TM1110

Integrated Motion Control – Axis Groups

PERFECTION IN AUTOMATION
www.br-automation.com

Prerequisites and requirements

Training modules	TM410 - Working with Integrated Motion Control TM440 - Motion Control: Basic Functions
Software	Automation Studio 4.2 Automation Runtime 4.08 mapp Technology 1.00.0
Hardware	-

Table of contents

1 소개

축 그룹을 사용하면 단일 기능 블록을 사용하여 여러 축을 그룹화하고 응용 프로그램에서 제어 할 수 있습니다. 여기에는 축 켜기와 같은 준비 작업과 실제 이동을 수행하고 그룹의 개별 축에서 정보를 검색하는 작업이 포함됩니다.

여러 축과 관련된 경로 제어 이동은 설정 점 생성을 사용하여 컨트롤러에서 중앙에서 조정하거나 개별 드라이브에서 독립적으로 수행할 수 있습니다.

축 그룹을 사용하면 단일 기능 블록을 사용하여 여러 축을 그룹화하고 응용 프로그램에서 제어 할 수 있습니다.

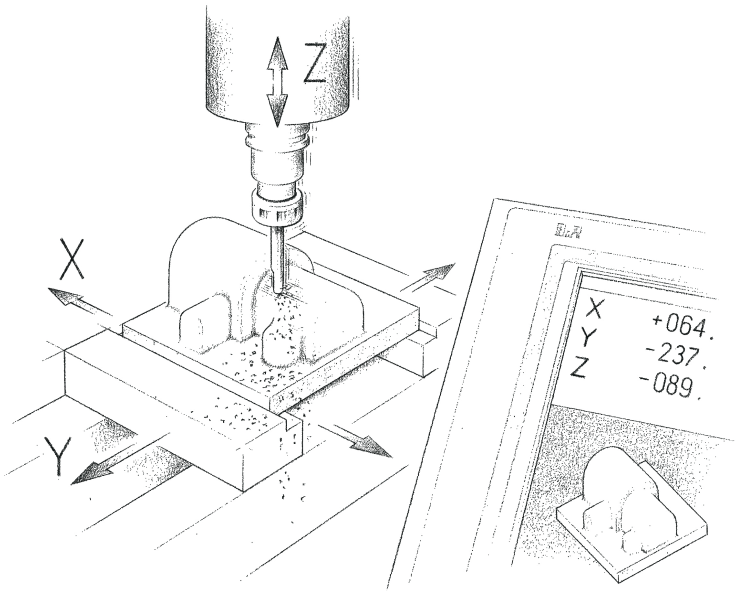


Figure 1: CNC milling machine

1.1 학습 목표

선택된 실습을 통해, 이 교육 모듈은 Automation Studio에서 축 그룹을 생성, 작동 및 프로그래밍하는 과정을 안내합니다.

- Generic Motion Control의 개념에 익숙해지고 각 라이브러리가 함께 작동하는 방식을 이해합니다.
- 축 그룹을 사용할 수 있는 위치와 그들이 제공하는 이점을 알게 됩니다.
- Automation Studio에서 축 그룹을 설정하고 축 그룹을 사용하여 자체 프로젝트를 생성하는 방법을 배우게 됩니다.
- 축 그룹을 시운전하기 위한 절차와 그룹 동작을 진단하는 방법을 배우게 됩니다.
- mapp 기술을 사용하여 축 그룹 프로그래밍을 학습하면 축 그룹 응용 프로그램을 생성 할 수 있습니다.
- PLCopen 기능 블록의 동작을 알게 되고 이 기능을 사용하여 축 그룹 어플리케이션을 확장 할 수 있습니다.
- 기존 축 그룹에 축을 추가하는 방법을 학습합니다.

2 The Generic Motion Control 개념

2.1 The B&R drive solution

B&R 제품의 범위는 모든 일반적인 유형의 드라이브 기술을 포괄합니다. 제품 제조 프로세스 및 기술적 요구 사항에 따라 자동화에 많은 다른 유형의 드라이브 개념을 사용할 수 있습니다.

축 그룹에 속하는 모든 구성 요소는 필드 버스를 통해 기계 어플리케이션을 실행하는 컨트롤러와네트워크로 연결됩니다.

컨트롤러, 공정 시각화 및 다양한 구동 기술은균일하게 혼합 될 수 있습니다.

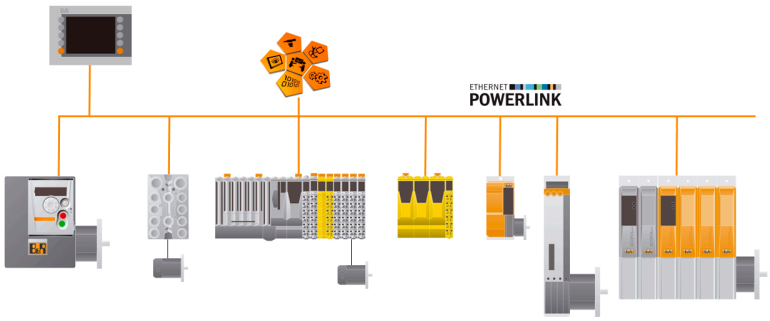


Figure 2: B&R hardware topology

스테퍼 또는 서보 모터가 필요한지 여부와 관계없이 드라이브는 POWERLINK를 통해 다른 제어 구성 요소에 항상 연결될 수 있으며 축 그룹에서 사용될 수 있습니다.

2.2 Automation Studio

Automation Studio는 로직, HMI, 안전 및 모션 어플리케이션의 계획과 설계를 결합합니다. 내장 진단 및 시운전 인터페이스는 프로젝트 계획 설계를 용이하게 하며 모션 구성 요소를 시운전 할 때 도움이 됩니다. 모든 설정 및 매개 변수는 Automation Studio 프로젝트에 저장됩니다.

Figure 3: Automation Studio

2.3 Generic Motion Control

Generic Motion Control을 살펴보면 모션 구성 요소를 결합하고 구현하는 옵션이 얼마나 유연한지를 분명히 알 수 있습니다. 또한 포괄적인 소프트웨어 도구를 사용하여 다양한 유형의 모션 어플리케이션을 개발할 수 있습니다.

CNC 기계에 대한 기본 요구 사항은 간단한 속도 매개 변수, 위치 매개 변수 및 축 결합과 같은 요구 사항 뿐 아니라 구동 시스템의 다양한 전기적 특성 그리고 균일한 소프트웨어 인터페이스를 통해 경로 제어 이동을 수행 할 수 있는 기능을 포함합니다.

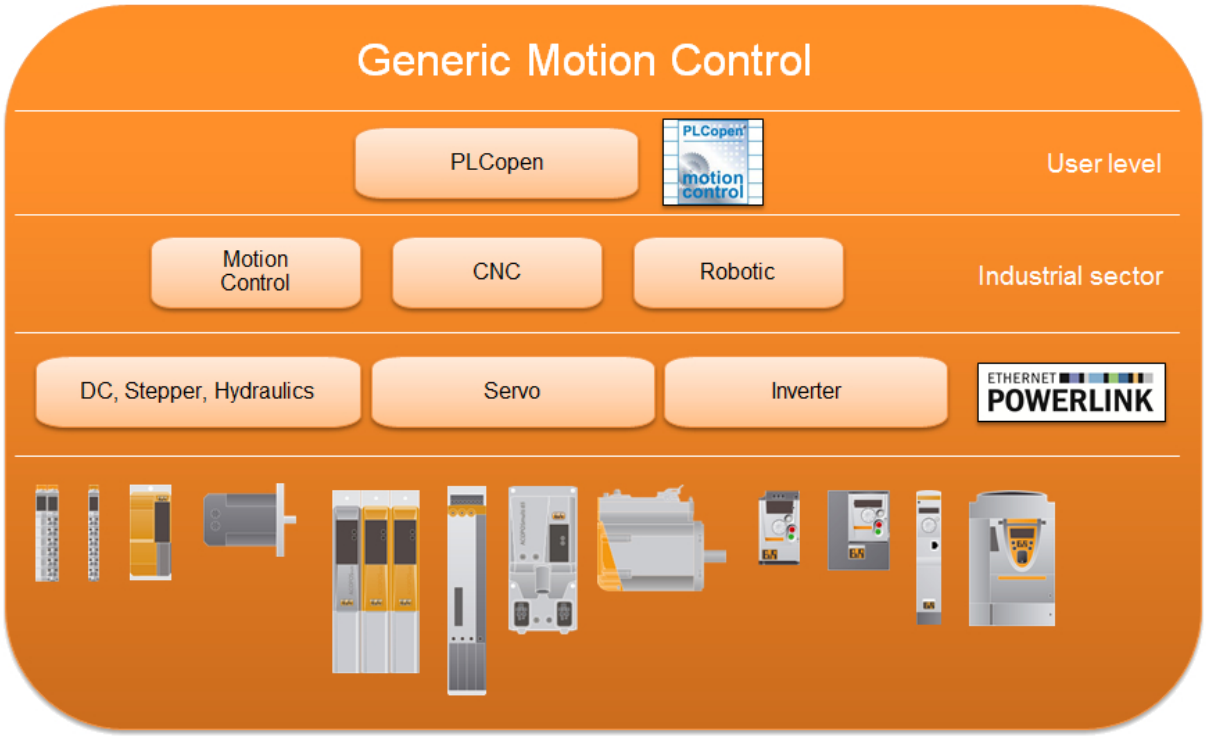


Figure 4: Generic Motion Control

모든 개별 축과 축 그룹은 PLCopen 호환 기능 블록을 사용하여 제어됩니다. 따라서 사용 된 드라이브 하드웨어와 관계없이 반복되는 위치 시퀀스에 대해 동일한 응용 프로그램을 사용할 수 있습니다.

로봇 또는 CNC 기계를 제어하는 축 그룹과 스텝 또는 서보 모터, 주파수 인버터 및 유압 축의 컨버터를 제어하는 개별 축을 동일한 방식으로 작동시키고 어떤 조합으로 결합 할 수 있습니다.

2.4 라이브러리

GMC 라이브러리는 축 그룹 기능과 사용자 인터페이스를 서로 독립적으로 만들 수 있게 해줍니다. 그룹 구현을 위한 별도의 라이브러리와 사용자 인터페이스가 있습니다. 관리자 라이브러리는 이들 사이의 연결을 제공합니다.

