

SIEMENS

Industry Online Support

Home

Verwendung von "REF_TO" und "DB_ANY" mit Technologieobjekten

SIMATIC S7-1500(T) / TIA Portal / Motion Control

<https://support.industry.siemens.com/cs/ww/de/view/109822593>

Siemens
Industry
Online
Support



Rechtliche Hinweise

Nutzung der Anwendungsbeispiele

In den Anwendungsbeispielen wird die Lösung von Automatisierungsaufgaben im Zusammenspiel mehrerer Komponenten in Form von Text, Grafiken und/oder Software-Bausteinen beispielhaft dargestellt. Die Anwendungsbeispiele sind ein kostenloser Service der Siemens AG und/oder einer Tochtergesellschaft der Siemens AG („Siemens“). Sie sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit und Funktionsfähigkeit hinsichtlich Konfiguration und Ausstattung. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern bieten lediglich Hilfestellung bei typischen Aufgabenstellungen. Sie sind selbst für den sachgemäßen und sicheren Betrieb der Produkte innerhalb der geltenden Vorschriften verantwortlich und müssen dazu die Funktion des jeweiligen Anwendungsbeispiels überprüfen und auf Ihre Anlage individuell anpassen.

Sie erhalten von Siemens das nicht ausschließliche, nicht unterlizenzierbare und nicht übertragbare Recht, die Anwendungsbeispiele durch fachlich geschultes Personal zu nutzen. Jede Änderung an den Anwendungsbeispielen erfolgt auf Ihre Verantwortung. Die Weitergabe an Dritte oder Vervielfältigung der Anwendungsbeispiele oder von Auszügen daraus ist nur in Kombination mit Ihren eigenen Produkten gestattet. Die Anwendungsbeispiele unterliegen nicht zwingend den üblichen Tests und Qualitätsprüfungen eines kostenpflichtigen Produkts, können Funktions- und Leistungsmängel enthalten und mit Fehlern behaftet sein. Sie sind verpflichtet, die Nutzung so zu gestalten, dass eventuelle Fehlfunktionen nicht zu Sachschäden oder der Verletzung von Personen führen.

Haftungsausschluss

Siemens schließt seine Haftung, gleich aus welchem Rechtsgrund, insbesondere für die Verwendbarkeit, Verfügbarkeit, Vollständigkeit und Mangelfreiheit der Anwendungsbeispiele, sowie dazugehöriger Hinweise, Projektierungs- und Leistungsdaten und dadurch verursachte Schäden aus. Dies gilt nicht, soweit Siemens zwingend haftet, z.B. nach dem Produkthaftungsgesetz, in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der schuldhaften Verletzung des Lebens, des Körpers oder der Gesundheit, bei Nichteinhaltung einer übernommenen Garantie, wegen des arglistigen Verschweigens eines Mangels oder wegen der schuldhaften Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegen oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist mit den vorstehenden Regelungen nicht verbunden. Von in diesem Zusammenhang bestehenden oder entstehenden Ansprüchen Dritter stellen Sie Siemens frei, soweit Siemens nicht gesetzlich zwingend haftet.

Durch Nutzung der Anwendungsbeispiele erkennen Sie an, dass Siemens über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden kann.

Weitere Hinweise

Siemens behält sich das Recht vor, Änderungen an den Anwendungsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in den Anwendungsbeispielen und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang. Ergänzend gelten die Siemens Nutzungsbedingungen (<https://support.industry.siemens.com>).

Securityhinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen einen Bestandteil eines solchen Konzepts.

Die Kunden sind dafür verantwortlich, unbefugten Zugriff auf ihre Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Diese Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und nur wenn entsprechende Schutzmaßnahmen (z.B. Firewalls und/oder Netzwerksegmentierung) ergriffen wurden.

Weiterführende Informationen zu möglichen Schutzmaßnahmen im Bereich Industrial Security finden Sie unter <https://www.siemens.com/industrialsecurity>.

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Produkt-Updates anzuwenden, sobald sie zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter <https://www.siemens.com/cert>.

Inhaltsverzeichnis

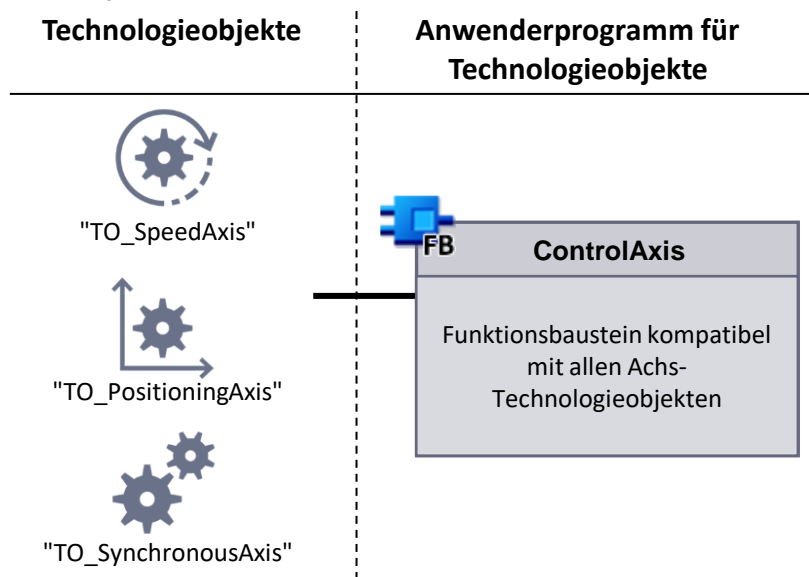
| | |
|---|-----------|
| Rechtliche Hinweise | 2 |
| 1 Allgemeines | 4 |
| 1.1 Überblick..... | 4 |
| 1.2 "DB_ANY" oder "REF_TO"..... | 4 |
| 1.2.1 Allgemein..... | 4 |
| 1.2.2 Vorteile der Nutzung von "DB_ANY"..... | 5 |
| 1.2.3 Vorteile der Nutzung von "REF_TO"..... | 6 |
| 2 Direkte Übergabe von Technologieobjekten | 7 |
| 2.1 Bausteinschnittstelle für Technologieobjekte..... | 7 |
| 2.1.1 Datentypen von Technologieobjekten..... | 7 |
| 2.1.2 Aufbau von Technologieobjekten..... | 9 |
| 2.2 Technologieobjekt-Schnittstelle der Motion Control-Anweisungen..... | 12 |
| 3 Übergabe von Technologieobjekten mit "DB_ANY" | 15 |
| 3.1 Funktionsweise..... | 15 |
| 3.2 Anlegen der Bausteine und Variablen..... | 16 |
| 3.3 Verwenden von Technologieobjekten mit "DB_ANY"..... | 17 |
| 4 Technologieobjekte mit "REF_TO" | 20 |
| 4.1 Funktionsweise..... | 20 |
| 4.2 Anlegen der Variablen im Bausteininterface..... | 20 |
| 4.3 Verwenden der Referenzen mit Technologieobjekten..... | 21 |
| 4.4 Verwenden von Referenzen für verschiedene Technologieobjekte..... | 23 |
| 4.5 Tipps für die Verwendung von "REF_TO"..... | 25 |
| 4.5.1 Nutzung von "RETURN"..... | 25 |
| 4.5.2 Gruppieren von Zugriffen über Referenzen..... | 27 |
| 5 Programmier Tipps | 29 |
| 5.1 Organisation von Technologieobjekten in einem Array..... | 29 |
| 5.2 Massenoperationen über den Array-Index..... | 30 |
| 5.2.1 Massenoperation "MC_Reset" mit "DB_ANY"..... | 30 |
| 5.2.2 Massenoperation "MC_Reset" mit "REF_TO"..... | 34 |
| 6 Anhang | 37 |
| 6.1 Service und Support..... | 37 |
| 6.2 Industry Mall..... | 38 |
| 6.3 Links und Literatur..... | 38 |
| 6.4 Änderungsdokumentation..... | 38 |

1 Allgemeines

1.1 Überblick

Auf den SIMATIC S7-1500(T) Steuerungen stehen Ihnen verschiedene Technologieobjekte zur Verfügung, um beispielsweise die Achsen Ihrer Maschine in der Steuerung abzubilden und anzusteuern. Bei größeren Maschinen kommen viele Achstechnologieobjekte und verschiedene Technologieobjekt-Typen zum Einsatz. Daher sollten Programmbausteine zur Ansteuerung und Diagnose der Technologieobjekte möglichst modular programmiert werden. So können Sie auch noch zur Laufzeit entscheiden, welches Technologieobjekt typübergreifend für eine Anwendung verwendet werden soll.

Abbildung 1-1 Modularer Achsbaustein



Um diese flexible Programmierung umzusetzen, stehen Ihnen auf den SIMATIC S7-1500(T) Steuerungen zwei Möglichkeiten zur Verfügung:

- Nutzung des Datentyps "DB_ANY" für die Übergabe der Technologieobjekte an den Funktionsbaustein.
- Verwendung von "REF_TO" innerhalb des Funktionsbausteins, um Referenzen auf Technologieobjekte zu erstellen.

Die Funktionsweise und die Anwendung der beiden Möglichkeiten werden in diesem Dokument erklärt.

1.2 "DB_ANY" oder "REF_TO"

1.2.1 Allgemein

In der Regel sollten Sie "REF_TO" für die Erstellung von eigenen Programmbausteinen verwenden, die flexibel mit mehreren Technologieobjekt-Typen eingesetzt werden sollen. Diese Aufgabe lässt sich zwar auch mit "DB_ANY" umsetzen, "REF_TO" bietet aber gegenüber der Verwendung von "DB_ANY" mehrere Vorteile.

Allerdings ist die Möglichkeit "REF_TO" auch für Technologieobjekte zu verwenden erst seit TIA Portal V18 mit entsprechenden Firmware- und Motion Control-Versionen vorhanden. Daher müssen folgende Voraussetzungen gegeben sein, um "REF_TO" für die Programmierung eines

flexibel mit mehreren Technologieobjekt-Typen verwendbaren Programmbausteins verwenden zu können:

- TIA Portal V18 oder höher
- CPU-Firmware V3.0 oder höher
- Motion Control-Version V7.0 oder höher

Sollten die von Ihnen verwendeten Produkte mindestens eine der Voraussetzungen nicht erfüllen, können Sie nur "DB_ANY" verwenden, um einen entsprechenden Programmbaustein zu erstellen.

Hinweis

Selbstverständlich ist die Nutzung von "DB_ANY" mit Technologieobjekten auch mit TIA Portal V18, CPU-Firmware V3.0 und Motion Control Version V7.0 möglich. So können Sie z. B. Bausteine aus früheren Versionen problemlos hochrüsten und weiterhin verwenden.

Informationen, wie Sie "DB_ANY" zusammen mit Technologieobjekten verwenden, finden Sie in Kapitel [3](#).

Informationen zur Verwendung von "REF_TO" mit Technologieobjekten finden Sie in Kapitel [4](#).

1.2.2 Vorteile der Nutzung von "DB_ANY"

Die Verwendung des Datentyps "DB_ANY" bei Motion Control Anwendungen bzw. bei der Nutzung von Technologieobjekten bietet folgende Vorteile:

- **Erzeugung von flexiblen und universellen Programmbausteinen**
Wird der Eingangsparameter eines Funktionsbausteins oder einer Funktion zur Übergabe eines Technologieobjekts mit dem Datentyp "DB_ANY" definiert, kann dieser Baustein für alle im Programm vorhandenen Technologieobjekte eingesetzt werden.
Der Datentyp des übergebenen Technologieobjekts wird innerhalb des Bausteins ermittelt und gegebenenfalls an typabhängige Programmabschnitte im Baustein übergeben.
- **Keine feste Zuordnung der Technologieobjekte im Programm**
Während der Erstellungsphase des Programms müssen Technologieobjekte nicht fest den einzelnen Programmteilen oder Funktionseinheiten zugeordnet werden, sondern können zur Laufzeit des Programms ermittelt und aufgerufen werden. Damit ist beispielsweise auch eine Auswahl eines Technologieobjekts über eine HMI-Bedienoberfläche möglich.
- **Organisation der Technologieobjekte in einem Array**
Verfügt das Programm über eine große Anzahl an Technologieobjekten, können diese beispielsweise in einem Array vom Datentyp "DB_ANY" organisiert werden (z.B. ARRAY[0..9] of DB_ANY).
Die Auswahl der Technologieobjekte im Programm oder an der HMI-Bedienoberfläche ist dann einfach über den Array-Index möglich.
- **Bearbeitung von Programmteilen in Schleifenstrukturen**
Sind in einem Projekt sehr viele Technologieobjekte enthalten und müssen auf diese Technologieobjekte gleiche Funktionen, wie z.B. das Quittieren von Fehlern über die Funktion "MC_Reset", angewendet werden, kann diese Funktion in einer Programmschleife (FOR, WHILE, REPEAT) ausgeführt werden.

Hinweis

Bei der Bearbeitung sehr vieler Technologieobjekte oder größerer Programme innerhalb der Schleifen, kann es sinnvoll sein, die Bearbeitung auf mehrere CPU-Zyklen aufzuteilen und in jedem CPU-Zyklus nur einen Teil des Arrays bzw. der Technologieobjekte zu bearbeiten, um die Zykluszeit gering zu halten.

