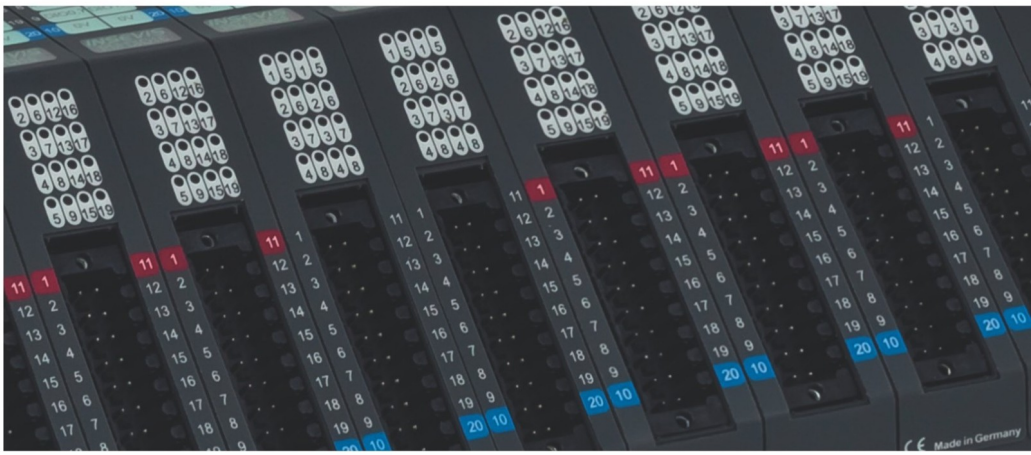
INSEVIS



Beispieldokumentation
Sample documentation

Überschrift / Thema deutsch

Überschrift / Thema englisch

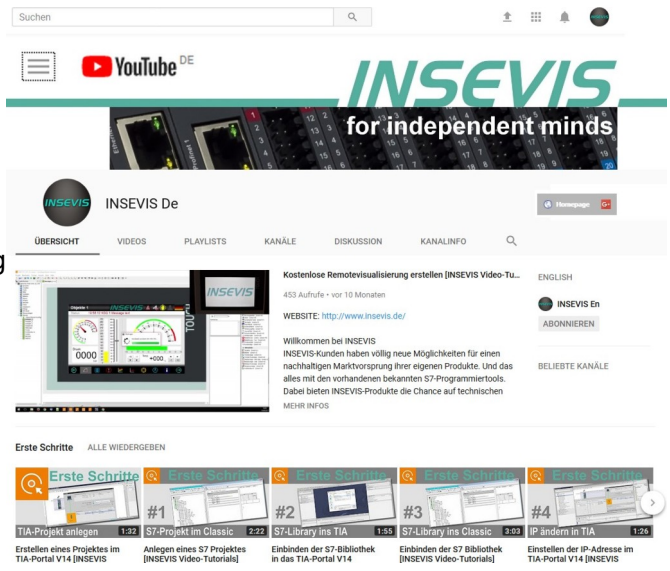
Hinweis zum besseren Verständnis durch Zusatzinformationen

Im deutschen INSEVIS-YouTube-Kanal INSEVIS DE stehen mehrere Playlists mit **Hantierungsvideos** für einzelne Details zur Verfügung.

Ebenfalls stehen **Handbücher** für die einzelnen Produktgruppen im Downloadbereich der Webseite insevis.de zur Verfügung

Bitte nutzen Sie diese Informationsquellen in Ergänzung zur vorliegenden Dokumentation. So können Sie sich noch leichter mit den INSEVIS-Funktionen vertraut machen.

Möchten Sie Erweiterungswünsche oder Fehler zu diesen Beispielen melden oder wollen Sie anderen eigene Bespielprogramme kostenlos zur Verfügung stellen? Gern werden Ihre Programme -auf Wunsch mit Benennung des Autors- allen INSEVIS- Kunden zur Verfügung gestellt.



Hinweis zu den verschiedenen Versionen der Beispielprogramme

Im Lieferumfang der Beispielprogramme können sich auch ältere Ausgabestände bzw. Versionen befinden. Diese wurden nicht aktualisiert und auf die neueste Siemens-Programmiersoftware angepasst, um einen Zugriff mit älteren Programmier-systemen weiterhin zu ermöglichen. Generell werden INSEVIS-Beispielprogramme immer mit dem aktuell neuesten Siemens-Programmiertools erstellt.

BEISPIELBESCHREIBUNG

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

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.

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.

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 (schnellstmöglich)
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