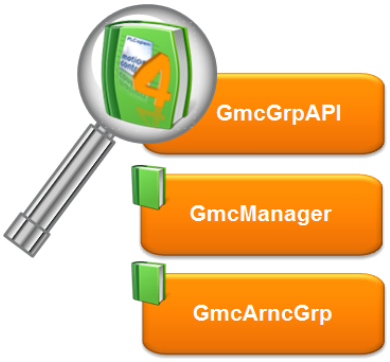


Figure 5: GMC libraries

Library	Description
GmcManager	구현과 사용자 인터페이스 사이의 연결을 제공함으로써, 다른 모든 GMC 라이브러리의 기초 역할을 합니다.
GmcGrpAPI	축 그룹을 제어하기위한 PLCopen 기능 블록 포함
GmcArncGrp	ARNC0을 기반으로하는 축 그룹 구현

Table 1: GMC libraries

GMC Libraries \ GmcArncGrp \ Technical information \ Dependencies of individual libraries

최신 버전의 GMC 라이브러리는 Automation Studio의 업그레이드 대화 상자를 사용하여 다운로드 할 수 있습니다. 그런 다음 프로젝트로 라이브러리로 추가해야 합니다.

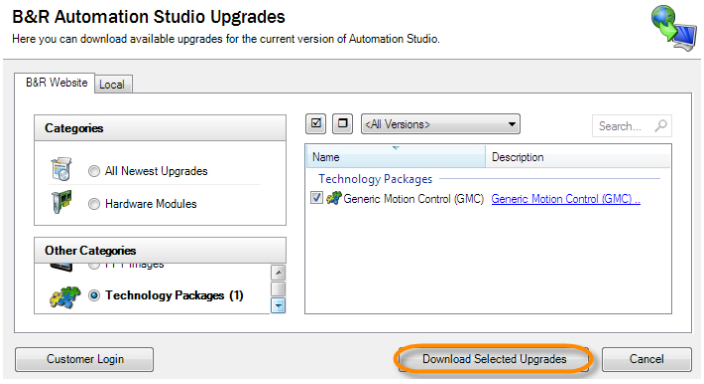


Figure 6: Generic Motion Control technology package

3 Axis group

축 그룹은 응용 프로그램을 단순화하기 위해 여러 축을 결합합니다. 경로 생성기를 사용하여 그룹의 축을 사용하여 경로 제어 이동을 수행 할 수 있습니다. 경로 제어 이동은 CNC 및 로봇 어플리케이션을 구현하는 데 필요합니다.

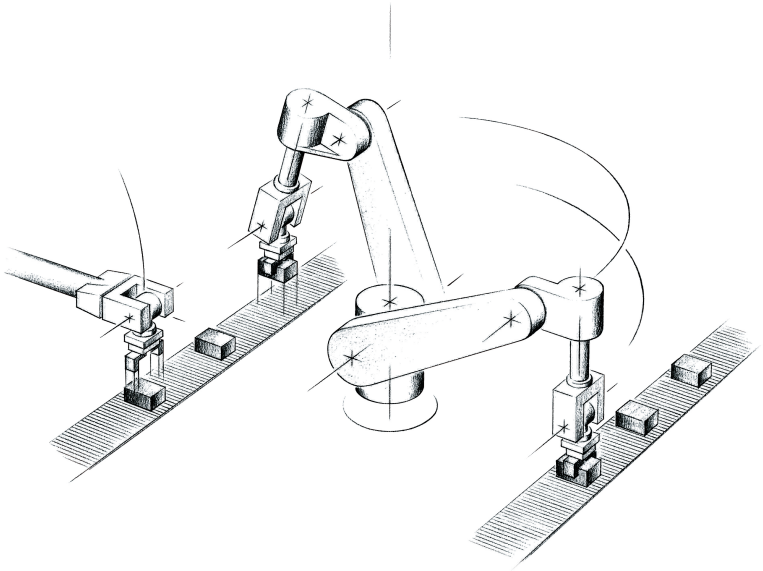


Figure 7: Robotics application

The path generator is described in greater depth in training module "TM1111 - Integrated Motion Control - Path-Controlled Movements."

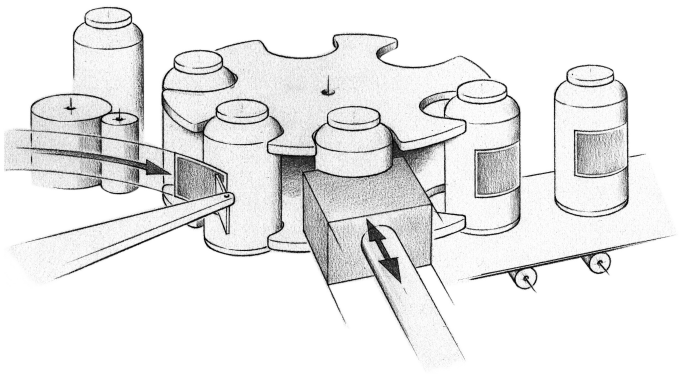


Figure 8: Labeling machines

많은 어플리케이션의 경우 하나의 축 그룹에 다른 축을 조립해야 합니다. 예를 들어 하나의 명령으로 스위치를 켜거나 축 그룹 내의 모든 축에 대해 간단한 오류 처리를 할 수 있습니다.

그러나 대부분의 경우, 축 그룹에서 선택한 축만 중앙 설정 점 생성을 사용하여 조작해야 하므로 선택된 축은 경로 제어를 사용하여 조정됩니다. 경로 계획에 포함되지 않은 축 그룹 내의 축은 경로 제어 이동과 별도로 단일 축 명령으로 이동할 수 있습니다.

Axis types

축 그룹은 일반적으로 여러 유형의 축으로 세분 될 수 있는 다양한 축으로 구성됩니다.

Axis type	Path-controlled	Description
Path axis	✓	경로 축은 경로 생성기에 의해 제어됩니다. 모든 경로 축의 조합은 공간에서 경로를 생성합니다.
Slave axis	✓	보조 축은 경로 생성기로 제어됩니다. 그러나 보조 축은 공간 내 경로와 무관합니다.
Independent axis	✗	독립 축은 경로 생성기에 의해 제어되지 않으며 컨트롤러에서 설정 위치를 수신하지 않습니다. 오직 관리적인 function block 만이 축에 작용합니다.

Table 2: Axis types

샘플 어플리케이션과 함께 축 그룹 및 축 유형에 대한 자세한 설명은 Automation Studio 도움말에서 찾을 수 있습니다.

? Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Technical information \ What is an axis group?

4 Axis group configuration

축 그룹 내의 개별 축의 특성은 축 그룹 구성을 사용하여 구성 할 수 있습니다. 다음은 Automation Studio 프로젝트에서 축 그룹을 만들고 구성하는 방법을 보여줍니다.

4.1 Your first axis group project

X / Y 테이블은 경로 제어를 사용하여 X 및 Y 축을 제어 할 수있는 공작 기계의 첫 번째 축 그룹 예제로 사용됩니다.

이 예제에서는 ARSim이 대상 시스템으로 사용됩니다.

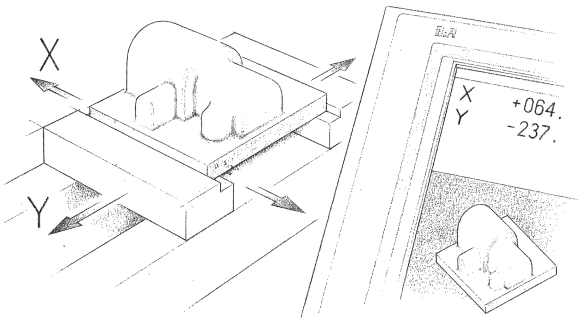
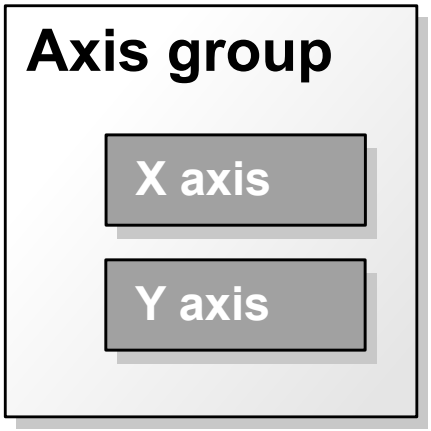


Figure 9: CNC 축 그룹 적용 : 공작 기계의 X / Y 테이블

무엇이 필요할까요?



다음 구성 요소가 필요합니다:

- 2 axes (X axis & Y axis)
- 1 axis group

이동을 시작하기 전에 축을 활성화하고 홈 동작이 완료되어야 합니다.

Figure 10: X, Y axis group

Exercise:

1) Automation Studio 프로젝트 만들기
2) 모션 구성 요소 삽입
3) CNC 프로그램 만들기
4) NC Test에서 작동

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Creating an axis group

" Commissioning and diagnostics"은 NC Test를 사용하여 축 그룹을 조작하는 방법에 대한 자세한 내용을 제공합니다. (Section 5.1 "NC Test" on page 13).

항상 parallel mode에서 NC Test를 실행하십시오!
NC Test가 exclusive mode에서 활성화 된 경우 축에 대한 액세스가 응용 프로그램에 의해 차단됩니다. Parallel mode에서는 NC Test 및 응용 프로그램을 통해 축의 병렬 제어가 가능합니다.

4.2 Axis group reference

Automation Studio에서 생성 된 축 그룹과 프로그래밍을 위한 function block 간의 관계를 설정하려면 각 축 그룹에 고유한 Axis group reference가 필요합니다.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Axis group reference

4.3 Axis configuration

그룹 내의 각 축의 동작을 정의해야 합니다. 이 목적을 위해 CNC init parameter table이 사용됩니다.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Axis configuration

4.4 Unit system

축 그룹의 단위 시스템은 축의 정확도가 상당히 높더라도 (예 : 축 해상도가 1 / 1000mm 일 수 있음) 대부분 밀리미터 또는 인치로 구성됩니다. 축 그룹의 축 단위는 PLCopen 단위에 해당합니다.
PLCopen units factor는 축 단위를 PLCopen 단위로 변환하는 데 사용됩니다.

드라이브상의 축 위치 = PLCopen factor * PLCopen position

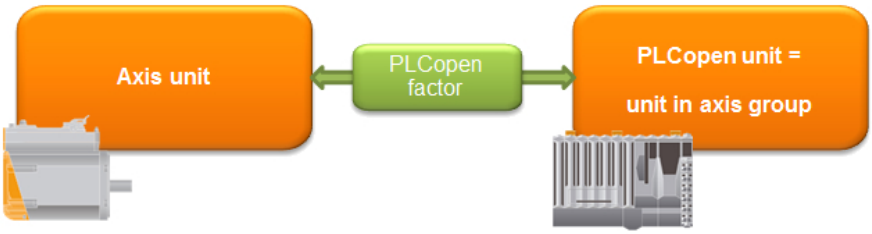


Figure 11: Unit system

응용 프로그램에서 사용 된 축 그룹 단위는 PLCopen unit factor를 사용하여 고해상도 축 단위로 변환됩니다.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Technical information \ Unit system

5 Commissioning and diagnostics

이 섹션에서는 Automation Studio에서 사용할 수 있는 기능을 사용하여 축 그룹을 단계적으로 작동시킬 수 있는 방법과 이러한 기능을 진단에 사용할 수 있는 방법을 설명합니다.