1.2.3 Vorteile der Nutzung von "REF_TO"

Die Verwendung von Referenzen auf Technologieobjekte mit "REF_TO" bei Motion Control Anwendungen bzw. bei der Nutzung von Technologieobjekten bietet alle Vorteile der Nutzung des Datentyps "DB_ANY", die in Kapitel [1.2.2](#) genannt wurden. Die Nutzung von Referenzen auf Technologieobjekte mit "REF_TO" bietet jedoch noch weitere Vorteile:

- **Wegfall der Technologieobjekt-spezifischen Bausteine**

Bei Verwendung von "DB_ANY" für Technologieobjekte werden für jeden Technologieobjekt-Typen separate Funktionsbausteine, die die Motion Control-Programmierung enthalten, benötigt (s. Kapitel [3](#)). Diese werden bei der Verwendung von Referenzen mit "REF_TO" durch einen einzigen Baustein ersetzt, der zusammen mit allen gewünschten Technologieobjekt-Typen verwendet werden kann. Dies hat die folgenden Auswirkungen:

- Geringerer Projektierungsaufwand, da die Motion Control-Programmierung nur einmal und nicht für jeden Achstypen separat erstellt werden muss.
- Bessere Übersichtlichkeit des Anwenderprogramms, da bedeutend weniger Programmbausteine benötigt werden.
- Einsparung von Programmierspeicher, da weniger Bausteine benötigt werden

- **Wegfall des Schalenbausteins zur Typ-Angleichung**

Bei der Verwendung des Datentyps "DB_ANY" für Technologieobjekte wird ein zusätzlicher Programmbaustein benötigt, der den Typ des Technologieobjekts auswertet und einen entsprechenden Technologieobjekt-spezifischen Baustein aufruft. Auch dieser zusätzliche Baustein wird bei der Verwendung von Referenzen auf Technologieobjekte mit "REF_TO" nicht benötigt. Dadurch wird die CPU-Auslastung reduziert und weniger Programmspeicher benötigt.

2 Direkte Übergabe von Technologieobjekten

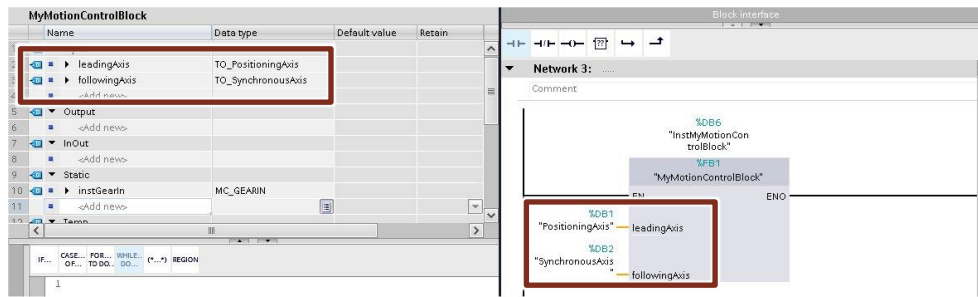
In diesem Kapitel erfahren Sie, wie Technologieobjekte direkt an Programmbausteine übergeben werden können und welche Besonderheiten es dabei gibt.

Diese Informationen schaffen die Grundlage für die Verwendung von "REF_TO" und "DB_ANY" bei der Übergabe von Technologieobjekten an Programmbausteine, um eine strukturierte und flexible Programmierung des Anwenderprogramms umsetzen zu können.

2.1 Bausteinschnittstelle für Technologieobjekte

Um Technologieobjekte direkt an einen Programmbaustein zu übergeben, können Sie innerhalb der Schnittstellendefinition des Bausteins einen Eingang für das Technologieobjekt definieren. Als Datentyp dieses Eingangs muss der Datentyp des entsprechenden Technologieobjekts angegeben werden.

Abbildung 2-1 Programmbeispiel – Selbst erstellter Funktionsbaustein



Nach der Definition des Eingangsparameters für die Übergabe eines Technologieobjekts können Sie innerhalb des Bausteins mit dem Eingangsparameter, stellvertretend für das Technologieobjekt, arbeiten. Beim Aufruf des Bausteins wird das verschaltete Technologieobjekt dem Eingangsparameter zugewiesen und somit im Programmbaustein verwendet.

Hinweis

Die Übergabe eines Technologieobjekts an einen Funktionsbaustein oder eine Funktion kann sowohl als Eingangsvariable (IN), als auch als DurchgangsvARIABLE (IN/OUT) erfolgen. In beiden Fällen wird – als Besonderheit der Technologieobjekte – eine Referenz des Technologieobjekts an den Baustein übergeben, weshalb sich beide Übergabevarianten in der Ausführungsgeschwindigkeit des Programms und im Ressourcenverbrauch für Technologieobjekte nicht unterscheiden.

Um eine einheitliche Programmierung in Bezug auf die Motion Control-Anweisungen im TIA Portal zu erhalten, ist in der Regel die Übergabe des Technologieobjekts als Eingangsvariable (IN) zu bevorzugen.

2.1.1 Datentypen von Technologieobjekten

Der Datentyp von Technologieobjekten kann in der Schnittstellendefinition nicht aus der Klappliste der Datentypen ausgewählt werden, sondern muss manuell eingegeben werden.

In der folgenden Tabelle sind ausgewählte Datentypen von Technologieobjekten aufgeführt, die überwiegend als Eingangsparameter von selbst erstellten Funktionsbausteinen oder Funktionen benutzt werden:

2 Direkte Übergabe von Technologieobjekten

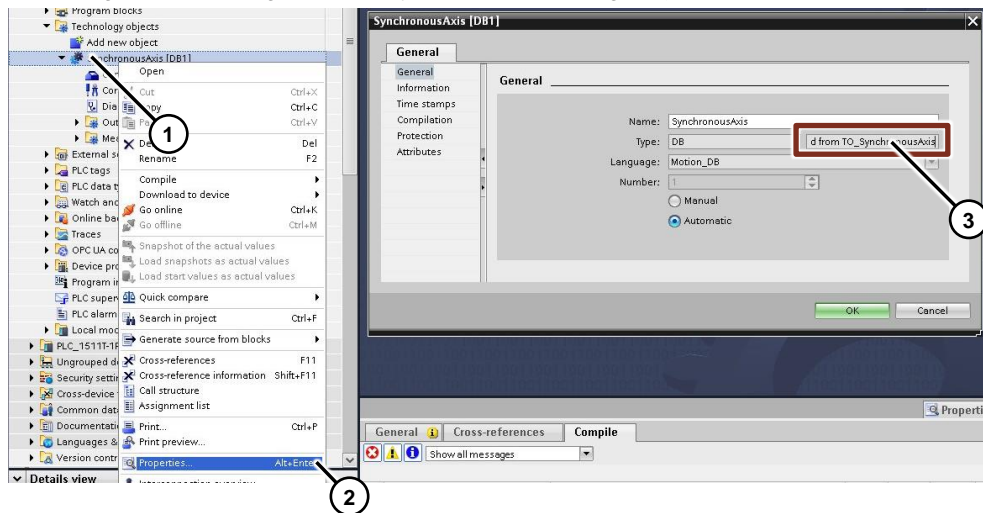
Tabelle 2-1 Ausgewählte Datentypen von Technologieobjekten zur Schnittstellendefinition

| Objekttyp | Technologieobjekt | Datentyp |
|-------------------------------|------------------------|---------------------|
| Achse | Drehzahlachse | TO_SpeedAxis |
| | Positionierachse | TO_PositioningAxis |
| | Gleichlaufachse | TO_SynchronousAxis |
| Externer Geber | Externer Geber | TO_ExternalEncoder |
| Nocken | Nocken | TO_OutputCam |
| | Nockenspur | TO_CamTrack |
| Messtaster | Messtaster | TO_MeasuringInput |
| Kurvenscheibe | Kurvenscheibe | TO_Cam |
| Kinematik | Kinematik | TO_Kinematics |
| PLC-übergreifender Gleichlauf | Leitwertstellvertreter | TO_LeadingAxisProxy |

Allgemein kann der Datentyp eines Technologieobjekts über den Eigenschaftendialog des Technologieobjekts ermittelt werden.

Öffnen Sie am Technologieobjekt (1) mit der rechten Maustaste das Kontextmenü und wählen Sie dort den Eigenschaftendialog (2) aus. Im Dialog kann auf der Seite der allgemeinen Informationen der Datentyp (3) abgelesen werden, von dem der Datenbaustein des Technologieobjekts abgeleitet ist.

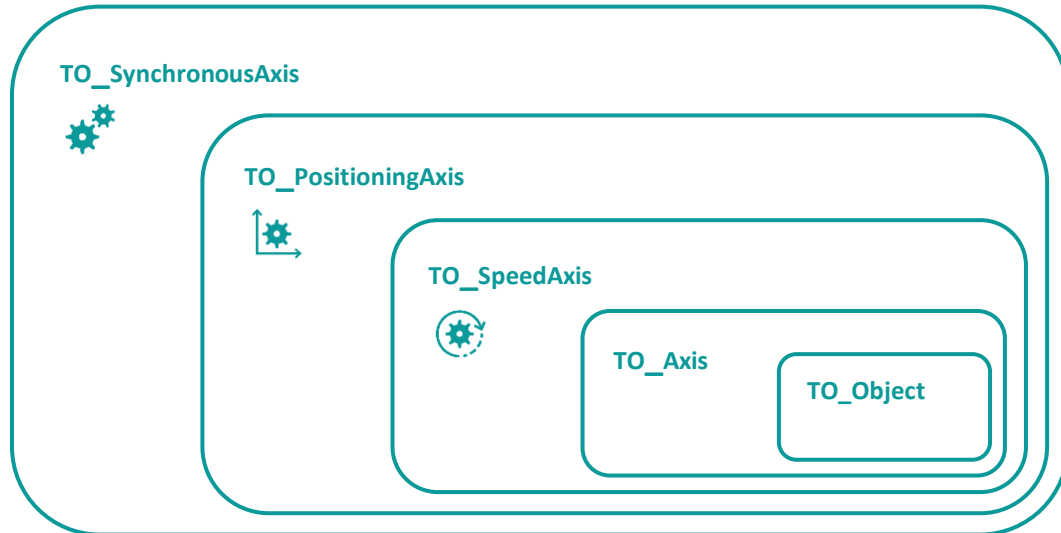
Abbildung 2-2 Ermittlung des Datentyps eines Technologieobjekts



2.1.2 Aufbau von Technologieobjekten

Datentypen von Achs-Technologieobjekten bauen aufeinander auf und enthalten daher auch die unterlagerten Datentypen.

Abbildung 2-3 "Schalenkonzept" der Achs-Technologieobjekte

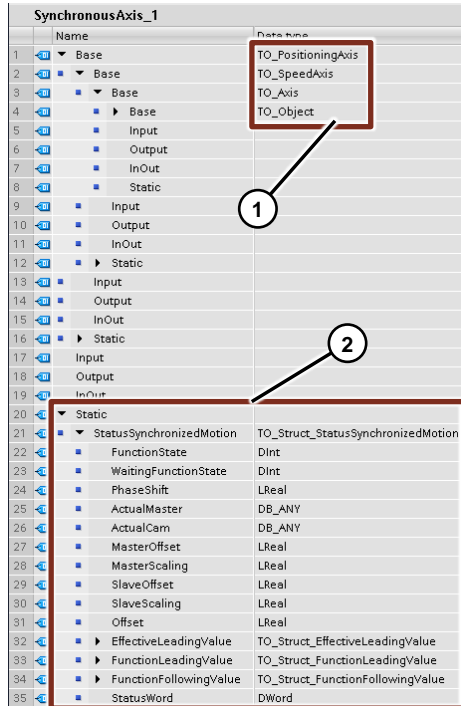


- Die Basis für alle Technologieobjekte bildet der Datentyp "TO_Object". Auf diesem Datentyp baut der Datentyp "TO_Axis" auf, der für alle Achs-Technologieobjekte die Basis bildet.
- Auf das Objekt "TO_Axis" baut wiederum der erste "echte" Achs-Datentyp "TO_SpeedAxis" für Drehzahlachsen auf. Dieser Datentyp enthält alle für die Konfiguration und den Betrieb von Drehzahlachsen notwendigen Variablen.
- Der Datentyp "TO_PositioningAxis" für den "nächsthöheren" Achs-Typ Positionierachse enthält den Datentypen "TO_SpeedAxis" und bringt noch weitere Variablen für die Positionierfunktionen mit.
- Der letzte Achs-Datentyp "TO_SynchronousAxis" für Gleichlaufachsen enthält sowohl den Datentyp "TO_PositioningAxis" für Positionierachsen als auch den Datentyp "TO_SpeedAxis" für Drehzahlachsen und weitere Variablen für die Gleichlauffunktionen. Damit kann dieser Achstyp sowohl als Gleichlaufachse, als auch als Positionier- oder Drehzahlachse eingesetzt werden.

2 Direkte Übergabe von Technologieobjekten

Im Technologiedatenbaustein einer Gleichlaufachse "TO_SynchronousAxis" sind sowohl die unterlagerten Achstypen (1), als auch die zusätzlichen Variablen für die Gleichlauffunktionen (2) erkennbar.

Abbildung 2-4 "Schalenkonzept" im Technologiedatenbaustein



Dieses Schalenkonzept wirkt sich auch auf die Bausteinschnittstelle aus. Speziell für Achstechnologieobjekte der SIMATIC ist es möglich, dass an den Eingängen von Programmbausteinen gleich- oder niederwertige Achsdatentypen bei direkter (absoluter) Übergabe benutzt werden können. So kann beispielsweise auch eine Positionierachse "TO_PositioningAxis" an einen Eingangsparameter eines Programmbausteins mit dem Datentyp "TO_SpeedAxis" übergeben und im Baustein verwendet werden.

Abbildung 2-5 "Schalenkonzept" beim Bausteinaufruf

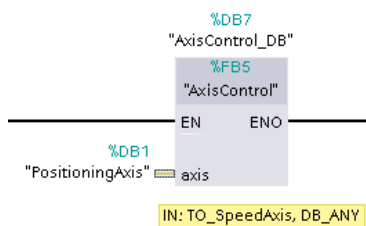


Tabelle 2-2 Schalenkonzept der Datentypen des Technologieobjekts "Achse"

| Technologieobjekte nutzbar an folgenden Eingängen | Datentyp des Eingangs: | | |
|---|------------------------|----------------------|----------------------|
| | "TO_SpeedAxis" | "TO_PositioningAxis" | "TO_SynchronousAxis" |
| Drehzahlachse | Ja | Nein | Nein |
| Positionierachse | Ja | Ja | Nein |
| Gleichlaufachse | Ja | Ja | Ja |

Dies ermöglicht schon ohne die Verwendung von "DB_ANY" oder "REF_TO" eine flexible Bausteinprogrammierung für Bausteine, die mit mehreren Achstypen verwendet werden können.