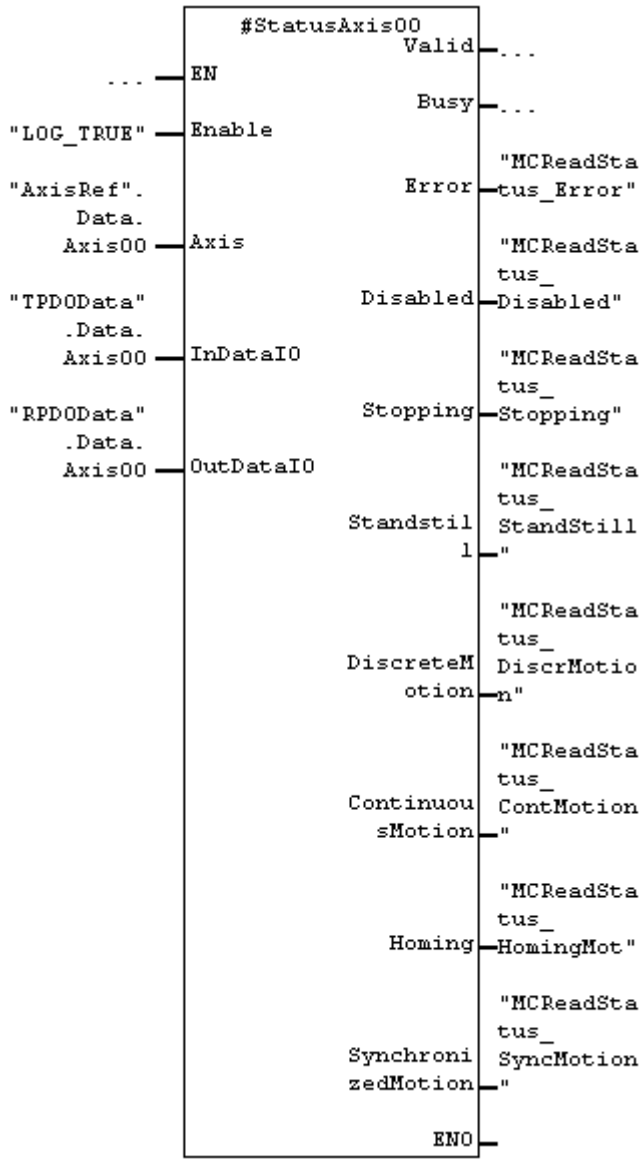
MC_ReadStatus_C3 (FB)

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.
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)

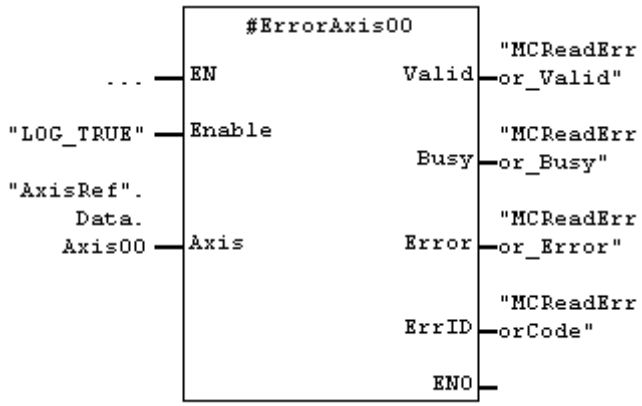


Im Stat-Bereich eines Container-FB's instanziierter MC_ReadStatus_C3 mit dem Instanznamen StatusAxis00.

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)

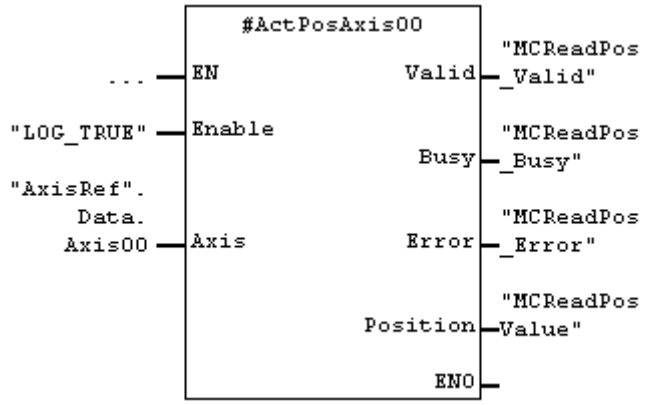


Im Stat-Bereich eines Container-FB's instanziierter MC_ReadAxisError_C3 mit dem Instanznamen ErrorAxis00.

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“



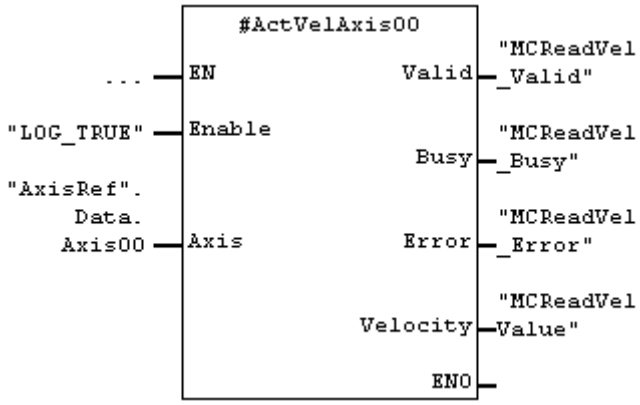
Im Stat-Bereich eines Container-FB's instanziierter MC_ReadActualPosition_C3 mit dem Instanznamen ActPosAxis00.

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

Name	Variablenbereich	Typ	Funktion
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“



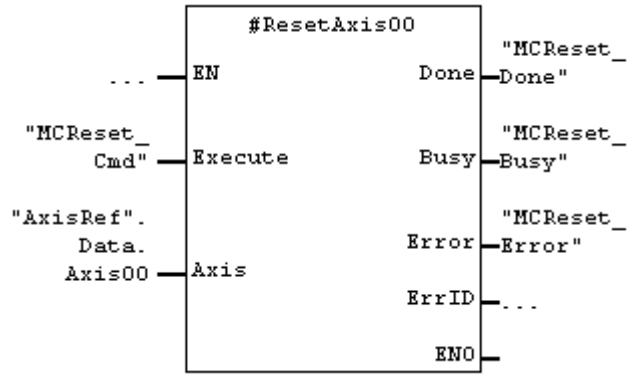
Im Stat-Bereich eines Container-FB's instanziierter MC_ReadActualVelocity_C3 mit dem Instanznamen ActVelAxis00.