5.1 NC Test

NC Test를 사용하여 응용 프로그램 없이 축 그룹을 조작 할 수 있습니다. 축 그룹의 경로 제어 이동을 이 창에서 수행하고 진단 할 수 있습니다.

NC Test 기능

- Command 실행
- Parameter 수정
- Watch 기능
- Trace 기능
  - Cyclic trace (see ~~5.2.1 "Cyclic trace" on page 14~~)
  - Network command trace (see 5.2.2 "Network command trace" on page 14)

Figure 12: Commissioning and diagnostics

NC Test는 parallel mode 및 exclusive mode에서 작동 할 수 있습니다. Exclusive mode에서는 응용 프로그램에서 축 그룹 이동을 시작할 수 없습니다. 이렇게 하면 NC 테스트와 응용 프로그램 간의 간섭을 방지 할 수 있습니다.

매핑 테이블에서 축 그룹을 선택한 후 <열기 / 테스트>의 바로 가기 메뉴에서 NC Test를 호출 할 수 있습니다. NC Test를 시작해야하는 모드에 대한 요청이 발생합니다.

Motion \ Diagnostics \ NC Test

5.2 NC Trace

매핑 테이블에서 축 그룹을 선택한 후 <Open / Trace>의 바로 가기 메뉴에서 NC Trace를 호출 할 수 있습니다. 그러면 cyclic trace 데이터를 보여주는 창이 열립니다. Trace network command으로 전환하려면 NC Trace 창을 마우스 오른쪽 버튼으로 클릭하십시오.

Automation Studio에서 NC Trace는 NC Test 창의 하위 섹션으로 표시되지만 자체 창으로 별도로 열 수도 있습니다.

5.2.1 Cyclic trace

Cyclic Trace는 경로 및 축의 움직임 (예: 위치 및 속도)을 기록하고 분석하는 데 사용할 수 있습니다.

데이터가 컨트롤러에 기록됩니다. 기록이 완료되면 데이터를 Automation Studio에서 로드하여 표시 할 수 있습니다.

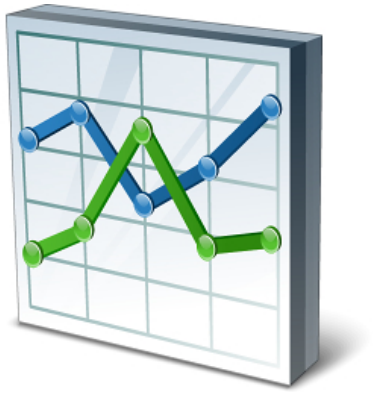


Figure 13: Cyclic trace

Motion \ Diagnostics \ NC Trace

5.2.2 Network command trace

Network command trace는 축 통신을 기록합니다. 또한 축 및 축 그룹으로 전송 된 명령, PLCopen 상태 변경 및 오류도 기록합니다. 이 기능은 코드에 의해 명령이 발행되는 순서를 분석하게 합니다. 따라서 오류를 빠르고 쉽게 찾을 수 있습니다.

데이터는 연속적으로 기록되며, 컨트롤러에서 Automation Studio로 로드하여 진단 목적으로 표시 할 수 있습니다.

Figure 14: Command trace

Motion \ Diagnostics \ Network command trace

6 mapp technology

mapp technology[1] 통해, 우리는 사용자에게 포괄적인 기능을 구현하기 위한 사용하기 쉬운 인터페이스를 제공합니다. 레시피 데이터의 로드 및 저장, 드라이브 축 제어 그리고 프로세스 데이터 기록과 같은 많은 복잡한 작업들은 사용하기 쉬운 mapp technology 구성 요소를 사용하여 수행됩니다.

Figure 15: mapp technology logo

mapp 기술은 환경설정 및 프로그래밍을 통합합니다. 기능은 표준 라이브러리를 사용하여 응용 프로그램에서 구현됩니다. 또한 mapp은 응용 프로그램 소프트웨어에서 해당 구현을 작성하지 않고도 mapp 구성 요소의 기능을 수행 할 수 있는 환경설정 인터페이스를 제공합니다.

Application layer - mapp technology

- Concept
- Getting started
- Components

6.1 mapp technology components 사용 지침

mapp component를 처음 사용할 때는 다음 단계를 수행해야 합니다.

- Configuration View로 이동
- 도구 상자에서 "mapp"기술 패키지를 추가하십시오.
- 도구 상자에서 사용 할 mapp component의 표준 설정을 추가하십시오.
- 필요에 따라 Configuration View에서 MpLink의 이름을 변경합니다.

Configuration View는 다음과 같습니다:

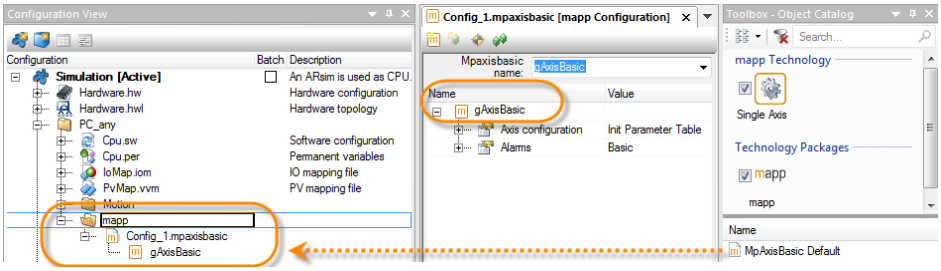


Figure 16: Configuration View에서 mapp technology package 및 MpAxisBasic 기능의 표준 설정

Configuration View에서 MpLink는 프로그램 내에서 ADR() function을 사용하여 function block으로 전달됩니다. 이것은 환경설정에서 mapp component의 프로그래밍 코드로의 연결을 설정합니다.

[1] mapp technology stands for "Modular APPlication technology".

Application layer - mapp technology \ Concept \ Component design \ Adding mapp components

mapp components 호출

mapp component function block은 모든 컨트롤러 사이클에서 호출되어야 합니다. 고급 프로그래밍 (Ansi C, ST 등)을 사용하는 경우, 프로그램 마지막에 모든 mapp 구성 요소를 호출하는 것이 좋습니다.

모든 mapp function block은 "Enable" 입력이 있습니다. 이 입력은 mapp 구성 요소를 사용 가능하게 하는 데 사용되며, 각 mapp 구성 요소의 설정이 자동으로 로드 됩니다. mapp component의 성공적인 초기화는 "Active = TRUE" 출력으로 표시됩니다.

Application layer - mapp technology \ Concept \ Component design \ Using mapp components

Download behavior

프로그램이 전송 될 때마다 Automation Studio의 표준 설정에 의해 컨트롤러는 다시 시작됩니다. 드라이브 응용 프로그램의 구현 단계에서는 "복사 모드" 전송 방법을 사용하는 것이 좋습니다. 그렇게함으로써 컨트롤러가 몇 번씩 다시 시작할 필요가 없습니다.
전송 유형에 대한 설정은 Configuration View의 활성 설정의 바로 가기 메뉴를 사용하여 열 수 있습니다. 전송 방법은 고급 단추를 사용하여 전송 탭에서 설정합니다.

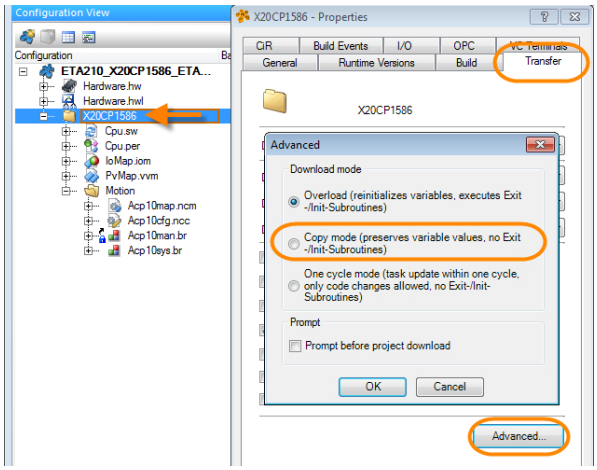


Figure 17: Configuration of the transfer method in the Properties window in Configuration View

Application layer - mapp technology \ Concept \ Component design \ Using mapp components

Real-time operating system \ Target systems \ SG4 \ Download

Configuration Files

모든 mapp component는 사용 가능한 설정이 있습니다. 환경설정은 Automation Studio의 Configuration View, WebXs 웹 기반 인터페이스 또는 응용 프로그램에서 생성되고 수정됩니다. mapp 설정에 대한 추가 정보는 Automation Studio 도움말 시스템에서 찾을 수 있습니다.

Application layer - mapp technology \ mapp \ Concept

- Component design \ Adding mapp components
- Configuring components

6.2 Diagnostic options for mapp technology components

mapp technology components 여러 가지 다른 방법을 통해 모니터링하고 진단 할 수 있습니다. 다음은 Automation Studio, 웹 기반 진단 및 시각화 응용 프로그램의 진단 옵션 목록입니다.

Programming languages in monitor mode

대부분의 경우 첫 번째 액세스는 응용 프로그램 소프트웨어 프로그래밍 중 모니터 모드입니다. 프로세스 변수의 값은 프로그램 코드와 직접적으로 관계가 있습니다.
모든 mapp technology components 에는 초기 진단을 수행하는 데 사용할 수 있는 "Error"및 "Status ID" 출력이 있습니다.

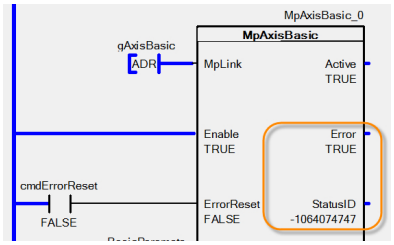


Figure 18: Ladder Diagram program in monitor mode

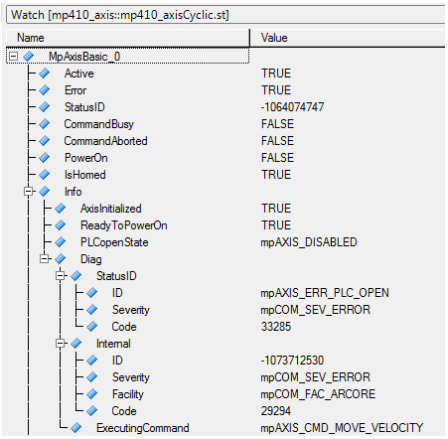


Figure 19: Instance variable of the MpAxisBasic function block in the Watch window

Watch window

Watch window는 Logical View에서 프로그램 바로가기 메뉴를 사용하여 열거나, 소프트웨어 설정에서 **<Watch>**를 선택하여 열 수 있습니다.
사용하는 function block의 인스턴스 변수는 툴바 또는 바로가기 메뉴를 사용하여 추가됩니다. 예를 들어, Error, StatusID, CommandBusy 출력 및 정보 구조는 현재 상태를 진단하는 데 도움이 될 수 있습니다. 이러한 파라메타 및 오류 번호에 대한 설명은 해당 function block의 설명에서 읽을 수 있습니다.