Abbildung 2-6 Baustein zum Auslesen und quittieren von Achsfehlern mit mehreren Achsen

| Name | Data type | Default value | Retain | Accessible ... |
|----------------|--------------|---------------|------------|-------------------------------------|
| axis | TO_SpeedAxis | | | |
| execute | Bool | false | Non-retain | <input checked="" type="checkbox"/> |
| restart | Bool | false | Non-retain | <input checked="" type="checkbox"/> |
| -<Add new> | | | | |
| Output | | | | |
| ErrorTO | Bool | false | Non-retain | <input checked="" type="checkbox"/> |
| AlarmNoTO | UDInt | 0 | Non-retain | <input checked="" type="checkbox"/> |
| done | Bool | false | Non-retain | <input checked="" type="checkbox"/> |
| busy | Bool | false | Non-retain | <input checked="" type="checkbox"/> |
| commandAborted | Bool | false | Non-retain | <input checked="" type="checkbox"/> |
| resetError | Bool | false | Non-retain | <input checked="" type="checkbox"/> |
| resetErrorID | Word | 16#0 | Non-retain | <input checked="" type="checkbox"/> |
| -<Add new> | | | | |
| InOut | | | | |
| -<Add new> | | | | |
| Static | | | | |
| instReset | MC_RESET | | | <input checked="" type="checkbox"/> |

Jedoch gibt es hierbei einige Einschränkungen.

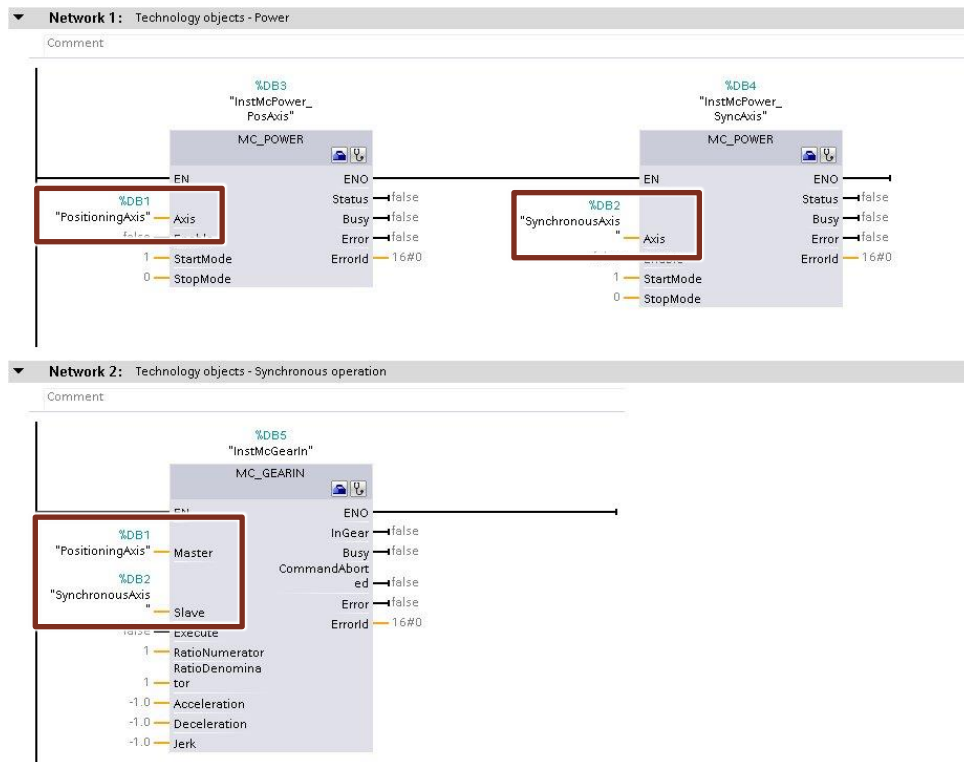
- Es können nur Motion Control-Anweisungen verwendet werden, die auch für den in der Bausteinschnittstelle angelegten Technologieobjekt-Typen verwendet werden können. Innerhalb des Bausteins ist nur der Technologieobjekt-Datentyp des Eingangsparameters bekannt. Wenn in der Bausteinschnittstelle beispielsweise eine Drehzahlachse "TO_SpeedAxis" definiert ist, können Sie innerhalb des Bausteins auch nur Motion Control-Anweisungen verwenden, die für Drehzahlachsen verwendet werden können. Die Verwendung von Motion Control-Anweisungen wie z. B. "MC_MoveAbsolute" zum absoluten Positionieren einer Achse ist hier nicht möglich. Dies gilt auch, wenn am Baustein eine Positionierachse verschaltet ist.
- Es kann nur auf Variablen der Technologieobjekte zugegriffen werden, die auch in dem in der Bausteinschnittstelle angelegten Technologieobjekt-Typen vorhanden sind. Wenn in der Bausteinschnittstelle also beispielsweise eine Drehzahlachse "TO_SpeedAxis" definiert ist, können Sie innerhalb des Bausteins nicht über den Eingangsparameter auf die Variablen einer Positionierachse wie z. B. die aktuelle

Position zugreifen. Dies gilt auch, wenn am Baustein eine Positionierachse verschaltet ist.

2.2 Technologieobjekt-Schnittstelle der Motion Control-Anweisungen

Die Motion Control-Anweisungen, die zum Steuern der Technologieobjekte verwendet werden, besitzen alle einen Eingangsparameter, an dem das Technologieobjekt übergeben wird, auf das die Motion Control-Anweisung angewendet werden soll.

Abbildung 2-7 Programmbeispiel – Aufruf der Motion Control-Anweisungen



Dabei kann allerdings nicht jedes Technologieobjekt an jeder Motion Control-Anweisung verschaltet werden. Um herauszufinden, welche Technologieobjekt-Typen an den verschiedenen Motion Control-Anweisungen verwendet werden können, haben Sie mehrere Möglichkeiten:

- **Online-Hilfe**

In den Einträgen zu den Motion Control-Anweisungen in der TIA Portal Online-Hilfe finden Sie unter anderem eine Übersicht über die Ein- und Ausgangsparameter der jeweiligen Anweisung. In dieser Übersicht finden Sie in der Spalte "Data type" für den Parameter, an dem das Technologieobjekt verschaltet wird, eine Auflistung aller Technologieobjekte, die mit der entsprechenden Motion Control-Anweisung verwendet werden können.

Abbildung 2-8 Online-Hilfe Eintrag "MC_Reset"

MC_Reset: Acknowledge alarms, restart technology object V7

Parameters

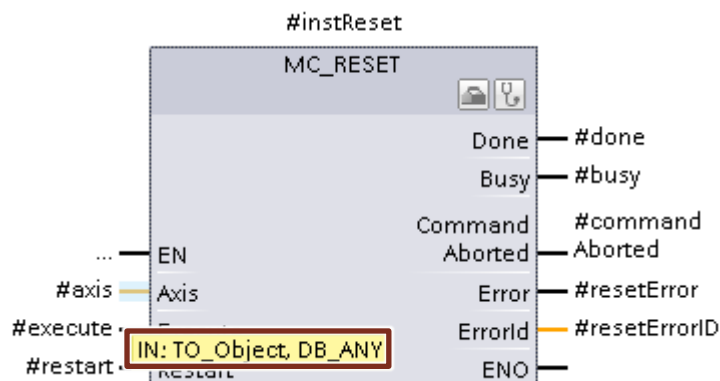
The following table shows the parameters of the Motion Control instruction "MC_Reset":

| Parameter | Declaration | Data type | Default value | Description |
|-----------|-------------|--|---------------|-------------------|
| Axis | INPUT | TO_SpeedAxis TO_PositioningAxis TO_SynchronousAxis TO_ExternalEncoder TO_LeadingAxisProxy (S7-1500T) TO_OutputCam TO_CamTrack TO_MeasuringInput TO_Cam (S7-1500T) TO_Cam_10k (S7-1500T) TO_Kinematics (S7-1500T) | - | Technology object |

- Datentyp des Eingangsparameters**

Wenn Sie den Mauszeiger auf den Eingangsparameter des Aufrufs einer Motion Control-Anweisung bewegen, erscheint ein Tool-Tip mit dem Datentypen des Eingangsparameters. Auch hier greift das Schalenkonzept der Technologieobjekte (s. Kapitel [2.1.2](#)) und es können am entsprechenden Eingang alle Technologieobjekte verschaltet werden, die diesen Datentypen beinhalten.

Abbildung 2-9 Eingangsdatentyp "axis" an der Motion Control-Anweisung "MC_Reset"



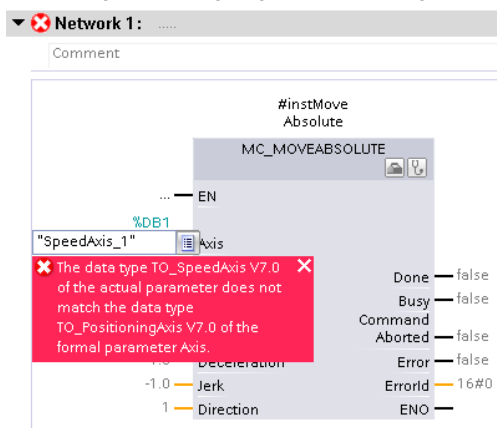
Hinweis

Bei der Verschaltung von "DB_ANY" Datentypen funktioniert das Schalenkonzept nicht. Die Verschaltung eines Achs-Datentyps über "DB_ANY" an einer Motion Control-Anweisung, die nicht genau diesen Achsdatentyp erwartet, führt zur Ausgabe eines Programmierfehlers und ggf. zu einem Übergang der CPU in den Betriebszustand "STOP".

- Compiler im TIA Portal**

Der Compiler im TIA Portal erkennt ungültige absolute Verschaltungen und markiert diese im Programmierer. Sie können die Technologieobjekte also während des Engineerings beliebig an Motion Control-Anweisungen verschalten und mögliche Übersetzungsfehler, die vom Compiler ausgegeben werden, im Anschluss beheben.

Abbildung 2-10 Ungültige Verschaltung im Programmierer



Hinweis

Der Compiler erkennt nur ungültige absolute Verschaltungen. Wird beispielsweise eine Variable mit dem Datentypen "DB_ANY" an einer Motion Control-Anweisung verschalten, kann der Compiler die Gültigkeit dieser Verschaltung nicht bewerten, da die Variable mit dem Datentypen "DB_ANY" erst zur Laufzeit zugewiesen wird.

3 Übergabe von Technologieobjekten mit "DB_ANY"

Hinweis

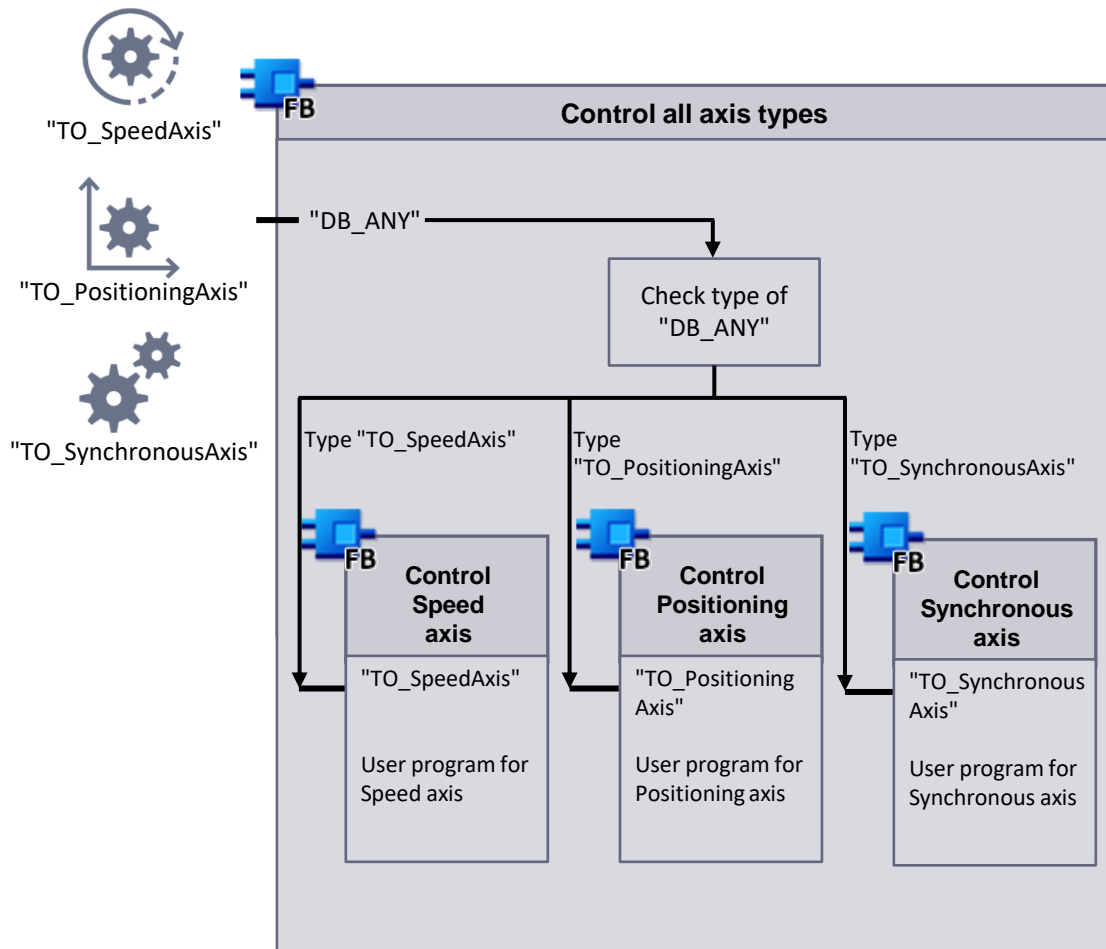
Die in diesem Kapitel vorgestellte Möglichkeit den Datentypen "DB_ANY" für Technologieobjekte zu verwenden, ist für die Verwendung der TIA Portal Version V17 zusammen mit CPU-Firmware V2.9 und Motion Control-Version V6.0 oder älteren Versionen vorgesehen. Wenn Sie TIA Portal V18, CPU Firmware V3.0 und die Motion Control-Version V7.0 oder neuere Versionen verwenden, sollten Sie die effizientere Möglichkeit, über Referenzen auf Technologieobjekte mit "REF_TO" zuzugreifen, einsetzen (s. Kapitel [4](#)).

