누산기의 상승 에지에서 16비트 누산기 내용을 인수 1에 의해 정의된 비휘발성 레지스터에 저장합니다.
- 논리 명령에 적용되지 않는 인수입니다.				

## Timer Logic 명령

타이머는 0~65,535의 범위를 가지며 초 또는 10분의 1초 간격으로 시간을 측정합니다.

- 인수 1은 기간을 지정합니다.
- 인수 2는 계산된 종료 시간입니다.
- 인수 3은 타이머 상태 레지스터입니다.

타이머 명령에는 다음이 포함됩니다.

명령	인수 1	인수 2	인수 3	설명
TIMER_SEC	임시 레지스터(기간)	임시 레지스터(계산된 종료 시간)	임시 레지스터(상태)	상태 레지스터 비트로 설명된 대로 인수 1에 입력된 기간을 초 단위로 카운트합니다.
TIMER_TENTHS	임시 레지스터(기간)	임시 레지스터(계산된 종료 시간)	임시 레지스터(상태)	상태 레지스터 비트로 설명된 대로 인수 1에 입력된 기간을 10분의 1초 단위로 카운트합니다.

## Latch logic 명령

래치 명령에는 다음이 포함됩니다.

명령	인수 1	인수 2	인수 3	설명
LATCH	임시 레지스터(상태)	-	-	임시 레지스터에 신호 이력을 기록하고 유지합니다.
LATCH_NV	비휘발성 레지스터(상태)	-	-	신호 이력을 비휘발성 레지스터에 기록하고 유지합니다.
- 논리 명령에 적용되지 않는 인수입니다.				

## Counter Logic 명령

카운터는 0에서 65,535까지의 범위이며 최대값인 65,535까지 카운트하면 0으로 전환됩니다.

카운터 명령에는 다음이 포함됩니다.

명령	인수 1	인수 2	인수 3	설명
COUNTER	임시 레지스터(카운터 값)	상수 값 0~65,535(사전 설정 값)	임시 레지스터(상태)	비교 카운트를 수행하고, 카운터 값과 상태를 모두 임시 레지스터에 저장합니다.
COUNTER_NV	비휘발성 레지스터(카운터 값)	상수 값 0~65,535(사전 설정 값)	비휘발성 레지스터(상태)	비교 카운트를 수행하고, 카운터 값과 상태를 모두 비휘발성 레지스터에 저장합니다.

## Math Logic 명령

수학 명령은 16비트 누산기와 임시 레지스터를 사용하여 부호 없는 수학 함수를 수행합니다. 수학 명령은 1비트 누산기의 상승 에지에서 실행됩니다. 수학 명령에는 다음이 포함됩니다.

명령	인수 1	인수 2	인수 3	설명
ON_ADD	임시 레지스터(값)	임시 레지스터(상태)	-	인수 1 = 인수 1 + 16비트 누산기
ON_SUB	임시 레지스터(값)	임시 레지스터(상태)	-	인수 1 = 인수 1 - 16비트 누산기
ON_MUL	임시 레지스터(최상위 워드)	임시 레지스터(최하위 워드)	임시 레지스터(상태)	인수 1:인수 2 = 16비트 누산기 x 인수 2
ON_DIV	임시 레지스터(최상위 워드)	임시 레지스터(최하위 워드)	임시 레지스터(상태)	인자 1:인자 2 = 인자 1:인자 2 / 16비트 누산기.
- 논리 명령에 적용되지 않는 인수입니다.				

# 논리 명령

## 이 장의 내용

프로그램 논리 명령 .....	129
Boolean Logic 명령 .....	131
Register Logic 명령 .....	140
Timer Logic 명령 .....	148
Latch logic 명령 .....	150
Counter Logic 명령 .....	152
Math Logic 명령 .....	154

# 프로그램 논리 명령

## 개요

프로그램 논리 명령은 다음과 같은 작업에 사용됩니다.

- Custom Logic Editor에 대한 프로그램 파일 식별
- 사전 정의 작동 모드 실행

다음 명령을 사용할 수 있습니다.

- LOGIC\_ID
- CALL\_EOM
- NOP

## LOGIC\_ID

LOGIC\_ID 문은 프로그램 파일의 식별자 역할을 합니다.

LOGIC\_ID 값은 256~511 범위의 정수 값을 갖습니다.

인수	표현
1	LOGIC_ID ID#

입력 인수	유형	범위	설명
ID#	UINT	256~511	사용자 정의 프로그램의 논리 ID

출력 인수가 없습니다.

## CALL\_EOM

CALL\_EOM 문은 사용자 정의 프로그램에서 사전 정의 작동 모드를 실행합니다.

인수	표현
1	CALL_EOM OP_MODE#

입력 인수	유형	범위	표현
OP_MODE#	INT	12~22	내장형 작동 모드(EOM): <ul style="list-style-type: none"> <li>• 12: 과부하</li> <li>• 13: Direct Online</li> <li>• 14: Reverse Direct Online</li> <li>• 15: Star Delta</li> </ul>

출력 인수가 없습니다.

## NOP

NOP 명령은 작업을 수행하지 않습니다.

NOP 명령은 프로그램 파일에서 기존 명령을 바꾸거나 향후 명령을 위해 공간을 예약하기 위한 자리 표시자로 사용됩니다.

인수	표현
0	NOP

NOP 명령은 인수가 없습니다.

## Boolean Logic 명령

### 개요

Custom Logic Editor는 다음과 같은 부울 논리 명령을 사용합니다.

- LOAD\_K\_BIT
- LOAD\_BIT
- LOAD\_TMP\_BIT
- LOAD\_NV\_BIT
- LOAD\_NOT\_BIT
- LOAD\_NOT\_TMP\_BIT
- LOAD\_NOT\_NV\_BIT
- AND\_BIT
- AND\_TMP\_BIT
- AND\_NV\_BIT
- AND\_NOT\_BIT
- AND\_NOT\_TMP\_BIT
- AND\_NOT\_NV\_BIT
- OR\_BIT
- OR\_TMP\_BIT
- OR\_NV\_BIT
- OR\_NOT\_BIT
- OR\_NOT\_TMP\_BIT
- OR\_NOT\_NV\_BIT
- SET\_BIT
- SET\_TMP\_BIT
- SET\_NV\_BIT
- SET\_NOT\_BIT
- SET\_NOT\_TMP\_BIT
- SET\_NOT\_NV\_BIT

### LOAD\_K\_BIT

LOAD\_K\_BIT 명령은 1비트 누산기에 상수 부울 값(0 또는 1)을 로드합니다.

인수	표현
1	LOAD_K_BIT KValue

입력 인수	유형	범위	설명
KValue	BOOL	0/1	상수 값

출력 인수가 없습니다.

### LOAD\_BIT

LOAD\_BIT 명령은 1비트 누산기에 레지스터 비트의 부울 값(0 또는 1)을 로드합니다.

인수	표현
2	LOAD_BIT RegAddr BitNo

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## LOAD\_TMP\_BIT

LOAD\_TMP\_BIT 명령은 1비트 누산기에 임시 레지스터 비트의 부울 값(0 또는 1)을 로드합니다.

인수	표현
2	LOAD_TMP_BIT TmpReg BitNo

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## LOAD\_NV\_BIT

LOAD\_NV\_BIT 명령은 1비트 누산기에 비휘발성 레지스터 비트의 부울 값(0 또는 1)을 로드합니다.

인수	표현
2	LOAD_NV_BIT NVReg BitNo

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## LOAD\_NOT\_BIT

LOAD\_NOT\_BIT 명령:

- 지정된 레지스터 비트의 부울 값(0 또는 1)을 반전한 다음
- 반전된 값을 1비트 누산기에 로드합니다.

인수	표현
2	LOAD_NOT_BIT RegAddr BitNo

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## LOAD\_NOT\_TMP\_BIT

LOAD\_NOT\_TMP\_BIT 명령:

- 지정된 임시 레지스터 비트의 부울 값(0 또는 1)을 반전한 다음
- 반전된 값을 1비트 누산기에 로드합니다.

인수	표현
2	LOAD_NOT_TMP_BIT TmpReg BitNo

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## LOAD\_NOT\_NV\_BIT

LOAD\_NOT\_NV\_BIT 명령:

- 선택된 비휘발성 레지스터 비트의 부울 값(0 또는 1)을 반전한 다음
- 반전된 값을 1비트 누산기에 로드합니다.

인수	표현
2	LOAD_NOT_NV_BIT NVReg BitNo

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## AND\_BIT

AND\_BIT 명령은 논리 메모리에서 레지스터 비트 값과 누산기 내용 간의 논리 AND 링크를 만듭니다.

- 1비트 누산기가 1이고 링크로 연결된 레지스터 비트가 1이면 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 AND 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	AND_BIT RegAddr BitNo

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## AND\_TMP\_BIT

AND\_TMP\_BIT 명령은 논리 메모리에서 임시 레지스터 비트 값과 누산기 내용 간의 논리 AND 링크를 만듭니다.

- 1비트 누산기가 1이고 링크로 연결된 임시 레지스터 비트가 1이면 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 AND 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	AND_TMP_BIT TmpReg BitNo

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## AND\_NV\_BIT

AND\_NV\_BIT 명령은 논리 메모리에서 비휘발성 레지스터 비트 값과 누산기 내용 간의 논리 AND 링크를 만듭니다.

- 1비트 누산기가 1이고 링크로 연결된 비휘발성 레지스터 비트가 1이면 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 AND 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	AND_NV_BIT NVReg BitNo

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## AND\_NOT\_BIT

AND\_NOT\_BIT 명령은 지정된 레지스터 비트의 부울 값(0 또는 1)을 반전한 다음 논리 메모리에서 반전된 값과 누산기 내용 간의 논리 AND 링크를 만듭니다.

- 1비트 누산기가 1이고 링크로 연결된 레지스터 비트가 0이면 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 AND 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	AND_NOT_BIT RegAddr BitNo

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## AND\_NOT\_TMP\_BIT

AND\_NOT\_TMP\_BIT 명령은 지정된 임시 레지스터 비트의 부울 값(0 또는 1)을 반전한 다음 논리 메모리에서 반전된 값과 누산기 내용 간의 논리 AND 링크를 만듭니다.

- 1비트 누산기가 1이고 링크로 연결된 임시 레지스터 비트가 0이면 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 AND 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	AND_NOT_TMP_BIT TmpReg BitNo

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## AND\_NOT\_NV\_BIT

AND\_NOT\_NV\_BIT 명령은 선택된 비휘발성 레지스터 비트의 부울 값(0 또는 1)을 반전한 다음 논리 메모리에서 반전된 값과 누산기 내용 간의 논리 AND 링크를 만듭니다.

- 1비트 누산기가 1이고 링크로 연결된 비휘발성 레지스터 비트가 0이면 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 AND 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	AND_NOT_NV_BIT NVReg BitNo

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## OR\_BIT

OR\_BIT 명령은 논리 메모리에서 레지스터 비트 값과 누산기 내용 간의 논리 OR 링크를 만듭니다.

- 1비트 누산기 또는 레지스터 비트 중 하나라도 값이 1이면 OR 프로세스의 결과도 1입니다.
- 모든 비교 비트의 값이 0이면 OR 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	OR_BIT RegAddr BitNo

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## OR\_TMP\_BIT

OR\_TMP\_BIT 명령은 논리 메모리에서 임시 레지스터 비트 값과 누산기 내용 간의 논리 OR 링크를 만듭니다.

- 1비트 누산기 또는 임시 레지스터 비트 중 하나라도 값이 1이면 OR 프로세스의 결과도 1입니다.
- 모든 비교 비트의 값이 0이면 OR 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	OR_TMP_BIT TmpReg BitNo

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## OR\_NV\_BIT

OR\_NV\_BIT 명령은 논리 메모리에서 비휘발성 레지스터 비트 값과 누산기 내용 간의 논리 OR 링크를 만듭니다.

- 1비트 누산기 또는 비휘발성 레지스터 비트 중 하나라도 값이 1이면 OR 프로세스의 결과도 1입니다.
- 모든 비교 비트의 값이 0이면 OR 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	OR_NV_BIT NVReg BitNo

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## OR\_NOT\_BIT

OR\_NOT\_BIT 명령은 지정된 레지스터 비트의 부울 값(0 또는 1)을 반전한 다음 논리 메모리에서 반전된 값과 누산기 내용 간의 논리 OR 링크를 만듭니다.

- 1비트 누산기 또는 레지스터 비트 중 하나라도 값이 0이면 OR 프로세스의 결과도 1입니다.
- 모든 비교 비트의 값이 0이면 OR 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	OR_NOT_BIT RegAddr BitNo

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## OR\_NOT\_TMP\_BIT

OR\_NOT\_TMP\_BIT 명령은 지정된 임시 레지스터 비트의 부울 값(0 또는 1)을 반전한 다음 논리 메모리에서 반전된 값과 누산기 내용 간의 논리 OR 링크를 만듭니다.

- 1비트 누산기 또는 임시 레지스터 비트 중 하나라도 값이 0이면 OR 프로세스의 결과도 1입니다.
- 모든 비교 비트의 값이 0이면 OR 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	OR_NOT_TMP_BIT TmpReg BitNo

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## OR\_NOT\_NV\_BIT

OR\_NOT\_NV\_BIT 명령은 선택된 비휘발성 레지스터 비트의 부울 값(0 또는 1)을 반전한 다음 논리 메모리에서 반전된 값과 누산기 내용 간의 논리 OR 링크를 만듭니다.

- 1비트 누산기 또는 비휘발성 레지스터 비트 중 하나라도 값이 0이면 OR 프로세스의 결과도 1입니다.
- 모든 비교 비트의 값이 0이면 OR 프로세스의 결과는 0입니다.

결과는 1비트 누산기에 저장됩니다.

인수	표현
2	OR_NOT_NV_BIT NVReg BitNo

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호
BitNo	UINT	0~15	비트 번호

출력 인수가 없습니다.

## SET\_BIT

SET\_BIT 명령은 1비트 누산기의 값을 지정된 레지스터 비트에 설정합니다.

인수	표현
2	SET_BIT RegAddr BitNo

입력 인수가 없습니다.

출력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소
BitNo	UINT	0~15	비트 번호

## SET\_TMP\_BIT

SET\_TMP\_BIT 명령은 1비트 누산기의 값을 지정된 임시 레지스터 비트에 설정합니다.