Logger

오류가 발생하면 mapp technology 구성 요소의 추가정보가 "$mapp"라는 이름의 logger file에 추가됩니다. 오류 번호는 AutomationStudio 도움말 시스템에서 직접 검색하거나 **<F1>** 키를 눌러 호출 할 수 있습니다. 자세한 정보는 강조 표시된 logger 항목의 세부 정보 섹션에서 확인할 수 있습니다. 예를 들어, PLCopen 오류가 있는 경우 영향을 받는 기능과 오류의 원인은 logger 항목의 세부 정보 섹션에 명확하게 설명되어 있습니다.

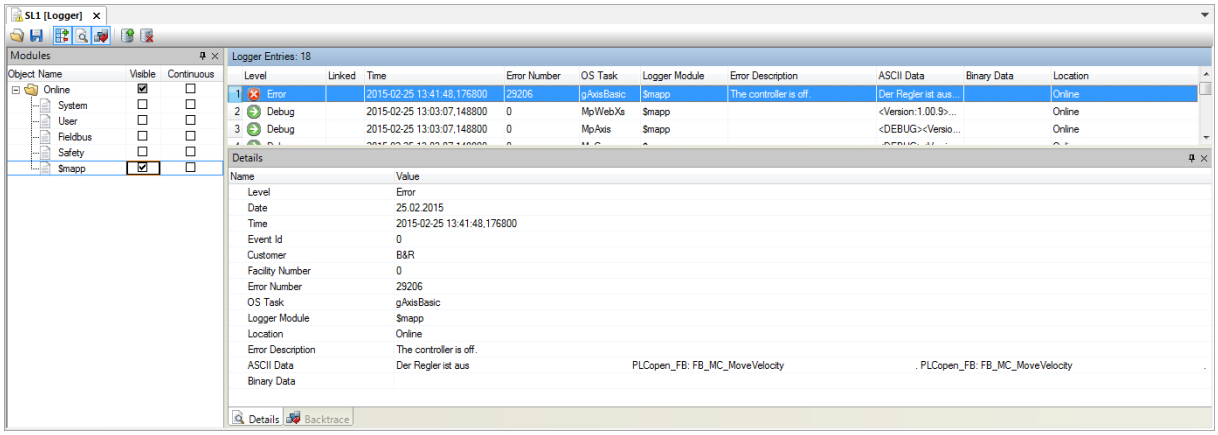


Figure 20: PLCopen error in the logger window

Trace

Automation Studio에서 trace 기능을 사용하면, 프로세스 변수의 값이 실시간으로 기록되고 저장됩니다. 무엇보다 입력 파라메타와 상태 변수의 타이밍을 시각화 할 수 있습니다.

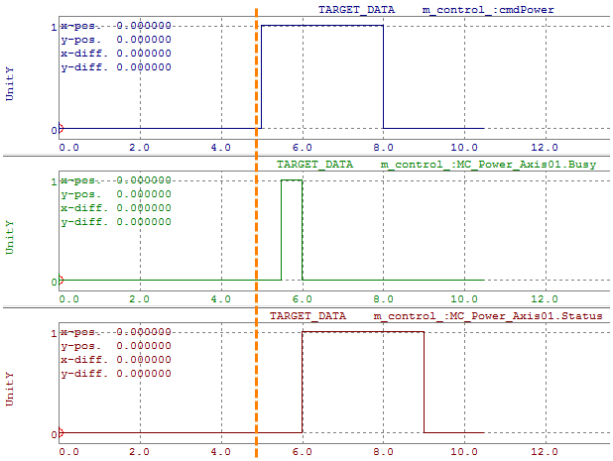


Figure 21: Switching on the controller with cmdPower: the time relationship between commands and status information

System Diagnostics Manager

"Application Status" 버튼은 System Diagnostics Manager를 사용하여 호출 할 수 있습니다. 이것은his directly ope for mapp technology를 위한 WebXs를 직접 엽니다. 또한, SDM에서는 mapp components에 대한 logger 데이터 저장이 가능합니다. 그리고 사용 가능한 드라이브 축에 대한 기본 진단도 있습니다. SDM은 HTML 컨트롤을 사용하여 Visual Components 응용 프로그램에 직접 포함시킬 수 있습니다.

mapp WebXs

mapp technology WebXs를 사용하여 실해되는 모든 mapp components 웹 기반 인터페이스에 표시됩니다. mapp components의 인스턴스 변수를 통한 진단 외에도, components 설정과 alarms 또한 제공됩니다[2].

Integration of Visual Components using the MpAlarm component

mapp technology component는 사전 정의된 alarm이 있습니다. 사용자 별 alarm도 구성 할 수 있습니다. 비주얼 컴포넌트 alarm 시스템의 mapp Alarm 출력은 MpAlarm 구성요소를 사용하여 활성화됩니다.

2 웹 기반 구성을 수행 할 수 있는지 여부는 사용 된 구성 요소에 따라 다릅니다.

Integration of Visual Components using the MpComLoggerUI component

mapp technology의 이벤트 관리 기능은 모든 이벤트를 logger에 저장합니다. 이러한 logger 항목은 MpComLoggerUI 구성 요소를 사용하여 Visual Components에 쉽게 통합 될 수 있습니다. 필터 기능을 사용하면 개별 mapp components, 특정 error numbers 또는 event types을 검색 할 수 있습니다. 따라서 mapp logger 항목을 필터링하는 데 추가 프로그래밍이 필요하지 않습니다.

Diagnostics and service \ Diagnostic tool \

- Logger
- Watch window
- Monitors \ programming languages in monitor mode
- Trace
- System Diagnostics Manager

Application layer - mapp technology \

- WebXs
- Components \ Infrastructure \
  - MpAlarm - Support for alarm management
  - MpCom - mapp management \ function blocks \ MpComLoggerUI
- Diagnostics \ Logger window

7 Integrating an axis group in the control program

기계 유형에 따라 mapp으로 축 그룹을 제어하기 위한 여러 가지 Component가 있습니다. 다양한 축 수를 가진 CNC 시스템부터 다른 유형의 로봇까지 다양합니다.

Figure 22: mapp components for controlling axis groups

다음 장에서는 축 그룹을 제어하는 응용 프로그램을 작성하는 방법을 단계별로 보여줍니다.

MpCnc2Axis mapp component는 2축 CNC 기계에 사용됩니다. MpAxisBasic mapp function block 으로 작동하는 두 개의 축이 MpCnc2Axis에 기본적으로 필요합니다.

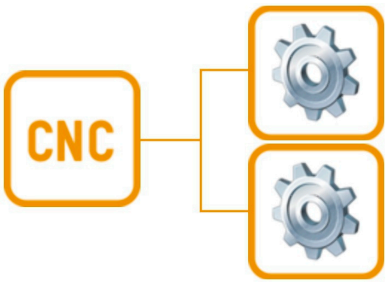


Figure 23: mapp 2-axis CNC

Task: Creating mapp-control of the axes of an X/Y-CNC machine using MpAxisBasic

1. Go to the Configuration View
2. Add mapp technology MpAxisBasic component from the toolbox for the X and Y axes
3. Add the new "cnc_ctrl" Ladder Diagram program
4. Add MpAxisBasic function blocks
5. Connect the "MpLink", "Enable", "Axis" and "Parameters" inputs
6. Transfer project and test axes by switching on "Power"

7.1 The MpCnc2Axis component

MpCnc2Axis mapp technology component는 2축 CNC 기계를 제어, 구성 및 진단하기 위한 표준 기능을 제공합니다.
CNC 기계의 다음과 같은 기본 기능은 MpCnc2Axis function block에 포함됩니다:

- Group preparation
- CNC movements
- Error handling
- Jog mode
- Single-step operation

Figure 24: mapp CNC

MpCnc 라이브러리는 MpDelta, MpRoboArm 및 MpScara 라이브러리와 마찬가지로 GmcGrpAPI 라이브러리의 함수 블록을 기반으로 합니다. 따라서 두 라이브러리는 서로 호환 가능하며 응용 프로그램에서 공동으로 구현할 수 있습니다.

모든 Mapp component는 애플리케이션에서 사용자가 직접 사용할 수 있는 개방형 표준, 기술 기능 및 라이브러리를 기반으로 합니다.

사용 된 기능 블록 및 기능에 관한 세부 사항은 Automation Studio 도움말 시스템의 해당 섹션에서 가져올 수 있습니다.

Figure 25: mapp components are based on open standards, technology functions and libraries

7.2 Creating a program and adding MpCnc2Axis

이제 MpCnc2Axis function block을 호출하여 Automation Studio 프로젝트를 확장해야 합니다.
준비 단계에서 Configuration View의 프로젝트는 mapp technology package를 추가하여 확장됩니다. 다음으로 MpCnc2Axis component의 표준 설정이 Configuration View에 추가됩니다. MpCnc 라이브러리가 Logical View에 추가됩니다.

마지막으로, MpCnc2Axis function block이 프로그램에 추가되고 Configuration View에 이미 설정된 MpLink에 연결됩니다.

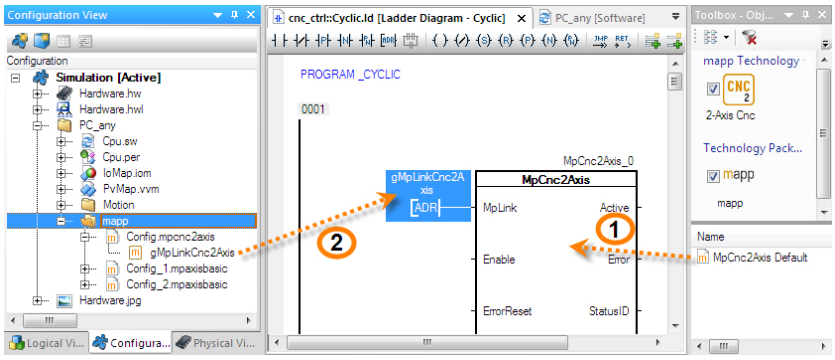


Figure 26: (1) Add MpCnc2Axis from the toolbox and (2) transfer the MpLink address from the Configuration View

Application layer - mapp technology \ Concept \ Component design \ Adding mapp components

Exercise: Adding the mapp technology configuration for MpCnc2Axis

이제 mapp technology package를 추가하여 Automation Studio 프로젝트를 확장 한 다음 MpCnc2Axis Configuration View의 표준 설정을 삽입하십시오. MpCnc 라이브러리를 가져온 다음 MpCnc2Axis function block을 cnc_ctrl 프로그램에 추가 한 다음 ADR() 함수를 사용하여 Configuration View에서의 MpLink를 연결합니다.

