Beispieldokumentation

NUTZUNGSBEDINGUNGEN
Die Verwendung der Beispielprogramme erfolgt ausschließlich unter Anerkennung folgender Bedingungen durch den Benutzer:
INSEVIS bietet kostenlose Beispielprogramme für die optimale Nutzung der S7-Programmierung und zur Zeitersparnis bei der Programmerstellung. Für direkte, indirekte oder Folgeschäden des Gebrauchs dieser Software schließt INSEVIS jegliche Gewährleistung genauso aus, wie die Haftung für alle Schäden, die aus die aus der Weitergabe der die Beispielinformationen beinhaltenden Software resultieren.

Inhaltsverzeichnis

1 Motivation....2
2 Grundprinzipien des Software-Entwurfs....2
3 Antriebsfunktionen (MC-Bausteine und -Typen)....3
3.1 MC_ReadStatus_C3 (FB)....3
3.2 MC_ReadAxisError_C3 (FB)....5
3.3 MC_ReadActualPosition_C3 (FB)....5
3.4 MC_ReadActualVelocity_C3 (FB)....6
3.5 MC_Reset_C3 (FB)....6
3.6 MC_Power_C3 (FB)....7
3.7 MC_Stop_C3 (FB)....8
3.8 MC_MoveAbsolute_C3 (FB)....8
3.9 MC_MoveRelative_C3 (FB)....9
3.10 MC_MoveAdditive_C3 (FB)....10
3.11 MC_MoveVelocity_C3 (FB)....11
3.12 MC_GearIn_C3 (FB)....12
3.13 MC_Home_C3 (FB)....13
3.14 MC_Jog_C3 (FB)....14
3.15 C3_Input (FB)....14
3.16 C3_Output (FB)....15
3.17 InDataC3Type (UDT)....16
3.18 OutDataC3Type (UDT)....16
3.19 AxisRefC3Type....16
4 Datenfluss am Beispiel einer MC-Block-Instanz....17
5 CANopen-Konfiguration mit dem C3-ServoManager....18
5.1 Kommunikation konfigurieren....18
5.2 C3-Gerät konfigurieren....18
6 Slave-Konfiguration mit ConfigStage....19
6.1 Mapping T-PDO1....20
6.2 Mapping T-PDO2....20
6.3 Mapping R-PDO1....20
6.4 Mapping R-PDO2....20
6.5 Mapping R-PDO3....20
6.6 Zusätzliche SDO-Übertragung nach PDO-Mapping....21
7 S7-Beispiel-Programm....21

1 Motivation

Seit Jahren werden von Firmen der Antriebstechnikbranche herstellerspezifische S7-Bausteine zur leichteren Einbindung ihrer Antriebstechnik in die Steuerungswelt der Simatic- und kompatiblen SPSen angeboten. Dies geschieht oft mit einem effektiven, den Möglichkeiten des Antriebs angepassten, monolithischen Funktionsbaustein, der eine optimierte, jedoch willkürliche Schnittstelle aufweist und zudem meist auf ein Bussystem zugeschnitten ist (in der Regel Profibus-DP, aber auch Interbus-S und CANopen mittels Feldbus-Master-Baugruppen anderer Hersteller).

Die PLCopen (http://www.plcopen.org) als internationale Organisation hat sich unter anderem zum Ziel gesetzt, Engineering-Aufwand durch einheitliche Software-Schnittstellen zu reduzieren. Im Antriebsbereich wurden darum Standards mit Einzelfunktionen für Antriebe definiert, eine Zertifizierung von Antrieben und implementierten Schnittstellen ist möglich. Bei Verwendung von Bussystemen wie CANopen mit Antriebsschnittstellen (DS402 Antriebsprofil) ist zudem der Aufwand zur Anpassung an eine konkretes Busprotokoll gering.

Im folgenden wird der Betrieb an einem Servodrive Parker C3I21T11 (http://www.parker-eme.com) beschrieben. Die erstellte S7-Software wurde für INSEVIS-SPS'n erstellt und ist an den PLCopen-Standard angelehnt.

An folgenden Geräten erfolgte der Test der Software:

C3I21T11

Testgerät	:	C3I21T11
Software-Version	:	2011 R09-11
C3ServoManager	:	V 2.9.2.49 (Juni 2011)

INSEVIS

Testgerät	:	CC300V
Betriebssystem	:	2.0.23
S7-Bibliothek	:	Insevis_S7-library_from_2_0_22

Die Firma inmotec Automation GmbH (support@inmotec.de) erstellt und erweitert antriebsnahe Software für INSEVIS-Steuerungen.

2 Grundprinzipien des Software-Entwurfs

1. Alle Antriebsfunktionen (sogenannte Motion-Control-Bausteine MC_) werden als einzelne Funktionsbausteine implementiert, z.B. ist der Funktionsbaustein „MC_Power_C3“ ein S7-FB, der zum Bestromen des Servomotors dient. Da der Servomotor nicht nur bestromt werden muss, sondern auch Bewegungsfunktionen ausführen soll, sind weitere Funktionsbausteine erforderlich, auch werden selbstverständlich mehrere Achsen unterstützt. Um die Vielzahl von Instanzen von Funktionsbausteinen mit einem separaten Instanzdatenbaustein zu vermeiden, empfiehlt sich die Instanziierung von Funktionsbausteinen im STAT-Bereich der Variablendefinition des „Container“-Funktionsbausteins.
2. Die MC-Bausteine verwenden keine globalen Ressourcen wie M-Merker, T-Zeiten oder Z-Zähler, sondern deren instanziierbaren IEC-Varianten.
3. Alle Antriebsfunktionen auf der INSEVIS-SPS kommunizieren über asynchrone CANopen-PDO's nach DS301, so dass der Kommunikationsaufwand (Busauslastung) reduziert ist. Beim Antriebsprofil DS402 werden ausschließlich Betriebsarten verwendet, die keine äquidistante Übertragung von Sollwerten erfordern. Der sogenannte „Interpolated mode“ wird nicht verwendet.
4. Die Funktionsbausteine werden im Original mit SCL (Structured Control Language), einer Engineering-Option zu Step7 der Firma Siemens erstellt, die Verwendung der Funktionsbausteine erfordert jedoch kein installiertes SCL-Paket auf dem Entwicklungsrechner des Anwenders.
5. Um Diversitäten bei Antrieben abzufangen und Namenskonflikte mit bereits vorhandenen Bausteinen aus Bibliotheken zu vermeiden (z.B. bei Technologie-SPSen der Firma Siemens), erhalten die MC-Bausteine einen Postfix wie „_C3“ in Abhängigkeit vom jeweiligen Antrieb. Es bleibt zu bemerken, dass der Instanz-Name (im Beispiel „Axis00“ bei Tausch von Antrieben unberührt bleibt).
6. Da sich Bausteine nicht gegenseitig referenzieren, können Baustein-Adressen (absolute Nummern) dem Bedarf des Anwenderprogramms angepasst werden.