인수	표현
2	SET_TMP_BIT TmpReg BitNo

입력 인수가 없습니다.

출력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호
BitNo	UINT	0~15	비트 번호

## SET\_NV\_BIT

SET\_NV\_BIT 명령은 1비트 누산기의 값을 지정된 비휘발성 레지스터 비트에 설정합니다.

인수	표현
2	SET_NV_BIT NVReg BitNo

입력 인수가 없습니다.

출력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호
BitNo	UINT	0~15	비트 번호

## SET\_NOT\_BIT

SET\_NOT\_BIT 명령은 1비트 누산기의 반전된 값을 지정된 레지스터 비트에 설정합니다.

인수	표현
2	SET_NOT_BIT RegAddr BitNo

입력 인수가 없습니다.

출력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소
BitNo	UINT	0~15	비트 번호

## SET\_NOT\_TMP\_BIT

SET\_NOT\_TMP\_BIT 명령은 1비트 누산기의 반전된 값을 지정된 임시 레지스터 비트에 설정합니다.

인수	표현
2	SET_NOT_TMP_BIT TmpReg BitNo

입력 인수가 없습니다.

출력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호
BitNo	UINT	0~15	비트 번호

## SET\_NOT\_NV\_BIT

SET\_NOT\_NV\_BIT 명령은 1비트 누산기의 반전된 값을 지정된 비휘발성 레지스터 비트에 설정합니다.

인수	표현
2	SET_NOT_NV_BIT NVReg BitNo

입력 인수가 없습니다.

출력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호
BitNo	UINT	0~15	비트 번호

## Register Logic 명령

### 개요

레지스터 명령은 16비트 값을 평가하고 제어합니다.

Custom Logic Editor는 다음과 같은 레지스터 명령을 사용합니다.

- LOAD\_K\_REG
- LOAD\_REG
- LOAD\_TMP\_REG
- LOAD\_NV\_REG
- COMP\_K\_REG
- COMP\_REG
- COMP\_TMP\_REG
- COMP\_NV\_REG
- AND\_K
- AND\_REG
- AND\_TMP\_REG
- AND\_NV\_REG
- OR\_K
- OR\_REG
- OR\_TMP\_REG
- OR\_NV\_REG
- XOR\_K
- XOR\_REG
- XOR\_TMP\_REG
- XOR\_NV\_REG
- ON\_SET\_REG
- ON\_SET\_TMP\_REG
- ON\_SET\_NV\_REG

### LOAD\_K\_REG

LOAD\_K\_REG 명령은 논리 메모리의 16비트 누산기에 16비트 상수 값을 로드합니다.

인수	표현
1	LOAD_K_REG KValue

입력 인수	유형	범위	설명
KValue	UINT	0~65,535	상수 값

출력 인수가 없습니다.

### LOAD\_REG

LOAD\_REG 명령은 논리 메모리의 16비트 누산기에 레지스터 사본을 로드합니다.

인수	표현
1	LOAD_REG RegAddr

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소

출력 인수가 없습니다.

## LOAD\_TMP\_REG

LOAD\_TMP\_REG 명령은 논리 메모리의 16비트 누산기에 임시 레지스터 사본을 로드합니다.

인수	표현
1	LOAD_TMP_REG TmpReg

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호

출력 인수가 없습니다.

## LOAD\_NV\_REG

LOAD\_NV\_REG 명령은 논리 메모리의 16비트 누산기에 비휘발성 레지스터 사본을 로드합니다.

인수	표현
1	LOAD_NV_REG NVReg

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호

출력 인수가 없습니다.

## COMP\_K\_REG

COMP\_K\_REG 명령은 16비트 누산기의 내용을 인수 1 상수 값과 비교하고 비교 결과를 인수 2 임시 레지스터의 한 비트에 설정합니다.

인수	표현
2	COMP_K_REG KValue TmpReg

입력 인수	유형	범위/비트	설명
KValue	UINT	0~65,535	상수 값

출력 인수	유형	범위/비트	설명
TmpReg	UINT	Bit1	16비트 누산기 < KValue
		Bit2	16비트 누산기 = KValue
		Bit3	16비트 누산기 > KValue

## COMP\_REG

COMP\_REG 명령은 16비트 누산기의 내용을 인수 1에 의해 정의된 레지스터의 내용과 비교하고 비교 결과를 인수 2 임시 레지스터의 한 비트에 설정합니다.

인수	표현		
2	COMP_REG RegAddr TmpReg		

입력 인수	유형	범위/비트	설명
RegAddr	UINT	0~9249	레지스터 주소

출력 인수	유형	범위/비트	설명
TmpReg	UINT	Bit1	16비트 누산기 < RegAddr
		Bit2	16비트 누산기 = RegAddr
		Bit3	16비트 누산기 > RegAddr

## COMP\_TMP\_REG

COMP\_TMP\_REG 명령은 16비트 누산기의 내용을 인수 1에 의해 정의된 임시 레지스터의 내용과 비교하고 비교 결과를 인수 2 임시 레지스터의 한 비트에 설정합니다.

인수	표현		
2	COMP_TMP_REG TmpReg1 TmpReg2		

입력 인수	유형	범위/비트	설명
TmpReg1	UINT	0~299	임시 레지스터 번호

출력 인수	유형	범위/비트	설명
TmpReg2	UINT	Bit1	16비트 누산기 < TmpReg1
		Bit2	16비트 누산기 = TmpReg1
		Bit3	16비트 누산기 > TmpReg1

## COMP\_NV\_REG

COMP\_NV\_REG 명령은 16비트 누산기의 내용을 인수 1에 의해 정의된 비휘발성 레지스터의 내용과 비교하고 비교 결과를 인수 2 임시 레지스터의 한 비트에 설정합니다.

인수	표현		
2	COMP_NV_REG NVReg TmpReg		

입력 인수	유형	범위/비트	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호

출력 인수	유형	범위/비트	설명
TmpReg	UINT	Bit1	16비트 누산기 < NVReg
		Bit2	16비트 누산기 = NVReg
		Bit3	16비트 누산기 > NVReg

## AND\_K

AND\_K 명령은 논리 메모리에서 16비트 상수 값과 16비트 누산기 내용 간의 논리 AND 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

AND 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 16비트 상수 값의 해당 비트와 비교합니다.

- 두 비트가 모두 1이면 해당 비트 번호에 대한 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 해당 비트 번호에 대한 AND 프로세스의 결과는 0입니다.

인수	표현
1	AND_K KValue

입력 인수	유형	범위	설명
KValue	UINT	0~65,535	상수 값

출력 인수가 없습니다.

## AND\_REG

AND\_REG 명령은 논리 메모리에서 레지스터 값과 16비트 누산기 내용 간의 논리 AND 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

AND 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 레지스터의 해당 비트와 비교합니다.

- 두 비트가 모두 1이면 해당 비트 번호에 대한 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 해당 비트 번호에 대한 AND 프로세스의 결과는 0입니다.

인수	표현
1	AND_REG RegAddr

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소

출력 인수가 없습니다.

## AND\_TMP\_REG

AND\_TMP\_REG 명령은 논리 메모리에서 임시 레지스터 값과 16비트 누산기 내용 간의 논리 AND 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

AND 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 임시 레지스터의 해당 비트와 비교합니다.

- 두 비트가 모두 1이면 해당 비트 번호에 대한 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 해당 비트 번호에 대한 AND 프로세스의 결과는 0입니다.

인수	표현
1	AND_TMP_REG TmpReg

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호

출력 인수가 없습니다.

## AND\_NV\_REG

AND\_NV\_REG 명령은 논리 메모리에서 비휘발성 레지스터 값과 16비트 누산기 내용 간의 논리 AND 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

AND 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 비휘발성 레지스터의 해당 비트와 비교합니다.

- 두 비트가 모두 1이면 해당 비트 번호에 대한 AND 프로세스의 결과도 1입니다.
- 그 외 모든 경우에는 해당 비트 번호에 대한 AND 프로세스의 결과는 0입니다.

인수	표현
1	AND_NV_REG NVReg

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호

출력 인수가 없습니다.

## OR\_K

OR\_K 명령은 논리 메모리에서 16비트 상수 값과 16비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

OR 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 16비트 상수 값의 해당 비트와 비교합니다.

- 두 비트 중 하나라도 1이면 해당 비트 번호에 대한 OR 프로세스의 결과도 1입니다.
- 두 비트 모두 0이면 해당 비트 번호에 대한 OR 프로세스의 결과는 0입니다.

인수	표현
1	OR_K KValue

입력 인수	유형	범위	설명
KValue	UINT	0~65,535	상수 값

출력 인수가 없습니다.

## OR\_REG

OR\_REG 명령은 논리 메모리에서 레지스터 값과 16비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

OR 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 레지스터의 해당 비트와 비교합니다.

- 두 비트 중 하나라도 1이면 해당 비트 번호에 대한 OR 프로세스의 결과도 1입니다.
- 두 비트 모두 0이면 해당 비트 번호에 대한 OR 프로세스의 결과는 0입니다.

인수	표현
1	OR_REG RegAddr

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소

출력 인수가 없습니다.

## OR\_TMP\_REG

OR\_TMP\_REG 명령은 논리 메모리에서 임시 레지스터 값과 16비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

OR 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 임시 레지스터의 해당 비트와 비교합니다.

- 두 비트 중 하나라도 1이면 해당 비트 번호에 대한 OR 프로세스의 결과도 1입니다.
- 두 비트 모두 0이면 해당 비트 번호에 대한 OR 프로세스의 결과는 0입니다.

인수	표현
1	OR_TMP_REG TmpReg

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호

출력 인수가 없습니다.

## OR\_NV\_REG

OR\_NV\_REG 명령은 논리 메모리에서 비휘발성 레지스터 값과 16비트 누산기 내용 간의 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

OR 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 비휘발성 레지스터의 해당 비트와 비교합니다.

- 두 비트 중 하나라도 1이면 해당 비트 번호에 대한 OR 프로세스의 결과도 1입니다.
- 두 비트 모두 0이면 해당 비트 번호에 대한 OR 프로세스의 결과는 0입니다.

인수	표현
1	OR_NV_REG NVReg

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호

출력 인수가 없습니다.

## XOR\_K

XOR\_K 명령은 논리 메모리에서 16비트 상수 값과 16비트 누산기 내용 간의 배타적 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

XOR 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 16비트 상수 값의 해당 비트와 비교하여 다음과 같은 결과를 생성합니다.

- 한 비트는 1이고 다른 비트는 0이면 XOR 프로세스의 결과는 1입니다.
- 그 외 모든 경우에는 XOR 프로세스의 결과는 0입니다.

인수	표현
1	XOR_K KValue

입력 인수	유형	범위	설명
KValue	UINT	0~65,535	상수 값

출력 인수가 없습니다.

## XOR\_REG

XOR\_REG 명령은 논리 메모리에서 레지스터 값과 16비트 누산기 내용 간의 배타적 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

XOR 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 레지스터의 해당 비트와 비교하여 다음과 같은 결과를 생성합니다.

- 한 비트는 1이고 다른 비트는 0이면 XOR 프로세스의 결과는 1입니다.
- 그 외 모든 경우에는 XOR 프로세스의 결과는 0입니다.

인수	표현
1	XOR_REG RegAddr

입력 인수	유형	범위	설명
RegAddr	UINT	0~9249	레지스터 주소

출력 인수가 없습니다.

## XOR\_TMP\_REG

XOR\_TMP\_REG 명령은 논리 메모리에서 임시 레지스터 값과 16비트 누산기 내용 간의 배타적 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

XOR 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 임시 레지스터의 해당 비트와 비교하여 다음과 같은 결과를 생성합니다.

- 한 비트는 1이고 다른 비트는 0이면 XOR 프로세스의 결과는 1입니다.
- 그 외 모든 경우에는 XOR 프로세스의 결과는 0입니다.

인수	표현
1	XOR_TMP_REG TmpReg

입력 인수	유형	범위	설명
TmpReg	UINT	0~299	임시 레지스터 번호

출력 인수가 없습니다.

## XOR\_NV\_REG

XOR\_NV\_REG 명령은 논리 메모리에서 비휘발성 레지스터 값과 16비트 누산기 내용 간의 배타적 논리 OR 링크를 만듭니다. 결과는 16비트 누산기에 저장됩니다.

XOR 프로세스는 16비트 누산기의 각 비트를 링크로 연결된 비휘발성 레지스터의 해당 비트와 비교하여 다음과 같은 결과를 생성합니다.

- 한 비트는 1이고 다른 비트는 0이면 XOR 프로세스의 결과는 1입니다.
- 그 외 모든 경우에는 XOR 프로세스의 결과는 0입니다.

인수	표현
1	XOR_NV_REG NVReg

입력 인수	유형	범위	설명
NVReg	UINT	0~63	비휘발성 레지스터 번호

출력 인수가 없습니다.

## ON\_SET\_REG

ON\_SET\_REG 명령은 16비트 누산기의 값을 1비트 누산기의 상승 에지 시 지정된 레지스터에 복사합니다.

인수	표현
2	ON_SET_REG RegAddr TmpReg

입력 인수가 없습니다.

출력 인수	유형	범위/비트	설명
RegAddr	UINT	0~9249	레지스터 주소
TmpReg	UINT	Bit3	1비트 누산기 이력 비트

## ON\_SET\_TMP\_REG

ON\_SET\_TMP\_REG 명령은 16비트 누산기의 값을 1비트 누산기의 상승 에지 시 지정된 임시 레지스터에 복사합니다.

인수	표현
2	ON_SET_TMP_REG TmpReg1 TmpReg2

입력 인수가 없습니다.

출력 인수	유형	범위/비트	설명
TmpReg1	UINT	0~299	임시 레지스터 번호
TmpReg2	UINT	Bit3	1비트 누산기 이력 비트

## ON\_SET\_NV\_REG

ON\_SET\_NV\_REG 명령은 16비트 누산기의 값을 1비트 누산기의 상승 에지 시 지정된 비휘발성 레지스터에 복사합니다.

인수	표현
1	ON_SET_NV_REG NVReg1 NVReg2

입력 인수가 없습니다.