Der Datentyp "DB_ANY" bietet bei den Technologieobjekten der SIMATIC Steuerungen bei Motion Control Anwendungen die Möglichkeit, die direkte (absolute) Zuordnung der Technologieobjekte im Programm, während der Programmierphase, durch eine flexiblere Übergabe der Technologieobjekte zur Laufzeit des Programms zu ersetzen. Dabei müssen jedoch ein paar Besonderheiten berücksichtigt werden, um ein optimales Ergebnis zu erzielen.

3.1 Funktionsweise

Für einen Programmbaustein, der mit verschiedenen (Achsen-)Technologieobjekt-Typen verwendet werden kann, werden mehrere ineinander aufgerufene Funktionsbausteine benötigt. Zuerst wird das jeweilige Technologieobjekt als Datentyp "DB_ANY" an einen Funktionsbaustein übergeben, der den Technologieobjekt-Typ prüft und einen entsprechenden Funktionsbaustein für den jeweiligen Technologieobjekt-Typen aufruft. Beim Aufruf dieses Funktionsbausteins wird das Technologieobjekt mit dem Datentypen "DB_ANY" als Datentyp des jeweiligen Technologieobjekts z. B. "TO_PositioningAxis" übergeben.

Abbildung 3-1 Funktionsweise "DB_ANY" mit Technologieobjekten



© Siemens AG 2023 All rights reserved

3.2 Anlegen der Bausteine und Variablen

Um den Datentypen "DB_ANY" für Technologieobjekte zu verwenden, müssen Sie zuerst die entsprechenden Bausteine und Variablen in den Bausteininterfaces anlegen. Sie benötigen die folgenden Bausteine:

- **"Schalenbaustein" zur Typkonvertierung**

Dieser Baustein wird, unabhängig davon wie viele Technologieobjekte Sie verwenden möchten, nur einmal benötigt. Legen Sie im Bausteininterface eine Eingangsvariable mit dem Datentypen "DB_ANY" an. An diese Eingangsvariable können Sie ein beliebiges Technologieobjekt übergeben. Bei der Übergabe wird das Technologieobjekt der Variable mit dem Datentypen "DB_ANY" zugewiesen, sodass sie den Datenbaustein des Technologieobjekts adressiert.

Abbildung 3-2 Eingangsparameter "DB_ANY" für Technologieobjekt

| ControlAllAxes | | |
|----------------|-----------|-----------|
| | Name | Data type |
| 1 | Input | |
| 2 | axis | DB_ANY |
| 3 | <Add new> | |

- **Funktionsbausteine für jeden verwendeten Technologieobjekt-Typ**

Da Sie das jeweilige Technologieobjekt über die Eingangsvariable mit dem Datentypen "DB_ANY" im Schalenbaustein nicht direkt adressieren können, werden weitere Programmbausteine benötigt, die aus der Variablen mit dem Datentypen "DB_ANY" wieder ein direkt adressierbares Objekt machen.

Legen Sie für jeden Technologieobjekt-Typen, den Sie in Ihrem Anwenderprogramm verwenden wollen, einen Funktionsbaustein an. Definieren Sie im Interface der Bausteine jeweils eines der gewünschten Technologieobjekte. Beim Aufruf dieses Bausteins wird die Variable mit dem Datentypen "DB_ANY" aus dem Schalenbaustein an den hier erstellten Programmbaustein übergeben und steht über die im Interface angelegte Variable des Technologieobjekt-spezifischen Bausteins als direkt adressierbares Objekt zur Verfügung.

Abbildung 3-3 Technologieobjekt-Spezifische Funktionsbausteine

| ControlSpeedAxis | | ControlPositioningAxis | | ControlSynchronousAxis | |
|------------------|--------------|------------------------|--------------------|------------------------|--------------------|
| Name | Data type | Name | Data type | Name | Data type |
| Input | | Input | | Input | |
| axis | TO_SpeedAxis | axis | TO_PositioningAxis | axis | TO_SynchronousAxis |
| <Add new> | | <Add new> | | <Add new> | |

Hinweis

Falls Sie die Technologieobjekte in diesen Bausteinen nicht über die Motion Control-Anweisungen steuern wollen, also keine Instanzen im Funktionsbaustein für das Technologieobjekt benötigen, sondern beispielsweise nur Aktualwerte aus dem Technologieobjekt auslesen wollen und auch sonst keine statischen Variablen für die Programmierung benötigen, können Sie hier auch eine Funktion (FC) statt eines Funktionsbausteins (FB) anlegen.

3.3 Verwenden von Technologieobjekten mit "DB_ANY"

Programmierung im "Schalenbaustein"

Nachdem Sie die benötigten Bausteine und Variablen angelegt haben, können Sie mit der Programmierung des "Schalenbausteins" zur Typkonvertierung beginnen.

Mit Hilfe der Anweisung "TypeOfDB(...)" ermitteln Sie den Datentyp des Datenbausteins, den die Variable vom Datentyp DB_ANY adressiert (1). Die Funktion gibt die Bezeichnung des Datentyps als "Type_ID" im Klartext an das Programm zurück (2).

Für Motion Control Anwendungen können Sie diese Funktion auf die Eingangsvariable eines Funktionsbausteins bzw. einer Funktion anwenden und sofort über eine CASE-Anweisung auswerten. In den einzelnen Sprungbereichen der CASE-Anweisung können Sie nun die Technologieobjekt-spezifischen Funktionsbausteine aufrufen (3). Für die Aufrufe der Funktionsbausteine empfiehlt es sich, Multiinstanzen (4) im "Schalenbaustein", statt globaler Instanzen zu verwenden.

Hinweis

Sie müssen sicherstellen, dass eine Variable mit dem Datentyp "DB_ANY" **immer** an einen Programmbaustein übergeben wird, der genau den zum Aktualwert des "DB_ANY" passenden Datentypen erwartet. Adressiert der "DB_ANY" einen anderen Datentyp, als am Programmbaustein erwartet wird, wird von der Steuerung ein Programmierfehler ausgegeben und die Steuerung wechselt ggf. in den Betriebszustand "STOP".

Dies trifft auch auf Technologieobjekte zu. Das Schalenkonzept kann hier nicht angewendet werden. Ein "DB_ANY" der auf eine Positionierachse zeigt, kann also nur an einen Programmbaustein übergeben werden, der auch den Datentypen "TO_PositioningAxis" erwartet. Eine Übergabe an einen Programmbaustein, der den Datentypen "TO_SpeedAxis" erwartet ist nicht möglich, auch wenn der Datentyp "TO_SpeedAxis" im Datentypen "TO_PositioningAxis" enthalten ist.

Hinweis

Sie können den Baustein zur Typkonvertierung auch als Funktion (FC) anlegen, wenn in diesem Baustein keine Multiinstanzen für die Technologieobjekt-spezifischen Programmbausteine angelegt werden sollen.

Abbildung 3-4 Ermittlung des TO-Datentyps zur Programmlaufzeit

| Name | Data type | Default value |
|--------------------------|--------------------------|---------------|
| ControlAllAxes | | |
| Input | | |
| axis | DB_ANY | 0 |
| Output | | |
| error | Bool | false |
| InOut | | |
| <Add new> | | |
| Static | | |
| instControlSpeedAxis | "ControlSpeedAxis" | |
| instControlPositionin... | "ControlPositioningAxis" | |
| instControlSynchrono... | "ControlSynchronousAxis" | |
| Temp | | |


```

1 CASE TypeOfDB(#axis) OF
2   TO_SpeedAxis
3     //Call user program for speed axis
4     #instControlSpeedAxis(axis := #axis);
5   TO_PositioningAxis
6     //Call user program for positioning axis
7     #instControlPositioningAxis(#axis);
8   TO_SynchronousAxis
9     //Call user program for synchronous axis
10    #instControlSynchronousAxis(#axis);
11  ELSE
12    //Error handling
13    //unexpected DB-type connected at axis input
14    #error := TRUE;
15  END_CASE;

```

© Siemens AG 2023 All rights reserved

Hinweis

Der ELSE-Zweig am Ende der CASE-Anweisung (5) dient der Fehlerrückmeldung innerhalb Ihres Anwenderprogramms, falls ein nicht in der CASE-Anweisung aufgeführter Datentyp verwendet wird. Hierfür wurde im Beispielpogramm zusätzlich die Ausgangsvariable "error" (Bool) angelegt.

Programmierung der Motion Control-Anwendung

Im Anschluss können Sie das Anwenderprogramm für Ihre Motion Control-Anwendung in den Technologieobjekt-spezifischen Programmbausteinen erstellen. Dabei können Sie, wie gewohnt, absolut auf Variablen der über die Eingangsvariablen dargestellten spezifischen Technologieobjekte zugreifen (1) oder Motion Control-Anweisungen auf die Technologieobjekte ausführen (2).

3 Übergabe von Technologieobjekten mit "DB_ANY"

Abbildung 3-5 Anwenderprogramm im Technologieobjekt-spezifischen Programmbaustein

| Name | Data type | Default value | Retain |
|---------------|--------------|---------------|------------|
| Input | | | |
| axis | TO_SpeedAxis | | |
| <Add new> | | | |
| Output | | | |
| speedOutput | LReal | 0.0 | Non-retain |
| <Add new> | | | |
| InOut | | | |
| <Add new> | | | |
| Static | | | |
| instPowerAxis | MC_POWER | | |
| <Add new> | | | |

| IF... | CASE... OF... | FOR... TO DO... | WHILE... DO... | (*...*) | REGION |
|-------|---------------|-----------------|----------------|---------|--------|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |

```
1 //write actual axis speed to output
2 #speedOutput := #axis.ActualSpeed;
3
4
5 //call MC_Power for axis
6 #instPowerAxis(Axis := #axis);
```

4 Technologieobjekte mit "REF_TO"

Referenzen bieten bei den Technologieobjekten der SIMATIC Steuerungen bei Motion Control Anwendungen die Möglichkeit, die direkte (absolute) Zuordnung der Technologieobjekte im Programm während der Programmierphase durch eine flexible Übergabe der Technologieobjekte zur Laufzeit des Programms in der SIMATIC Steuerung zu ersetzen. Dabei müssen jedoch ein paar Besonderheiten berücksichtigt werden, um ein optimales Ergebnis zu erzielen.

Hinweis

Referenzen auf Technologieobjekte sind erst ab der TIA Portal Version V18 mit SIMATIC S7-1500(T) Steuerungen mit der Firmware V3.0 und Motion Control-Version V7.0 nutzbar. Wenn Sie eine frühere TIA Portal Version, CPU-Firmware oder Motion Control-Version verwenden, können Sie keine Referenzen auf Technologieobjekten nutzen. Verwenden Sie in diesem Fall den Datentyp "DB_ANY" wie in Kapitel 3 beschrieben.

4.1 Funktionsweise

In einem Programmbaustein, der mit verschiedenen Technologieobjekten verwendet werden soll, wird eine Referenz angelegt, die auf ein für den jeweiligen Bausteinaufruf gewünschtes Technologieobjekt zeigt. Diese Referenz kann für die Verschaltung an den Motion Control-Anweisungen und auch zum Auslesen von Aktualwerten aus dem Technologieobjekt verwendet werden. Die genaue Vorgehensweise zum Anlegen und zum Arbeiten mit dieser Referenz ist in den folgenden Kapiteln beschrieben.

Hinweis

Eine Referenz ist eine Variable, die selbst keinen Wert enthält, sondern auf den Speicherort einer anderen Variablen zeigt.

4.2 Anlegen der Variablen im Bausteininterface

Bevor Sie ein Technologieobjekt über eine Referenz verwenden können, müssen Sie erst die benötigten Variablen im Bausteininterface des jeweiligen Programmbausteins anlegen.

Abbildung 4-1 Interface Funktionsbaustein

| MotionProgram_REF | | |
|-------------------|------------------|---------------------|
| | Name | Data type |
| 1 | Input | |
| 2 | axis | DB_ANY |
| 3 | Output | |
| 4 | <Add new> | |
| 5 | InOut | |
| 6 | <Add new> | |
| 7 | Static | |
| 8 | <Add new> | |
| 9 | Temp | |
| 10 | tempRefSpeedAxis | REF_TO TO_SpeedAxis |
| 11 | <Add new> | |

Um ein Technologieobjekt an den Programmbaustein zu übergeben, legen Sie eine Eingangsvariable mit dem Datentypen "DB_ANY" im Bereich der Eingangsparameter "Input" an (1).

Hinweis

Auch bei der Verwendung von "REF_TO" wird das Technologieobjekt über den Datentyp "DB_ANY" an den Baustein übergeben. Zur besseren Unterscheidung der beiden Varianten wird die Methode in diesem Dokument trotzdem nur als "REF_TO" bezeichnet.

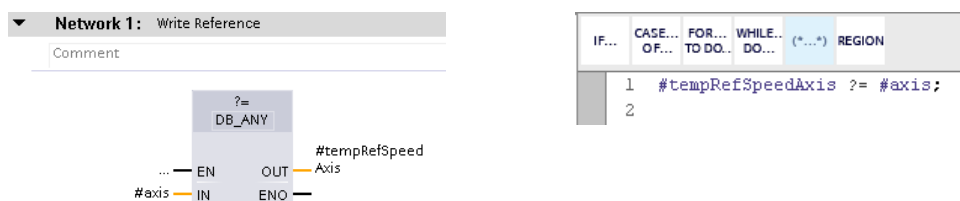
Legen Sie zusätzlich im Bereich "Temp" eine temporäre Variable mit dem Datentypen "REF_TO TO_SpeedAxis", also eine Referenz auf ein Technologieobjekt Drehzahlachse, an (2).