3 Antriebsfunktionen (MC-Bausteine und -Typen)

MC-Baustein/Symbol	Adresse	Funktionalität
MC_ReadStatus_C3	FB40	Visualisierung der Antriebszustände (entstromt, stoppend, stillstehend, profilbasierte Bewegungsfunktionen aktiv, Endlosbewegung aktiv, Synchronisierte Bewegungsfunktionen aktiv, Referenzfahrt aktiv)
MC_ReadAxisError_C3	FB41	Visualisierung des Fehlercodes des Antriebs
MC_ReadActualPosition_C3	FB42	Visualisierung der aktuellen Position des Motors
MC_ReadActualVelocity_C3	FB43	Visualisierung der aktuellen Geschwindigkeit des Motors
MC_Reset_C3	FB44	Fehler im Antrieb rücksetzen
MC_Power_C3	FB45	Motor bestromen/entstromen mit Schnellstmögliche
MC_Stop_C3	FB46	Bestromten Motor stoppen
MC_MoveAbsolute_C3	FB47	Absolute Position anfahren
MC_MoveRelative_C3	FB48	Relative Distanz abfahren
MC_MoveAdditive_C3	FB49	Relative Distanz an Bewegung anhängen
MC_MoveVelocity_C3	FB50	Endlosbewegung
MC_GearIn_C3	FB51	Synchronisierte Bewegungsfunktionen ausführen (Elektronisches Getriebe), z.B. einem Leitantrieb via Encoder-Pulsen folgen
MC_Home_C3	FB52	Referenzfahrt ausführen
MC_Jog_C3	FB53	Hand+/- fahren, stoppt an den Software-Endgrenzen
C3_Input	FB54	C3-Eingänge auslesen (Standard-Eingänge)
C3_Output	FB55	C3-Ausgänge schreiben (Standard-Ausgänge
InDataC3Type	UDT100	Datentyp für Eingangsdaten CANopen, pro Achse einmal instanziieren
OutDataC3Type	UDT101	Datentyp für Ausgangsdaten CANopen, pro Achse einmal instanziieren
SWPosC3Type	UDT102	Datentyp Statuswort CANopen, NUR IINTERNE VERWENDUNG
CWPosC3Type	UDT103	Datentyp Steuerwort CANopen, NUR IINTERNE VERWENDUNG
AxisRefC3Type	UDT104	Datentyp Achsreferenz, pro Achse einmal instanziieren

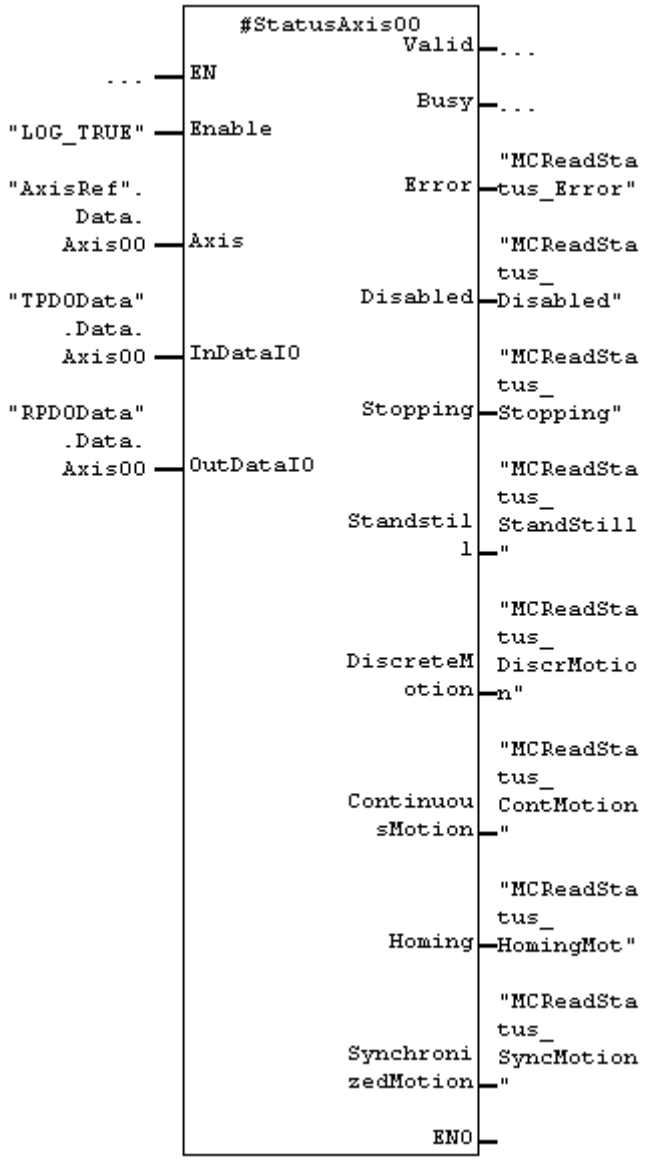
3.1 MC_ReadStatus_C3 (FB)

Der MC_ReadStatus_C3 wird zur Visualisierung (Statusbildung) verschiedener Antriebszustände verwendet. Anhand dieser Informationen kann das SPS-Programm Aktivitäten des Antriebs verfolgen.

ⓘ	Dieser Baustein ist zwingend in das SPS-Programm einzubinden, da neben der Statusbildung auch die komplette Achsreferenz bearbeitet wird. Daher ist die Bausteingröße auch deutlich größer als bei den andern MC-Bausteinen. Erläuterungen findet man im Abschnitt zur Achsreferenz. Es ist nur eine Instanz dieses FB's sinnvoll und erlaubt.

Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	Statusbildung aktivieren Für die Abarbeitung der Achsreferenz ist der Enable-Eingang unbedeutend, der FB muss aber trotzdem aufgerufen werden.

Name	Variablenbereich	Typ	Funktion
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
InDataIO	IN_OUT	InDataC3Type (UDT)	Referenz auf IO-Daten (Eingangsdaten CANopen)
OutDataIO	IN_OUT	OutDataC3Type (UDT)	Referenz auf IO-Daten (Ausgangsdaten CANopen)
Valid	OUT	Bool	Daten sind gültig
Busy	OUT	Bool	Funktion läuft
Error	OUT	Bool	Achse mit Fehler
Disabled	OUT	Bool	Achse stromlos
Stopping	OUT	Bool	Achse stoppt
DiscreteMotion	OUT	Bool	Achse positioniert
ContinuousMotion	OUT	Bool	Achse positioniert endlos
Homing	OUT	Bool	Achse führt Homing-Fahrt aus
SynchronizedMotion	OUT	Bool	Achse positioniert synchronisiert (z.B. via Elektronisches Getriebe)

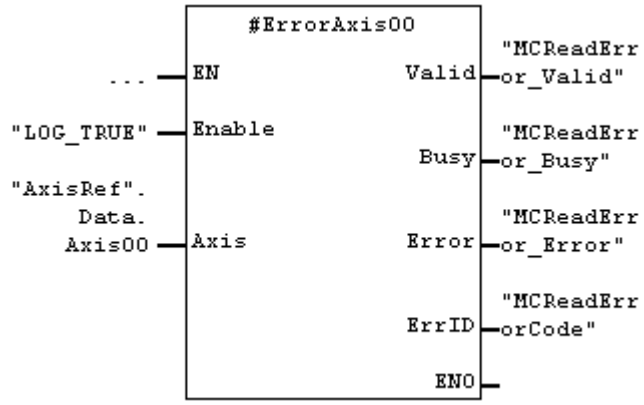


Im Stat-Bereich eines Container-FB's instanziierter MC_ReadStatus_C3 mit dem Instanznamen StatusAxis00.

3.2 MC_ReadAxisError_C3 (FB)

Der MC_ReadAxisError_C3 wird zur Visualisierung des Fehlercodes des Achse verwendet. Die Bedeutung des Fehlercodes ist der Hilfeanleitung zu entnehmen.

Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	Fehlercode lesen
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Valid	OUT	Bool	Daten sind gültig
Busy	OUT	Bool	Funktion läuft
Error	OUT	Bool	Achse mit Fehler
ErrorID	OUT	WORD	Fehlercode Achse (hier 16-Bit)

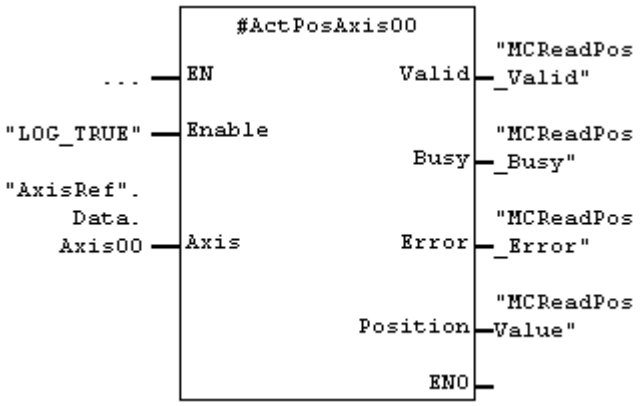


Im Stat-Bereich eines Container-FB's instanziierter MC_ReadAxisError_C3 mit dem Instanznamen ErrorAxis00.

3.3 MC_ReadActualPosition_C3 (FB)

Der MC_ReadActualPosition_C3 stellt die Absolutposition der Achse bereit. Die Position kann durch jede Art Positionierung, Handfahren, mechanische Verschiebung, Referenzfahrt oder Regelschwingung verändert werden.

Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	Fehlercode lesen
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Valid	OUT	Bool	Daten sind gültig
Busy	OUT	Bool	Funktion läuft
Error	OUT	Bool	Achse mit Fehler
Position	OUT	REAL	Achsposition in Nutzereinheiten, z.B. „mm“

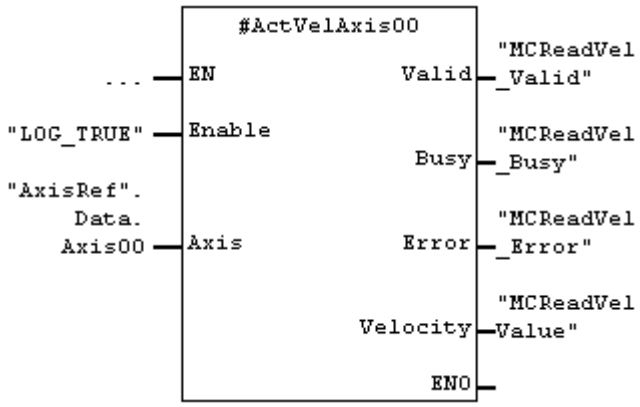


Im Stat-Bereich eines Container-FB's instanziierter MC_ReadActualPosition_C3 mit dem Instanznamen ActPosAxis00.

3.4 MC_ReadActualVelocity_C3 (FB)

Der MC_ReadActualVelocity_C3 stellt die Geschwindigkeit der Achse bereit. Die Geschwindigkeit kann durch jede Art Positionierung, Handfahren, mechanische Verschiebung, Referenzfahrt oder Regelschwingung verändert werden.

Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	Fehlercode lesen
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Valid	OUT	Bool	Daten sind gültig
Busy	OUT	Bool	Funktion läuft
Error	OUT	Bool	Achse mit Fehler
Velocity	OUT	REAL	Achsgeschwindigkeit in Nutzereinheiten, z.B. „mm/s“



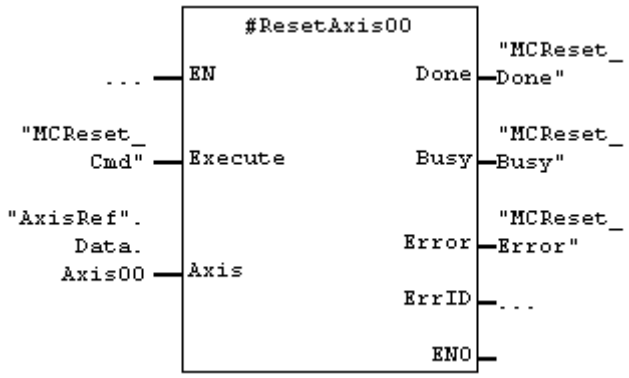
Im Stat-Bereich eines Container-FB's instanziierter MC_ReadActualVelocity_C3 mit dem Instanznamen ActVelAxis00.

3.5 MC_Reset_C3 (FB)

Mit dem Baustein MC_Reset_C3 wird die Servoachse bei Fehler quittiert.

ⓘ	Es ist nur eine Instanz dieses FB's sinnvoll und erlaubt.

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	0-1-Flanke quittiert Achse
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	Achse quittiert
Busy	OUT	Bool	Funktion läuft
Error	OUT	Bool	Achse mit Fehler
ErrorID	OUT	WORD	Fehlercode Achse (hier 16-Bit)



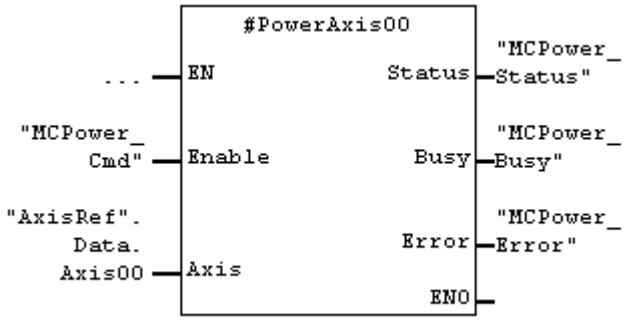
Im Stat-Bereich instanziierter MC_Reset_C3 mit dem Instanznamen ResetAxis00.

3.6 MC_Power_C3 (FB)

Mit dem Baustein MC_Power_C3 wird die Achse bestromt (Motor hat Drehmoment bzw. Kraft) oder entstromt (Schnellstopp, dann stromlos).

ⓘ	Es ist nur eine Instanz dieses FB's sinnvoll und erlaubt. Die Schnellstopprampe und Schnellstoppruck werden mit dem C3-ServoManager konfiguriert.

Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	0-1-Flanke bestromt Achse, 1-0-Flanke führt Schnellstopp mit anschließendem Stromlosschalten aus
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Status	OUT	Bool	1 bestromt 0 stromlos
Busy	OUT	Bool	Funktion Bestromen gerade aktiv
Error	OUT	Bool	Achse mit Fehler



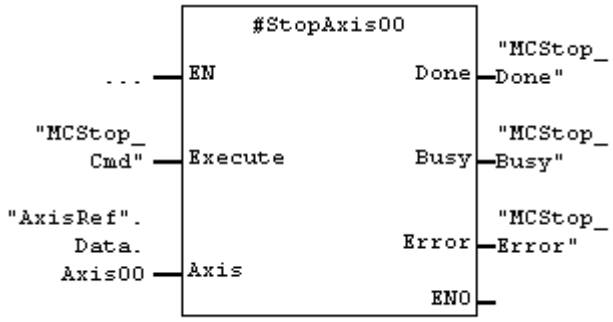
Im Stat-Bereich instanziierter MC_Power_C3 mit dem Instanznamen PowerAxis00.

3.7 MC_Stop_C3 (FB)

Mit dem Baustein MC_Stop_C3 wird die Achse gestoppt. Ein Stoppen ist nur bei bestromter Achse durchführbar.

ⓘ	Es ist nur eine Instanz dieses FB's sinnvoll und erlaubt. Die Stopprampe und der Stoppruck werden mit dem C3-ServoManager konfiguriert. Bei 0-1-Flanke am Enable-Eingang und bestromter Achse wird einmalig ein Stoppbefehl übertragen. Weitere Achsbewegungen (Neustarten) werden bei aktiviertem Stopp-Execute (=1) grundsätzlich geblockt.