출력 인수	유형	범위/비트	설명
NVReg1	UINT	0~63	비휘발성 레지스터 번호
NVReg2	UINT	Bit3	1비트 누산기 이력 비트

# Timer Logic 명령

## 개요

Custom Logic Editor는 다음과 같은 타이머 명령을 사용합니다.

- TIMER\_SEC
- TIMER\_TENTHS

**주의:** 사용자 정의 논리 파일을 시뮬레이터에 업로드하고 타이머를 활성화하면 예상대로 작동합니다. 그러나 타이머를 활성화하고 실행한 후에는 다시 활성화할 수 없습니다. 이는 TIMER\_SEC 및 TIMER\_TENTHS의 제한 사항입니다. 타이머를 다시 활성화하려면 사용자 정의 논리 파일을 다시 업로드해야 합니다. 타이머로 작업하는 사용자 정의 논리 파일을 설계하고 테스트할 때는 이 제한 사항을 고려하는 것이 좋습니다.

## TIMER\_SEC

TIMER\_SEC 명령:

- 임시 레지스터에 의해 지정된 카운트 수까지 시간(초)을 카운트합니다.
- 두 번째 임시 레지스터에서 종료 시간을 계산합니다.
- 세 번째 임시 레지스터에 의해 활성화된 경우 해당 레지스터에 카운트 상태를 보고합니다.

인수	표현
3	TIMER_SEC TmpReg1 TmpReg2 TmpReg3

입력 인수	유형	범위/비트	설명
TmpReg1	UINT	0~65,535	타이머 사전 설정 값
TmpReg3	UINT	Bit0	<ul style="list-style-type: none"> <li>• 상승 에지에서 타이머 시작</li> <li>• 하강 에지에서 타이머 중지</li> </ul>

출력 인수	유형	범위/비트	설명
TmpReg2	UINT	0~65,535	계산된 종료 시간
TmpReg3	UINT	Bit1	타이머 완료: <ul style="list-style-type: none"> <li>• 타이머가 TmpReg2에 도달할 때 비트 설정</li> <li>• 다음 경우에 비트 초기화:               <ul style="list-style-type: none"> <li>◦ TmpReg3.Bit0 초기화 시</li> <li>◦ 전원 주기 발생 시</li> </ul> </li> </ul>
		Bit2	타이머 실행 진행 중 타이머가 TmpReg2에 도달할 때 비트 초기화
		Bit3	TmpReg3.Bit0 이력 비트
		비트4	예약됨

## TIMER\_TENTHS

TIMER\_TENTHS 명령:

- 시간을 0.1초 단위로 카운트하며, 임시 레지스터로 지정된 카운트 수까지 진행합니다.
- 종료 시간을 두 번째 임시 레지스터에 계산합니다.

- 제3의 임시 레지스터에 의해 활성화되며, 그 카운팅 상태를 해당 레지스터에 보고합니다.

**주의:** TIMER\_TENTHS 명령의 곱셈 계수는 10입니다. 예를 들어, 필요한 값이 6초이면 입력 값에 계수 10을 곱해야 합니다. 즉, 제공된 입력 값은 60초여야 합니다.

인수	표현
3	TIMER_TENTHS TmpReg1 TmpReg2 TmpReg3

입력 인수	유형	범위/비트	설명
TmpReg1	UINT	0~65,535	타이머 사전 설정 값
TmpReg3	UINT	Bit0	<ul style="list-style-type: none"> <li>상승 에지에서 타이머 시작</li> <li>하강 에지에서 타이머 중지</li> </ul>

출력 인수	유형	범위/비트	설명
TmpReg2	UINT	0~65,535	계산된 종료 시간
TmpReg3	UINT	Bit1	타이머 완료: <ul style="list-style-type: none"> <li>타이머가 도달하면 해당 비트가 설정됩니다. TmpReg2</li> <li>다음 경우에 비트 초기화:                             <ul style="list-style-type: none"> <li>TmpReg3.Bit0 초기화 시</li> <li>전원 주기 발생 시</li> </ul> </li> </ul>
		Bit2	타이머 실행 진행 중 타이머가 TmpReg2에 도달할 때 비트 초기화
		Bit3	TmpReg3.Bit0 이력 비트
		비트4	예약됨

## Latch logic 명령

### 개요

Custom Logic Editor는 다음과 같은 래치 명령을 사용합니다.

- LATCH
- LATCH\_NV

## LATCH

LATCH 명령:

- 부울 값(0 또는 1)을 임시 레지스터에 저장합니다.
- 저장된 값을 설정하고 초기화하는 방법을 제공합니다.
- 이전 스캔의 해제 및 설정 상태를 저장합니다.

인수	표현
1	LATCH TmpReg

입력 인수	유형	비트	설명
TmpReg	UINT	Bit1	상승 에지에서 TmpReg.Bit0를 1로 설정
		Bit2	하강 에지에서 TmpReg.Bit1를 0으로 초기화

출력 인수	유형	비트	설명
TmpReg	UINT	Bit0	래치 상태
		Bit3	TmpReg.Bit1 이력 비트
		비트4	TmpReg.Bit2 이력 비트

## LATCH\_NV

LATCH\_NV 명령:

- 부울 값(0 또는 1)을 비휘발성 레지스터에 저장합니다.
- 저장된 값을 설정하고 초기화하는 방법을 제공합니다.
- 이전 스캔의 해제 및 설정 상태를 저장합니다.

전원 주기 동안 래치 상태를 유지하려면 LATCH 명령 대신 LATCH\_NV 명령을 사용합니다.

인수	표현
1	LATCH_NV NVReg

입력 인수	유형	비트	설명
NVReg	UINT	Bit1	상승 에지에서 TmpReg.Bit0을 1로 설정
		Bit2	상승 에지에서 TmpReg.Bit0을 0으로 초기화

출력 인수	유형	비트	설명
NVReg	UINT	Bit0	래치 상태
		Bit3	TmpReg.Bit1 이력 비트
		비트4	TmpReg.Bit2 이력 비트



# Counter Logic 명령

## 개요

Custom Logic Editor는 다음과 같은 카운터 논리 명령을 사용합니다.

- COUNTER
- COUNTER\_NV

## COUNTER

COUNTER 명령:

- 카운트 값을 증가 또는 감소시킵니다.
- 카운트 값을 사전 설정 값으로 지정하는 방법을 제공합니다.
- 카운트 값이 0일 때를 나타냅니다.
- 카운트 값과 사전 설정 값의 관계(같은지, 큰지, 작은지)를 나타냅니다.
- 이전 스캔에서 증가, 감소 및 설정 상태를 저장합니다

인수	표현
3	COUNTER TmpReg1 KValue TmpReg2

입력 인수	유형	범위/비트	설명
KValue	UINT	0~65,535	카운터 사전 설정 값
TmpReg2	UINT	비트4	상승 에지에서 카운터 현재 값을 증가시킵니다. 카운터 현재 값은 0에서 65,535까지 증가한 후 다시 0에서 시작합니다.
		비트5	하강 에지에서 카운터 현재 값을 감소시킵니다. 카운터 현재 값은 65,535에서 0까지 감소한 후 다시 65,535에서 시작합니다.
		비트6	상승 에지에서 현재 카운터 값을 사전 설정 값으로 지정합니다.

출력 인수	유형	범위/비트	설명
TmpReg1	UINT	0~65,535	카운터 현재 값
TmpReg2	UINT	비트0	카운터 현재 값이 0: TmpReg1=0
		비트1	카운터 현재 값이 사전 설정 값보다 작음: TmpReg1<KValue
		비트2	카운터 현재 값이 사전 설정 값과 같음: TmpReg1=KValue
		비트3	카운터 현재 값이 사전 설정 값보다 큼: TmpReg1>KValue
		비트7	TmpReg2.Bit4 이력 비트
		비트8	TmpReg2.Bit5 이력 비트
		비트9	TmpReg2.Bit6 이력 비트

## COUNTER\_NV

COUNTER\_NV 명령:

- 카운트 값을 증가 또는 감소시킵니다.
- 카운트 값을 사전 설정 값으로 지정하는 방법을 제공합니다.
- 카운트 값이 0일 때를 나타냅니다.

- 카운트 값과 사전 설정 값의 관계(같은지, 큰지, 작은지)를 나타냅니다.
- 이전 스캔에서 증가, 감소 및 설정 상태를 저장합니다.

전원 주기 동안 카운트를 유지하려면 COUNTER 명령 대신 COUNTER\_NV 명령을 사용합니다.

인수	표현
3	COUNTER NVReg1 KValue NVReg2

입력 인수	유형	범위/비트	설명
KValue	UINT	0~65,535	카운터 사전 설정 값
NVReg2	UINT	비트4	상승 에지에서 카운터 현재 값을 증가시킵니다. 카운터 현재 값은 0에서 65,535까지 증가한 후 다시 0에서 시작합니다.
		비트5	하강 에지에서 카운터 현재 값을 감소시킵니다. 카운터 현재 값은 65,535에서 0까지 감소한 후 다시 65,535에서 시작합니다.
		비트6	상승 에지에서 현재 카운터 값을 사전 설정 값으로 지정합니다.

출력 인수	유형	범위/비트	설명
NVReg1	UINT	0~65,535	카운터 현재 값
NVReg2	UINT	비트0	카운터 현재 값이 0: NVReg1=0
		비트1	카운터 현재 값이 사전 설정 값보다 작음: NVReg1<KValue
		비트2	카운터 현재 값이 사전 설정 값과 같음: NVReg1=KValue
		비트3	카운터 현재 값이 사전 설정 값보다 큼: NVReg1>KValue
		비트7	NVReg2.Bit4 이력 비트
		비트8	NVReg2.Bit5 이력 비트
		비트9	NVReg2.Bit6 이력 비트

# Math Logic 명령

## 개요

Custom Logic Editor는 다음과 같은 수학 명령을 사용합니다.

- ON\_ADD
- ON\_SUB
- ON\_MUL
- ON\_DIV

## ON\_ADD

ON\_ADD 명령은 1비트 누산기가 0에서 1로 전환될 때 부호 없는 덧셈을 수행합니다. 인수 1의 값을 16비트 누산기 값에 더한 다음 인수 1의 값에 결과를 다시 씁니다.

상태 레지스터:

- 덧셈 프로세스의 결과가 65,535를 초과할 경우 오버플로를 나타냅니다.
- 이전 스캔의 1비트 누산기 상태를 나타냅니다.

인수	표현
2	ON_ADD TmpReg1 TmpReg2

입력 인수	유형	범위/비트	설명
TmpReg1	UINT	0~65,535	16비트 누산기에 추가할 값

출력 인수	유형	범위/비트	설명
TmpReg1	UINT	0~65,535	덧셈 작업 결과
TmpReg2	UINT	Bit0	오버플로: 덧셈 결과가 65,535보다 큼니다. 이 경우 덧셈의 결과는 $\text{인수 1} + 65,536$ 과 같습니다.
		Bit3	1비트 누산기 이력 비트

## ON\_SUB

ON\_SUB 명령은 1비트 누산기가 0에서 1로 전환될 때 부호 없는 뺄셈을 수행합니다. 인수 1의 값에서 16비트 누산기 값을 뺀 다음 인수 1의 값에 결과를 다시 씁니다.

상태 레지스터:

- 뺄셈 프로세스의 결과가 0보다 작은 경우 오버플로를 나타냅니다.
- 이전 스캔의 1비트 누산기 상태를 나타냅니다.

인수	표현
2	ON_SUB TmpReg1 TmpReg2

입력 인수	유형	범위/비트	설명
TmpReg1	UINT	0~65,535	16비트 누산기에서 뺄 값

출력 인수	유형	범위/비트	설명
TmpReg1	UINT	0~65,535	빨셈 연산 결과
TmpReg2	UINT	Bit0	언더플로: 빨셈 결과가 0보다 작습니다. 이 경우 실제 연산 결과는 인수 1의 값 출력 - 65,536과 같습니다.
		Bit3	1비트 누산기 이력 비트

## ON\_MUL

ON\_MUL 명령은 1비트 누산기가 0에서 1로 전환될 때 부호 없는 곱셈을 수행합니다. ON\_MUL 프로시저는 인수 2의 값과 16비트 누산기 값을 곱한 다음 인수 1(최상위 워드)과 인수 2(최하위 워드)에 결과를 다시 씁니다.

상태 레지스터는 이전 스캔의 1비트 누산기의 상태를 나타냅니다.

인수	표현
3	ON_MUL TmpReg1 TmpReg2 TmpReg3

입력 인수	유형	범위/비트	설명
TmpReg2	UINT	0~65,535	16비트 누산기와 곱할 값

출력 인수	유형	범위/비트	설명
TmpReg1 및 TmpReg2	UINT	0~65,535	곱셈 연산 결과: • TmpReg1: 최상위 워드 유지 • TmpReg2: 최하위 워드 유지
TmpReg3	UINT	Bit3	1비트 누산기 이력 비트

## ON\_DIV

ON\_DIV 명령은 1비트 누산기가 0에서 1로 전환될 때 부호 없는 나눗셈을 수행합니다. ON\_DIV 프로시저는 인수 1과 인수 2의 결합된 값을 16비트 누산기 값으로 나눈 다음 인수 1(최상위 워드)과 인수 2(최하위 워드)에 결과를 다시 씁니다.

상태 레지스터는 다음을 나타냅니다.

- 0으로 나눈 경우 오버플로
- 이전 스캔의 1비트 누산기 상태

인수	표현
3	ON_DIV TmpReg1 TmpReg2 TmpReg3

입력 인수	유형	범위/비트	설명
TmpReg1 및 TmpReg2	UINT	0~65,535	16비트 누산기로 나눌 값

출력 인수	유형	범위/비트	설명
TmpReg1 및 TmpReg2	UINT	0~65,535	나눗셈 연산 결과: • TmpReg1: 최상위 워드 유지 • TmpReg2: 최하위 워드 유지
TmpReg3	UINT	Bit0	0으로 나누기
		Bit3	1비트 누산기 이력 비트

# 사용자 정의 논리 프로그램 예

## 이 장의 내용

타이머 및 곱하기 명령을 확인하는 방법 .....	157
진리표를 생성하는 방법 .....	158

## 타이머 및 곱하기 명령을 확인하는 방법

### 개요

애플리케이션을 사용자 정의할 때 타이머 및 곱하기 명령을 확인해야 할 수도 있습니다.