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)



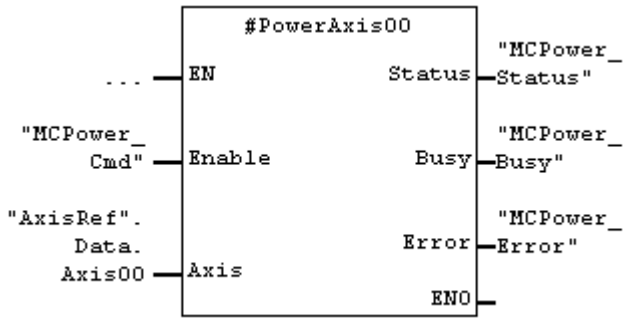
Im Stat-Bereich instanziierter MC_Reset_C3 mit dem Instanznamen ResetAxis00.

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.

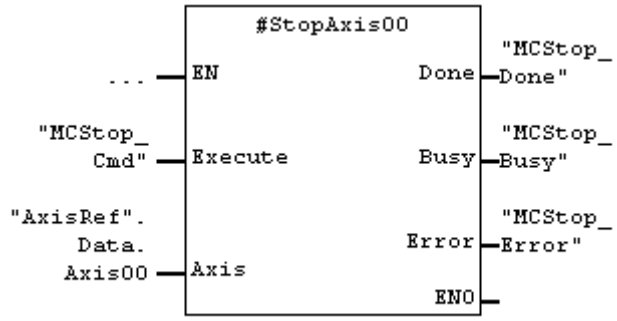
MC_Stop_C3 (FB)

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



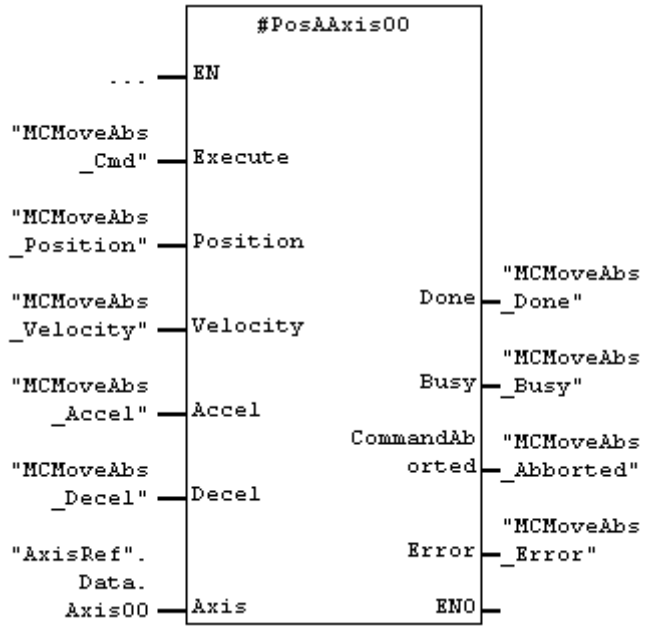
Im Stat-Bereich instanziierter MC_Stop_C3 mit dem Instanznamen StopAxis00.

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,

Name	Variablenbereich	Typ	Funktion
			Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler



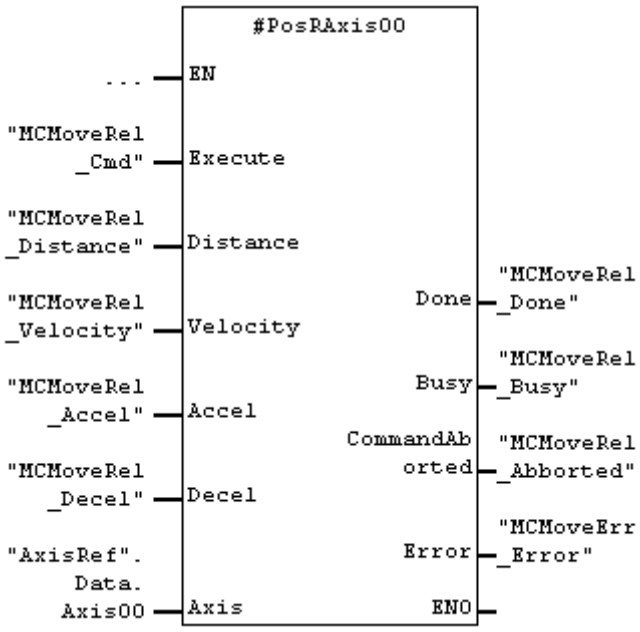
Im Stat-Bereich instanziierter MC_MoveAbsolute_C3 mit dem Instanznamen PosAAxis00.

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.

Name	Variablenbereich	Typ	Funktion
Error	OUT	Bool	Achse mit Fehler

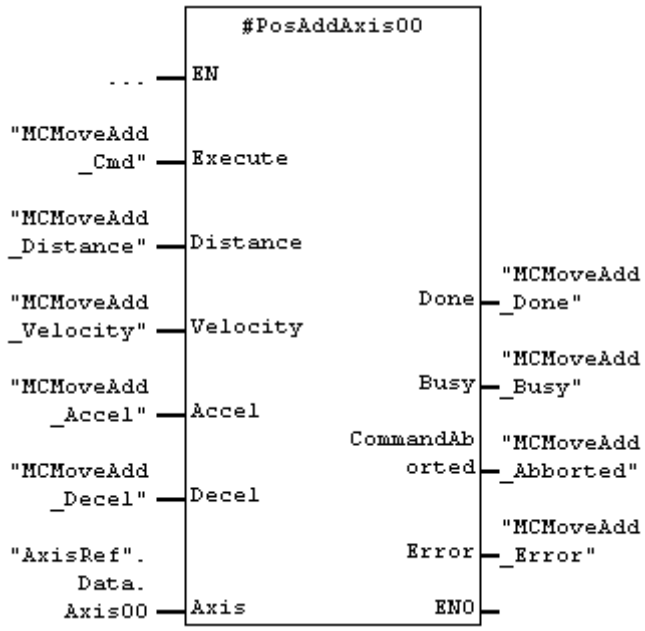


Im Stat-Bereich instanziierter MC_MoveRelative_C3 mit dem Instanznamen PosRAxis00.