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	1 Stoppt Achse 0 Bewegungsfreigabe Achse
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	Achse gestoppt
Busy	OUT	Bool	Funktion läuft
Error	OUT	Bool	Achse mit Fehler

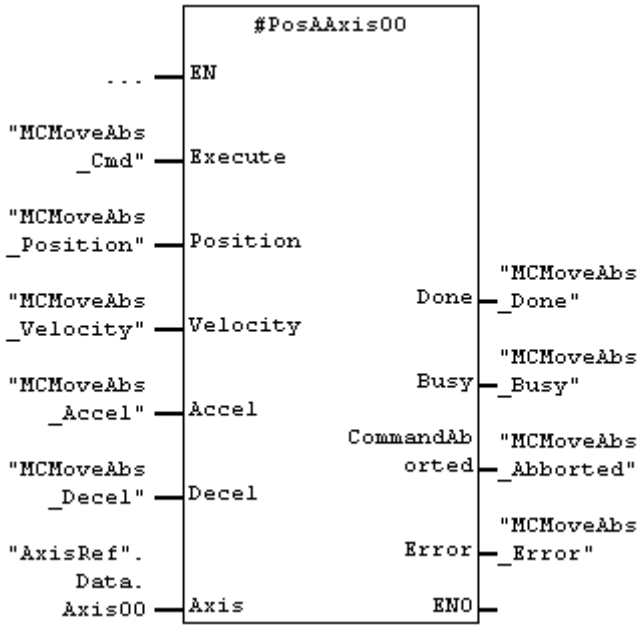


Im Stat-Bereich instanziierter MC_Stop_C3 mit dem Instanznamen StopAxis00.

3.8 MC_MoveAbsolute_C3 (FB)

Der Baustein MC_MoveAbsolute_C3 wird für absolute Positionierungen verwendet. Bezugspunkt der absoluten Position ist der durch Teachen (Absolutwertgeber) oder eine Referenzfahrt (bei einfacheren Messsystemen wie Resolver, Sinus-Cosinus-Encoder, RS422-Encoder) festgelegte bzw. ermittelte absolute Bezugspunkt (auch mathematischer Nullpunkt).

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	0-1-Flanke startet die Bewegung
Position	IN	Real	Absolute Position in Nutzereinheiten, z.B. „mm“
Velocity	IN	Real	Positioniergeschwindigkeit in Nutzereinheiten, z.B. „mm/s“
Accel	IN	Dint	Beschleunigung in Nutzereinheiten, z.B. „mm/s2“
Decel	IN	Dint	Verzögerung in Nutzereinheiten, z.B. „mm/s2“
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	Achse hat Zielposition erreicht
Busy	OUT	Bool	Funktion läuft
CommandAbborted	OUT	Bool	Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler

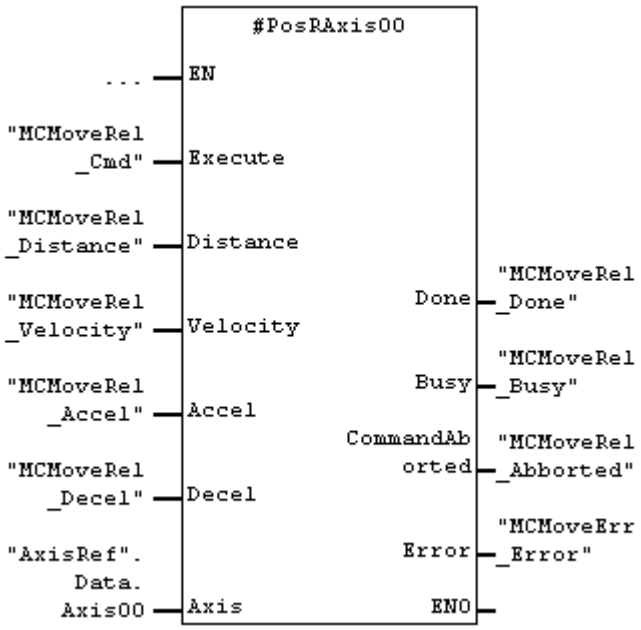


Im Stat-Bereich instanziierter MC_MoveAbsolute_C3 mit dem Instanznamen PosAAxis00.

3.9 MC_MoveRelative_C3 (FB)

Der Baustein MC_MoveRelative_C3 wird für relative Positionierungen (um eine Distanz) verwendet. Bezugspunkt für den Distanz ist die aktuelle Sollposition. Diese Art der Positionierung wird auch als Kettenpositionierung bezeichnet.

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	0-1-Flanke startet die Bewegung
Distance	IN	Real	Distanz in Nutzereinheiten, z.B. „mm“
Velocity	IN	Real	Positioniergeschwindigkeit in Nutzereinheiten, z.B. „mm/s“
Accel	IN	Dint	Beschleunigung in Nutzereinheiten, z.B. „mm/s2“
Decel	IN	Dint	Verzögerung in Nutzereinheiten, z.B. „mm/s2“
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	Achse hat Zielposition erreicht (Distanz abgefahren)
Busy	OUT	Bool	Funktion läuft
CommandAbborted	OUT	Bool	Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler

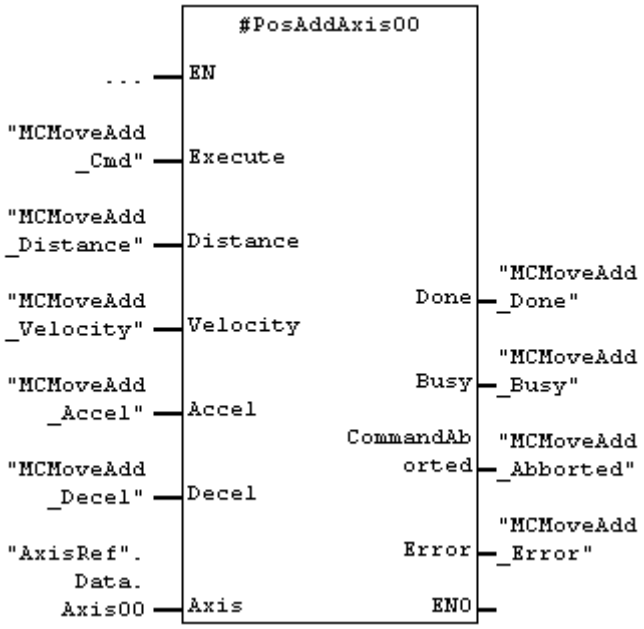


Im Stat-Bereich instanziierter MC_MoveRelative_C3 mit dem Instanznamen PosRAxis00.

3.10 MC_MoveAdditive_C3 (FB)

Der Baustein MC_MoveAdditive_C3 wird für additive Positionierungen (um eine Distanz) verwendet, wobei im Unterschied zur relativen Positionierung die Distanz an die laufende Positionierung angehängt wird.