### 사용자 정의 논리 프로그램을 사용하여 타이머 및 곱하기 명령 확인하기

다음 다이어그램은 타이머 확인 및 명령어 곱셈을 수행하는 사용자 정의 논리 프로그램의 텍스트 뷰를 보여줍니다:

```

LOGIC_ID 356
// A very simple test that checks timers and MUL (multiply command)
// It should switch LO1 and LO2 ON OFF if OK !!
//
LOAD_K_BIT 1
SET_TMP_BIT 115 3
LOAD_TMP_REG 115
ON_SET_TMP_REG 105 111
ON_SET_TMP_REG 108 112
LOAD_NOT_TMP_BIT 110 2 // timer 2 not timing
SET_TMP_BIT 107 0
TIMER_TENTHS 105 106 107
LOAD_NOT_TMP_BIT 107 2 // timer 1 not timing
SET_TMP_BIT 110 0
TIMER_TENTHS 108 109 110
LOAD_TMP_BIT 107 2
SET_BIT 1200 12 // Switch LO1 if timer 1 is working
LOAD_K_REG 50 // Load value of 50
LOAD_K_BIT 1
SET_NOT_TMP_BIT 123 3 // Clear history bit
ON_SET_TMP_REG 122 123 // Save the 50 in temporary register 22
LOAD_K_REG 2 // Load value of 2
SET_NOT_TMP_BIT 123 3
ON_MUL 121 122 123 // Multiply 50x2
LOAD_TMP_REG 122
COMP_K_REG 100 101 // Is result 100?
LOAD_TMP_BIT 110 2 // timer 2 timing
AND_TMP_BIT 101 2 // =100?
SET_BIT 1200 13 // Don't switch LO2 if MUL did not work OK

```

## 진리표를 생성하는 방법

### 개요

애플리케이션을 사용자 정의할 때 진리표를 생성해야 할 수도 있습니다.

### 사용자 정의 논리 프로그램을 사용하여 진리표 생성하기

다음 다이어그램에서는 진리표를 생성 중인 텍스트 보기의 사용자 정의 논리 프로그램을 보여 줍니다.

```

LOGIC_ID 444
//
//
// Truth table example
//
//   I1  I2  I3   Output
//   0   0   0     0   (0)
//   0   0   1     1   (1)
//   0   1   0     1   (2)
//   0   1   1     0   (3)
//   1   0   0     1   (4)
//   1   0   1     0   (5)
//   1   1   0     0   (6)
//   1   1   1     0   (7)

LOAD_BIT 457.0           //SET INPUTS
SET_TMP_BIT 1.1
LOAD_BIT 457.1
SET_TMP_BIT 1.2
LOAD_BIT 457.2
SET_TMP_BIT 1.3

//
//**** 3x1 TRUTH TABLE TEMPLATE
//**** Inputs defined as bits 1.1 through 1.3)
//**** Output defined as bit 1.15
//
LOAD_K_BIT 0             //default output OFF
SET_TMP_BIT 1.15        //save partial result

//*****0** Inputs 1-2-3 are OFF OFF OFF
//
LOAD_NOT_TMP_BIT 1.1    //include this SECTION
AND_NOT_TMP_BIT 1.2    //if output is to be ON
AND_NOT_TMP_BIT 1.3    //REMOVE if output to be OFF
SET_TMP_BIT 1.15       //save partial result
//

```

## 사용자 정의 논리 프로그램으로 진리표 만들기 (계속)

```

LOAD_NOT_TMP_BIT 1.1 //include this SECTION
AND_NOT_TMP_BIT 1.2 //if output is to be ON
AND_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result
//
//*****2** Inputs 1-2-3 are OFF ON OFF
//
LOAD_NOT_TMP_BIT 1.1 //include this SECTION
AND_TMP_BIT 1.2 //if output is to be ON
AND_NOT_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result
//
//*****3** Inputs 1-2-3 are OFF ON ON
//
LOAD_NOT_TMP_BIT 1.1 //include this SECTION
AND_TMP_BIT 1.2 //if output is to be ON
AND_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result
//
//*****4** Inputs 1-2-3 are ON OFF OFF
//
LOAD_TMP_BIT 1.1 //include this SECTION
AND_NOT_TMP_BIT 1.2 //if output is to be ON
AND_NOT_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result
//
//*****5** Inputs 1-2-3 are ON OFF ON
//
LOAD_TMP_BIT 1.1 //include this SECTION
AND_NOT_TMP_BIT 1.2 //if output is to be ON
AND_TMP_BIT 1.3 //REMOVE if output to be OFF
OR_TMP_BIT 1.15 //include previous result
SET_TMP_BIT 1.15 //save partial result

```

## 사용자 정의 논리 프로그램으로 진리표 만들기 (계속)

```
//  
//*****6** Inputs 1-2-3 are ON ON OFF  
//  
LOAD_TMP_BIT 1.1 //include this SECTION  
AND_TMP_BIT 1.2 //if output is to be ON  
AND_NOT_TMP_BIT 1.3 //REMOVE if output to be OFF  
OR_TMP_BIT 1.15 //include previous result  
SET_TMP_BIT 1.15 //save partial result  
//  
//*****7** Inputs 1-2-3 are ON ON ON  
//  
LOAD_TMP_BIT 1.1 //include this SECTION  
AND_TMP_BIT 1.2 //if output is to be ON  
AND_TMP_BIT 1.3 //REMOVE if output to be OFF  
OR_TMP_BIT 1.15 //include previous result  
SET_TMP_BIT 1.15 //save partial result  
  
LOAD_TMP_BIT 1.15 //SET OUTPUT  
SET_BIT 1200.14
```

# 기능 블록 다이어그램 언어

## 이 파트의 내용

FBD 언어 개요 .....	162
FBD 요소 .....	165
FBD 언어를 사용하여 프로그래밍하기 .....	180
FBD 블록 조작하기 .....	186
FBD 편집기 표시 옵션 .....	190

# FBD 언어 개요

## 이 장의 내용

FBD 편집기 소개 ..... 163

# FBD 편집기 소개

## 개요

FBD 편집기는 TeSys Tera DTM Library의 기능입니다. FBD 편집기를 사용하여 기존 FBD 프로그램 파일을 보거나 명령문 기반 텍스트 프로그래밍 언어 대신 FBD 언어를 사용하여 FBD 프로그램 파일을 작성할 수 있습니다.

## FBD 프로그램 작성하기

FBD 편집기를 열려면 **Device(장치) > FB Diagram(FB 다이어그램) > New FB Diagram(새 FB 다이어그램)**을 선택하거나 **FB Diagram(FB 다이어그램)** 탭을 선택합니다. FBD 편집기가 주 창에 나타납니다.

## FBD 프로그램 저장하기

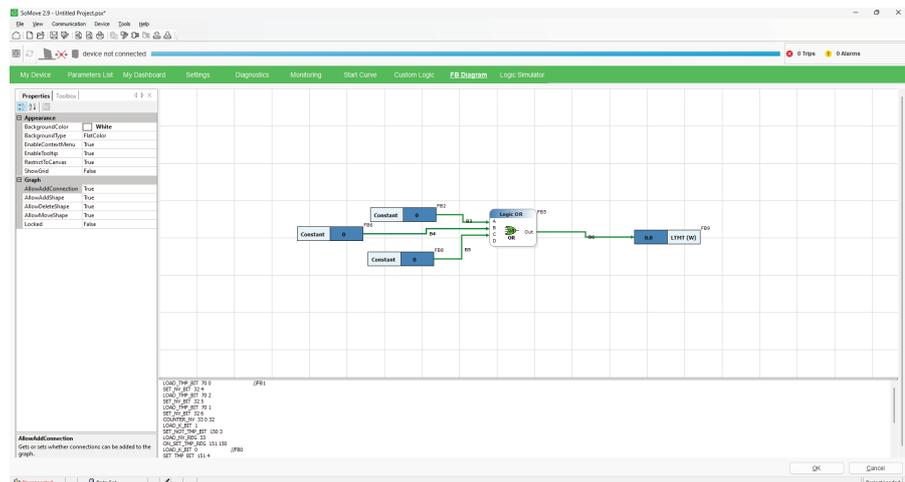
FBD 프로그램을 컴파일하려면 먼저 저장해야 합니다. 작성하거나 편집한 프로그램을 저장하려면 **Device(장치) > FB Diagram(FB 다이어그램) > Save FB Diagram as(다른 이름으로 FB 다이어그램 저장)**를 선택합니다.

주의: 이 파일은 \*.Gef 확장명으로 저장됩니다.

## FBD 편집기 사용자 인터페이스

FBD 편집기는 TeSys Tera DTM Library가 연결 모드일 때도 사용할 수 있습니다. 그러나 많은 메뉴 항목은 FBD 프로그램이 FBD 편집기에 열려 있는 경우에만 활성화됩니다.

아래 창은 FBD 편집기에 열려 있는 FBD 프로그램을 보여 줍니다.



## 작업 영역

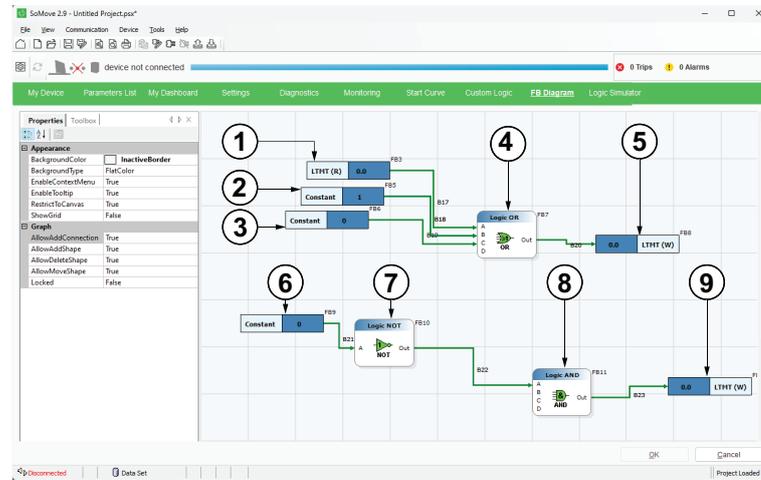
작업 영역에서 FBD 프로그램을 편집하고 작성합니다.

작업 영역은 다음 두 요소로 구성됩니다.

- 블록
- 블록을 연결하는 배선

## FBD 프로그램 실행하기

FBD 프로그램은 왼쪽에서 오른쪽으로, 위에서 아래로 한 라인씩 실행됩니다. 아래 예에서 명령문들은 숫자로 표시된 순서대로, 명령문 1에서 명령문 5까지 그리고 명령문 6에서 명령문 9까지 실행되거나 수행됩니다.



# FBD 요소

## 이 장의 내용

Starter Types .....	166
Computation Blocks .....	168
Inputs Blocks .....	171
Function Blocks.....	174
Logic Blocks .....	177
Outputs Blocks.....	177

# Starter Types

## 개요

FBD 편집기를 사용하면 도구 상자의 **Starter Types(스타터 유형)** 표시줄을 통해 액세스할 수 있는 스타터 블록을 사용할 수 있습니다.

블록	설명
	Direct Online
	Reversible Direct Online
	Star Delta
	Overload

**주의:** 아이콘 위에 커서를 놓으면 아이콘을 정의하는 도구 설명이 나타납니다. 이 정보를 통해 해당 아이콘으로 표시되는 블록의 유형을 구별할 수 있습니다.

**주의:** 스타터 유형의 블록은 하나만 사용할 수 있습니다.

## Direct Online



블록을 사용하면 스타터 유형이 Direct Online으로 설정됩니다.

FBD 표시	인수	설명
	디지털 입력 신호	<ul style="list-style-type: none"> <li>DI1(5629.0): 로컬-시작&gt; DI</li> <li>DI2(5629.1): 로컬-정지 DI</li> <li>DI3(5629.3): DI 모드 선택 1</li> <li>DI4(5629.6): 리모트-시작&gt; DI</li> <li>DI5(5629.7): 리모트-정지 DI</li> <li>DI6(5629.9): DI 모드 선택 2</li> <li>DI7(5629.12): 실행 DI</li> </ul>
	HMI 신호	<ul style="list-style-type: none"> <li>HMI1(9875.0): HMI_START</li> <li>HMI2(9875.12): HMI_STOP</li> <li>HMI3(9875.2): HMI 모드 선택 1</li> <li>HMI4(9875.4): HMI 모드 선택 2</li> </ul>
	통신 신호	<ul style="list-style-type: none"> <li>RI1(704.0): 통신 시작</li> <li>RI2(704.2): 리모트 모드 선택 1</li> <li>RI3(704.4): 리모트 모드 선택 2</li> <li>RI4(704.12): 통신 정지</li> </ul>
	출력 신호	<ul style="list-style-type: none"> <li>OP1(32.0): CL Run1 Cde 정보</li> </ul>

# 과부하



블록을 사용하면 스타터 유형이 Overload로 설정됩니다.

FBD 표시	인수	설명
	OP1	출력 신호 1

# Reversible Direct Online



블록을 사용하면 스타터 유형이 Reversible Direct Online으로 설정됩니다.

FBD 표시	인수	설명
	디지털 입력 신호	<ul style="list-style-type: none"> <li>DI1(5629.0): 로컬-시작&gt; DI</li> <li>DI2(5629.1): 로컬-정지 DI</li> <li>DI3(5629.3): DI 모드 선택 1</li> <li>DI4(5629.4): 로컬-시작&lt; DI</li> <li>DI5(5629.6): 리모트-시작&gt; DI</li> <li>DI6(5629.7): 리모트-정지 DI</li> <li>DI7(5629.9): DI 모드 선택 2</li> <li>DI8(5629.10): 리모트-시작&lt; DI</li> <li>DI9(5629.12): 실행 DI</li> </ul>
	HMI 신호	<ul style="list-style-type: none"> <li>HMI1(9875.0): HMI_START &gt;</li> <li>HMI2(9875.12): HMI_STOP,</li> <li>HMI3(9875.1): HMI_START &lt;</li> <li>HMI4(9875.2): HMI 모드 선택 1</li> <li>HMI5(9875.4): HMI 모드 선택 2</li> </ul>
	통신 신호	<ul style="list-style-type: none"> <li>R1(704.0): 통신 시작 &gt;</li> <li>R2(704.1): 통신 시작 &lt;</li> <li>R3(704.2): 리모트 모드 선택 1</li> <li>R4(704.4): 리모트 모드 선택 2</li> <li>R5(704.12): 통신 정지</li> </ul>
	출력 신호	<ul style="list-style-type: none"> <li>OP1(32.0): CL Run1 Cde 정보</li> <li>OP2(32.1): CL Run2 Cde 정보</li> </ul>

# Star Delta



블록을 사용하면 스타터 유형이 Star Delta로 설정됩니다.