1) Go to the Configuration View.
2) Add the mapp technology package from the toolbox.
3) Add the MpCnc2Axis standard configuration in the Configuration View.
4) Add the MpCnc library in the Logical View.
5) Add MpCnc function block from the toolbox into the "cnc_ctrl" into the Ladder Diagram program.
6) Assign MpLink from the Configuration View to the MpCnc2Axis function block using the ADR() function.

7.3 Connecting the axis group reference and the movement parameters

축 그룹에 액세스하고 동작을 수행 할 수 있도록 axis group reference와 이동 파라메타를 전송해야 합니다.

Using the axis group reference

MCAxesGroupType_Arnc type의 전역 프로세스 변수가 설정됩니다. 프로세스 변수의 이름은 NC mapping table의 그룹에 이름과 일치해야 합니다. 이러한 방식으로 축 그룹과 응용 프로그램 간의 연결을 생성 할 수 있습니다.

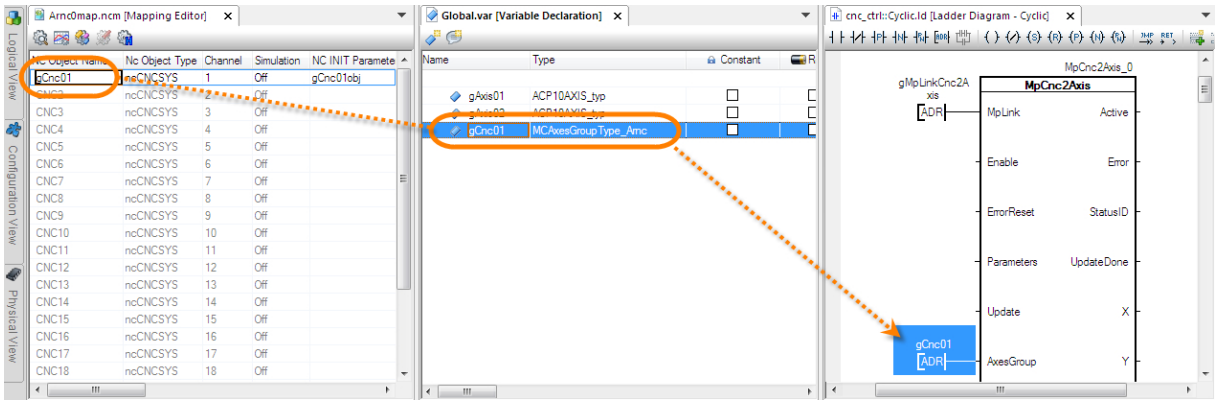


Figure 27: Connecting the axis group reference

Axis reference 주소는 "AxesGroup" 파라미타를 사용하여 모든 function block에 지정됩니다.래더 다이어그램에서 주소 기능(ADR) 또는 접근 주소 중 하나를 사용할 수 있습니다. 축 그룹으로 액세스는 다른 모든 function block과 동일한 방식으로 발생합니다.

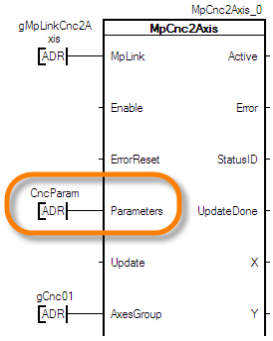


Figure 28: Parameter data structure transfer (MpCnc2AxisParType)

Transferring CNC parameters

MpCnc2Axis function block의 경우 파라메타가 있는 데이터 구조를 전송해야 합니다. 데이터 구조에는 표준 값이 사전 설치되어 있습니다. 다음 파라메타를 전송할 수 있습니다.:

- CNC program name
- Initialization subroutine
- CNC single-step
- Jog parameter
- Tool parameters

고려해야 할 사항은 변수 선언에 데이터 구조를 만들 때 데이터 구조의 표준 값을 받기 위해 초기화 값을 입력 할 수 없다는 것입니다.

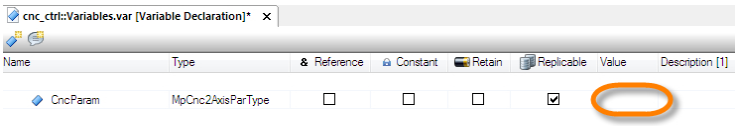


Figure 29: Empty "Value" field to initialize the structure with standard values

Application layer - mapp technology \ Components \ Mechatronics \ MpCnc - CNC system controller \ Function blocks \ MpCnc2Axis

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Axis group reference

Exercise: Assigning axis group reference and assign parameters

이제 기존 프로그램에서 확장되어 axis group reference 및 파라메타수가 있는 데이터 구조가 MpCnc2Axis function block으로 전송 될 수 있습니다.

1) ADR() function을 사용하여 axis group reference를 "AxesGroup" 입력에 지정합니다..

2) "CncParam" 구조를 "Parameters" 입력과 연결하십시오..

"CncParam" 구조는 MpCnc2AxisParType 데이터 형식을 기반으로 합니다. 기본값으로 사전에 초기화됩니다.

7.4 mapp hierarchy

축 그룹을 성공적으로 작동 시키려면 MpCnc2Axis 및 MpAxisBasic mapp component 사이에 계층을 만들어야 합니다. CNC와 축의 연결은 MpCom 라이브러리의 Mapp MpComLinkToParent function block을 사용하여 생성됩니다.

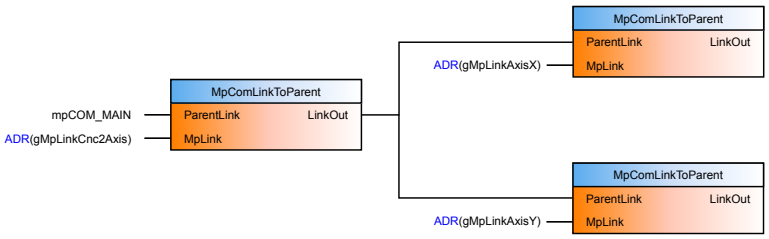


Figure 30: mapp hierarchy with MpComLinkToParent

MpComLinkToParent function block은 응용 프로그램 초기화 중에 호출되어야 합니다.

Application layer - mapp technology \ Components \ Infrastructure \ MpCom \ mapp management \ MpComLinkToParent

Application layer - mapp technology \ Components \ Mechatronics \ MpCnc - controller of a CNC system \ Technical information \ CNC hierarchy concept

Create mapp link hierarchy and switch on axis group with MpCnc2Axis

1. MpComLinkToParent 함수 블록을 사용하여 "cnc_ctl"프로그램의 초기화 부분에 mapp 계층 구조를 추가합니다.
2. 프로그램을 컨트롤러로 전송하십시오
3. MpAxisBasic mapp function block의 "Enable"입력을 TRUE로 설정합니다.
4. "Active"출력과 "Info.ReadyToPowerOn"출력이 TRUE가 될 때까지 기다리십시오.
5. MpCnc2Axis의 "Enable"입력을 TRUE로 설정하십시오.
6. MpCnc2Axis 기능 블록에서 "Power"를 통해 축 그룹을 켭니다.
7. MpCnc2Axis 및 MpAxisBasic의 상태 출력을 관찰합니다.

7.5 Function block operation and status evaluation

모든 mapp function block은 균일한 조작 파라메타를 사용하여 액세스되며 균일한 상태 파라메타를 리턴합니다. 이를 통해 응용 프로그램이 단순화되고 프로그래밍 중 명료성이 추가됩니다.

Application layer - mapp technology \ Concept \ Component design \ Inputs and outputs

Timing diagrams

Automation Studio 도움말 시스템의 타이밍 다이어그램은 다른 입력 상태에서 function block이 어떻게 작동하는지 보여줍니다. 도움말 시스템은 구성 요소의 기능을 설명하고 응용 프로그램을 올바르게 구현하는 데 도움을 줍니다.

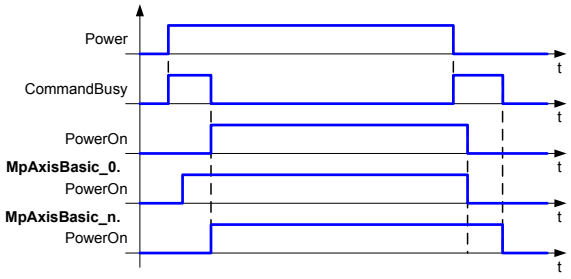


Figure 31: Input "Power" timing diagram at MpCnc2Axis

Application layer - mapp technology \ Components \ Mechatronics \ MpCnc - controller of a CNC system \ Function blocks \ MpCnc2Axis \ Timing diagrams

Status information

오류가 발생하면 "오류" 출력의 값은 TRUE가 됩니다. "StatusID" 출력에는 Automation Studio 도움말 시스템에서 검색 할 수 있는 숫자 정보가 들어 있습니다. 현재 상태에 대한 추가 정보는 "info" 출력 구조에서 확인할 수 있습니다. 이것은 브라우저에서 WebXs를 사용할 때 표시 될 수 있습니다. 오류는 Logger에도 입력됩니다. 이것은 Automation Studio 또는 System Diagnostics Manager에 선택적으로 표시 될 수 있습니다.

아래 이미지는 WebXs에서 확장된 "info" 구조가 있는 MpCnc2Axis component를 보여줍니다. "Error" 출력은 TRUE이고 값은 "StatusID" 출력에 주어집니다. "info" 구조체는 축 그룹 오류가 발생했음을 나타냅니다. 오류 코드는 15131입니다. Automation Studio 도움말 시스템에서 이 오류 번호를 찾으면 설명을 확인할 수 있습니다. 이 경우 오류 설명은 존재하지 않는 NC 프로그램을 시작하려고 시도했음을 나타냅니다.