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	0-1-Flanke startet die Bewegung
Distance	IN	Real	Distanz in Nutzereinheiten, z.B. „mm“
Velocity	IN	Real	Positioniergeschwindigkeit in Nutzereinheiten, z.B. „mm/s“
Accel	IN	Dint	Beschleunigung in Nutzereinheiten, z.B. „mm/s2“
Decel	IN	Dint	Verzögerung in Nutzereinheiten, z.B. „mm/s2“
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	Achse hat Zielposition erreicht (Distanz abgefahren)
Busy	OUT	Bool	Funktion läuft
CommandAbborted	OUT	Bool	Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler

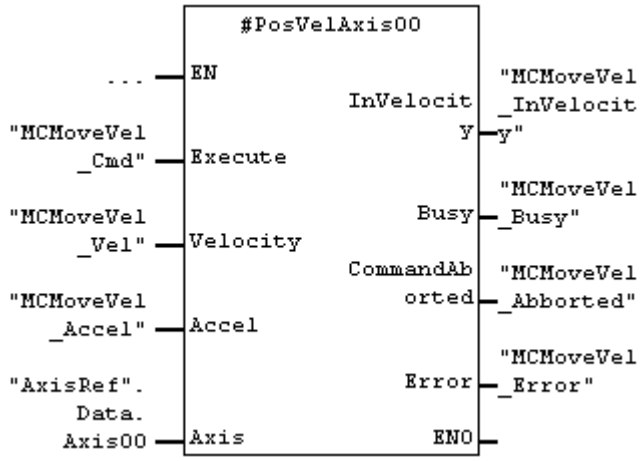


Im Stat-Bereich instanziierter MC_MoveAdditive_C3 mit dem Instanznamen PosAddAxis00.

3.11 MC_MoveVelocity_C3 (FB)

Der Baustein MC_MoveVelocity_C3 wird für Endlosbewegungen verwendet, wobei der Lageregler aktiviert bleibt, so dass bei Lageschleppfehler auch kleine Geschwindigkeitsüberhöhungen bzw. -erniedrigungen zur Lagefehlerminimierung realisiert werden. Die Bewegung muss mit MC_Stop_C3 gestoppt werden.

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	0-1-Flanke startet die Bewegung
Velocity	IN	Real	Positioniergeschwindigkeit in Nutzereinheiten, z.B. „mm/s“
Accel	IN	Dint	Beschleunigung und Verzögerung in Nutzereinheiten, z.B. „mm/s2“
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
InVelocity	OUT	Bool	Achse hat Zielgeschwindigkeit erreicht
Busy	OUT	Bool	Funktion läuft
CommandAbborted	OUT	Bool	Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler



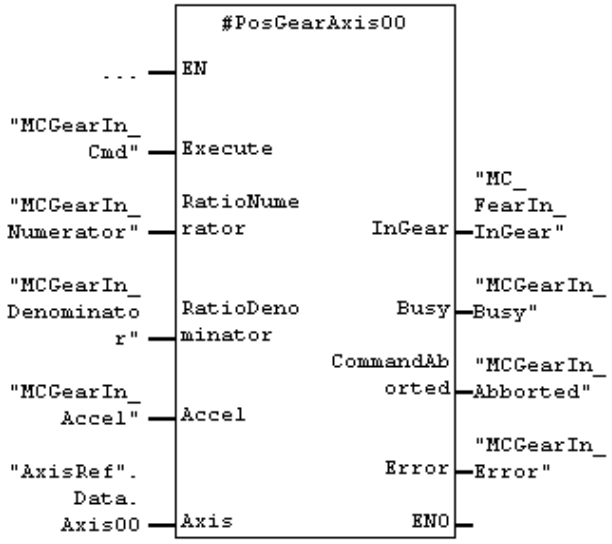
Im Stat-Bereich instanziierter MC_MoveVelocity_C3 mit dem Instanznamen PosVelAxis00.

3.12 MC_GearIn_C3 (FB)

Der Baustein MC_GearIn_C3 wird verwendet, wenn die Achse einem Leitantrieb (z.B. einer anderen Achse oder einem Encoder) folgen soll. In diesem Fall wird für eine Positionierung keine zeitbasiertes Bewegungsprofil erstellt, sondern unter Beschränkung auf eine maximale Beschleunigung bzw. Verzögerung einem Leitwert, was ein RS422-Encodersignal oder ein Analogwert oder ein Motion-Bus-Signal (bei C3 der sogenannte HEDA-Bus) sein kann, gefolgt. Art und Auflösung des Leitwertes wird mit dem C3-ServoManager konfiguriert.

Das Verhältnis Zähler/Nenner wird an den Servoantrieb übertragen, es ist zu beachten, dass als minimale Untersetzung 0.001 übertragen werden kann, auch ist das Verhältnis mit einer Genauigkeit nicht besser als 0.001 einstellbar.

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	0-1-Flanke startet die Bewegung
RatioNumerator	IN	Real	Zähler Getriebefaktor
RatioDenominator	IN	Dint	Nenner Getriebefaktor
Accel	IN	Dint	Maximale Beschleunigung und Verzögerung in Nutzereinheiten, z.B. „mm/s2“
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
InVelocity	OUT	Bool	Achse hat Zielgeschwindigkeit erreicht
Busy	OUT	Bool	Funktion läuft
CommandAbborted	OUT	Bool	Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler



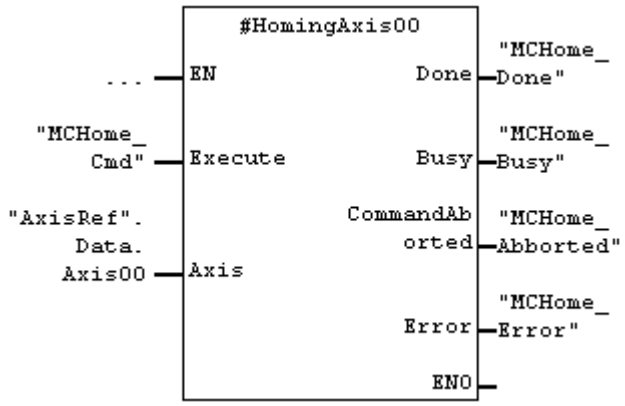
Im Stat-Bereich instanziierter MC_GearIn_C3 mit dem Instanznamen PosGearAxis00.

3.13 MC_Home_C3 (FB)

Der Baustein MC_Home_C3 wird verwendet, um den mathematischen Bezugspunkt der Achse zu definieren. Die Referenzierarten (Modi) sind der C3-Beschreibung zu entnehmen. Wird ein Absolutgeber verwendet oder eine Absolutgebersimulation, muss nach dem Teachen der Modus wieder auf 0 (keine Referenzierung erforderlich) gesetzt werden. Die Profilwerte (Geschwindigkeit, Beschleunigung, Ruck) sind Bestandteil der C3-Konfiguration.

ⓘ	Es ist nur eine Instanz dieses FB's sinnvoll und erlaubt. (1) Bei der Konfiguration ist der Homing-Offset einzustellen. Nach dem eigentlichen Referenzieren wird die Achse mit einem Wert von -1.0 * Homing-Offset als Istposition gemeldet. (2) Man kann dann noch im C3-ServoManager einstellen, ob danach im Rahmen der Referenzfahrt noch der mathematische Nullpunkt angefahren wird.

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	0-1-Flanke startet die Referenzfahrt
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	Referenzfahrt beendet, Setzposition gesetzt, der mathematische Nullpunkt und gegebenenfalls die Software-Endgrenzen sind gültig
Busy	OUT	Bool	Funktion läuft
CommandAbborted	OUT	Bool	Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler



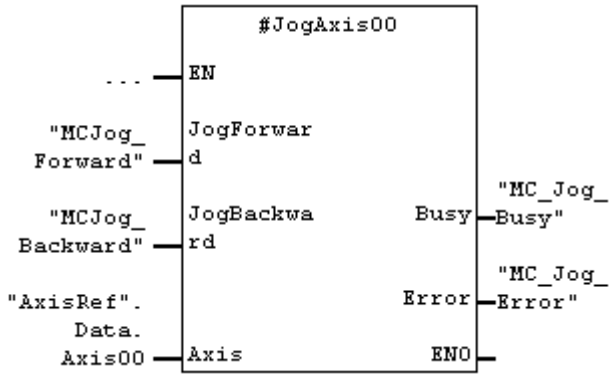
Im Stat-Bereich instanziierter MC_Home_C3 mit dem Instanznamen HomingAxis00.