FBD 표시	인수	설명
	디지털 입력 신호	<ul style="list-style-type: none"> <li>DI1(5629.0): 로컬-시작&gt; DI</li> <li>DI2(5629.1): 로컬-정지 DI</li> <li>DI3(5629.3): DI 모드 선택 1</li> <li>DI4(5629.6): 리모트-시작&gt; DI</li> <li>DI5(5629.7): 리모트-정지 DI</li> <li>DI6(5629.9): DI 모드 선택 2</li> <li>DI7(5629.12): 실행 DI</li> </ul>
	HMI 신호	<ul style="list-style-type: none"> <li>HMI1(9875.0): HMI_START &gt;</li> <li>HMI2(9875.12): HMI_STOP,</li> <li>HMI3(9875.2): HMI 모드 선택 1</li> <li>HMI4(9875.4): HMI 모드 선택 2</li> </ul>
	통신 신호	<ul style="list-style-type: none"> <li>R11(704.0): 통신 시작 &gt;</li> <li>R12(704.2): 리모트 모드 선택 1</li> <li>R13(704.4): 리모트 모드 선택 2</li> <li>R14(704.12): 통신 정지</li> </ul>
	출력 신호	<ul style="list-style-type: none"> <li>OP1(32.0): CL Run1 Cde 정보</li> <li>OP2(32.1): CL Run2 Cde 정보</li> <li>OP3(32.2): CL Run3 Cde 정보</li> </ul>

## Computation Blocks

### 개요

FBD 편집기는 도구 상자의 **Computation(계산)** 표시줄을 통해 액세스할 수 있는 다양한 계산 블록을 사용합니다.

블록	설명
	Compare
	Add
	Division
	Multiplication
	Subtraction

**주의:** 아이콘 위에 커서를 놓으면 아이콘을 정의하는 도구 설명이 나타납니다. 이 정보를 통해 해당 아이콘으로 표시되는 블록의 유형을 구별할 수 있습니다.

### Compare Block



블록은 두 16비트 레지스터 값을 비교합니다.

FBD 기호	인수	설명
	입력	<ul style="list-style-type: none"> <li>X: 부호 없는 16비트 레지스터 값(0~65,535)</li> <li>Y: 부호 없는 16비트 레지스터 값(0~65,535)</li> </ul>
	출력	<ul style="list-style-type: none"> <li>X &lt; Y: X 값이 Y 값보다 작은 경우 ON인 ON/OFF 임시 비트입니다.</li> <li>X = Y: X 값이 Y 값과 같은 경우 ON인 ON/OFF 임시 비트입니다.</li> <li>X &gt; Y: X 값이 Y 값보다 큰 경우 ON인 ON/OFF 임시 비트입니다.</li> </ul>

## Add Block



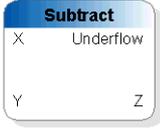
블록은 두 16비트 레지스터 값의 부호 없는 덧셈을 수행합니다.

FBD 기호	인수 또는 예	설명
	입력	<ul style="list-style-type: none"> <li>X: 부호 없는 16비트 레지스터 값(0~65,535)</li> <li>Y: 부호 없는 16비트 레지스터 값(0~65,535)</li> </ul>
	출력	<ul style="list-style-type: none"> <li>Z: 부호 없는 16비트 레지스터 결과(<math>Z = X + Y</math>)</li> <li>오버플로: ON으로 설정되면 65,536 값을 수반하는 ON 또는 OFF 값입니다. 값은 OFF로 초기화됩니다.</li> </ul>
	예	X = 60,000, Y = 7,000이라고 가정하면 $60,000 + 7,000 = 67,000$ 으로 65,536보다 크기 때문에 오버플로가 ON이 됩니다. 그러면 결과 Z는 1,464( $1,464 + 65,356 = 67,000$ )와 같습니다.

## Subtraction Block



블록은 두 16비트 레지스터 값의 부호 없는 뺄셈을 수행합니다.

FBD 기호	인수 또는 예	설명
	입력	<ul style="list-style-type: none"> <li>X: 부호 없는 16비트 레지스터 값(0~65,535)</li> <li>Y: 부호 없는 16비트 레지스터 값(0~65,535)</li> </ul>
	출력	<ul style="list-style-type: none"> <li>Z: 부호 없는 16비트 레지스터 결과(<math>Z = X - Y</math>)</li> <li>언더플로: ON으로 설정되면 음수 65,536 값을 수반하는 ON 또는 OFF 값입니다. 값은 OFF로 초기화됩니다.</li> </ul>
	예	X = 5, Y = 10이라고 가정하면 결과는 음수이므로 언더플로가 ON이 됩니다. 그러면 결과 Z는 65,531( $65,531 - 65,536 = -5$ )와 같습니다.

## Multiplication Block



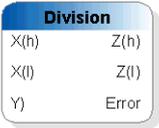
블록은 두 16비트 레지스터 값의 부호 없는 곱셈을 수행합니다.

FBD 기호	인수 또는 예	설명
	입력	<ul style="list-style-type: none"> <li>X: 부호 없는 16비트 레지스터 값(0~65,535)</li> <li>Y: 부호 없는 16비트 레지스터 값(0~65,535)</li> </ul>
	출력	<ul style="list-style-type: none"> <li>Z(h): 32비트 결과 중 최상위 16비트 <math>Z(h) = (X * Y) / 65,536</math></li> <li>Z(l): 32비트 결과 중 최하위 16비트 <math>Z(l) = (X * Y) - Z(h) * 65,536</math></li> </ul>
	예	X = 20,000, Y = 10이라고 가정하면 $200,000 = 3 * 65,536 + 3,392$ 이므로 결과는 Z(h) = 3 및 Z(l) = 3,392가 됩니다.

## Division Block



블록은 두 16비트 레지스터 값의 부호 없는 나눗셈을 수행합니다.

FBD 기호	인수 또는 예	설명
	입력	<ul style="list-style-type: none"> <li>X(h): 부호 없는 레지스터 값의 최상위 16비트(0~65,535)</li> <li>X(l): 부호 없는 레지스터 값의 최하위 16비트(0~65,535)</li> <li>Y: 부호 없는 16비트 레지스터 나눗수(0~65,535)</li> </ul>
	출력	<ul style="list-style-type: none"> <li>Z(h): 32비트 몫의 최상위 16비트 <math>Z(h) = (X/Y) / 65,536</math></li> <li>Z(l): 32비트 몫의 최하위 16비트 <math>Z(l) = (X/Y) - Z(h) * 65,536</math></li> <li>Detected Error: 0으로 나누기가 발생할 때 ON으로 설정되는 ON 또는 OFF 값입니다. 이 값은 OFF로 초기화됩니다.</li> </ul>
	예	X(h) = 3, X(l) = 3,392, Y = 40이라고 가정하면, 결과는 Z(h) = 0 및 Z(l) = 5,000이 됩니다. 왜냐하면 $X(h) * 65,536 + X(l) = 3 * 65,536 + 3,392 = 200,000$ 이고 $200,000 / Y = 5,000 = 0 * 65,536 + 5,000$ 이므로

# Inputs Blocks

## 개요

FBD 편집기는 도구 상자의 **Inputs(입력)** 표시줄을 통해 액세스할 수 있는 다양한 입력 블록을 사용합니다.

블록	설명
	Constant Bit
	Constant Word
	Register Bit In
	Register Word In
	Register NV Bit In
	Register NV Word In
	Register Temp Bit In
	Register Temp Word In

주의: 아이콘 위에 커서를 놓으면 아이콘을 정의하는 도구 설명이 나타납니다. 이 정보를 통해 해당 아이콘으로 표시되는 블록의 유형을 구별할 수 있습니다.

## Constant Bit Block

이  블록은 다른 블록의 입력을 0 또는 1로 설정하는 데 사용됩니다.

FBD 기호	인수	설명
	특성	• 상수 비트 값 0 또는 1(True = 1 및 False = 0)
	출력	• 상수 값 0 또는 1(True = 1 및 False = 0)

## Constant Word Block

이  블록은 다른 블록의 입력값(0부터 65,535까지)을 설정하는 데 사용됩니다.

FBD 기호	인수	설명
Constant 0	특성	• 0에서 65,535까지의 상수 값
	출력	• 0에서 65,535까지의 상수 값

## Register Bit In Block

**1**  
LTMT

블록을 사용하면 LTMT main unit 레지스터의 레지스터 비트 값을 읽고 사용할 수 있습니다.

FBD 기호	인수	설명
LTMT (R) 0.0	특성	• 0~9249(0x0000~0x2421) 사이, 9875(0x2693) 또는 9876(0x2694)의 레지스터 주소 x 0~15의 비트 번호
	출력	• 값 0 또는 1 (True = 1 및 False = 0)

## Register Word In Block

**16**  
LTMT

블록을 사용하면 LTMT main unit 레지스터의 레지스터 값을 읽고 사용할 수 있습니다.

FBD 기호	인수	설명
LTMT (R) 0	특성	• 0~9249(0x0000~0x2421) 사이, 9875(0x2693) 또는 9876(0x2694)의 레지스터 주소입니다.
	출력	• 0에서 65,535까지의 값

## Register NV Bit Block

**1**  
NV

블록을 사용하면 비휘발성 레지스터 비트 값을 읽고 사용할 수 있습니다.

FBD 기호	인수	설명
NV (R) 0.0	특성	• 0에서 63까지의 모든 비휘발성 레지스터 x 0에서 15까지의 비트 번호
	출력	• 값 0 또는 1(True = 1 및 False = 0)

## Register NV Word In Block

**16**  
NV

블록을 사용하면 비휘발성 레지스터 값을 읽고 사용할 수 있습니다.

FBD 기호	인수	설명
NV (R) 0	특성	• 0에서 63까지의 모든 비휘발성 레지스터입니다.
	출력	• 0에서 65,535까지의 값

## Register Temp Bit In Block



블록을 사용하면 임시 레지스터 비트 값을 읽고 사용할 수 있습니다.

FBD 기호	인수	설명
	특성	<ul style="list-style-type: none"> <li>0에서 299까지의 모든 임시 레지스터 x 0에서 15까지의 비트 번호</li> <li></li> </ul>
	출력	<ul style="list-style-type: none"> <li>값 0 또는 1(True = 1 및 False = 0)</li> </ul>

## Temp Word In Block



블록을 사용하면 임시 레지스터 값을 읽고 사용할 수 있습니다.

FBD 기호	인수	설명
	특성	<ul style="list-style-type: none"> <li>0에서 299까지의 모든 임시 레지스터입니다.</li> </ul>
	출력	<ul style="list-style-type: none"> <li>0에서 65,535까지의 값</li> </ul>

# Function Blocks

## 개요

FBD 편집기는 도구 상자의 **Function(기능)** 표시줄을 통해 액세스할 수 있는 다양한 기능 블록을 사용합니다.

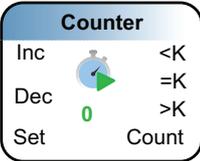
블록	설명
	Counter
	Counter NV
	Volatile Latch
	Non Volatile Latch
	Multiplexer
	TimerSeconds
	TimerTenthSeconds

**주의:** 아이콘 위에 커서를 놓으면 아이콘을 정의하는 도구 설명이 나타납니다. 이 정보를 통해 해당 아이콘으로 표시되는 블록의 유형을 구별할 수 있습니다.

## Counter Block



기능은 비교 카운트를 수행하여 카운터 현재 값과 카운터 사전 설정 값을 모두 임시 레지스터에 저장합니다.

FBD 기호	인수	설명
	특성	K: 카운터 사전 설정 값(UINT 0~65,535)
	입력	<ul style="list-style-type: none"> <li>Inc: 상승 에지에서 카운터 현재 값을 증가시킵니다. 카운터 현재 값은 0에서 65,535까지 증가한 후 다시 0에서 시작합니다.</li> <li>Dec: 하강 에지에서 카운터 현재 값을 감소시킵니다. 카운터 현재 값은 65,535에서 0까지 감소한 후 다시 65,535에서 시작합니다.</li> <li>Set: 상승 에지에서 현재 카운터 값을 사전 설정 값으로 지정합니다.</li> </ul>
	출력	<ul style="list-style-type: none"> <li>Count: 카운터 현재 값(UINT 0~65,535) 전원 공급 시 카운트는 0으로 초기화됩니다.</li> <li>&lt;K: 카운터 현재 값이 사전 설정 값 K보다 작습니다.</li> <li>=K: 카운터 현재 값이 사전 설정 값 K와 같습니다.</li> <li>&gt;K: 카운터 현재 값이 사전 설정 값 K보다 큼니다.</li> </ul>

**주의:** 카운터 사전 설정 값 범위는 0에서 65,535까지입니다. 더 큰 값이나 여러 사전 설정 값이 필요한 경우 연속형 카운터 및 비교 기능을 사용할 수 있습니다.

## Counter NV Block



기능은 비교 카운트를 수행하여 카운터 현재 값과 카운터 사전 설정 값을 모두 비휘발성 레지스터에 저장합니다.