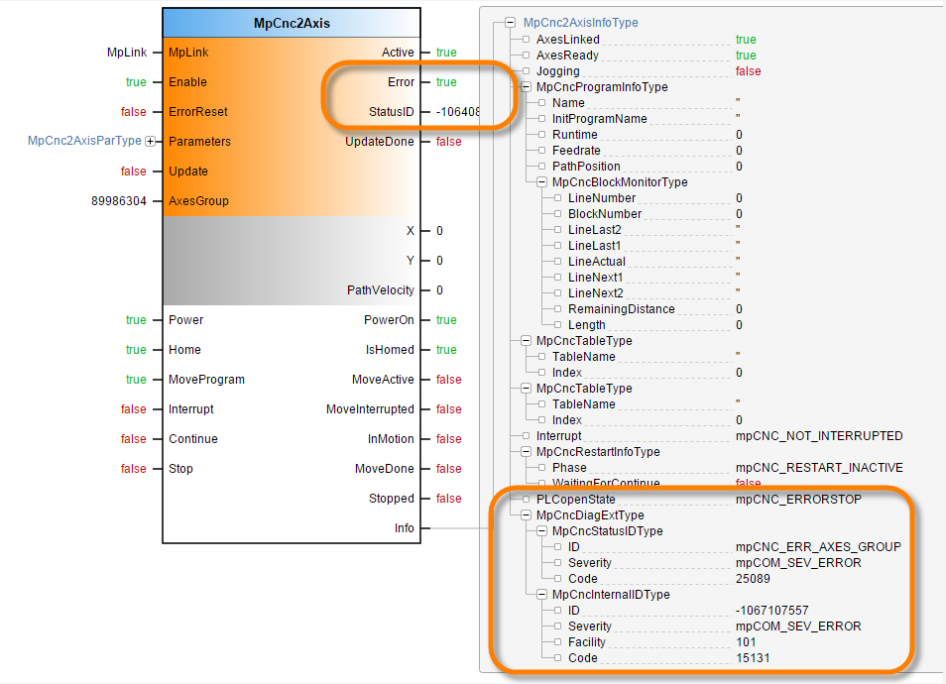


Figure 32: Comprehensive "info" structure - Representation in WebXs

Automation Studio logger에서 "$ mapp"및 "GMC"라는 항목도 생성됩니다.
오류의 원인을 식별하기 위한 오류 번호 외에 추가 세부 사항을 읽을 수 있습니다.

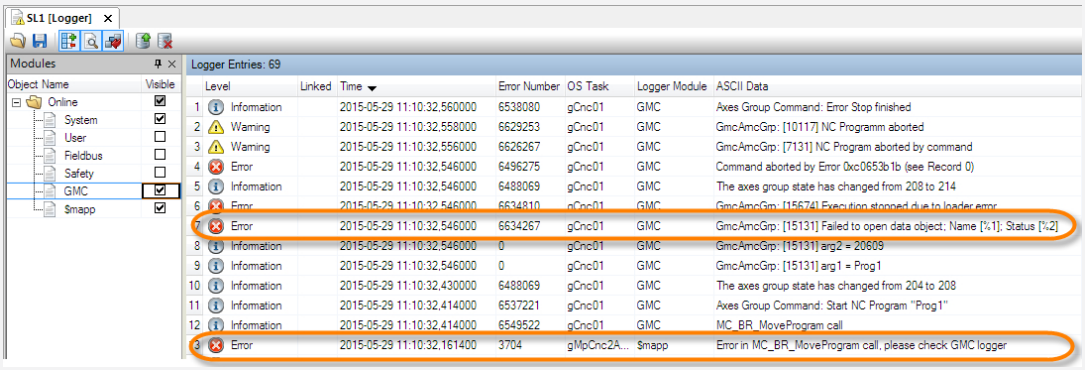


Figure 33: "$mapp" and "GMC" logger modules

오류는 "ErrorReset" 입력을 통해 확인 될 수 있습니다. 오류가 수정 된 후, "MoveProgram"입력에서 새로운 상승 에지를 통해 프로그램을 시작할 수 있습니다.

8 Axis group states

PLCopen 상태는 축 그룹을 조작하는 데 사용됩니다. 이러한 상태는 복잡한 이동 절차의 개요를 단순화하고 오류 상황을 보다 쉽게 처리하도록 합니다.

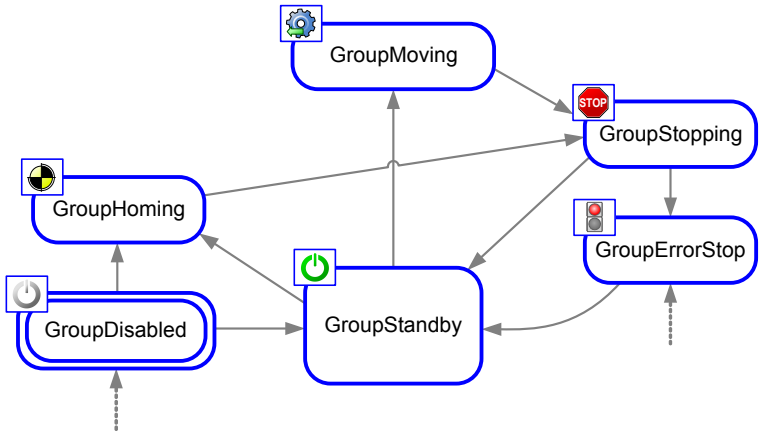


Figure 34: PLCopen axis group state diagram

이러한 상태 사이의 전환은 특정 function block을 호출하여 초기화 할 수 있습니다. GroupErrorStop 상태는 그룹의 축 중 하나에 오류가 발생하면 다른 모든 상태에서 입력 될 수 있습니다.

State	Description
GroupDisabled	그룹이 비활성화되었습니다. 축 그룹 이동을 수행 할 수 없습니다.
GroupStandby	그룹이 켜져 있지만 이동이 수행되고 있지 않습니다.
GroupHoming	원점 복귀는 전체 그룹에 대해 활성화됩니다.
GroupMoving	그룹의 하나 이상의 축이 이동 중입니다. 이 이동은 그룹 이동 명령 또는 단일 축 이동 명령으로 인해 발생할 수 있습니다.
GroupErrorStop	그룹이 오류 상태입니다. 동작이 중지됩니다.
GroupStopping	그룹이 중지되었습니다.

Table 3: PLCopen axis group states

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Technical information \ State diagram

축 그룹의 현재 상태는 MpCnc2Axis 기능 블록의 "info" 구조의 "PLCopenState" 요소를 사용하여 읽을 수 있습니다.

Exercise: Command execution and condition monitoring

다른 명령을 실행하고 상태 출력과 축 그룹 상태를 항상 확인하십시오.

1 MpWebXs 라이브러리를 추가하고 프로젝트를 전송하십시오.
2 WebXs뿐만 아니라 Watch 창에서 "info" 구조를 확인하십시오. .
3 축 그룹을 켜고 축 그룹의 상태를 확인합니다.
4 "Home" 명령으로 축 그룹을 홈에 넣고 "IsHomed" 상태를 확인합니다. 그런 다음 명령을 재설정할 수 있습니다..
5 "Override" 입력을 100% 값으로 설정하십시오..
6 MpCnc2Axis 기능 블록의 파라미타 구조에서 "ProgramName" 아래에 프로그램 이름을 입력하고 "MoveProgram" 명령을 사용하여 CNC 프로그램을 시작합니다.
7 MpCnc2Axis function block의 상태 출력과 축 위치를 확인합니다.
8 "Interrupt", "Continue"및 "Stop"과 같은 추가 명령을 사용하고, MpCnc2Axis function blockd의 출력 동작을 관찰합니다.

Exercise: Generating and deleting axis group errors

존재하지 않는 CNC 프로그램을 시작하고 상태 출력을 관찰하십시오. 그런 다음 "ErrorReset" 입력을 통해 오류를 삭제하십시오.

1 MpCnc2Axis 기능 블록의 파라미터 구조에서 "ProgramName" 아래에 유효하지 않은 프로그램 이름을 입력하고 "MoveProgram" 명령을 사용하여 CNC 프로그램을 시작하십시오.
2 "Error"출력과 "StatusID"를 확인하십시오.
3 "info" 구조와 logger 항목을 확인하십시오..
4 "ErrorReset" 입력을 통해 오류를 확인하십시오..
5 "Error" 와 "StatusID" 출력 그리고 "info" 구조를 점검하십시오.

Exercise: Single-step operation

1 "SingleStep" 입력을 사용하여 단일 단계 작동을 활성화하고 "SingleStepActive" 출력을 확인하십시오.
2. "Continue", "MovementActive" 및 "MoveInterrupted" 변수를 플롯하도록 variable trace을 구성합니다.
3 Trace를 시작한 다음 CNC 프로그램을 시작합니다..
4 "MovementInterrupted"가 TRUE로 전환 된 후 Trace를 중지하고 레코드를 분석합니다.
5 "info" 구조의 "BlockMonitor" 및 "Interrupt" 값을 관찰하십시오.
6 "Continue"를 사용하여 CNC 프로그램에서 다음 동작을 시작하십시오.

9 PLCopen axis group library

GmcGrpAPI 라이브러리의 function block은 mapp technology와 함께 축 그룹을 제어하기 위한 기초로 사용됩니다. 이 라이브러리의 function block에 대한 인터페이스는 PLCopen의 정의를 기반으로 합니다. 이 섹션에서는 PLCopen이 무엇이고, mapp과 함께 이 블록을 사용함으로 사용자에게 제공되는 이점에 대해 설명합니다.

9.1 PLCopen library

축 그룹을 제어하기 위해 B&R에서 제공하는 PLCopen function block은 PLCopen 모션 제어 파트 4 표준에 지정된 기능을 기반으로 합니다. 축 그룹의 모든 요구 사항을 충족하도록 기능을 확장하기 위해 표준에 지정된 function block 외에도 B&R 관련 function block이 제공됩니다.

Function block이 PLCopen 표준 function block 인지 B&R 특정 function block 인지는 이름으로 표시됩니다. 표준 function block은 항상 이름 시작 부분에 접두어 "MC_"를 포함합니다. e.g. MC_MoveDirectAbsolute(). B&R 관련 function block 및 확장 이름은 "MC_BR_"로 시작합니다. e.g. MC_BR_GroupPower().

Figure 35: PLCopen @ B&R

축 그룹을 제어하기 위한 function block은 GmcGrpAPI 라이브러리에서 찾을 수 있습니다.

> Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI

9.2 Using the function blocks

이 절에서는 PLCopen function block 사용에 대한 기본 정보를 제공합니다. Function block 을 작동하는 방법과 작동을 모니터링 하는데 사용할 수 있는 옵션에 대해 살펴 보겠습니다.