3.14 MC_Jog_C3 (FB)

Der Baustein MC_Jog_C3 wird verwendet, um die Achse „manuell“ zu bewegen (auch tippen genannt). Beide Richtungen sind möglich, die Profilwerte (Beschleunigung, Verzögerung, Ruck) werden mit der C3-Konfiguration festgelegt. Sind Software-Endgrenzen festgelegt, hält die Achse auf den definierten Software-Endgrenzen.

ⓘ	Es ist nur eine Instanz dieses FB's sinnvoll und erlaubt.

Name	Variablenbereich	Typ	Funktion
JogForward	IN	Bool	0-1-Flanke startet Joggen im Uhrzeigersinn, 1-0-Flanke stoppt die Achse
JogBackward	IN	Bool	0-1-Flanke startet Joggen entgegen dem Uhrzeigersinn, 1-0-Flanke stoppt die Achse
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
Busy	OUT	Bool	Funktion läuft
Error	OUT	Bool	Achse mit Fehler

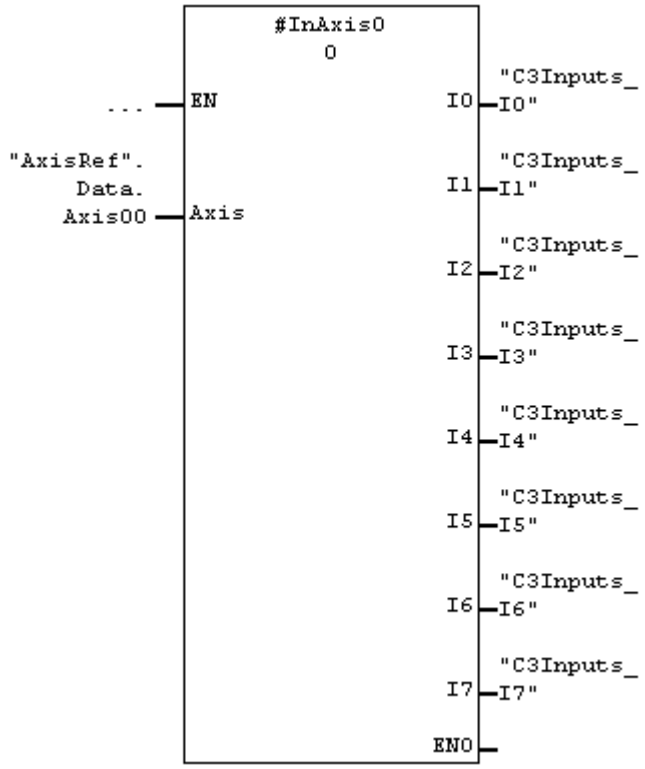


Im Stat-Bereich instanziierter MC_Jog_C3 mit dem Instanznamen JogAxis00.

3.15 C3_Input (FB)

Der Baustein C3_Input ist eine Spezialversion für die C3-Achse, die die „unteren“ C3-Eingänge 0 bis 7 als Status bereit stellt. Eingang 5 und 6 sind für Endschalter reserviert, so dass sich dort bei entsprechender Konfiguration die Polarität umdrehen kann. Eingang E7 ist für den Referenzschalter reserviert.

Name	Variablenbereich	Typ	Funktion
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
I0 bis I7	OUT	Bool	Eingänge 0 bis 7 als Status

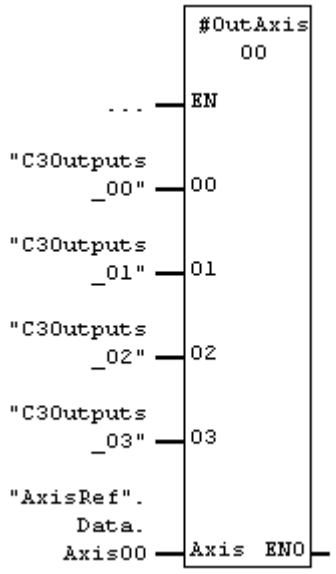


Im Stat-Bereich instanziierter C3_Input mit dem Instanznamen InAxis00.

3.16 C3_Output (FB)

Der Baustein C3_Output ist eine Spezialversion für die C3-Achse, die die „unteren“ C3-Ausgänge 0 bis 3 zum Setzen/Rücksetzen bereit stellt.

Name	Variablenbereich	Typ	Funktion
Axis	IN_OUT	AxisRefC3Type (UDT)	Achsreferenz (Achsverweis)
O0 bis O3	IN	Bool	Ausgänge 0 bis 3 zum Setzen/Rücksetzen



Im Stat-Bereich instanziierter C3_Output mit dem Instanznamen OutAxis00.

3.17 InDataC3Type (UDT)

Dieser Datentyp ist bei Verwendung in einem Datenbaustein mit einem Namen, z.B. Axis00 zu instanziieren. Pro Achse wird genau 1 Instanz benötigt. Die Instanz-Daten entsprechen T-PDO-Daten der C3-Achse.

ⓘ	Am besten werden alle Instanzen (der verschiedenen Achsen) von InDataC3Type in einem separaten DB (z.B. „TPDODATA“) angelegt.

```
wStatusWord    :   WORD;      // TPDO1, async, 0x6041 + 0x00, Status word
iActModeOfOp   :   INT;       // TPDO1, async, 0x6061 + 0x00, Actual mode of operation
wDigInWord     :   WORD;      // TPDO1, async, 0x6100 + 0x01, Digital inputs (Standard)
wErrCode       :   WORD;      // TPDO1, async, 0x603f + 0x00, Error code
diActPosition  :   DINT;      // TPDO2, async, 0x6064 + 0x00, Actual position [units * 1000]
diActVelocity  :   DINT;      // TPDO2, async, 0x606C + 0x00, Actual velocity [units/s * 1000]
```

3.18 OutDataC3Type (UDT)

Dieser Datentyp ist bei Verwendung in einem Datenbaustein mit einem Namen, z.B. Axis00 zu instanziieren. Pro Achse wird genau 1 Instanz benötigt. Die Instanz-Daten entsprechen R-PDO-Daten der C3-Achse.

ⓘ	Am besten werden alle Instanzen (der verschiedenen Achsen) von OutDataC3Type in einem separaten DB (z.B. „RPDODATA“) angelegt.

```
wControlWord   :   WORD;      // RPDO1, async, 0x6040 + 0x00, Control word
iModeOfOp      :   INT;       // RPDO1, async, 0x6060 + 0x00, Mode of operation
diTarget       :   DINT;      // RPDO1, async, 0x607a + 0x00, Target [units * 1000]
diProfVelocity :   DINT;      // RPDO2, async, 0x6081 + 0x00, Profile velocity [units/s * 1000]
wDigOutWord    :   WORD;      // RPDO2, async, 0x6300 + 0x01, Digital outputs (Standard)
diProfAccel    :   DINT;      // RPDO3, async, 0x6083 + 0x00, Prof. acceleartion [units/s2]
diProfDecel    :   DINT;      // RPDO3, async, 0x6084 + 0x00, Prof. deceleration [units/s2]
```