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

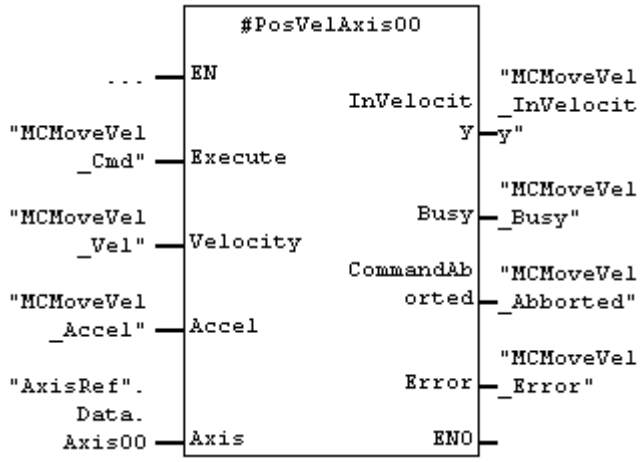


Im Stat-Bereich instanziierter MC_MoveAdditive_C3 mit dem Instanznamen PosAddAxis00.

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



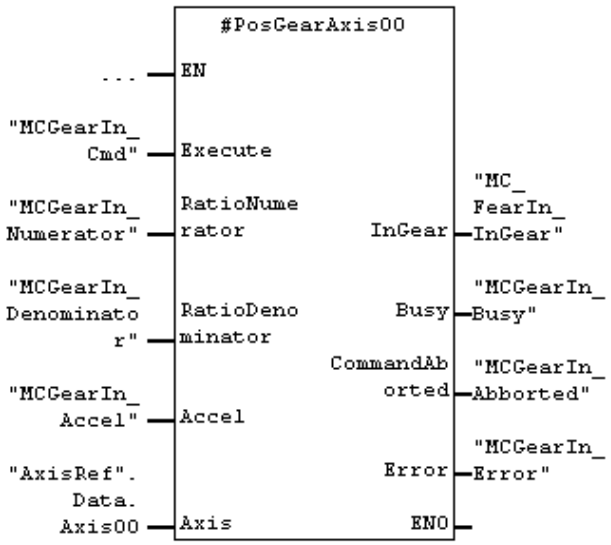
Im Stat-Bereich instanziierter MC_MoveVelocity_C3 mit dem Instanznamen PosVelAxis00.

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



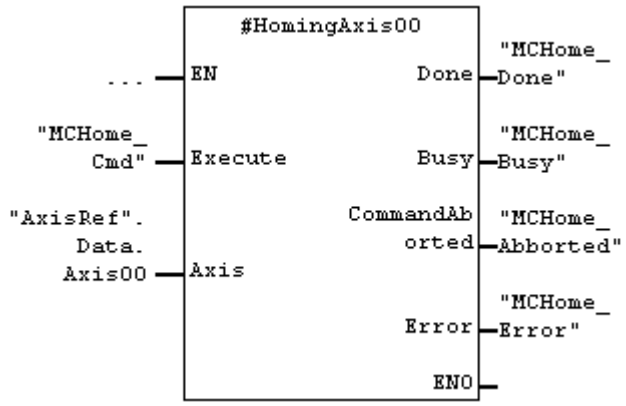
Im Stat-Bereich instanziierter MC_GearIn_C3 mit dem Instanznamen PosGearAxis00.

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



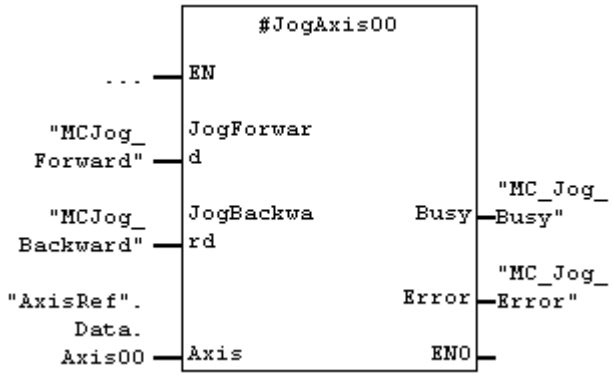
Im Stat-Bereich instanziierter MC_Home_C3 mit dem Instanznamen HomingAxis00.