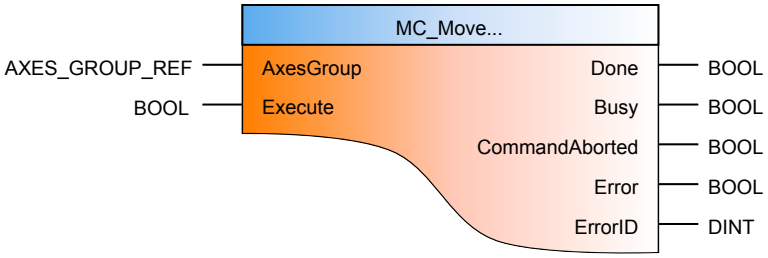


Figure 36: Standard function block parameters

모든 function block은 일정한 작동 방법과 상태 파라메타를 사용하여 액세스 됩니다. 이를 통해 응용 프로그램이 단순화되고 프로그래밍은 명료성이 추가됩니다.

Axis group reference

Axis group reference는 축 그룹과 function block 간의 링크를 설정합니다.

Parameter	Description
AxesGroup	Function block이 사용해야 하는 축 그룹을 지정합니다.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Axis group reference

Begin execution

Function block에는 두 가지 기본 유형이 있습니다. function block은 Enable 입력과 Execute 입력이 있습니다.

Parameter	Description
Enable	Function block은 Enable이 설정되어 있는 동안 활성화됩니다. Enable이 Reset 되면 function block은 비활성화됩니다.
Execute	Function block은 Execute input으로 상승 에지가 수신되면 동작을 시작합니다. Execute input이 Reset 될 때 아무런 작업도 시작되지 않습니다.

Status outputs

Function block에는 다음과 같은 상태 출력이 있습니다. 이 출력 중 하나만 설정될 수 있습니다.

Parameter	Description
Busy	Function block이 활성화되어 계속 호출되어야합니다.
Done	작업이 성공적으로 완료되었습니다.
CommandAborted	다른 function block 호출에 의해 작업이 중단되었습니다.
Error	오류로 인해 작업이 중단되었습니다.

Error information

Error 상태 출력이 설정되면 ErrorID 출력에 오류 번호가 제공됩니다.

Parameter	Description
ErrorID	오류의 원인에 대한 정보. 가능한 오류 번호 목록은 Automation Studio 도움말 시스템에서 찾을 수 있습니다.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ *Name of the function block* \ Error numbers

Done, Aborted, Error 및 ErrorID 출력의 값은 Execute 또는 Enable 입력이 Reset 될 때까지 동일하게 유지됩니다.

이러한 신호가 도착하기 전에 Execute input이 이미 비활성 상태 인 경우 이 출력은 한 사이클 동안만 설정됩니다.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Technical information \ Function block interface

9.3 Function blocks

PLCopen function block은 관리 function block과 비관리 function block으로 나눌 수 있습니다. 관리 function block은 호출 될 때 이동을 발생시키지 않습니다. 비관리 function block은 호출 될 때 동작을 시작합니다.

축 그룹으로 작업 할 때 사용되는 가장 중요한 function block이 이 섹션에 나열되어 있습니다.

나열된 기능 블록의 대부분은 MpCnc2Axis mapp component 내에서 사용됩니다. 필요한 기능이 mapp component에 매핑되어 있지 않으면 PLCopen function block을 MpCnc2Axis와 쉽게 결합 시킬 수 있습니다.

Figure 37: PLCopen Motion Control logo

Preparing the group

다음 표는 축 그룹을 준비하는 데 사용되는 가장 중요한 기능 블록을 보여줍니다.

Function block	Description
MC_BR_GroupPower()	Switches on the axis group and the axis controllers
MC_GroupHome()	Homes the axes

Table 4: Important function blocks used for preparation

Executing movements

다음 표는 이동을 실행하는 데 사용되는 가장 중요한 기능 블록을 보여줍니다.

Function block	Description
MC_MoveDirectAbsolute_15()	Performs a direct movement to a defined end position
MC_MoveDirectRelative_15()	Performs a direct movement over a defined distance
MC_MoveLinearAbsolute_15()	Performs a linear movement to a defined end position
MC_MoveLinearRealtive_15()	Performs a linear movement over a defined distance
MC_BR_MoveBlock()	Executes a single interpretable line (block)
MC_BR_MoveProgram()	Executes an interpretable program (CNC program)
MC_GroupStop()	Stops an active movement

Table 5: Important function blocks used to execute movements

Error handling

다음 표는 오류를 처리하는 데 사용되는 가장 중요한 기능 블록을 보여줍니다.

Function block	Description
MC_GroupReadStatus()	Reads the state of the group
MC_GroupReset()	Takes the axis group out of the GroupErrorStop and state and acknowledges all errors
MC_GroupReadError()	Reads the current group error and acknowledges it with a command.

Table 6: Important function blocks used for error handling

사용 가능한 모든 function block의 개요는 Automation Studio 도움말 시스템에서 찾을 수 있습니다.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Function blocks

10 Programming

응용 프로그램은 축 그룹을 제어하기 위한 자동 순서를 설정해야 합니다. 시퀀스는 운영자의 작업에 의해 시작됩니다. 오류는 또한 고려되어야 하며 적절하게 응답해야 합니다.

이러한 요구 사항을 구현하는 데는 여러 가지 가능성이 있습니다. 한편으로는 서로 다른 프로그래밍 언어를 선택할 수 있으며, 다른 하나는 소프트웨어를 구조화하는 다양한 접근법이 있습니다.

Figure 38: A more in-depth look at the application program

10.1 Automating tasks

MpCnc2Axis function block은 축 그룹을 제어할 수 있는 기능을 제공합니다. 프로세스는 명령을 사용하여 프로그램에서 정의된 지점에서 시작할 수 있습니다. Function block의 상태 출력 및 출력 파라메타는 축 그룹의 현재 상태에 대한 정보를 제공합니다.

특정 시퀀스가 자동으로 실행되는 프로그램을 생성하기 위해 상태 출력을 정보 소스로 사용하여 어떤 추가 명령을 실행할지 결정할 수 있습니다. 이는 변수 또는 조건의 논리적 링크를 통해 달성 될 수 있습니다. Automation Studio 도움말 시스템의 타이밍 다이어그램은 올바른 프로그래밍에 대한 지원을 제공합니다.

Application layer - mapp technology \ Components \ Mechatronics \ MpCnc - controller of a CNC system \ Function blocks \ MpCnc2Axis \ Timing diagrams

Exercise: Automatic homing after switching on the axis group

기존 프로그램은 이제 자동화 되어야 합니다. MpCnc2Axis 및 MpAxisBasic component는 컨트롤러를 켠 직후에 활성화 되어야 합니다. 또한 축 그룹이 아직 홈 위치에 있지 않은 경우 프로그램을 성공적으로 켠 후에 홈 지정 절차를 시작해야 합니다.

1. 시작한 후 MpCnc2Axis를 자동으로 구성하고 활성화해야 합니다.
2. "PowerOn" 출력을 설정 한 후 원점 복귀 절차를 시작해야 합니다.
3. 축 그룹이 홈 위치가 아닌 경우만, 원점 복귀 절차를 시작해야 합니다
4. 원점 복귀 동작이 성공적으로 실행되었거나, 오류가 발생하면 "Home" 명령은 Reset 되어야 합니다.

10.2 Uses of control structures

이제 우수한 기계 논리가 추가 프로그램에서 작성되어야 합니다. 여기서 사용자에 의한 동작은 축 그룹에서 특정 명령을 트리거하고 기계 상태들을 전환해야 합니다.

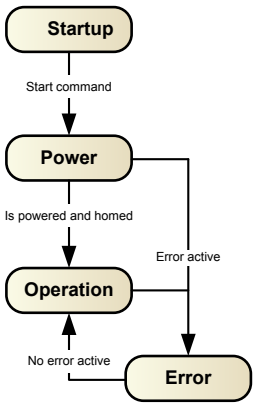


Figure 39: Control structure diagram

MpCnc2Axis의 다음 정보를 사용하여 시퀀스를 제어 할 수 있습니다:

- 명령 실행이 성공적이었습니까?
- 그렇지 않은 경우 어떤 오류가 발생 했습니까?
- 프로세스 상태:
  - 축 그룹이 움직입니까?
  - 축 그룹이 목적지에 도착 했습니까?
  - 원점 복귀가 성공적으로 수행 되었습니까?

이 정보는 프로그램 순서의 단계를 제어하는 데 사용될 수 있습니다. 프로그램은 오류 발생 여부에 따라 다르게 응답해야합니다.

단계 시퀀서는 이러한 유형의 기능 순서를 관리하는 데 특히 적합한 제어 구조입니다.

이 유형의 구조는 단계 색인을 사용하여 순서가 결정될 수있는 개별 단계의 구현을 허용합니다.

필요한 명령은 제어 구조의 개별 단계에서 실행할 수 있습니다. Error, StatusID 및 "info" 구조와 같은 상태 파라메타를 사용하여 응용 프로그램을 계속 진행할 단계를 결정할 수 있습니다.

이렇게하면 응용 프로그램의 구조가 명확해지고 향후 확장을 위해 기회가 열립니다.

Exercise: Creating machine logic

작동 상태가 포함된 기계 논리를 새 프로그램에서 작성해야 합니다. MpCnc2Axis function blokck에 액세스 할 수 있도록 전역 변수로 선언해야 합니다. 명령을 사용하여 그룹을 켜야 합니다. 성공적으로 켜지고 홈으로 전환 된 후, 로직은 "작동"상태로 전환됩니다. 이 상태에서만 운전 시작을 위한 추가 명령이 내려질 수 있습니다. 그룹은 스위치 오프 되고 시작 명령을 재설정하여 "시작" 상태로 전환됩니다..

1 새로운 "machine" 프로그램 만들기
2 상태 머신 생성 및 필요한 단계 추가 (고급 프로그래밍 언어 사용시 선택 사항)
3 전역 변수로 "cnc_ctrl" 프로그램의 MpCnc2Axis function block 만들기
4 자동 시퀀스는 "Info.ReadyToPowerOn"= TRUE 인 경우 "cmdOn"명령에 의해 활성화됩니다.
5 "Power"를 설정하고 "PowerOn" 출력 = TRUE 및 "IsHomed" = TRUE가 될 때까지 기다립니다.
6 "Operation"상태에서 추가 명령 전송
7 "cmdOn"= FALSE이면 그룹을 끄고, "Start" 상태로 전환하십시오.
8 동작은 "Operation"상태에서 "cmdMoveProgram"을 사용하여 시작할 수 있습니다.

10.3 Error handling

응용 프로그램을 프로그래밍 할 때 프로그램 순서에 오류 평가를 포함시키는 것이 항상 중요합니다. 다음 오류가 발생할 수 있습니다.