4.3 Verwenden der Referenzen mit Technologieobjekten

Damit Sie das Technologieobjekt über die Referenz im Programm verwenden können, muss der Referenz zuerst das über den Eingangsparameter "axis" mit dem Datentypen "DB_ANY" übergebene Technologieobjekt zugewiesen werden. Hierfür verwenden Sie die Funktion "Zuweisungsversuch" ("?=").

Der Datentyp einer Referenzvariablen wird zum Zeitpunkt der Deklaration festgelegt. Der Datentyp einer DB_ANY-Variablen wird erst zur Laufzeit ermittelt. Eine implizierte Datentypkonvertierung ist daher bei Referenzvariablen nicht zulässig. Um eine Variable mit dem Datentyp "DB_ANY" einer Referenz zuzuweisen, muss daher der "Zuweisungsversuch" ("?=") verwendet werden. Beim Zuweisungsversuch wird zur Laufzeit geprüft, ob die zugewiesene Variable den zur Referenz passenden Datentypen hat. Ist dies der Fall, wird die Zuweisung ausgeführt und es steht eine gültige Referenz in der Zielvariablen. Andernfalls steht "NULL" in der Zielvariablen (Referenz).

Abbildung 4-2 Zuweisung der Technologieobjekte auf Referenzen im FUP- und SCL-Programm



Hinweis

Bei Referenzen gilt auch das Schalenkonzept für Technologieobjekte (s. Kapitel 2.1.2). Daher können Sie einer Referenz mit dem Datentypen "TO_SpeedAxis" auch die anderen Achs-Datentypen "TO_PositioningAxis" und "TO_SynchronousAxis" zuweisen.

Nachdem Sie den Zuweisungsversuch des Technologieobjekts auf die Referenz durchgeführt haben, müssen Sie die Zuweisung vor der ersten Verwendung auf den Wert "NULL" überprüfen. Die Referenz behält den Wert "NULL", wenn die Zuweisung nicht funktioniert hat. Ist sie "ungleich NULL", können Sie die Referenz im Programm verwenden, ohne einen Programmierfehler im Programm zu riskieren.

Hinweis

Bei Zugriffen auf Referenzen mit dem Wert "NULL" wird von der Steuerung ein Programmierfehler ausgegeben und die Steuerung wechselt ggf. in den Betriebszustand "STOP".

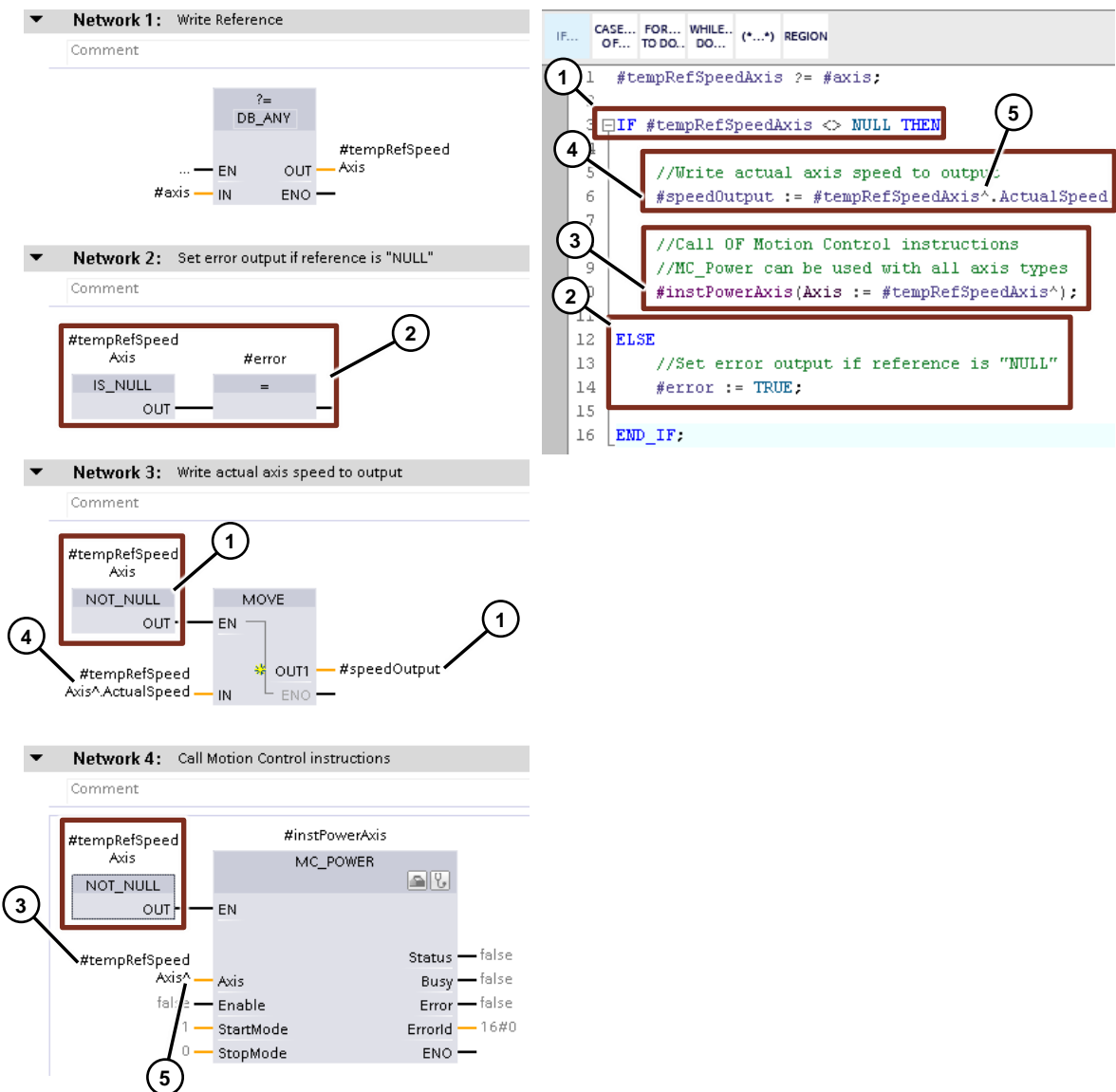
Um diese Überprüfung durchzuführen, steht Ihnen im TIA Portal für die FUP-Programmierung die Anweisung "NOT_NULL" zur Verfügung, die "TRUE" ausgibt, wenn die Referenz einen Wert

ungleich "<>" "NULL" hat. Für die SCL-Programmierung können Sie die Referenz über eine Vergleichsanweisung einen Wert ungleich "<>" oder gleich "=" "NULL" abfragen. Wenn die Zuweisung fehlgeschlagen ist, weil beispielsweise kein Technologieobjekt am entsprechenden Bausteineingang verschaltet ist, und die Referenz nach dem Zuweisungsversuch noch den Wert "NULL" hat, können Sie im Programm entsprechend darauf reagieren und z. B. einen Fehler am Baustein ausgeben (2).

Um die Referenz anschließend im Programm einsetzen zu können, muss sie jeweils dereferenziert werden, um auf die referenzierten Daten zuzugreifen. Hierzu verwenden Sie das Caret-Zeichen "^".

Beispielsweise kann die Referenz mit Hilfe des Caret-Zeichens (5) direkt an der Motion Control-Anweisung "MC_Power" verschaltet werden (3). Weiterhin werden beim Variablenzugriff über das Caret-Zeichen (5) die Daten der Referenz für den Zugriff adressierbar. So kann beispielsweise direkt auf die aktuelle Geschwindigkeit der Drehzahlachse (4) zugegriffen werden.

Abbildung 4-3 Verwenden der Referenzierten Variablen im FUP- und SCL-Programm



Hinweis

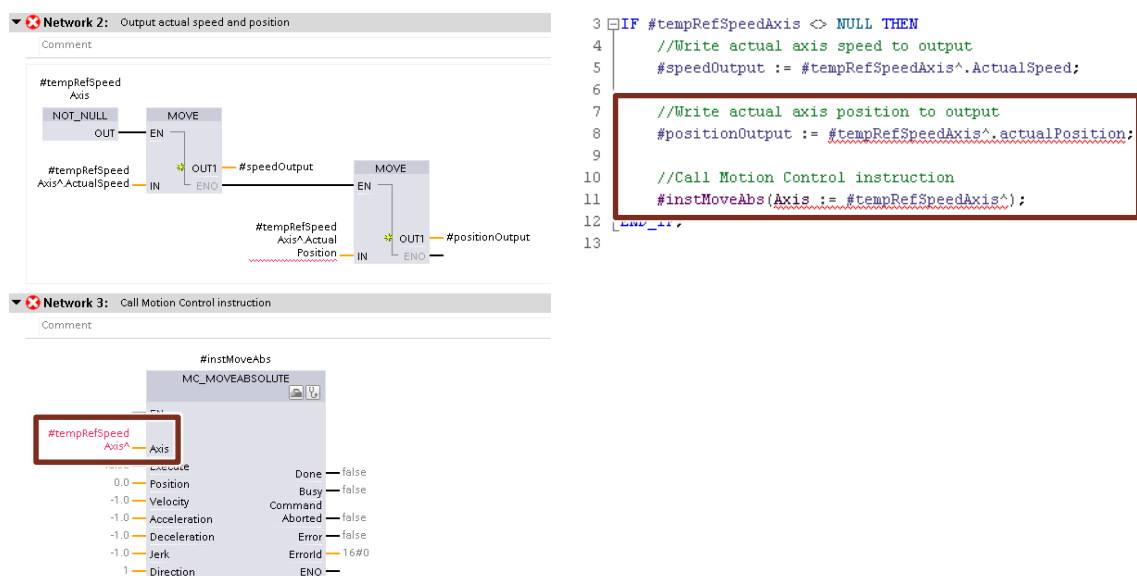
Über das Dereferenzieren können Referenzen auf Technologieobjekte im Programm, genau wie absolut adressierte Technologieobjekte bzw. Variablen aus dem Technologie-Datenbaustein, direkt verwendet werden.

4.4 Verwenden von Referenzen für verschiedene Technologieobjekte

Referenzen können nach dem Dereferenzieren genau wie die absolut adressierten Technologieobjekte verwendet werden. Dadurch ergeben sich allerdings auch die folgenden Einschränkungen, wenn z. ein Technologieobjekt Positionierachse "TO_PositioningAxis" einer Referenz mit dem Datentypen "REF_TO TO_SpeedAxis" zugewiesen ist:

- Die Referenz kann nur für den Zugriff auf Variablen verwendet werden, die im Datentypen der Drehzahlachse "TO_SpeedAxis" enthalten sind. Der Zugriff auf die spezifischen Variablen einer Positionierachse, wie beispielsweise die Istposition, ist mit dieser Referenz nicht möglich.
- Die Referenz kann nur an Motion Control-Anweisungen verschaltet werden, an denen auch eine Drehzahlachse verschaltet werden kann. Die Verschaltung dieser Referenz an Motion Control-Anweisung wie z. B. "MC_MoveAbsolute" ist nicht möglich.

Abbildung 4-4 Aufruf und Variablengriff nicht möglich mit Referenz auf Drehzahlachse



Soll der Programmbaustein also unter anderem mit Drehzahlachsen verwendet werden können, aber der Variablengriff auf die Variablen einer Positionier- oder Gleichlaufachse und die Verwendung entsprechender Motion Control-Anweisungen möglich sein, wird, zusätzlich zu der bereits vorhandenen Referenz mit dem Datentypen "TO_SpeedAxis", eine Referenz mit den entsprechenden Datentypen "TO_PositioningAxis" oder "TO_SynchronousAxis" benötigt.

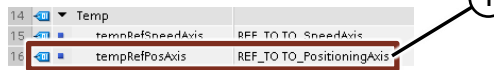
Legen Sie in diesem Fall im Bausteininterface eine weitere Referenz mit dem Datentypen "REF_TO TO_PositioningAxis" an (1). Führen Sie für diese Referenz wieder einen Zuweisungsversuch der Achse ("DB_ANY") im Programm durch (2).

Sie können den im vorherigen Kapitel erstellten Programmteil (3), der die Anweisungen und Variablengriffe der Referenz mit dem Datentypen "REF_TO TO_SpeedAxis" beinhaltet, weiterhin verwenden, da dieser auch mit den übergeordneten Achstypen "TO_PositioningAxis" und "TO_SynchronousAxis" funktioniert. Es müssen also nur die Positionierachs-spezifischen Variablengriffe (4) und (Motion Control-) Anweisungen (5) im Programm ergänzt werden. Für

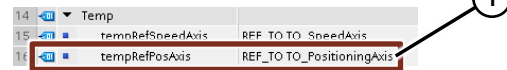
diesen Programmteil verwenden Sie die neu angelegte Referenz mit dem Datentypen "REF_TO TO_PositioningAxis".
 Vor dem Zugriff auf das über diese Referenz referenzierte Technologieobjekt müssen Sie allerdings auch wieder die Gültigkeit der jeweiligen Referenz durch die Abfrage auf "NULL" (6) prüfen.