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

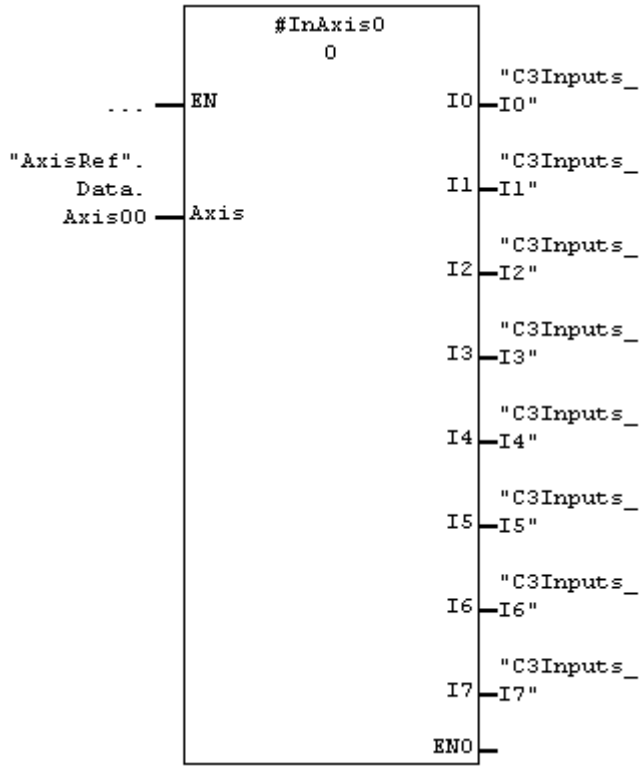


Im Stat-Bereich instanziierter MC_Jog_C3 mit dem Instanznamen JogAxis00.

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

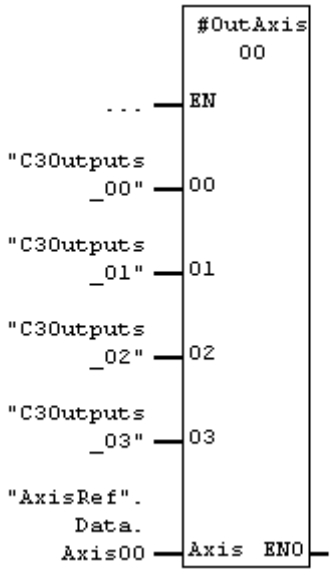


Im Stat-Bereich instanziierter C3_Input mit dem Instanznamen InAxis00.

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.

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]
```

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]
```

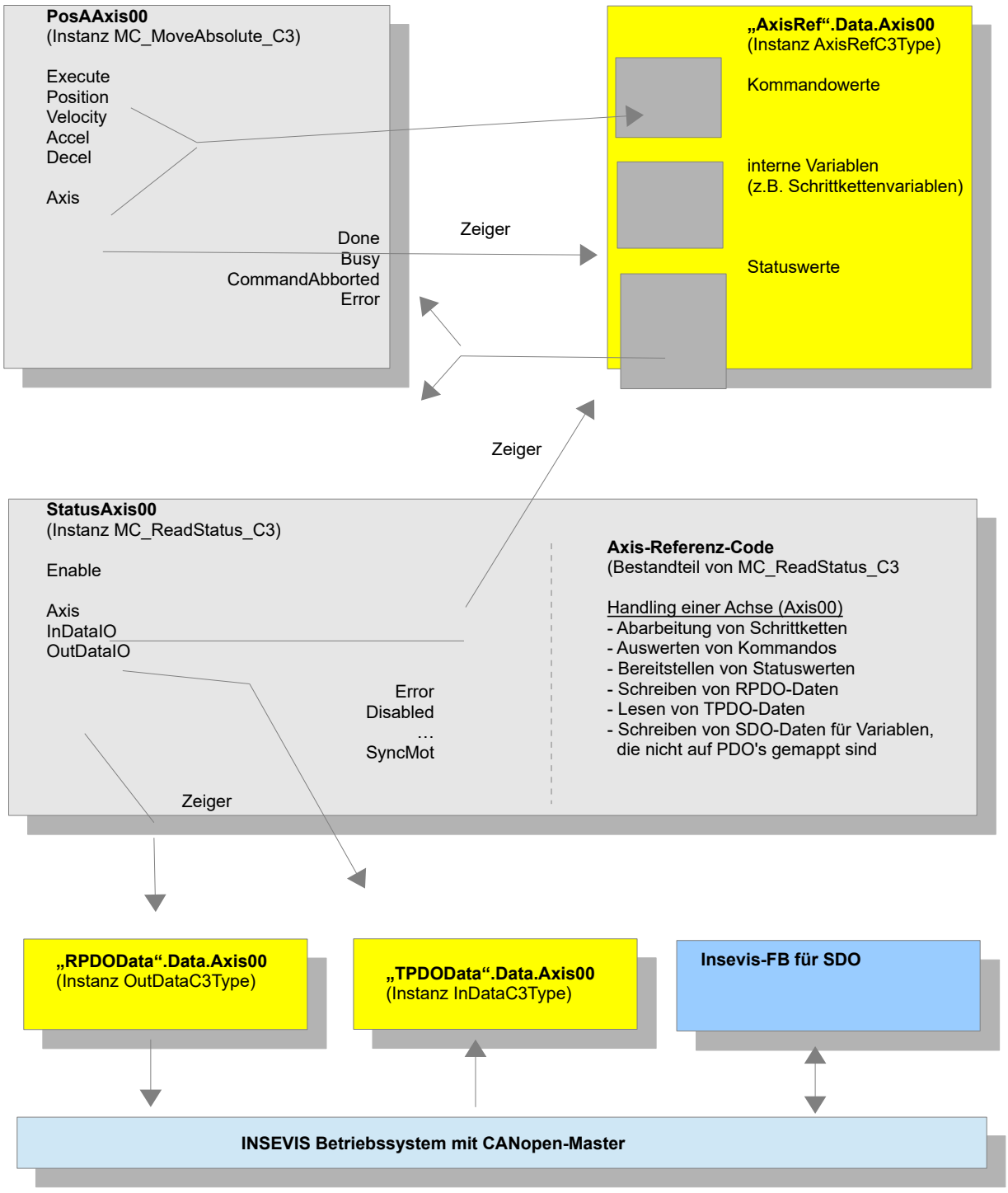
AxisRefC3Type

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.