- Error calling a function block
- Axis error
- Group error

오류 처리는 "Error" 및 "StatusID" 상태 출력과 "info"구조를 사용하여 응용 프로그램에서 프로그래밍 할 수 있습니다. 오류가 발생하면 기능 블록의 실행이 중단됩니다. 그런 다음 Automation Studio 도움말 시스템의 오류 메시지 설명을 고려하여 오류를 수정해야 합니다. "ErrorReset" 입력을 사용하면 MpCnc2Axis를 다시 작동 상태로 만들 수 있으며 추가 명령을 실행할 수 있습니다.

Figure 40: Error handling in the application program

Logger 및 비주얼 컴포넌트 경보 시스템은 오류를 표시하는 데 사용할 수 있습니다.

Logger

오류 원인에 대한 자세한 데이터는 자동으로 "$ mapp"및 "GMC" logger 파일에 입력됩니다. Logger 파일은 Automation Studio 및 System Diagnostics Manager를 사용하여 열 수 있으며 PC에 저장할 수 있습니다. 또는 "AsArLog"라이브러리를 사용하여 logger 파일을 읽을 수 있습니다.

비주얼 컴포넌트에 HTML 컨트롤을 사용하면 logger를 표시하는 SDM 페이지를 머신 시각화 응용 프로그램에 직접 포함시킬 수 있습니다.

Application layer - mapp technology \ Diagnostics

Diagnostics and service \ Diagnostic tool

- Logger
- System Diagnostics Manager

Visualization \ Visual Components VC4 \ Control reference \ HTML view

Programming \ Libraries \ Configuration, system information, runtime control

- AsArLog
- ArEventLog

MpAlarm component and Visual Components alarm system

MpAlarm 구성 요소를 사용하면 MpAlarmUI 기능 블록을 사용하여 MpCnc2Axis 구성 요소의 미리 정의된 경보를 비주얼 구성 요소 경보 시스템으로 전달할 수 있습니다. 거기에 표시된 항목은 그룹, 시간 및 우선 순위별로 필터링 할 수 있습니다. 표시된 텍스트에 언어 전환을 사용할 수 있습니다
(참조 training module TM640 – Alarm System, Trends and Diagnostics).

Application layer - mapp technology \ Components \ Infrastructure \ MpAlarm - Support for alarm management

Visualization \ Visual Components VC4 \ Shared Resources \ Alarm System

Exercise: Programming error handling

오류가 발생하면, "Error" 상태로 전환해야 합니다. Error 확인이 완료되면 축 그룹을 다시 켜고, 이동 시작을 위한 모든 명령이 비활성화 되도록 모든 명령은 Reset 됩니다.

1 오류 상태를 인식하기 위해 "Error" 출력을 평가하십시오.
2 Reset the command
3 축 그룹 Acknowledgment는 "cmdReset"(운영자의 요청)을 통해 이루어져야합니다.
4 필요한 경우 다시 켜고 새로운 원점 복귀 절차를 수행하십시오.

10.4 Programming PLCopen function blocks

응용 프로그램에서 추가 기능을 얻으려면 PLCopen function block 추가 할 수 있습니다. 그들은 PLCopen function block은 mapp component와 마찬가지로 프로그램에서 주기적으로 호출됩니다.

프로그램에서 PLCopen function block의 오류 평가도 "Error" 출력을 사용하여 이루어지며 오류 번호는 "ErrorID" 출력에 표시됩니다. PLCopen function block 실행 중에 축 오류 또는 축 그룹 오류가 발생하면 두 개의 출력을 통해 신호가 전송됩니다. 또한 오류 정보는 MpCnc2Axis를 통해 사용할 수 있으며 이 component의 "ErrorReset" 입력으로 Acknowledge 할 수 있습니다. 이러한 이유로 축 그룹은 GroupErrorStop 상태를 가져옵니다. PLCopen function block의 오류를 Reset하려면 "Execute" 또는 " Enable" 입력을 Reset해야 합니다.

축 그룹에서 발생하지 않는 오류가 PLCopen function block에서는 발생할 수 있습니다. 예를 들어 유효하지 않은 축 그룹 참조를 사용하는 경우가 이에 해당됩니다. 이 경우 오류는 PLCopen function block에서 사용할 수 있지만 mapp component에서는 사용할 수 없습니다.

Exercise: Programming additional PLCopen functionality

이 기능은 MC_MoveDirectAbsolute_15 PLCopen function block 확장되어야 합니다. 주기적으로 호출되며 명령을 통해 활성화 할 수 있어야 합니다. 또한 function block은 응용 프로그램의 오류 처리에 통합되어야 합니다.

1 MC_MoveDirectAbsolute_15를 "cnc_ctrl" 프로그램에 삽입하십시오.
2 "cmdMoveDirectAbsolute"명령을 사용하여 "Operation" 단계에서 이동 시작 가능
3 "Error" 출력이 설정된 경우 오류 단계로 전환하십시오.
4 Acknowledge 오류 그리고 명령을 다시 설정하십시오.

11 Summary

축 그룹을 사용하면 여러 축을 포함하는 모션 어플리케이션을 만드는 프로세스를 상당히 단순화 할 수 있습니다. CNC 또는 로보틱스 어플리케이션이든 간단한 축 그룹이든, 축 그룹 기능을 사용하면 애플리케이션에 필요한 function block 수를 크게 줄일 수 있습니다. 그 결과 응용 프로그램이 더 쉽고 빠를 뿐만 아니라 읽기 쉽고 유지 관리가 더 쉽습니다.

MpCnc 라이브러리의 구성 요소는 축 그룹을 제어하는 데 사용할 수 있습니다. 포함된 function block은 PLCopen 표준을 준수하며 효율적인 디자인과 유용성으로 구분됩니다.

컨트롤러 애플리케이션의 통합은 사용자가 mapp technology에 확신을 가지자마자 시작됩니다 (이는 매우 간단합니다). 정교한 구성 관리, 웹 기반 진단, Automation Studio Logger 창과 System Diagnostics Manager에서 오류 정보를 직접 통합하는 것은 구성 요소 아키텍처를 통해 사용할 수 있습니다. 시각화 응용 프로그램에 드라이브 경보를 통합하는 것은 MpAlarm 구성 요소에 의해 활성화됩니다.

Figure 41: mapp technology offers a comprehensive portfolio of functions

PLCopen function block과 기능 호환이 가능하기 때문에 GmcGrpAPI 라이브러리의 function block을 통해 아무런 문제없이 드라이브 애플리케이션을 향상시킬 수 있습니다. 이 방법으로 축 그룹 어플리케이션을 기계의 요구 사항에 맞게 최적으로 조정할 수 있습니다.

추가적인 function block은 표준에서 정의된 것을 보완하기 위해 제공됩니다. B&R 축 그룹 솔루션과 관련된 기능을 처리합니다. 이를 통해 프로그래머는 모든 작업을 해결할 수 있는 모든 범위의 기능에 액세스 할 수 있습니다.

Figure 42: PLCopen Motion Control logo

Seminars and training modules

The Automation Academy provides targeted training courses for our customers as well as our own employees.
At the Automation Academy, you'll develop the skills you need in no time!
Our seminars make it possible for you to improve your knowledge in the field of automation engineering.
Once completed, you will be in a position to implement efficient automation solutions using B&R technology. This will make it possible for you to secure a decisive competitive edge by allowing you and your company to react faster to constantly changing market demands.

Automation Studio seminars and training modules

Programming and configuration	Diagnostics and service
SEM210 – Basics SEM246 – IEC 61131-3 programming language ST* SEM250 – Memory management and data storage SEM410 – Integrated motion control* SEM441 – Motion control (multi-axis systems) ** SEM480 – Hydraulics** SEM1110 – Axis groups and path-controlled movements** SEM510 – Integrated safety technology* SEM540 – Safe motion control*** SEM610 – Integrated visualization*	SEM920 – Diagnostics and service for end users SEM920 – Diagnostics and service with Automation Studio SEM950 – POWERLINK configuration and diagnostics* If you don't happen to find a seminar on our website that suits your needs, keep in mind that we also offer customized seminars that we can set up in coordination with your sales representatives: SEM099 – Individual training day Please visit our website for more information****: **www.br-automation.com/academy**

Overview of training modules	
TM210 – Working with Automation Studio TM213 – Automation Runtime TM223 – Automation Studio Diagnostics TM230 – Structured Software Development TM240 – Ladder Diagram (LD) TM241 – Function Block Diagram (FBD) TM242 – Sequential Function Chart (SFC) TM246 – Structured Text (ST) TM250 – Memory Management and Data Storage TM400 – Introduction to Motion Control TM410 – Working with Integrated Motion Control TM440 – Motion Control: Basic Functions TM441 – Motion Control: Multi-axis Functions TM1110 – Integrated Motion Control (Axis Groups) TM1111 – Integrated Motion Control (Path Controlled Movements) TM450 – Motion Control Concept and Configuration TM460 – Initial Commissioning of Motors TM500 – Introduction to Integrated Safety TM510 – Working with SafeDESIGNER TM540 – Integrated Safe Motion Control	TM600 – Introduction to Visualization TM610 – Working with Integrated Visualization TM630 – Visualization Programming Guide TM640 – Alarm System, Trends and Diagnostics TM670 – Advanced Visual Components TM920 – Diagnostics and service TM923 – Diagnostics and Service with Automation Studio TM950 – POWERLINK Configuration and Diagnostics TM261 – Closed-loop Control with LOOPCONR TM280 – Condition Monitoring for Vibration Measurement TM480 – The Basics of Hydraulics TM481 – Valve-based Hydraulic Drives TM482 – Hydraulic Servo Pump Drives TM490 – Printing machine technology In addition to a printed version, our training modules are also available on our website for download as electronic documents (login required): Visit our website for more information: **www.br-automation.com/academy**

Process control seminars and training modules

Process control standard seminars	Process control training modules
SEM841 – Process Control Training: Basic 1 SEM842 – Process Control Training: Basic 2 SEM890 – Advanced Process Control Solutions	TM800 – APROL System Concept TM811 – APROL Runtime System TM812 – APROL Operator Management TM813 – APROL XML Queries and Audit Trail TM830 – APROL Project Engineering TM890 – The Basics of LINUX Visit our website for more information: **www.br-automation.com/academy**

* SEM210 - Basics is a prerequisite for this seminar.
** SEM410 - Integrated motion control is a prerequisite for this seminar.
*** SEM410 - Integrated motion control and SEM510 - Integrated safety technology are prerequisites for this seminar.
**** Our seminars are listed in the Academy\Seminars area of the website. Seminar titles may vary by country. Not all seminars are available in every country.