Abbildung 4-5 Zugriff auf Variablen verschiedener Technologieobjekt-Typen

Bausteininterface:

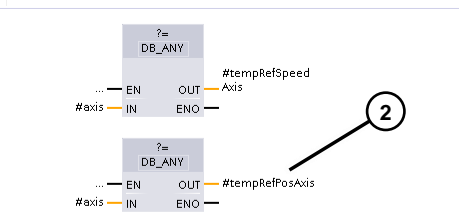


Bausteininterface:

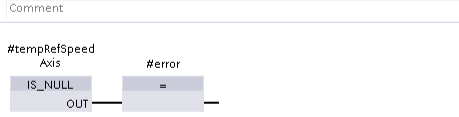


Programmierung:

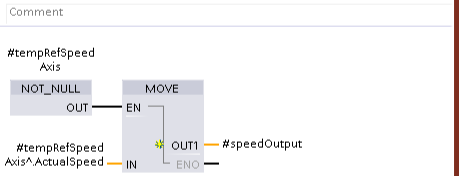
Network 1: Write Reference



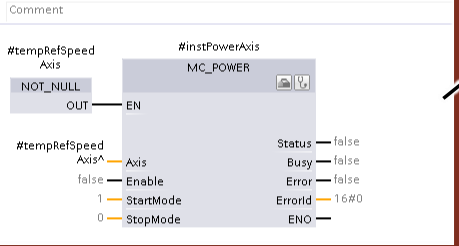
Network 2: Set error output if reference is "NULL"



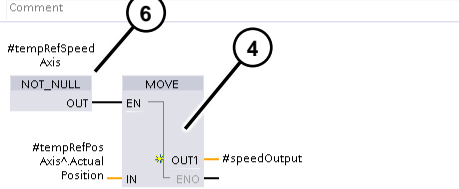
Network 3: Write actual axis speed to output



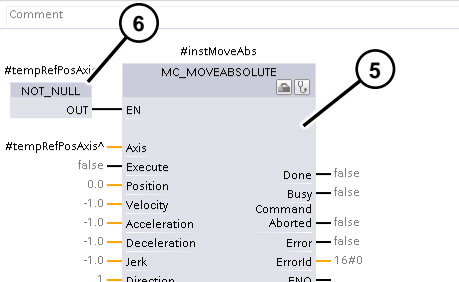
Network 4: Call Motion Control instructions with reference "TO_SpeedAxis"



Network 5: Write actual axis position to output



Network 6: Call Motion Control instr... #positionOutput "TO_PositioningAxis"



Programmierung:

```

1  #tempRefSpeedAxis ?= #axis;
2
3  IF #tempRefSpeedAxis <> NULL THEN
4
5      //Write actual axis speed to output
6      #speedOutput := #tempRefSpeedAxis^.ActualSpeed;
7
8      //Call OF Motion Control instructions
9      //MC_Power can be used with all axis types
10     #instPowerAxis(Axis := #tempRefSpeedAxis^);
11
12     //Assign PosAxis reference
13     #tempRefPosAxis ?= #axis;
14
15     IF #tempRefPosAxis <> NULL THEN
16
17         //Write actual position to output
18         #positionOutput := #tempRefPosAxis^.Position;
19
20         //Call MC_MoveAbsolute
21         #instMoveAbs(Axis := #tempRefPosAxis^);
22     END_IF;
23 ELSE
24     //Set error output if reference is "NULL"
25     #error := TRUE;
26
27 END_IF;
    
```


Verschalten Sie nun am Baustein eine Drehzahlachse, kann nur die Referenz mit dem Datentypen "TO_SpeedAxis" zugewiesen werden und es werden nur die Variablenzuweisung der aktuellen Geschwindigkeit und die Motion Control-Anweisung "MC_Power" ausgeführt (3). Die Referenz mit dem Datentypen "TO_PositioningAxis" behält den Wert "NULL" und der Programmteil für die Positionierachse wird in diesem Fall nicht ausgeführt.

Wenn Sie am Baustein eine Positionier- oder Gleichlaufachse verschalten, können beide Referenzen zugewiesen werden und das ganze Programm wird durchlaufen. Es werden also zusätzlich die Variablenzuweisung der Sollposition (4) und die Motion Control-Anweisung "MC_MoveAbsolute" (5) ausgeführt.

Eine Übersicht der Referenzwerte bei Verwendung der verschiedenen Achs-Datentypen finden Sie in der folgenden Tabelle:

Tabelle 4-1 Zuweisungsergebnisse bei verschiedenen Achs-Technologieobjekten

| Technologieobjekt | REF_TO TO_SpeedAxis | REF_TO TO_PositioningAxis | REF_TO TO_SynchronousAxis |
|--|-----------------------------------|-----------------------------------|-----------------------------------|
| Kein Achs-Technologieobjekt | NULL (nicht verwendbar) | NULL (nicht verwendbar) | NULL (nicht verwendbar) |
| Drehzahlachse "TO_SpeedAxis" | Referenz auf Technologieobjekt | NULL (nicht verwendbar) | NULL (nicht verwendbar) |
| Positionierachse "TO_PositioningAxis" | Referenz auf Technologieobjekt | Referenz auf Technologieobjekt | NULL (nicht verwendbar) |
| Gleichlaufachse "TO_SynchronousAxis" | Referenz auf Technologieobjekt | Referenz auf Technologieobjekt | Referenz auf Technologieobjekt |

Hinweis

Es müssen immer nur Referenzen mit den Datentypen angelegt werden, von denen später auch Technologieobjekte mit dem Programmbaustein verwendet werden sollen. Soll ein Baustein beispielsweise nur mit Positionier- und Gleichlaufachsen verwendet werden, ist es nicht zwingend nötig, eine Referenz mit dem Datentypen "TO_SpeedAxis" anzulegen. Es kann stattdessen eine Referenz auf eine Positionierachse "TO_PositioningAxis" verwendet werden.

4.5 Tipps für die Verwendung von "REF_TO"

4.5.1 Nutzung von "RETURN"

Um nicht einzeln vor jeder Verwendung zu prüfen, ob der Zuweisungsversuch erfolgreich war, können Sie, einmalig nach dem Zuweisungsversuch, prüfen, ob die Zuweisung erfolgreich war und die Referenz den Wert "NULL" hat oder gültig ist und ein Technologieobjekt referenziert.

- Enthält die Referenz den Wert "NULL", können Sie einen Fehler am Baustein ausgeben, da kein gültiges Technologieobjekt verschaltet wurde und die Bausteinbearbeitung mit "RETURN" beenden (1), sodass die Programmstellen, an denen die Referenz verwendet wird, nicht ausgeführt werden.
- Ist die Referenz gültig und hat nicht den Wert "NULL" können Sie die Bausteinbearbeitung fortsetzen und müssen die Überprüfung im Folgeprogramm nicht mehr durchführen, da durch die Prüfung nach dem Zuweisungsversuch bereits sichergestellt ist, dass die Referenz ein Technologieobjekt referenziert (2).

Bei der Verwendung von mehreren Referenzen, die verschiedene Technologieobjekt-Typen referenzieren (s. Kapitel 4.4), können Sie diese Überprüfung am Bausteinbeginn nur für eine der Referenzen durchführen.

Führen Sie die Überprüfung in diesem Fall immer für den Technologieobjekt-Typ mit dem kleinsten Variablen- bzw. Funktionsumfang durch, da wenn diese nicht zugewiesen werden

kann auch die übergeordneten Technologieobjekt-Typen nicht zugeordnet werden können. Bei der Verwendung der weiteren Referenzen müssen Sie weiterhin vor jeder Verwendung sicherstellen, dass die jeweilige Referenz gültig ist und ein Technologieobjekt referenziert (3).

Abbildung 4-6 Überprüfung der Referenz nur zu Bausteinbeginn

Network 1: Write Reference

Comment

DB_ANY

EN OUT #tempRefSpeed

#axis IN ENO

DB_ANY

EN OUT #tempRefPosAxis

#axis IN ENO

Network 2: Set error output if reference is "NULL"

Comment

#tempRefSpeed Axis

IS_NULL OUT

#error =

RLO RET

Network 3: Write actual axis speed to output

Comment

MOVE

EN OUT1 #tempRefSpeed Axis^ ActualSpeed

#speedOutput

ENO

Network 4: Call Motion Control instructions with reference "TO_SpeedAxis"

Comment

#instPowerAxis

MC_POWER

EN

#tempRefSpeed Axis^ Axis

Status false

Busy false

Enable false

Error false

StartMode 1 ErrorId 16#0

StopMode 0 ENO

Network 5: Write actual axis position to output

Comment

#tempRefSpeed Axis

NOT_NULL OUT

MOVE

EN OUT1 #tempRefPos Axis^ Actual Position

#speedOutput

ENO

Network 6: Call Motion Control instructions with reference "TO_PositioningAxis"

Comment

#instMoveAbs

MC_MOVEABSOLUTE

EN

#tempRefPosAxis Axis

Execute false

Done false

Position 0.0 Busy false

Velocity -1.0 Command false

Acceleration -1.0 Aborted false

Deceleration -1.0 Error false

Jerk -1.0 ErrorId 16#0

Direction 1 ENO

```

1 #tempRefSpeedAxis ?= #axis;
2
3 IF #tempRefSpeedAxis = NULL THEN
4   #error := TRUE;
5   RETURN;
6 END_IF;
7
8 //Write actual axis speed to output
9 #speedOutput := #tempRefSpeedAxis^.ActualSpeed;
10
11 //Call OF Motion Control instructions
12 //MC_Power can be used with all axis types
13 #instPowerAxis(Axis := #tempRefSpeedAxis^);
14
15 //Assign PosAxis reference
16 #tempRefPosAxis ?= #axis;
17
18 IF #tempRefPosAxis <> NULL THEN
19
20 //Write actual position to output
21 #positionOutput := #tempRefPosAxis^.Position;
22
23 //Call MC_MoveAbsolute
24 #instMoveAbs(Axis := #tempRefPosAxis^);
25 END_IF;
26
27 //Further program....
28

```

4.5.2 Gruppieren von Zugriffen über Referenzen

Um nicht einzeln vor jedem Zugriff überprüfen zu müssen, ob die Referenz gültig ist, können Sie die Zugriffe auf Variablen über die gleiche Referenz gruppieren und die Abfrage nur einmal für jede Gruppe durchführen.

Hinweis

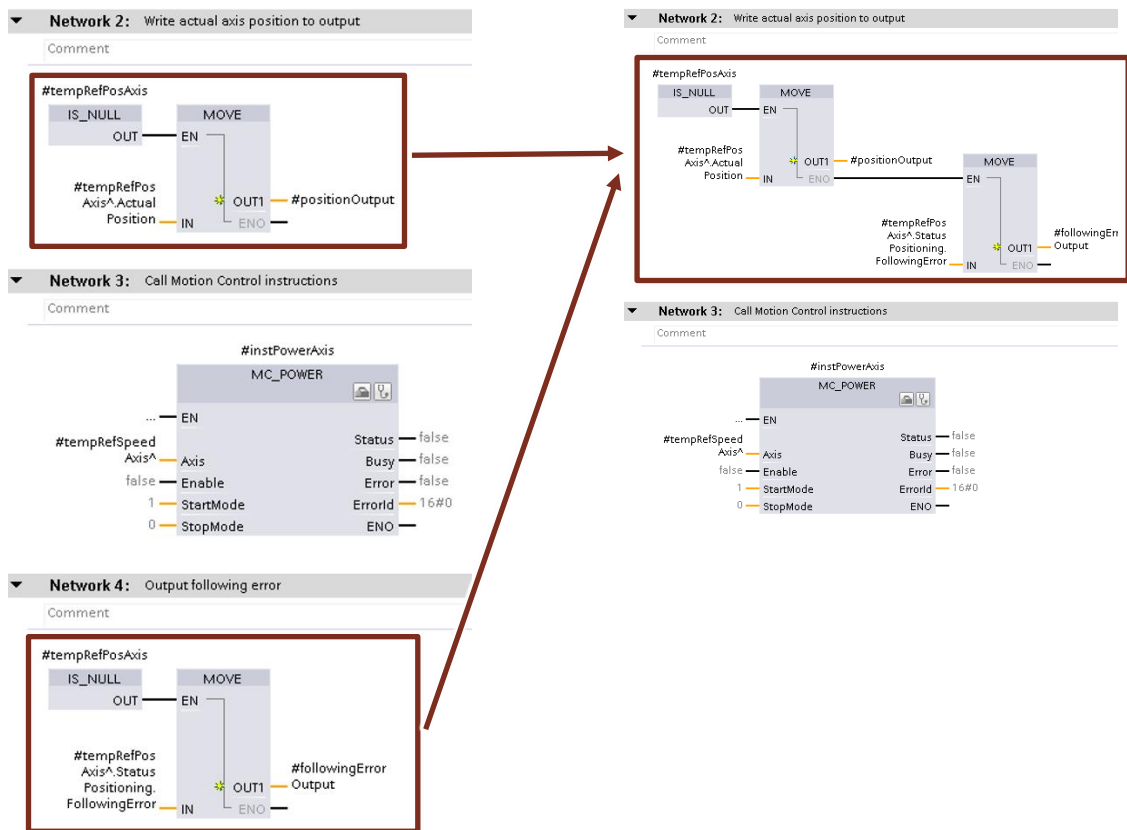
Es empfiehlt sich, Variablen, die im Programm öfter verwendet werden, auf lokale Variablen, beispielsweise im temporären Bereich eines Programmbausteins, zu kopieren und in der weiteren Bearbeitung die Kopien zu verwenden. Dies hat zwei Vorteile:

- Für die temporären Variablen muss keine Prüfung auf Gültigkeit durchgeführt werden.
- Der Zugriff auf die (System-)Variablen eines Technologieobjekts dauert länger als der Zugriff auf lokale Variablen eines Programmbausteins. Daher kann hier bei mehrfachem Zugriff auf Variablen die Bearbeitungszeit eines Bausteins reduziert werden.

(KOP-/)FUP-Programmierung

Nutzen Sie für die Gruppierung der Zugriffe bei der KOP- oder FUP-Programmierung die EN- und ENO-Parameter der Programmanweisungen.

Abbildung 4-7 Gruppierung von Variablenzugriffen über eine Referenz bei FUP-Programmierung



SCL-Programmierung

Nutzen Sie für die Gruppierung der Zugriffe bei der SCL-Programmierung eine IF-Anweisung, in der Sie alle benötigten Zugriffe durchführen.