Datenfluss am Beispiel einer MC-Block-Instanz

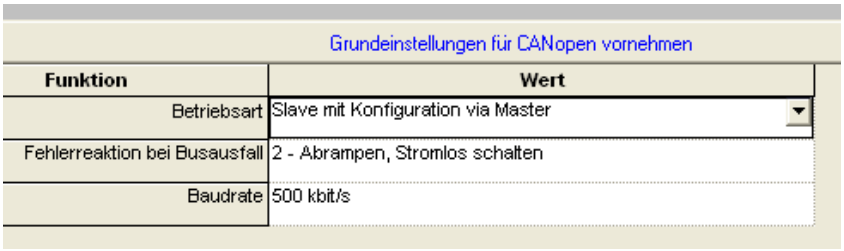
Folgende Grafik illustriert die Verwendung und den Datenfluss eines MC-Bausteines für eine Achse mit dem Namen Axis00.



CANopen-Konfiguration mit dem C3-ServoManager

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.

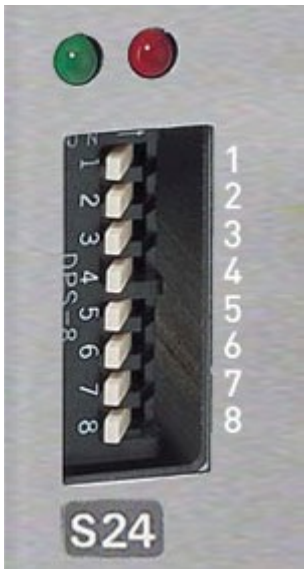


Betriebsart
Slave mit Konfiguration via Master
→ Die PDO's werden vom CANopen-Master konfiguriert und belegt.

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

Baudrate
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).

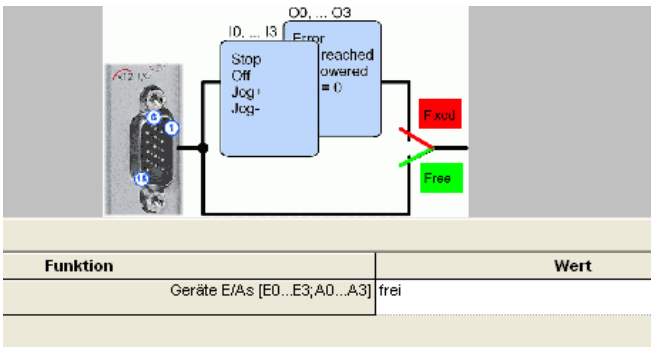


Einstellen der Node-ID
→ DIP-Schalter 1..7: Binäre Einstellung (OFF/ON) der Node-ID von 1 bis 127, DIP-Schalter 8: OFF

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:



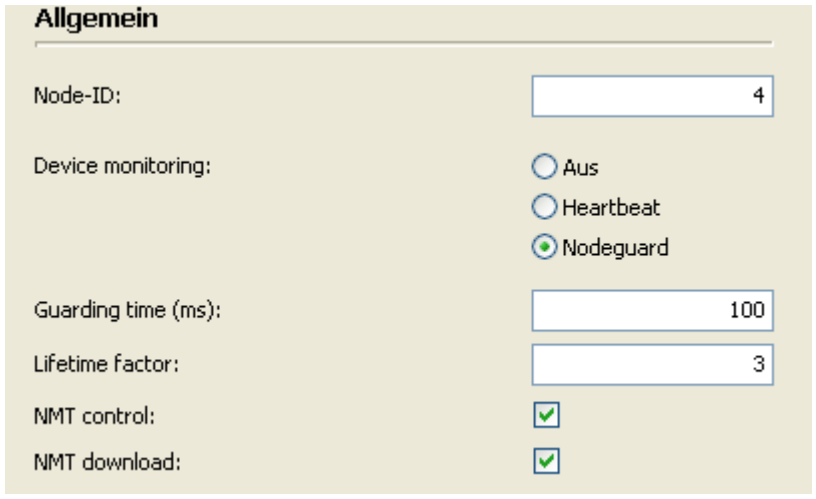
Die 24V-Eingänge E0..E7 können mit dieser Einstellung am C3 „frei“ genutzt und als zusätzliche Ressourcen für die SPS eingebunden werden. Wenn allerdings Endschalter genutzt werden, sind E5 bzw. E6 dafür fest vorzusehen. Wird bei einer Referenzierung ein Schalter benutzt, muss dieser auf E7 verwendet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

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

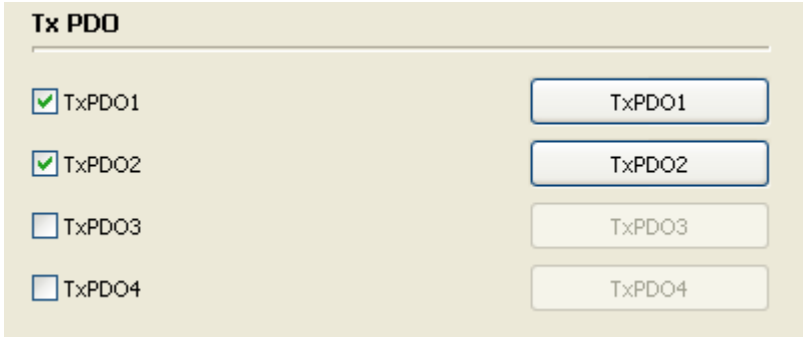
Slave-Konfiguration mit ConfigStage

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!