FBD 기호	인수	설명
	특성	K: 카운터 사전 설정 값(UINT 0~65,535)
	입력	<ul style="list-style-type: none"> <li>Inc: 상승 에지에서 카운터 현재 값을 증가시킵니다. 카운터 현재 값은 0에서 65,535까지 증가한 후 다시 0에서 시작합니다.</li> <li>Dec: 하강 에지에서 카운터 현재 값을 감소시킵니다. 카운터 현재 값은 65,535에서 0까지 감소한 후 다시 65,535에서 시작합니다.</li> <li>Set: 상승 에지에서 현재 카운터 값을 사전 설정 값으로 지정합니다.</li> </ul>
	출력	<ul style="list-style-type: none"> <li>Count: 카운터 현재 값(UINT 0~65,535) 이 값은 비휘발성 메모리에 저장되며 전원 공급 시 이전 값으로 초기화됩니다.</li> <li>&lt;K: 카운터 현재 값이 사전 설정 값 K보다 작습니다.</li> <li>=K: 카운터 현재 값이 사전 설정 값 K와 같습니다.</li> <li>&gt;K: 카운터 현재 값이 사전 설정 값 K보다 큼니다.</li> </ul>

**주의:** 카운터 사전 설정 값 범위는 0에서 65,535까지입니다. 더 큰 값이나 여러 사전 설정 값이 필요한 경우 연속형 카운터 및 비교 기능을 사용할 수 있습니다.

## Volatile Latch Block



기능은 임시 레지스터에 신호 이력을 기록하고 유지합니다.

FBD 기호	인수	설명
	입력	<ul style="list-style-type: none"> <li>Set: 입력 값 ON/OFF입니다. 이 입력이 OFF에서 ON으로 전환될 때 래치 값은 ON으로 설정됩니다.</li> <li>Clear: 입력 값 ON/OFF입니다. 이 입력이 OFF에서 ON으로 전환될 때 래치 값은 OFF로 설정됩니다.</li> </ul>
	출력	<ul style="list-style-type: none"> <li>Q: 이 래치의 상태를 나타내는 ON 또는 OFF 래치 값입니다. 이 값은 다음 번 Set 또는 Clear 상승 에지까지 ON/OFF로 유지됩니다. 이 값은 OFF로 초기화됩니다.</li> </ul>

## Non Volatile Latch Block



기능은 비휘발성 레지스터에 신호 이력을 기록하고 유지합니다.

FBD 기호	인수	설명
	입력	<ul style="list-style-type: none"> <li>Set: 입력 값 ON/OFF입니다. 이 입력이 OFF에서 ON으로 전환될 때 래치 값은 ON으로 설정됩니다.</li> <li>Clear: 입력 값 ON/OFF입니다. 이 입력이 OFF에서 ON으로 전환될 때 래치 값은 OFF로 설정됩니다.</li> </ul>
	출력	<ul style="list-style-type: none"> <li>Q: 이 래치의 상태를 나타내는 ON 또는 OFF 비휘발성 레지스터 비트 값입니다. 이 값은 다음 번 Set 또는 Clear 상승 에지까지 ON/OFF로 유지됩니다. 이 값은 비휘발성 메모리에 저장되며 전원 공급 시 이전 상태로 초기화됩니다.</li> </ul>

## Multiplexer Block



기능을 사용하면 두 부호 없는 16비트 값 중에서 선택할 수 있습니다.

FBD 기호	인수	설명
	입력	<ul style="list-style-type: none"> <li>A: 부호 없는 16비트 값(0~65,535)</li> <li>B: 부호 없는 16비트 값(0~65,535)</li> <li>A/B: A 또는 B 값을 선택하는 ON/OFF 입력 값</li> </ul>
	출력	<ul style="list-style-type: none"> <li>출력: 선택한 16비트 값:                             <ul style="list-style-type: none"> <li>A/B가 ON이면 Out = A입니다.</li> <li>A/B가 OFF이면 Out = B입니다.</li> </ul> </li> </ul>

## Timer Seconds Block



기능은 초 간격으로 시간을 측정합니다.

FBD 기호	타이밍 다이어그램	인수	설명
		입력	<ul style="list-style-type: none"> <li>Time: 기간을 초 단위로 지정하는 부호 없는 16비트 값 (0~65,535)입니다.</li> <li>Enable: 입력 값 ON/OFF입니다. 기간은 Enable 입력의 상승 에지에서 로드됩니다. Enable이 ON인 동안 시간 측정이 계속됩니다. Enable이 OFF이면 타이밍이 중지되고 출력이 OFF입니다.</li> </ul>
		출력	<ul style="list-style-type: none"> <li>Timed - 기간이 만료된 후 ON으로 전환되는 ON/OFF 값입니다. Enable이 OFF인 동안은 OFF입니다.</li> <li>Timing - Enable이 ON이고 시간을 측정하는 동안 ON인 ON/OFF 값입니다. 측정 시간이 만료된 후에는 OFF입니다.</li> </ul> <p>주의: 두 출력이 동시에 ON일 수 없습니다.</p>

## Timer Tenths Seconds Block



기능은 10분의 1초 간격으로 시간을 측정합니다.

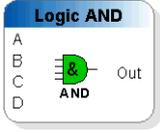
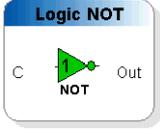
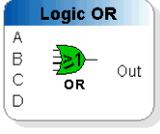
**주의:** TIMER TENTHS 명령의 곱셈 계수는 10입니다. 예를 들어, 필요한 값이 6초이면 입력 값에 계수 10을 곱해야 합니다. 즉, 제공된 입력 값은 60초여야 합니다.

FBD 기호	타이밍 다이어그램	인수	설명
		입력	<ul style="list-style-type: none"> <li>Time: 기간을 10분의 1초 단위로 지정하는 부호 없는 16비트 값(0~65,535)입니다.</li> <li>Enable: 입력 값 ON/OFF입니다. 기간은 Enable 입력의 상승 에지에서 로드됩니다. Enable이 ON인 동안 시간 측정이 계속됩니다. Enable이 OFF이면 타이밍이 중지되고 출력이 OFF입니다.</li> </ul>
		출력	<ul style="list-style-type: none"> <li>Timed - 기간이 만료된 후 ON으로 전환되는 ON/OFF 값입니다. Enable이 OFF인 동안은 OFF입니다.</li> <li>Timing - Enable이 ON이고 시간을 측정하는 동안 ON인 ON/OFF 값입니다. 측정 시간이 만료된 후에는 OFF입니다.</li> </ul> <p>주의: 두 출력이 동시에 ON일 수 없습니다.</p>

# Logic Blocks

## 개요

FBD 편집기는 도구 상자의 **Logic blocks(논리 블록)** 표시줄을 통해 액세스할 수 있는 다양한 논리 블록을 사용합니다.

기능	아이콘	FBD 기호	설명
AND			모든 입력(ON 또는 OFF 값, 각각 1 또는 0)이 ON이면 출력은 ON입니다. 하나 이상의 입력이 OFF이면 출력은 OFF입니다. <b>주의:</b> 연결되지 않은 입력은 ON으로 간주됩니다.
NOT			입력(ON 또는 OFF 값, 각각 1 또는 0)이 ON이면 출력은 OFF입니다. 입력이 OFF이면 출력이 ON입니다.
OR			하나 이상의 입력(ON 또는 OFF 값, 각각 1 또는 0)이 ON이면 출력은 ON입니다. 모든 입력이 OFF이면 출력은 OFF입니다. <b>주의:</b> 연결되지 않은 입력은 OFF로 간주됩니다.

**주의:** 아이콘 위에 커서를 놓으면 아이콘을 정의하는 도구 설명이 나타납니다. 이 정보를 통해 해당 아이콘으로 표시되는 블록의 유형을 구별할 수 있습니다.

# Outputs Blocks

## 개요

FBD 편집기는 도구 상자의 **Outputs(출력)** 표시줄을 통해 액세스할 수 있는 다양한 출력 블록을 사용합니다.

블록	설명
	Register Bit Out
	Register Word Out
	Register NV Bit Out
	Register NV Word Out
	Register Temp Bit Out
	Temp Word Out

**주의:** 아이콘 위에 커서를 놓으면 아이콘을 정의하는 도구 설명이 나타납니다. 이 정보를 통해 해당 아이콘으로 표시되는 블록의 유형을 구별할 수 있습니다.

## Register Bit Out Block



블록은 LTMT main unit 레지스터에서 LTMT main unit 레지스터 비트 값을 0 또는 1로 설정하는 데 사용됩니다.

FBD 기호	인수	설명
	특성	<ul style="list-style-type: none"> <li>a: 0~9249(0x0000~0x2421) 사이, 9875(0x2693) 또는 9876(0x2694)의 레지스터 주소입니다.</li> <li>b: 0에서 15까지의 비트 번호입니다.</li> </ul>
	입력	<ul style="list-style-type: none"> <li>0 또는 1(ON=1 및 OFF=0)</li> </ul>

## Register Word Out Block



블록은 LTMT main unit 레지스터에서 LTMT main unit 레지스터 값을 설정하는 데 사용됩니다.

FBD 기호	인수	설명
	특성	<ul style="list-style-type: none"> <li>a: 0~9249(0x0000~0x2421) 사이, 9875(0x2693) 또는 9876(0x2694)의 레지스터 주소입니다.</li> </ul>
	입력	<ul style="list-style-type: none"> <li>0에서 65,535까지의 부호 없는 16비트 값입니다.</li> </ul>

## Register NV Bit Out Block



블록은 비휘발성 레지스터 레지스터 비트 값을 0 또는 1로 설정하는 데 사용됩니다.

FBD 기호	인수	설명
	특성	<ul style="list-style-type: none"> <li>a: 0에서 63까지의 모든 비휘발성 레지스터입니다.</li> <li>b: 0에서 15까지의 비트 번호입니다.</li> </ul>
	입력	<ul style="list-style-type: none"> <li>0 또는 1(ON=1 및 OFF=0)</li> </ul>

## Register NV Word Out Block



블록은 비휘발성 레지스터 값을 설정하는 데 사용됩니다.

FBD 기호	인수	설명
	특성	<ul style="list-style-type: none"> <li>a: 0에서 63까지의 모든 비휘발성 레지스터입니다.</li> </ul>
	입력	<ul style="list-style-type: none"> <li>0에서 65,535까지의 부호 없는 16비트 값</li> </ul>

## Register Temp Bit Out Block



블록은 임시 레지스터 비트 값을 0 또는 1로 설정하는 데 사용됩니다.

FBD 기호	인수	설명
	특성	<ul style="list-style-type: none"> <li>a: 0에서 299까지의 모든 임시 레지스터입니다.</li> <li>b: 0에서 15까지의 비트 번호입니다.</li> </ul>
	입력	<ul style="list-style-type: none"> <li>0 또는 1(ON=1 및 OFF=0)</li> </ul>

## Temp Word Out Block



블록은 임시 레지스터 값을 설정하는 데 사용됩니다.

FBD 기호	인수	설명
	특성	<ul style="list-style-type: none"> <li>a: 0에서 299까지의 모든 임시 레지스터입니다.</li> </ul>
	입력	<ul style="list-style-type: none"> <li>0에서 65,535까지의 부호 없는 16비트 값</li> </ul>

# FBD 언어를 사용하여 프로그래밍하기

## 이 장의 내용

FBD 블록 삽입하기 .....	181
블록 간 링크 생성하기 .....	182
FBD 블록 속성 .....	184
FBD 리소스 관리 .....	185

# FBD 블록 삽입하기

## 개요

FBD 프로그램을 작성하려면 블록을 작업 영역에 삽입한 다음 함께 연결합니다. 모든 유형의 블록을 작업 영역에 배치할 수 있습니다.

## 도구 상자에서 블록 삽입하기

다음 절차에서는 도구 상자에서 작업 영역으로 블록을 삽입하는 방법에 대해 설명합니다.

1. **Device(장치) > FB Diagram(FB 다이어그램) > View(보기) > Toolbox(도구 상자)**를 선택하거나 왼쪽에 있는 **Toolbox(도구 상자)** 탭을 선택합니다.
2. 삽입할 블록 유형 선택:
  - Computation
  - Inputs
  - Function Blocks
  - Logic
  - Starter Types
  - Outputs
3. 삽입할 블록에 해당하는 아이콘을 마우스 왼쪽 버튼으로 클릭합니다.
4. 블록을 도구 상자에서 작업 영역으로 끌어 놓습니다.
5. 작업 영역의 필요한 위치에 블록을 배치합니다.
6. 2~5단계를 반복하여 프로그램에 필요한 모든 블록을 삽입합니다.

## 작업 영역에서 블록 삽입하기

다음 절차에서는 작업 영역에서 직접 블록을 삽입하는 방법에 대해 설명합니다.

1. 작업 영역의 빈 공간을 마우스 오른쪽 버튼으로 클릭합니다.  
**결과:** 삽입하려는 블록 유형을 선택할 수 있는 메뉴가 나타납니다.
2. 삽입할 블록 유형 선택:
  - Computation
  - Inputs
  - Function Blocks
  - Logic
  - Starter Types
  - Outputs
3. 삽입할 블록을 마우스 왼쪽 버튼으로 클릭합니다.
4. 작업 영역의 필요한 위치에 블록을 배치합니다.
5. 1~5단계를 반복하여 프로그램에 필요한 모든 블록을 삽입합니다.

## 블록 간 링크 생성하기

### 개요

작업 영역에 블록을 배치한 후 블록을 함께 연결할 수 있습니다. 이렇게 하려면 블록의 출력을 다른 블록의 입력에 연결합니다. 한 블록의 출력을 입력에 연결하여 루프를 형성할 수도 있습니다.

### 일반 규칙

블록을 배치하고 연결할 때 적용되는 기본 규칙이 있습니다.

- 함께 부착되는 하나 이상의 연결 배선은 배선 노드를 형성합니다. 이는 작업 영역에 빨간색 점으로 표시됩니다. 배선이 빨간색 연결 점 없이 교차하면 연결되지 않은 것입니다.
- 각 배선 노드에는 하나의 출력만 부착할 수 있습니다.
- 부울 데이터와 레지스터 데이터 간의 연결은 금지됩니다.
- 데이터는 일반적으로 왼쪽에서 오른쪽으로 흐릅니다.

### 블록 간 링크

다음 절차에서는 블록을 함께 연결하는 방법에 대해 설명합니다.

1. 마우스를 첫 번째 블록 위에 놓습니다.

**결과:** 블록 테두리에 하나 이상의 사각형이 표시되고 출력 유형(아날로그 또는 부울)이 나타납니다.



2. 마우스 왼쪽 버튼을 클릭한 채로 누르고 있습니다.
3. 버튼을 누른 채 커서를 연결하려는 블록의 입력 위로 이동합니다.