Abbildung 4-8 Gruppierung von Variablenzugriffen über eine Referenz bei SCL-Programmierung

```

11 //write actual axis position to output
12 IF #tempRefPosAxis <> NULL THEN
13     #positionOutput := #tempRefPosAxis^.ActualPosition;
14 END_IF;
15
16 //Call of Motion Control instructions
17 //MC_Power can be used with all axis types
18 #instPowerAxis(Axis := #tempRefSpeedAxis^);
19
20 IF #tempRefPosAxis <> NULL THEN
21     //Output following error
22     #followingErrorOutput := #tempRefPosAxis^.StatusPositioning.FollowingError;
23 END_IF;
24
25
26

```

```

17 IF #tempRefPosAxis <> NULL THEN
18     //Output actual position
19     #positionOutput := #tempRefPosAxis^.ActualPosition;
20     //Output following error
21     #followingErrorOutput := #tempRefPosAxis^.StatusPositioning.FollowingError;
22 END_IF;
23
24 //Call of Motion Control instructions
25 //MC_Power can be used with all axis types
26 #instPowerAxis(Axis := #tempRefSpeedAxis^);
27

```

5 Programmierertipps

5.1 Organisation von Technologieobjekten in einem Array

Möchten Sie die Technologieobjekte eines Projekts zentral zusammenfassen und verwalten, können Sie diese mit Hilfe des Datentyps "DB_ANY" in einem Array, beispielsweise in einem eigenen Datenbaustein, speichern. Den Zugriff auf die einzelnen Elemente können Sie dann sehr komfortabel über den Array-Index durchführen.

Auf diese Weise können Sie innerhalb des Anwenderprogramms die Gruppierung oder Reihenfolge der Technologieobjekte vorgeben und über den Array-Index auch in der HMI-Bedienoberfläche Technologieobjekte auswählen. Zusätzlich sind bei dieser Organisation auch Massenoperationen über den Array-Index auf alle enthaltenen Technologieobjekte sehr einfach möglich.

Die nachfolgende Abbildung zeigt ein Array von Technologieobjekten in einem Datenbaustein.

Abbildung 5-1 Organisation von Achs-Technologieobjekten in einem Array

| AxisDB | | | | | | | | | | | |
|--------|---------|-----------------------|-------------|--------|---------------|----------|----------------|----------|------------|---------|--|
| | Name | Data type | Start value | Retain | Accessible... | Writa... | Visible in ... | Setpoint | Supervi... | Comment | |
| 1 | Static | | | | | | | | | | |
| 2 | axis | Array[0..9] of DB_ANY | 0 | | | | | | | | |
| 3 | axis[0] | DB_ANY | 0 | | | | | | | | |
| 4 | axis[1] | DB_ANY | 0 | | | | | | | | |
| 5 | axis[2] | DB_ANY | 0 | | | | | | | | |
| 6 | axis[3] | DB_ANY | 0 | | | | | | | | |
| 7 | axis[4] | DB_ANY | 0 | | | | | | | | |
| 8 | axis[5] | DB_ANY | 0 | | | | | | | | |
| 9 | axis[6] | DB_ANY | 0 | | | | | | | | |
| 10 | axis[7] | DB_ANY | 0 | | | | | | | | |
| 11 | axis[8] | DB_ANY | 0 | | | | | | | | |
| 12 | axis[9] | DB_ANY | 0 | | | | | | | | |

Die Technologieobjekte können Sie beispielsweise im OB100 "Startup" einmalig den Werten im Datenbaustein zuweisen. Auf diese Weise entsteht keine CPU-Last durch zyklische Zuweisungen und Sie können die Technologieobjekte im Anwenderprogramm direkt verwenden.

Abbildung 5-2 Zuweisung des Achs-Datenbausteins im OB100 "Startup"

```

Startup
IF... CASE... FOR... WHILE... (*..*) REGION
OF... TO DO... DO...

1 //Assign axes to AxisDB
2 //Speed axes
3 "AxisDB".axis[0] := "SpeedAxis_1";
4 "AxisDB".axis[1] := "SpeedAxis_2";
5 "AxisDB".axis[2] := "SpeedAxis_3";
6
7 //Positioning axes
8 "AxisDB".axis[3] := "PositioningAxis_1";
9 "AxisDB".axis[4] := "PositioningAxis_2";
10 "AxisDB".axis[5] := "PositioningAxis_3";
11
12 //SynchronousAxes
13 "AxisDB".axis[6] := "SynchronousAxis_1";
14 "AxisDB".axis[7] := "SynchronousAxis_2";
15 "AxisDB".axis[8] := "SynchronousAxis_3";
16 "AxisDB".axis[9] := "SynchronousAxis_4";

```

Hinweis

Sie können das Array auch über einen eigenen PLC-Datentypen definieren und den einzelnen Einträgen zusätzlich zum Technologieobjekt ("DB_ANY"-Variable) noch weitere Informationen hinzufügen. Hier können Sie beispielsweise den Anzeigenamen des Technologieobjekts auf dem HMI oder eine Zugriffsoption für den Bediener, das Einricht- oder das Servicepersonal hinterlegen.

5.2 Massenoperationen über den Array-Index

Wenn Sie die Technologieobjekte zentral in einem Array organisiert haben, können Sie über den Array-Index sehr einfach Massenoperationen auf alle enthaltenen Technologieobjekte ausführen.

Die nachfolgenden beiden Beispiele zeigen das Quittieren von Fehlermeldungen an allen Technologieobjekten (im Beispiel sind nur Achs-Technologieobjekte berücksichtigt) eines Arrays über die Motion Control-Anweisung "MC_Reset".

5.2.1 Massenoperation "MC_Reset" mit "DB_ANY"

Programmieren der Funktion

Um Massenoperationen über "DB_ANY" auf Technologieobjekte auszuführen, muss zuerst ein Funktionsbaustein für jeden Achstypen angelegt werden, der die relevanten Signale auswertet und verarbeitet bzw. die relevanten Statussignale ausgibt. Dieser Baustein wird für jeden Achstypen benötigt, für den der Baustein verwendet werden soll.

Abbildung 5-3 Programm "MC_Reset"

| ResetPositioningAxis | | | |
|----------------------|-----------|--------------------|---------------|
| | Name | Data type | Default value |
| 1 | Input | | |
| 2 | axis | TO_PositioningAxis | |
| 3 | execute | Bool | false |
| 4 | <Add new> | | |
| 5 | Output | | |
| 6 | busy | Bool | false |
| 7 | done | Bool | false |
| 8 | error | Bool | false |
| 9 | errorID | Word | 16#0 |
| 10 | <Add new> | | |
| 11 | InOut | | |
| 12 | <Add new> | | |
| 13 | Static | | |
| 14 | instReset | MC_RESET | |
| 15 | <Add new> | | |


```

1 #instReset(Axis := #axis,
2             Execute:=#execute,
3             Done=>#done,
4             Busy=>#busy,
5             Error=>#error,
6             ErrorId=>#errorID);
7

```

Für das hier aufgeführte Beispiel, bei dem Fehler an allen Achsen quittiert werden sollen, ist es ausreichend, die Motion Control-Anweisung aufzurufen, mit dem jeweiligen Technologieobjekt-Typen zu verschalten und die benötigten Signale mit dem Eingangs- bzw. Ausgangsinterface des Funktionsbausteins zu verschalten.

Dieser Baustein wird für jeden Technologieobjekt-Typ benötigt. Der Datentyp des Eingangsparameters "axis" muss entsprechend angepasst werden. In diesem Fall sind das die Datentypen:

- "TO_SpeedAxis"
- "TO_PositioningAxis"
- "TO_SynchronousAxis"

Bausteininterface des "Schalenbausteins" zur Typkonvertierung

Abbildung 5-4 Bausteininterface Massenoperationen "DB_ANY"

| ResetAllAxes | | |
|--------------------|---|---------------|
| Name | Data type | Default value |
| Input | | |
| axes | Array[#ARRAY_MIN..#ARRAY_MAX] of DB_ANY | |
| execute | Bool | false |
| <Add new> | | |
| Output | | |
| busy | Bool | false |
| done | Bool | false |
| error | Bool | false |
| errorID | Word | 16#0 |
| <Add new> | | |
| InOut | | |
| <Add new> | | |
| Static | | |
| instResetPosAxis | Array[#ARRAY_MIN..#ARRAY_MAX] of "ResetPositioningAxis" | |
| instResetSpeedAxis | Array[#ARRAY_MIN..#ARRAY_MAX] of "ResetSpeedAxis" | |
| instResetSynchAxis | Array[#ARRAY_MIN..#ARRAY_MAX] of "ResetSynchronousAxis" | |
| <Add new> | | |
| Temp | | |
| tempIdx | USInt | |
| tempErrorArray | Array[#ARRAY_MIN..#ARRAY_MAX] of Bool | |
| <Add new> | | |
| Constant | | |
| ARRAY_MIN | USInt | 0 |
| ARRAY_MAX | USInt | 9 |

Über die Eingangsvariable "axes" können Sie das Array, das die im Projekt vorhandenen Achsen abbildet, an den Baustein übergeben. Über den Eingangsparameter "execute" können Sie das Rücksetzen der Fehler an allen Achsen starten.

Die angelegten Ausgangsvariablen bilden einen Teil der Ausgangsvariablen der Motion Control-Anweisung "MC_Reset" ab und geben Ihnen gesammelt Informationen über den Status der im Baustein für die Technologieobjekte verwendeten Instanzen der Motion Control-Anweisung "MC_Reset".

Im Bereich der statischen Variablen sind die Instanzen der achsspezifischen Funktionsbausteine, die den Aufruf der Motion Control-Anweisung "MC_Reset" enthalten, in einem Array mit der gleichen Länge, wie auch die an den Baustein übergebenen Achsen angelegt.

Im Bereich der temporären Variablen ist eine temporäre Zählvariable für den Array-Index angelegt, die Sie für die Programmierung der Schleifen verwenden können.

Um die Anzahl der Achsen flexibel anpassen zu können, empfiehlt es sich für die Array-Grenzen Konstanten anzulegen und diese für die Konfiguration und auch später im Programm zu verwenden. Somit können Sie die Array-Grenzen in allen Variablendefinitionen und im Programm durch Anpassung dieser Konstanten ändern.

Hinweis

Wenn Sie die Achsen bzw. die Array-Grenzen in verschiedenen Bausteinen im Programm verwenden, können Sie die Array-Grenzen auch als Anwenderkonstanten in den PLC-Variablen anlegen. In diesem Fall können Sie die Konstanten global in allen Bausteinen verwenden und müssen bei einer Änderung nur die Definition der Konstanten an einer Stelle anpassen.

Programmierung des Schalenbausteins zur Typkonvertierung

Abbildung 5-5 Programmierung Maschinerieoperation "DB_ANY"

```

1 //Set initial output values
2 #done := true;
3 #busy := false;
4 #error := false;
5 #errorID := 16#0;
6
7 FOR #tempIdx := #ARRAY_MIN TO #ARRAY_MAX DO
8   CASE TypeOfDB(#axes[#tempIdx]) OF
9     TO_SpeedAxis: //Speed axis program
10      //Call axis specific reset function block
11      #instResetSpeedAxis[#tempIdx](axis := #axes[#tempIdx],
12                                     execute := #execute);
13
14      //done should be set when all instances are done
15      //reset done output if at least one instance of MC_Reset is not done
16      IF NOT #instResetSpeedAxis[#tempIdx].done THEN
17        #done := false;
18      END_IF;
19
20      //busy output should be set if at least one instance of MC_Reset is busy
21      //set busy output if at least one instance of MC_Reset is busy
22      IF #instResetSpeedAxis[#tempIdx].busy THEN
23        #busy := true;
24      END_IF;
25
26      //set error if there was an error at the Motion Control instruction "MC_Reset"
27      IF #instResetSpeedAxis[#tempIdx].error THEN
28        #error := true;
29        #errorID := #instResetSpeedAxis[#tempIdx].errorID;
30      END_IF;
31
32
33     TO_PositioningAxis: //Positioning axis program
34      //Call axis specific reset function block
35      #instResetPosAxis[#tempIdx](axis := #axes[#tempIdx],
36                                   execute := #execute);
37
38      //done should be set when all instances are done
39      //reset done output if at least one instance of MC_Reset is not done
40      IF NOT #instResetPosAxis[#tempIdx].done THEN
41        #done := false;
42      END_IF;
43
44      //busy output should be set if at least one instance of MC_Reset is busy
45      //set busy output if at least one instance of MC_Reset is busy
46      IF #instResetPosAxis[#tempIdx].busy THEN
47        #busy := true;
48      END_IF;
49
50      //set error if there was an error at the Motion Control instruction "MC_Reset"
51      IF #instResetPosAxis[#tempIdx].error THEN
52        #error := true;
53        #errorID := #instResetPosAxis[#tempIdx].errorID;
54      END_IF;
55
56     TO_SynchronousAxis: //Synchronous axis program
57      //Call axis specific reset function block
58      #instResetSynchAxis[#tempIdx](axis := #axes[#tempIdx],
59                                     execute := #execute);
60
61      //done should be set when all instances are done
62      //reset done output if at least one instance of MC_Reset is not done
63      IF NOT #instResetSynchAxis[#tempIdx].done THEN
64        #done := false;
65      END_IF;
66
67      //busy output should be set if at least one instance of MC_Reset is busy
68      //set busy output if at least one instance of MC_Reset is busy
69      IF #instResetSynchAxis[#tempIdx].busy THEN
70        #busy := true;
71      END_IF;
72
73      //set error if there was an error at the Motion Control instruction "MC_Reset"
74      IF #instResetSynchAxis[#tempIdx].error THEN
75        #error := true;
76        #errorID := #instResetSynchAxis[#tempIdx].errorID;
77      END_IF;
78
79     ELSE
80       #error := true;
81     END_CASE;
82   END_FOR;

```


Im ersten Programmteil werden die Ausgangsvariablen initialisiert (1).

Anschließend wird eine FOR-Schleife von der unteren bis zur oberen Array-Grenze durchlaufen (2). In dieser FOR-Schleife wird für jedes Element des Arrays mit den Technologieobjekten der Typ geprüft (3) und der entsprechende achsspezifische Funktionsbaustein aufgerufen (4).

Falls das jeweilige Array-Element nicht einem der erwarteten Technologieobjekt-Typen entspricht, wird ein Fehler vom Baustein ausgegeben (5).