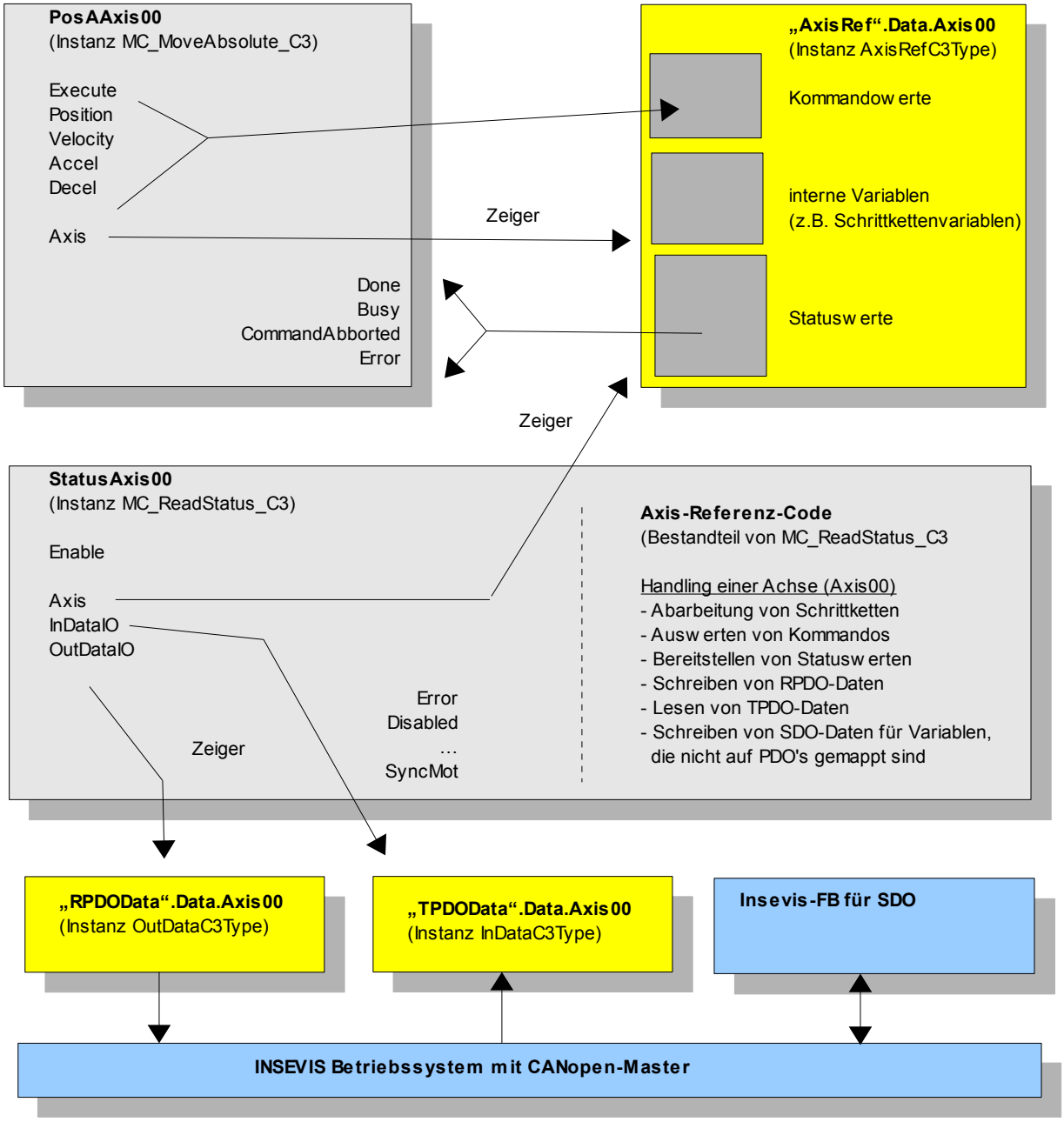
3.19 AxisRefC3Type

Dieser Datentyp wird intern zur Referenzierung der Achse benötigt. Alle Instanzen von MC-Bausteinen bestimmten damit die verbundene Achse. Da die instanziierten Variablen als IN_OUT übergeben werden, ist der Kopieraufwand gering. Der Baustein MC_ReadStatus_C3 verwendet die mit der Achsreferenz übergebenen Werte zur Abarbeitung der Schrittketten.

ⓘ	Am besten werden alle Instanzen (der verschiedenen Achsen) von AxisrefC3Type in einem separaten DB (z.B. „AXISREF“) angelegt.

4 Datenfluss am Beispiel einer MC-Block-Instanz

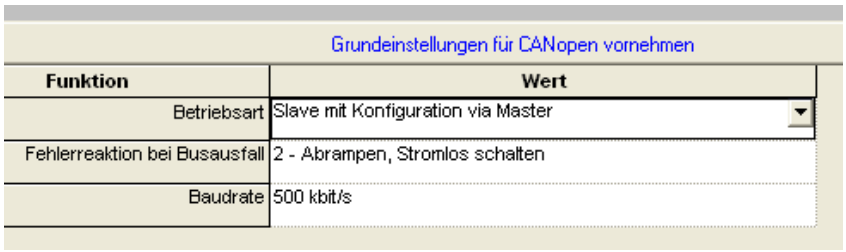
Folgende Grafik illustriert die Verwendung und den Datenfluss eines MC-Bausteines für eine Achse mit dem Namen Axis00.



5 CANopen-Konfiguration mit dem C3-ServoManager

5.1 Kommunikation konfigurieren

Grundsätzlich kann hier nicht auf die allgemeine Konfiguration eines C3-Servoantriebs eingegangen werden. Wichtig für den CANopen-Teil ist hier lediglich, dass C3I21 mit bis zu 4 PDO's in jede Richtung konfiguriert werden kann und SDO's unterstützt. Folgende Einstellungen sind im C3-Wizard für CAN zu treffen.



Grundeinstellungen für CANopen vornehmen

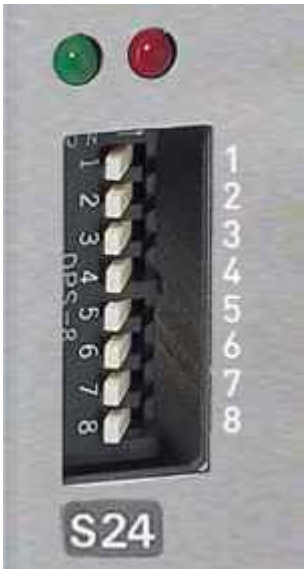
Funktion	Wert
Betriebsart	Slave mit Konfiguration via Master
Fehlerreaktion bei Busausfall	2 - Abrampen, Stromlos schalten
Baudrate	500 kbit/s

<u>Betriebsart</u>
Slave mit Konfiguration via Master
→ Die PDO's werden vom CANopen-Master konfiguriert und belegt.

<u>Fehlerreaktion bei Busausfall</u>
→ Hier wurde eine Variante gewählt, bei der der Servoantrieb bei Busausfall stoppt und dann stromlos schaltet

<u>Baudrate</u>
In Stufen einstellbar von 20 kBit/s bis 1 MBits/s

Am C3-Gerät selber muss man die Node-ID einstellen, optional kann auch die Baudrate eingestellt werden, was eigentlich nur sinnvoll ist, wenn man das Gerät über CANopen komplett konfigurieren würde, dafür ist der Aufwand aber sehr hoch (hierfür wird auf die Herstellerdokumentation verwiesen).