**결과:** 블록 테두리에 하나 이상의 사각형이 표시됩니다. 사각형이 녹색이면 두 블록 사이를 연결할 수 있습니다. 빨간색 사각형은 연결할 수 없다는 의미입니다. 출력 유형(아날로그 또는 부울)도 표시됩니다.



**주의:** 입력과 출력은 같은 유형이어야 합니다. 즉, 부울 출력은 다른 부울 출력에 연결됩니다. 입력 또는 출력이 같지 않으면 FBD 편집기에서 원본과 대상이 동일한 유형이 아님을 나타내는 팝업 창이 표시됩니다.

4. 마우스 버튼을 놓습니다.

**결과:** 연결된 두 블록 사이에 라인과 숫자가 표시됩니다.

5. 1~4 단계를 반복하여 모든 블록을 연결합니다.

### 링크 번호

배선에는 두 가지 유형이 있습니다.

- 번호가 B로 시작하는 부울 배선
  - 번호가 R로 시작하는 레지스터 배선
- 배선 번호는 시간 순서에 따라 자동으로 증가합니다.

# FBD 블록 속성

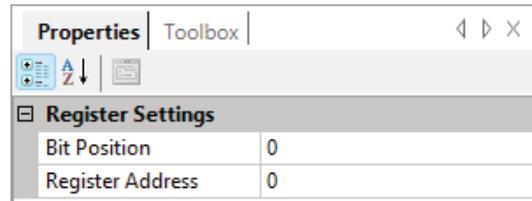
## 개요

각 블록에는 속성 창이 있습니다. 이 창을 표시하려면 블록을 마우스 왼쪽 버튼으로 클릭합니다.

The 속성 창은 블록 유형에 따라 1개 또는 2개 범주로 구분된 여러 탭으로 구성됩니다:

- 일반 설정 - 블록 ID 및 주석을 포함하며, 모든 유형의 블록에 공통입니다.
- 블록 유형에 따른 특정 설정(레지스터의 경우 레지스터 설정, 카운터의 경우 카운터 설정 등).

예를 들어, 비휘발성 레지스터 속성을 표시하려면 비휘발성 레지스터 블록을 선택하고 마우스 왼쪽 버튼으로 클릭합니다. 다음 창이 표시됩니다.



## 설명

설명 영역에서 오른쪽의 흰색 상자에 설명을 입력할 수 있습니다. 작업 영역에서 개체 또는 자유 위치를 선택하여 설명을 저장합니다.

## 설정

대부분의 블록에는 관련 설정 탭이 있습니다. 이 탭에서 블록의 특정 설정을 지정합니다. 이러한 설정은 각 FBD 블록에 대한 도움말에 자세히 설명되어 있습니다.

## 속성 표시

각 블록의 속성은 다음 두 방법으로 표시할 수 있습니다.

- 범주별 -  을 클릭합니다.
- 알파벳 순 -  을 클릭합니다.

## FBD 리소스 관리

### 개요

LTMT main unit 메모리에는 다음과 같은 리소스가 포함되어 있습니다.

- 9250 LTMT 레지스터
- 8,192워드(16비트) 크기의 논리 메모리 공간
- 300개의 임시 레지스터
- 64개의 비휘발성 레지스터

### 예약된 리소스

다음 표에는 모든 예약된 레지스터 및 해당 할당을 나와 있습니다. 또한 이러한 레지스터를 제어하는 방식을 보여 줍니다.

레지스터 유형	주소 범위	제어 주체	설명
임시	0~69	사용자	FBD 프로그램을 작성할 때 사용자가 할당한 비트 및 레지스터의 임시 저장
임시	70~299	FBD 컴파일러	컴파일러에서 사용하도록 예약된 임시 레지스터
비휘발성	0~31	사용자	FBD 프로그램을 작성할 때 사용자가 할당한 비휘발성 비트 또는 레지스터
비휘발성	32~63	FBD 컴파일러	컴파일러에서 사용하도록 예약된 비휘발성 레지스터

# FBD 블록 조작하기

## 이 장의 내용

블록 선택하기 .....	187
개체 삭제 또는 복제하기 .....	187

# 블록 선택하기

## 개요

작업 영역에 블록을 추가할 때 블록을 선택하여 작업 영역 내에서 위치를 변경할 수 있습니다.

## 하나 이상의 블록 선택하기

다음 표에서는 하나 이상의 블록을 선택하는 방법에 대해 설명합니다.

선택하려는 대상...	필요한 작업
격리된 블록	블록을 클릭합니다.
여러 연속 블록	선택 영역을 정의하여 선택할 블록을 둘러쌉니다. 결과: 선택한 모든 블록이 주황색 윤곽선으로 강조 표시됩니다. 
작업 영역의 여러 영역에 있는 여러 블록	Shift 키를 누른 채 선택할 블록을 클릭합니다. Shift 키를 계속 누르고 있어야 합니다. 결과: 선택한 모든 블록이 주황색 윤곽선으로 강조 표시됩니다.
배선을 포함한 모든 개체	<b>Device(장치) &gt; FB Diagram(FB 다이어그램) &gt; FBD Editor(FBD 편집기) &gt; Select All (모두 선택)</b> 을 선택합니다. 주의: 바로 가기 키 <b>Ctrl+A</b> 를 사용하여 모든 개체를 선택할 수도 있습니다.

# 개체 삭제 또는 복제하기

## 개요

작업 영역에서 블록을 삭제하거나 블록을 복제해야 하는 경우가 있습니다.

## 블록 삭제하기

다음 절차에서는 하나 이상의 블록을 삭제하는 방법에 대해 설명합니다.

1. 삭제할 블록을 선택합니다.

결과: 선택한 블록이 주황색 윤곽선으로 강조 표시됩니다.



2. Delete 키를 누르거나 **Device(장치) > FB Diagram(FB 다이어그램) > FBD Editor(FBD 편집기) > Delete(삭제)**를 선택합니다.

결과: 선택한 블록이 삭제됩니다.

1. 삭제할 블록을 선택합니다.

결과: 선택한 블록이 주황색 윤곽선으로 강조 표시됩니다.



2. 선택한 블록을 마우스 오른쪽 버튼으로 클릭하고 **Delete(삭제)**를 선택합니다.

결과: 선택한 블록이 삭제됩니다.

## 블록 잘라내기, 복사 또는 붙여넣기

다음 절차에서는 하나 이상의 블록을 잘라내거나 복사하거나 붙여넣는 방법에 대해 설명합니다.

1. 수정할 블록을 선택합니다.

결과: 선택한 블록이 주황색 윤곽선으로 강조 표시됩니다.



2. **Device(장치) > FB diagram(FB 다이어그램) > FBD editor(FBD 편집기)**를 선택하고 다음 명령 중 하나를 선택합니다.

- **Copy(복사)**
- **Cut(잘라내기)**
- **Paste(붙여넣기)**

결과: **Cut(잘라내기)**은 선택한 블록을 삭제하고 클립보드에 저장합니다. **Copy(복사)**는 클립보드에서 선택한 블록을 복제하고 **Paste(붙여넣기)**는 작업 영역에 클립보드 내용을 복제합니다.

주의: 바로 가기 키 **Ctrl+C** 및 **Ctrl+V**를 사용하여 선택한 블록을 복사하여 붙여넣을 수도 있습니다.

## 블록 숨기기 또는 표시

다음 절차에서는 하나 이상의 블록을 숨기는 방법에 대해 설명합니다.

1. 숨길 블록을 선택합니다.

결과: 선택한 블록이 주황색 윤곽선으로 강조 표시됩니다.

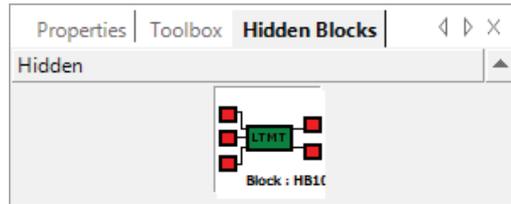


2. 선택한 블록을 마우스 오른쪽 버튼으로 클릭하고 **Hide/Show(숨기기/표시)**를 선택합니다.

결과: 선택된 블록이 다음과 같이 숨겨집니다.



3. 아래 그림처럼, 숨겨진 블록의 목록은 **Hidden Blocks(숨겨진 블록)**에 표시됩니다.



**주의:** 여러 블록을 선택하면 모든 블록이 동일한 블록으로 그룹화됩니다.

다음 절차에서는 하나 이상의 숨겨진 블록을 표시하는 방법에 대해 설명합니다.

1. 숨겨진 블록을 마우스 오른쪽 버튼으로 클릭하고 **Hide/Show(숨기기/표시)**를 선택합니다.

**결과:** 선택한 블록이 표시됩니다.

2. 또는 **Hidden Blocks(숨겨진 블록)** 섹션 아래에서 숨겨진 블록을 마우스 오른쪽 버튼으로 클릭하고 **Hide/Show(숨기기/표시)**를 선택합니다.

**결과:** 선택한 블록이 표시됩니다.

# FBD 편집기 표시 옵션

## 이 장의 내용

표시 옵션.....	191
작업 영역 모양새 및 그래프 옵션.....	192

# 표시 옵션

## 요약

요구 사항에 맞게 다음 표시 옵션을 사용자 정의할 수 있습니다.

- 확대/축소
- 링크
- 인수

## 확대/축소 표시 옵션

확대/축소 옵션에 액세스하려면 **Device(장치) > FB Diagram(FB 다이어그램) > View(보기)**를 선택합니다.

다음과 같은 세 가지 옵션이 제공됩니다.

- Zoom out(축소) - 한 번에 더 많은 프로그램을 볼 수 있습니다.
- Zoom in(확대) - 프로그램에 자세히 집중합니다.
- Zoom to(확대/축소 비율) - 50%, 75%, 100%, 150 %, 200% 또는 400%로 확대/축소하여 프로그램의 보기를 사용자 정의할 수 있습니다.

## 링크 표시 옵션

링크 표시 옵션에 액세스하려면 **Device(장치) > FB Diagram(FB 다이어그램) > Tools(도구)**를 선택합니다.

다음과 같은 세 가지 옵션이 제공됩니다. 기능은 다음과 같습니다.

- Renumber links(링크 번호 다시 매기기) - 프로그램 실행을 이해하는 데 도움이 됩니다.
- Show all links(모든 링크 표시) - 함께 연결되는 블록을 확인합니다.
- Hide all links(모든 링크 숨기기) - 블록의 전체 보기를 개선합니다.

링크를 클릭하면 해당 속성 창이 나타나고 링크 옆에 나타나는 텍스트를 사용자 정의할 수 있습니다.

## 인수 표시 옵션

다음 절차에서는 인수 표시 옵션에 액세스하고 변경하는 방법에 대해 설명합니다.

1. 마우스를 블록 위에 놓습니다.

**결과:** 블록 테두리에 하나 이상의 사각형이 표시됩니다. 또한 인수가 아날로그 또는 부울인지 여부가 나타납니다.



2. 이 사각형을 클릭합니다.

**결과:** 표시 옵션이 나타납니다.

3. 레이블을 표시할지 여부와 표시할 텍스트를 선택합니다.

## 작업 영역 모양새 및 그래프 옵션

### 요약

FBD 편집기를 사용하면 모양새 및 그래프 옵션을 변경하여 작업 영역을 사용자 정의할 수 있습니다.

### 모양새 및 그래프 옵션

모양새 및 그래프 옵션에 액세스하려면 작업 영역에서 개체를 제외한 아무 곳이나 마우스 왼쪽 버튼으로 클릭합니다.

### 모양새 옵션

다음 표에는 가능한 모든 모양새 사용자 정의 옵션이 나와 있습니다.

모양새 옵션	설명	가능한 선택
Background Color(배경색)	색상이 표시되는 상자를 클릭하여 작업 영역의 배경색을 설정할 수 있습니다.	<b>Custom(사용자 정의), Web(웹) 및 System(시스템)</b> 탭에서 사용할 수 있는 색상 중에서 선택할 수 있습니다.
Background Type(배경 유형)	배경 유형을 설정할 수 있습니다.	플랫 색상, 그라디언트 또는 이미지 배경 중에서 선택합니다.
Enable Context Menu(상황에 맞는 메뉴 사용)	상황에 맞는 메뉴를 표시하거나 숨깁니다.	True 또는 False
Enable Tooltip(도구 설명 사용)	도구 설명을 표시하거나 숨깁니다.	True 또는 False
Restrict to Canvas(캔버스로 제한)	FBD 프로그램을 캔버스 내부에 유지되도록 할지 여부를 선택할 수 있습니다.	True 또는 False
Show Grid(그리드 표시)	정확한 그리드가 표시되는지 여부를 선택할 수 있습니다.	True 또는 False <b>주의:</b> 이 그리드를 최상위 수준 <b>View(보기)</b> 메뉴 모음에서 액세스하는 그리드 라인과 혼동해서는 안 됩니다.

### 그래프 옵션

다음 표에는 가능한 모든 그래프 사용자 정의 옵션이 나와 있습니다.

그래프 옵션	설명	가능한 선택
Allow Add Connection(연결 추가 허용)	작업 영역에 연결을 추가할 수 있는지 여부를 선택할 수 있습니다.	True 또는 False
Allow Add Shape(모양 추가 허용)	작업 영역에 블록을 추가할 수 있는지 여부를 선택할 수 있습니다.	True 또는 False
Allow Delete Shape(모양 삭제 허용)	블록을 삭제할 수 있는지 여부를 선택할 수 있습니다.	True 또는 False
Allow Move Shape(모양 이동 허용)	작업 영역에서 블록을 이동할 수 있는지 여부를 선택할 수 있습니다.	True 또는 False
Locked(잠김)	FBD 프로그램을 편집할 수 있는지 여부를 선택할 수 있습니다.	True 또는 False

## Display Gridlines(눈금선 표시)

눈금선을 표시하고 싶을 수 있습니다. 이 경우 **Device(장치) > FB Diagram(FB 다이어그램) > View(보기) > Show Gridlines(눈금선 표시)**를 선택합니다.

# 프로그램 컴파일, 시뮬레이션 및 전송하기

## 이 파트의 내용

소개.....	195
LTMT Main Unit Logic Simulator .....	196
초기화 및 연결.....	198
LTMT Main Unit과 Custom Logic Editor 간에 논리 파일 전송하기 .....	199

## 소개

### 컴파일 개요

사용자 정의 프로그램을 LTMT main unit에 다운로드하려면 먼저 컴파일해야 합니다.