Außerdem werden nach der Typprüfung die Ausgangsvariablen entsprechend des aktuellen Bearbeitungsstatus gesetzt:

- Der Ausgang "done" wird auf "true" initialisiert und zurückgesetzt, wenn mindestens eine der Instanzen der Motion Control-Anweisung "MC_Reset" nicht den Wert "true" am Ausgang "done" hat (6).
- Der Ausgang "busy" wird auf "false" initialisiert und gesetzt, sobald mindestens eine der Instanzen der Motion Control-Anweisung "MC_Reset" den Wert "true" am Ausgang "busy" hat (7).
- Der Ausgang "error" wird auf "false" initialisiert und gesetzt, wenn mindestens eine der Instanzen der Motion Control-Anweisung "MC_Reset" den Wert "true" am Ausgang "error" hat. Zusätzlich wird die "errorID" der jeweiligen Instanz auf den Ausgang "errorID" des Funktionsbausteins kopiert (8).

Hinweis

Es wird in diesem Fall immer die "errorID" der letzten Instanz der Motion Control-Anweisung "MC_Reset" ausgegeben. Fehler können so nacheinander abgearbeitet werden. Alternativ können Sie auch ein Array mit "errorID" am Baustein ausgeben und die Fehlernummern für alle Instanzen ausgeben. Dies bedeutet ggf. allerdings einen erhöhten Aufwand bei der Fehlerauswertung im Programm außerhalb des Funktionsbaustein, da ein Array statt einer einzelnen Variable ausgewertet werden muss.

Hinweis

Diese Beispielprogrammierung zeigt eine mögliche Vorgehensweise für Massenoperationen, die Sie beliebig um weitere Eingangs- oder Statusvariablen sowie weitere Funktionen (z. B. weitere Motion Control-Anweisungen oder logische Abläufe) erweitern können.

Hinweis

Bei der Bearbeitung sehr vieler Technologieobjekte oder größerer Programme innerhalb der Schleifen, kann es sinnvoll sein, die Bearbeitung auf mehrere CPU-Zyklen aufzuteilen und in jedem CPU-Zyklus nur einen Teil des Arrays bzw. der Technologieobjekte zu bearbeiten, um die Zykluszeit gering zu halten.

Hinweis

Bei der Nutzung von Massenoperationen können Sie auch die Instanzen der genutzten Programmbausteine bzw. Motion Control-Anweisungen als Parameterinstanzen in einem Array organisieren. Somit können auch die Instanzen über den Array-Index zugewiesen und die Reaktionen bei der Ausführung der Motion Control-Anweisungen oder der Bausteine verfolgt werden.

5.2.2 Massenoperation "MC_Reset" mit "REF_TO"

Bausteininterface

Um Massenoperationen über Referenzen auf Technologieobjekte auszuführen, werden die im folgenden Bild zu sehenden Variablen benötigt:

Abbildung 5-6 Bausteininterface Massenoperationen "REF_TO"

| ResetAllAxes | | | |
|--------------|-----------------|---|---------------|
| | Name | Data type | Default value |
| 1 | Input | | |
| 2 | axes | Array[#ARRAY_MIN..#ARRAY_MAX] of DB_ANY | |
| 3 | execute | Bool | false |
| 4 | Output | | |
| 5 | busy | Bool | false |
| 6 | done | Bool | false |
| 7 | error | Bool | false |
| 8 | errorID | Word | 16#0 |
| 9 | InOut | | |
| 10 | -Add new> | | |
| 11 | Static | | |
| 12 | instReset | Array[#ARRAY_MIN..#ARRAY_MAX] of MC_RESET | |
| 13 | Temp | | |
| 14 | tempIdx | USInt | |
| 15 | tempRefSpeedAis | REF_TO_TO_SpeedAxis | |
| 16 | tempErrorArray | Array[#ARRAY_MIN..#ARRAY_MAX] of Bool | |
| 17 | Constant | | |
| 18 | ARRAY_MIN | USInt | 0 |
| 19 | ARRAY_MAX | USInt | 9 |

Über die Eingangsvariable "axes" können Sie das Array, das die im Projekt vorhandenen Achsen abbildet, an den Baustein übergeben. Über den Eingangsparameter "execute" können Sie das Rücksetzen der Fehler an allen Achsen starten.

Die Angelegten Ausgangsvariablen bilden einen Teil der Ausgangsvariablen der Motion Control-Anweisung "MC_Reset" ab und geben Ihnen gesammelt Informationen über den Status aller im Baustein verwendeten Instanzen der Motion Control-Anweisung "MC_Reset".

Im Bereich der statischen Variablen sind die Instanzen der Motion Control-Anweisung in einem Array mit der gleichen Länge, die auch das an den Baustein übergebene Array mit den Achsen hat, angelegt.

Im Bereich der temporären Variablen ist die Referenz auf eine Achse "TO_SpeedAxis" angelegt. Diese Referenz wird für die Verschaltung an der Motion Control-Anweisung "MC_Reset" verwendet. Zusätzlich ist eine temporäre Variable für Fehler angelegt, in der Fehler beim Zuweisen der Referenz gespeichert werden.

Hinweis

Referenzen können nicht in einem Array angelegt werden. Daher wird in diesem Beispiel nur eine Referenz angelegt, die für alle an den Baustein übergebenen Achsen verwendet werden kann.

Außerdem wird eine temporäre Zählvariable für den Array-Index angelegt, die Sie für die Programmierung der Schleifen verwenden können.

Um die Anzahl der Achsen flexibel anpassen zu können, empfiehlt es sich für die Array-Grenzen Konstanten anzulegen und diese für die Konfiguration und auch später im Programm zu verwenden. Somit können Sie die Array-Grenzen in allen Variablendefinitionen und im Programm durch Anpassung dieser Konstanten ändern.

Hinweis

Wenn Sie die Achsen bzw. die Array-Grenzen in verschiedenen Bausteinen im Programm verwenden, können Sie die Array-Grenzen auch als Anwenderkonstanten in den PLC-Variablen anlegen. In diesem Fall können Sie die Konstanten global in allen Bausteinen verwenden und müssen bei einer Änderung nur die Definition der Konstanten an einer Stelle anpassen.

Programmierung

Um Fehler an allen an den Baustein übergebenen Achsen zu quittieren, wurde die Programmierung wie folgt umgesetzt:

Abbildung 5-7 Programmierung Massenoperation "REF_TO"

```

1 //Call and execute MC_Reset
2 FOR #tempIdx := #ARRAY_MIN TO #ARRAY_MAX DO
3   #tempRefSpeedAxis := #axes[#tempIdx];
4   IF #tempRefSpeedAxis <> NULL THEN
5     #instReset[#tempIdx].Execute := #execute;
6     #instReset[#tempIdx](Axis := #tempRefSpeedAxis^);
7   ELSE
8     #tempErrorArray[#tempIdx] := true;
9   END_IF;
10 END_FOR;

13 //Set initial output values
14 #done := true;
15 #busy := false;
16 #error := false;
17 #errorID := 16#0;
18
19 //set output values according to status
20 FOR #tempIdx := #ARRAY_MIN TO #ARRAY_MAX DO
21   //done should be set when all instances are done
22   //reset done output if at least one instance of MC_Reset is not done
23   IF NOT #instReset[#tempIdx].Done THEN
24     #done := false;
25   END_IF;
26
27   //busy output should be set if at least one instance of MC_Reset is busy
28   //set busy output if at least one instance of MC_Reset is busy
29   IF #instReset[#tempIdx].Busy THEN
30     #busy := true;
31   END_IF;
32
33   //set error if there was an error assign one of the references
34   //or at the Motion Control-instruction "MC_Reset"
35   IF #tempErrorArray[#tempIdx] OR #instReset[#tempIdx].Error THEN
36     #error := true;
37     #errorID := #instReset[#tempIdx].#ErrorId;
38   END_IF;
39 END_FOR;

```

Im ersten Programmteil wird eine FOR-Schleife von der unteren bis zur oberen Array-Grenze durchlaufen (1). In dieser FOR-Schleife wird die Referenz jeweils für die Achse mit dem Index der FOR-Schleife zugewiesen (2). Bei erfolgreicher Zuweisung wird die Motion Control-Anweisung "MC_Reset" mit der zum Array-Index passenden Instanz aufgerufen (3).

Falls die Referenz nicht zugewiesen werden kann und den Wert "NULL" enthält, wird das entsprechende Element im Fehler-Array gesetzt (4).

Um die Ausgangsvariablen des Bausteins zu setzen, werden die sie zuerst auf einen festen Wert initialisiert (5). Nach dem Initialisieren wird eine Weitere FOR-Schleife von der unteren bis zur oberen Array-Grenze durchlaufen (6), in der die Ausgangsvariablen entsprechend des aktuellen Bearbeitungsstatus gesetzt werden:

- Der Ausgang "done" wird auf "true" initialisiert und zurückgesetzt, wenn mindestens eine der Instanzen der Motion Control-Anweisung "MC_Reset" nicht den Wert "true" am Ausgang "done" hat (7).
- Der Ausgang "busy" wird auf "false" initialisiert und gesetzt, sobald mindestens eine der Instanzen der Motion Control-Anweisung "MC_Reset" den Wert "true" am Ausgang "busy" hat (8).

- Der Ausgang "error" wird auf "false" initialisiert und gesetzt, wenn mindestens eine der Instanzen der Motion Control-Anweisung "MC_Reset" den Wert "true" am Ausgang "error" hat. Zusätzlich wird die "errorID" der jeweiligen Instanz auf den Ausgang "errorID" des Funktionsbausteins kopiert (9).

Hinweis

Es wird in diesem Fall immer die "errorID" der letzten Instanz der Motion Control-Anweisung "MC_Reset" ausgegeben. Fehler können so nacheinander abgearbeitet werden. Alternativ können Sie auch ein Array mit "errorID" am Baustein ausgeben und die Fehlernummern für alle Instanzen ausgeben. Dies bedeutet ggf. allerdings einen erhöhten Aufwand bei der Fehlerauswertung im Programm außerhalb des Funktionsbausteins, da ein Array statt einer einzelnen Variable ausgewertet werden muss.

Außerdem wird auch ein Fehler am Baustein ausgegeben, wenn mindestens eine der Referenzen nicht zugewiesen werden konnte. Hierfür wird das Array "tempErrorArray" ausgewertet.

Hinweis

Diese Beispielprogrammierung zeigt eine mögliche Vorgehensweise für Massenoperationen, die Sie beliebig um weitere Eingangs- oder Statusvariablen sowie weitere Funktionen (z. B. weitere Motion Control-Anweisungen oder logische Abläufe) erweitern können.

Hinweis

Bei der Bearbeitung sehr vieler Technologieobjekte oder größerer Programme innerhalb der Schleifen, kann es sinnvoll sein, die Bearbeitung auf mehrere CPU-Zyklen aufzuteilen und in jedem CPU-Zyklus nur einen Teil des Arrays bzw. der Technologieobjekte zu bearbeiten, um die Zykluszeit gering zu halten.

6 Anhang

6.1 Service und Support

SiePortal

Die integrierte Plattform für Produktauswahl, Einkauf und Support - und Verbindung von Industry Mall und Online Support. Die neue Startseite, ersetzt die bisherigen Startseiten der Industry Mall sowie des Online Support Portals (SIOS) und fasst diese zusammen.

- **Produkte & Services**
Unter Produkte & Services finden Sie alle unsere Angebote, die bisher im Mall Katalog verfügbar waren.
- **Support**
Im Bereich Support finden Sie alle Informationen, die für die Lösung technischer Probleme mit unseren Produkten hilfreich sind.
- **mySieportal**
mySiePortal ist Ihr persönlicher Bereich, der Funktionen, wie z.B. die Warenkorbverwaltung oder die Bestellübersicht anzeigt. Den vollen Funktionsumfang sehen Sie hier erst nach erfolgtem Login.

Das SiePortal rufen Sie über diese Adresse auf:
sieportal.siemens.com

Industry Online Support

Der Industry Online Support ist die bisherige Adresse für Informationen zu unseren Produkten, Lösungen und Services.

Produktinformationen, Handbücher, Downloads, FAQs und Anwendungsbeispiele - alle Informationen sind mit wenigen Mausklicks erreichbar:
support.industry.siemens.com

Technical Support

Der Technical Support von Siemens Industry unterstützt Sie schnell und kompetent bei allen technischen Anfragen mit einer Vielzahl maßgeschneiderter Angebote - von der Basisunterstützung bis hin zu individuellen Supportverträgen.

Anfragen an den Technical Support stellen Sie per Web-Formular:
support.industry.siemens.com/cs/my/src

SITRAIN - Digital Industry Academy

Mit unseren weltweit verfügbaren Trainings für unsere Produkte und Lösungen unterstützen wir Sie praxisnah, mit innovativen Lernmethoden und mit einem kundenspezifisch abgestimmten Konzept.

Mehr zu den angebotenen Trainings und Kursen sowie deren Standorte und Termine erfahren Sie unter:
siemens.de/sitrain

Industry Online Support App

Mit der App "Industry Online Support" erhalten Sie auch unterwegs die optimale Unterstützung. Die App ist für iOS und Android verfügbar:



6.2 Industry Mall



Die Siemens Industry Mall ist die Plattform, auf der das gesamte Produktportfolio von Siemens Industry zugänglich ist. Von der Auswahl der Produkte über die Bestellung und die Lieferverfolgung ermöglicht die Industry Mall die komplette Einkaufsabwicklung – direkt und unabhängig von Zeit und Ort:

mall.industry.siemens.com

6.3 Links und Literatur

Tabelle 6-1

| Nr. | Thema |
|-----|---|
| \1\ | Siemens Industry Online Support https://support.industry.siemens.com |
| \2\ | Link auf die Beitragsseite des Anwendungsbeispiels https://support.industry.siemens.com/cs/ww/de/view/109822593 |
| \3\ | |

6.4 Änderungsdocumentation

Tabelle 6-2

| Version | Datum | Änderung |
|---------|---------|---------------|
| V1.0 | 09/2023 | Erste Ausgabe |
| | | |