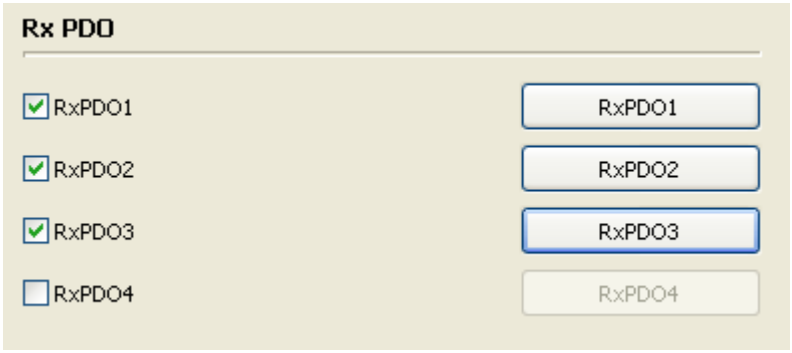
Festlegen der Node-ID und des Guardings
C3 unterstützt Nodeguarding
CANopen-Einstellungen (wie COB-ID's) zum C3 laden



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
	Zusätzliche Konfiguration über SDO 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.

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

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]

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]

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

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]
2	0x6084	0	32 Bit/DWord	Profilverzögerung [Einheiten/s^2 * 1000]

Zusätzliche SDO-Übertragung nach PDO-Mapping

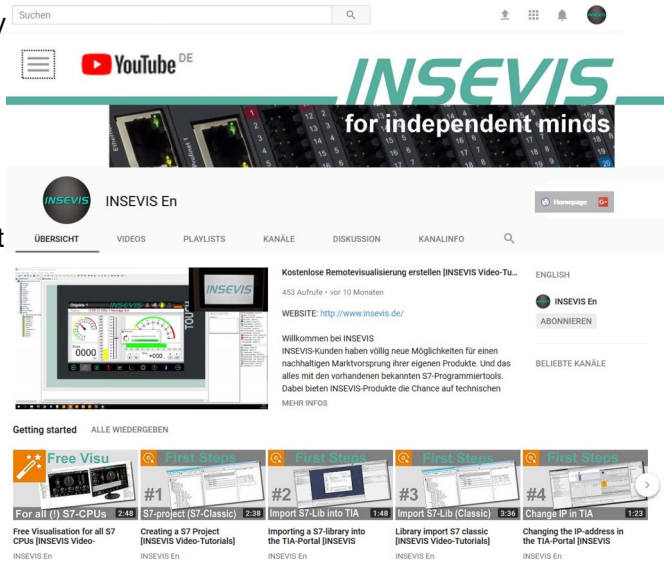
Nummer	Index	Subindex	Größe	Wert	Erklärung
1	0x605A	0	16 Bit/Word	6	„Quick stop mode“ so einstellen, dass ein Stopp zum Verwerfen der Position führt.

S7-Beispiel-Programm

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

Hint for better understanding by additional information

In the English YouTube-channel INSEVIS EN we supply different playlists with handling videos for single details. This will help you to get familiar with INSEVIS much faster.



Please download the referring manual from the download area of our English website insevis.com to get familiar with INSEVIS technology in detail.

Do you want to inform us about necessary increments or errors or do you want to provide us with your sample programs to offer it for free to all customers? Gladly we would provide your program -if you wish with the authors name- to all other customers of INSEVIS.

Hint to different versions of the sample programs

There could be older versions in delivery scope of the sample programs too. These were not updated and converted to the newest programming tool versions to allow access by older programming tools too. INSEVIS sample programs will be created in the present newest Siemens-programming tool always.

SAMPLE DESCRIPTION

index of content

1 Motivation....2
2 General principles of the software-design....2
3 Drive functions (MC-blocks and -types)....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)....7
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
4 MC_GearIn_C3 (FB)....11
4.1 MC_Home_C3 (FB)....12
4.2 MC_Jog_C3 (FB)....13
4.3 MC3_Input (FB)....14
4.4 C3_Output (FB)....14
4.5 InDataC3Type (UDT)....15
4.6 OutDataC3Type (UDT)....15
4.7 AxisRefC3Type....15
5 Sample of a MC-block-instance....16
6 CANopen-configuration with the C3-ServoManager....17

6.1 Configure communication....17
6.2 Configure C3-device....17
7 Slave-Configuration with ConfigStage....18
7.1 Mapping T-PDO1....19
7.2 Mapping T-PDO2....19
7.3 Mapping R-PDO1....19
7.4 Mapping R-PDO2....19
7.5 Mapping R-PDO3....19
7.6 Additional SDO-transfers after PDO-mapping....19
8 S7-Sample-program....19

Motivation

Manufacturer-specific S7-blocks will be offered from different vendors for an easy implementation of their own drive technology into the world of Simatic- and Simatic-compatible PLCs since years. This is often done by a very effective S7-block, adapted to the specials of the vendors drive, containing a monolithic and mostly customized interface and specialized in a certain bus system (normally Profibus DP, also Interbus S and CANopen based on fieldbus-master modules of other manufacturers).

The PLCopen (http://www.plcopen.org) as an international organisation is dedicated to reduce the effords for engineering by using general software interfaces. In the area of drive technology standards were defined, a certification of drives with implemented interfaces is possible. By using bus systems like CANopen with drive interfaces (DS402 drive profile) the effords for the adaption onto a certain bus protocol is unimportant.