- 사용자 정의 논리 언어로 된 프로그램은 직접 컴파일할 수 있습니다.
- FBD 언어로 된 프로그램은 먼저 사용자 정의 논리 언어 프로그램으로 변환해야 사용자 정의 논리 프로그램으로 컴파일할 수 있습니다.

컴파일에는 다음과 같은 프로그램 오류에 대한 검사가 포함됩니다.

- 구문 및 구조 오류
- 해당 주소가 없는 기호
- 사용할 수 없는 프로그램에서 사용하는 리소스
- 프로그램이 사용 가능한 LTMT main unit 메모리에 맞는 크기인지 여부

### FB Diagram을 사용자 정의 논리로 변환하기

**Device(장치) > FB Diagram(FB 다이어그램) > FB Diagram to Custom Logic(FB 다이어그램을 사용자 정의 논리로 변환)**을 선택하여 작성되거나 편집된 FBD를 사용자 정의 논리 프로그램으로 컴파일합니다.

프로그램은 자동으로 사용자 정의 로직 편집기 에 자동으로 복사됩니다.

**주의:** 사용자 정의 논리 프로그램 파일을 FBD 프로그램 파일로 변환할 수 없으므로 FBD 프로그램을 변환하려면 먼저 FBD 편집기에 저장해야 합니다.

### 사용자 정의 논리 컴파일하기

PCode로 작성한 사용자 정의 논리 프로그램을 컴파일하려면 다음 단계를 따릅니다.

1. **Device(장치) > Custom Logic(사용자 정의 논리)**을 선택합니다.
2. **Compile Custom Logic(사용자 정의 논리 컴파일)**을 선택합니다.

**주의:** 오류가 감지되지 않으면 PCode 창이 표시됩니다. 그렇지 않으면, 오류 감지 창이 표시됩니다.

# LTMT Main Unit Logic Simulator

## 개요

SoMove가 있는 TeSys Tera DTM Library software는 Logic Simulator와 함께 제공됩니다. 이 시뮬레이터를 사용하면 사용자 정의 프로그램을 LTMT main unit으로 전송하기 전에 사용자 정의 논리 언어로 기능을 테스트할 수 있습니다.

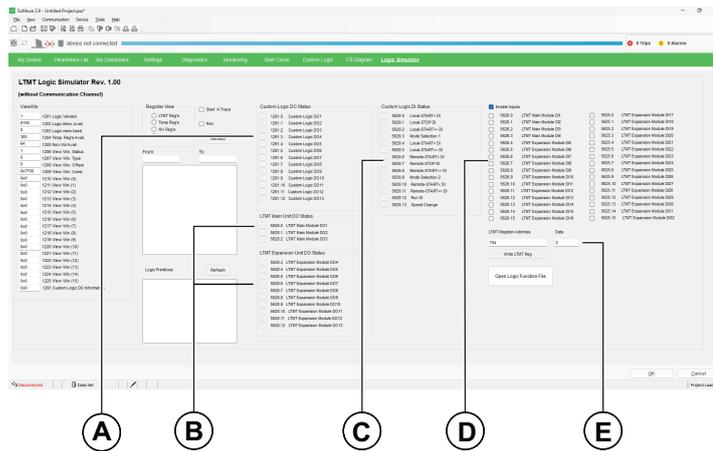
**주의:** FBD 프로그램을 시뮬레이션하려면 먼저 이 프로그램을 사용자 정의 논리 프로그램으로 변환하고 \*//확장명을 사용하여 저장해야 합니다.

## Logic Simulator 인터페이스

Logic Simulator를 열려면 **Logic Simulator** 탭을 선택합니다. 그러면 Logic Simulator 창이 표시됩니다. 오른쪽 하단 모서리에서 **Open Logic Function File(논리 기능 파일 열기)**를 선택하여 저장된 사용자 정의 논리 프로그램을 가져옵니다.

사용자 정의 프로그램 파일을 선택하고 **Open(열기)**를 선택합니다.

이제 Logic Simulator에 사용자 정의 프로그램이 로드됩니다. 사용자 정의 프로그램은 아래 표시된 대로 시뮬레이터에서 시뮬레이션할 수 있습니다.



- A 사용자 정의 논리 DO 상태를 업데이트합니다.
- B LTMT 모듈 DO 상태를 업데이트합니다.
- C 사용자 정의 논리 DI 상태를 업데이트합니다.
- D LTMT 모듈 DI 데이터를 시뮬레이션하는 데 사용됩니다.
- E LTMT 레지스터 주소를 시뮬레이션하는 데 사용됩니다.

## Logic Simulator 사용하기

Logic Simulator를 사용하여 사용자 정의 논리 파일을 시뮬레이션하려면 다음 단계를 따릅니다.

- 선택 로직 시뮬레이터 탭.
- **Open Logic Function File(논리 기능 파일 열기)**를 선택하고 PC에서 저장된 사용자 정의 논리 파일을 선택합니다.
- **Register View(레지스터 보기)** 아래에서 레지스터 유형을 선택하고 **Refresh(새로 고침)**를 클릭하여 레지스터 목록을 로드합니다.

- 요구 사항에 따라 **LTMT Register Address(LTMT 레지스터 주소)** 및 **Data(데이터)** 필드 각각에서 레지스터 및 데이터를 수정하고 **Write LTMT Reg(LTMT 레지스터 쓰기)**를 클릭합니다.
- **Refresh(새로 고침)**를 선택하여 레지스터 목록을 제공된 최신 값으로 업데이트합니다.

**주의:** 제공된 새 값으로 레지스터 목록을 업데이트하려면 **Refresh(새로 고침)**를 클릭합니다.

**주의:** 사용자 정의 논리 파일을 시뮬레이터에 업로드하고 타이머를 활성화하면 예상대로 작동합니다. 그러나 타이머를 활성화하고 실행한 후에는 다시 활성화할 수 없습니다. 이는 타이머 및 10분의 1 타이머의 제한 사항입니다. 타이머를 다시 활성화하려면 사용자 정의 논리 파일을 다시 업로드해야 합니다. 타이머로 작업하는 사용자 정의 논리 파일을 설계하고 테스트할 때는 이 제한 사항을 고려하는 것이 좋습니다.

# 초기화 및 연결

## 초기화

LTMT main unit을 PC에 연결하면 컨트롤러가 자동으로 초기화됩니다. 이러한 초기화 프로세스를 통해 컨트롤러와 PC가 식별 정보를 교환할 수 있습니다.

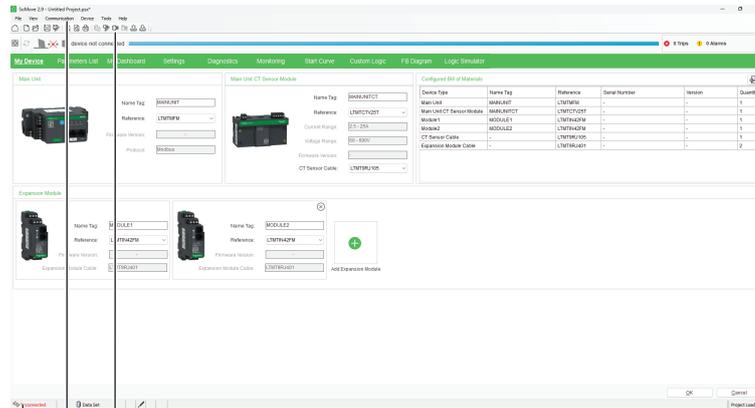
이 프로세스 동안 Custom Logic Editor는 초기화가 완료될 때까지 대기합니다.

## 연결

초기화한 후 LTMT main unit은 PC에 자동으로 연결되어야 합니다.

컨트롤러가 연결되었는지 확인하려면 Custom Logic Editor에서 상태 표시줄을 확인합니다.

상태 표시줄에 **Disconnected(연결 해제)**가 나타나면 **Communication(통신) > Connect to Device(장치에 연결)**를 선택하거나 **Connect to Device(장치에 연결)** 아이콘을 선택합니다.



- A      연결 상태
- B      통신 메뉴
- C      장치에 연결 아이콘

PC가 컨트롤러에 연결될 때 진행률 표시줄이 잠시 나타나고 연결 프로세스가 성공적으로 완료되면 상태 표시줄에 **Connected(연결)**라는 단어가 표시됩니다.

LTMT main unit이 연결된 경우 다음을 수행할 수 있습니다.

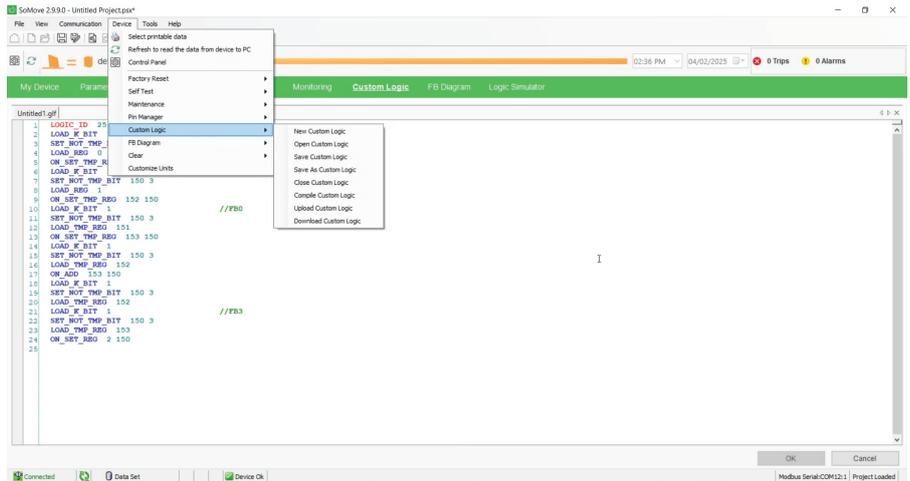
- 컨트롤러에서 TeSys Tera DTM Library가 있는 SoMove software로 편집할 사용자 정의 프로그램 파일을 업로드합니다.
- 편집된 사용자 정의 프로그램 파일을 TeSys Tera DTM Library가 있는 SoMove software에서 컨트롤러로 다운로드합니다.

# LTMT Main Unit과 Custom Logic Editor 간에 논리 파일 전송하기

## 장치에서 사용자 정의 논리를 PC로 파일 전송

LTMT main unit에서 Custom Logic Editor로 프로그램 파일을 전송하려면 다음과 같이 하십시오.

1. LTMT main unit이 PC에 연결되어 있는지 확인합니다.
2. **Device(장치) > Custom Logic(사용자 정의 논리) > Upload Custom Logic(사용자 정의 논리 업로드)**을 선택하여 프로그램 파일을 LTMT main unit에서 Custom Logic Editor로 전송합니다.



3. 사용자 정의 프로그램 파일이 전송되면 Custom Logic Editor를 사용하여 사용자 정의 논리 프로그램으로 편집할 수 있습니다.

**주의:** LTMT main unit에서 검색된 프로그램은 주석이 없는 사용자 정의 논리 형식입니다. FBD 프로그램과 같은 프로그램은 LTMT main unit에서 검색할 수 없습니다.

4. 프로그램 파일 편집이 완료되면 작업을 파일에 저장합니다.

**Device(장치) > Custom Logic(사용자 정의 논리) > Save Custom Logic(사용자 정의 논리 저장) 또는 Custom Logic(사용자 정의 논리)Save As Custom Logic(다른 이름으로 사용자 정의 논리 저장)**을 선택합니다.

## PC에서 사용자 정의 논리를 장치로 파일 전송하는 절차

프로그램 파일을 편집하고 컴파일한 후 파일을 LTMT main unit에 전송할 수 있습니다. TeSys Tera DTM Library가 있는 SoMove software에서 이 전송을 수행하려면 먼저 다음 조건이 충족되어야 합니다.

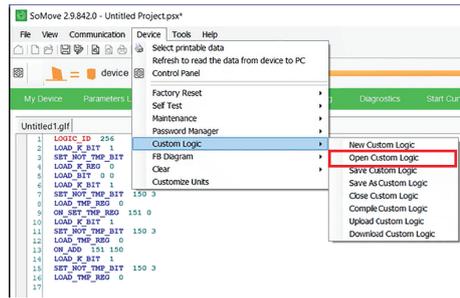
- 전송할 프로그램 파일은 LTMT main unit에 있는 프로그램 파일과 달라야 합니다. 즉, 소프트웨어는 동일한 프로그램을 전송하지 않습니다.
- 전류가 감지되어서는 안 됩니다. 즉, 온라인 전류가 FLC의 10% 미만이어야 합니다.

이러한 조건이 충족되지 않으면 파일을 컨트롤러로 전송할 수 없습니다.

Custom Logic Editor에서 LTMT main unit으로 프로그램 파일을 전송하려면 다음을 수행합니다.

1. LTMT main unit이 PC에 연결되어 있는지 확인합니다.

- 전송할 파일이 주 창에 있는지 확인합니다. 파일을 열려면 **Device(장치)** 메뉴의 **Custom Logic(사용자 정의 논리)** 하위 메뉴에서 **Open Custom Logic(사용자 정의 논리 열기)**을 선택합니다. 그런 다음 올바른 위치를 찾아 선택하고 **Open(열기)**을 선택합니다.



- Device(장치) > Custom Logic(사용자 정의 논리) > Compile Custom Logic(사용자 정의 논리 컴파일)**을 선택하여 사용자 정의 논리 프로그램을 컴파일합니다.
- 프로그램을 컴파일한 후 **Device(장치) > Custom Logic(사용자 정의 논리) > Download Custom Logic(사용자 정의 논리 다운로드)**를 선택하여 Custom Logic Editor에서 LTMT main unit으로 프로그램 파일을 다운로드합니다. 이제 전송에 성공했습니다.
- 새 대화 상자가 열리면 **Ok(확인)**를 선택하여 닫습니다.



Schneider Electric Industries SAS  
35 rue Joseph Monier  
92500 Rueil Malmaison  
France

[www.se.com](http://www.se.com)

표준, 사양 및 설계는 수시로 변경될 수 있으므로 이 출판물에서 제공하는 정보의 정확성을 확인하려면 당사료 문의하십시오..

© 2025 Schneider Electric. 무단 전재 금지.

DOCA0275KO-01