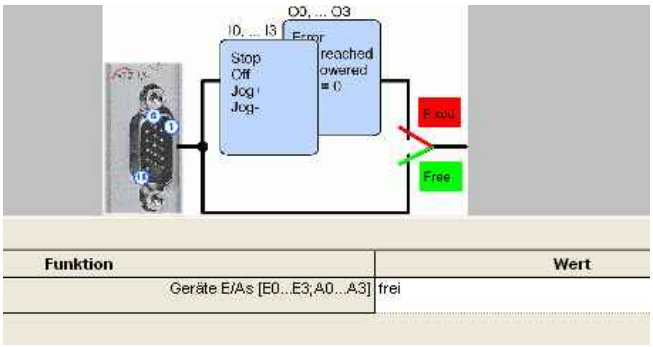


<u>Einstellen der Node-ID</u>
→ DIP-Schalter 1..7: Binäre Einstellung (OFF/ON) der Node-ID von 1 bis 127, DIP-Schalter 8: OFF

5.2 C3-Gerät konfigurieren

Das Gerät ist entsprechend den Erfordernissen Ihrer Hardware (Gerät, Motor) und Anwendung zu konfigurieren. Parameter, wie z.B. das Jog-Profil sind hier vorzugeben, da über die MC-Bausteine nicht jeder Parameter vorgegeben werden kann.

Beachten Sie folgende Einstellung für die Ein-/Ausgänge am C3:



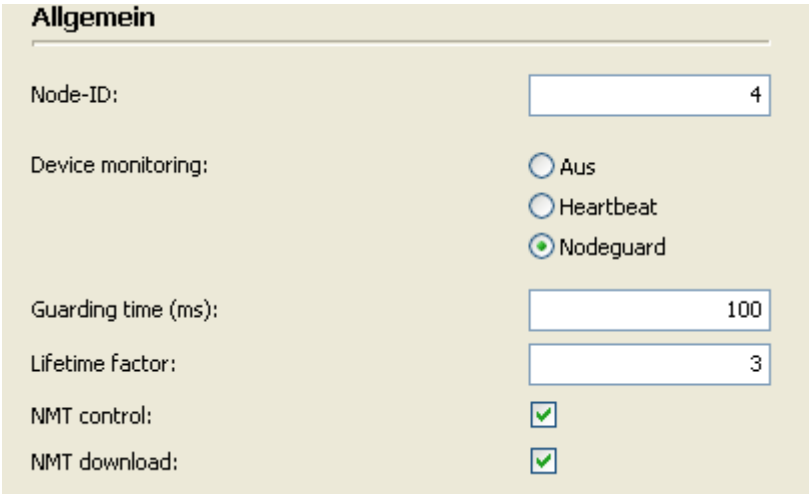
Die 24V-Eingänge E0..E7 können mit dieser Einstellung am C3 „frei“ genutzt und als zusätzliche Ressourcen für die SPS eingebunden werden. Wenn allerdings Endschalter genutzt werden, sind E5 bzw. E6 dafür fest vorzusehen. Wird bei einer Referenzierung ein Schalter benutzt, muss dieser auf E7 verwendet werden.

Die 24V-Ausgänge A0..A3 stehen als zusätzliche Ressourcen für die SPS bereit. Beachten Sie die Randbedingungen (Stromtragfähigkeit, Schalten von induktiven Lasten usw.).

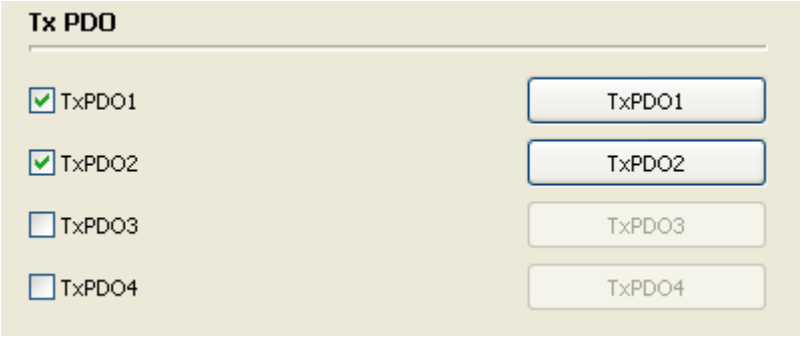
6 Slave-Konfiguration mit ConfigStage

Mit der ConfigStage-Software werden unter anderem der CANopen-Master und jeder CANopen-Slave konfiguriert. Zudem wird die Verbindung von SPS-Daten (z.B. Datenbaustein und Offset im Datenbaustein) zu den CANopen-Daten (R-PDO's, T-PDO's) definiert.

Die Achse kann als Typ in die Bibliothek der ConfigStage übernommen werden!



Festlegen der Node-ID und des Guardings
C3 unterstützt Nodeguarding
CANopen-Einstellungen (wie COB-ID's) zum C3 laden



TPDO (C3 → CANopen-Master)

Es werden für die Empfangsrichtung 2 T-PDO's benötigt. Download Kommunikationsparameter und des Mappings sind zu aktivieren.

Übertragungsverhalten TPDO1
Typ: 254
keine Sperrzeit

Übertragungsverhalten TPDO2
Typ: 254
Sperrzeit von z.B. 100ms definieren!

RPDO (CANopen-Master → C3)

Es werden für die Senderichtung 3 R-PDO's benötigt. Download Kommunikationsparameter und des Mappings sind zu aktivieren.

Übertragungsverhalten RPDO1
Typ: 254
keine Sperrzeit

	Übertragungsverhalten RPDO2 Typ: 254 keine Sperrzeit Übertragungsverhalten RPDO3 Typ: 254 keine Sperrzeit
	<u>Zusätzliche Konfiguration über SDO</u> Nach den vom SPS-Betriebssystem initiierten Download der Mapping-Parameter werden weitere Einstellungen an C3, die nicht über die C3-ServoManager-Konfiguration getätigt werden können, via SDO übertragen.

6.1 Mapping T-PDO1

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „InDataC3Type“: **0** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6041	0	16 Bit/Word	Statuswort DS402
2	0x6061	0	16 Bit/Word	Aktuelle Betriebsart DS402
3	0x6100	1	16Bit/Word	Basis-Eingänge C3
4	0x603F	0	16Bit/Word	Fehlercode C3

6.2 Mapping T-PDO2

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „InDataC3Type“: **8** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6064	0	32 Bit/DWord	Aktuelle Position {Einheiten * 1000]
2	0x606C	0	32 Bit/DWord	Aktuelle Geschwindigkeit [Einheiten/s * 1000]

6.3 Mapping R-PDO1

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „OutDataC3Type“: **0** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6040	0	16 Bit/Word	Steuerwort DS402
2	0x6060	0	16 Bit/Word	Betriebsart DS402
3	0x607A	0	32Bit/DWord	Sollwert 1 (variabel), [z.B. Einheiten * 1000]

6.4 Mapping R-PDO2

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „OutDataC3Type“: **8** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6081	0	32 Bit/DWord	Profilgeschwindigkeit [Einheiten/s * 1000]
2	0x6300	1	16Bit/Word	Basis-Ausgänge C3

6.5 Mapping R-PDO3

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „OutDataC3Type“: **12** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6083	0	32 Bit/DWord	Profilbeschleunigung [Einheiten/s^2 * 1000]

Nummer	Index	Subindex	Größe	Erklärung
2	0x6084	0	32 Bit/DWord	Profilverzögerung [Einheiten/s^2 * 1000]

6.6 Zusätzliche SDO-Übertragung nach PDO-Mapping

Nummer	Index	Subindex	Größe	Wert	Erklärung
1	0x605A	0	16 Bit/Word	6	„Quick stop mode“ so einstellen, dass ein Stopp zum Verwerfen der Position führt.

7 S7-Beispiel-Programm

Das Beispielprojekt besteht aus einem S7-Programm, das die Verwendung der MC-Blöcke veranschaulicht.