In the following the operation on a servo drive Parker C3I21T11 (http://www.parker-eme.com) is described. The software was created for INSEVIS-PLC and is based on the PLCopen-standard

On following devices the software test was done:

C3I21T11

Testing device	:	C3I21T11
Software-version	:	2011 R09-11
C3ServoManager	:	V 2.9.2.49 (June 2011)

INSEVIS

Testing device	:	CC300V
operating system	:	2.0.23
S7-Library	:	Insevis_S7-library_from_2_0_22

Company inmotec Automation GmbH (support@inmotec.de) creates and expands drive specific software for INSEVIS-S7-controllers.

General principles of the software-design

1. All drive functions (so called Motion-Control-Blocks MC_) will be implemented as single function blocks, e.g. the function block „MC_Power_C3“, a S7-FB, is used to enable the motor. Because the motor does need not only to be enabled but also has to do motion functions, more function blocks are neccessary. Of course multiple axes were supported too. To prevent a various number of instances of an function block with separate instance blocks, an instanciation of function blocks in the STAT area of the variables definiton of the „container“-funktion block is recommended.
2. The MC-Blocks use no global resources as M-merker, T-times or Z-counter, but their instanciable IEC-variantes.
3. All drive functions of the INSEVIS-PLC communicate via asynchronous CANopen-PDO's reg. DS301, so that the efford for communication (bus load) is reduced. At the drive profile DS402 will be used operating modes only, what do not require equidistant transfers of demand values. The so called „interpolated mode“ will not be used.
4. The function blocks will be created in origin with SCL (Structured Control Language), an engineering-option to Step7 of Siemens. The use of these function blocks does not need a preinstalled SCL-package on the programming PC of the user.
5. To absorb diversities of the drives and name conflicts of already existing blocks from custom libraries (e.g. at the technology- PLC of Siemens), the MC-Blocks get a postfix like „_C3“ in reference to the regarding drive. There needs to be notified, that the instance name (in the sample „Axis00“) is not touched while swapping drives.
6. Because blocks do not reference each other, block-addresses (absolut numbers) can be adapted to the demands of the users program.

Drive functions (MC-blocks and -types)

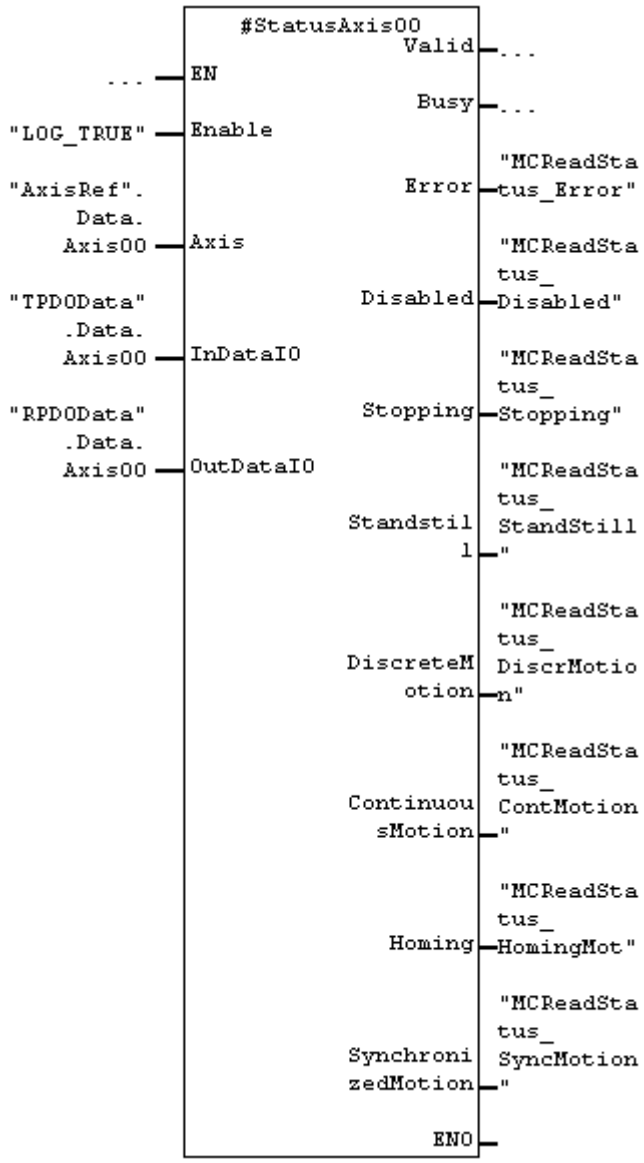
MC-Block/Symbol	Address	Function
MC_ReadStatus_C3	FB40	Visualization of drive states (currentless, stopping, remaining idle, profil based motion functions active, endlessmove active, synchronized motion functions active, reference move active)
MC_ReadAxisError_C3	FB41	Visualization of the error code of the drive
MC_ReadActualPosition_C3	FB42	Visualization of the actual postion of the motor
MC_ReadActualVelocity_C3	FB43	Visualization of actual velocity of the motor
MC_Reset_C3	FB44	Reset error in the drive
MC_Power_C3	FB45	Enable power stage to the motor or stop/disable fast as possible
MC_Stop_C3	FB46	Stop motor
MC_MoveAbsolute_C3	FB47	Move to an absolute position
MC_MoveRelative_C3	FB48	Move a relative distance
MC_MoveAdditive_C3	FB49	Append relative distance to a move
MC_MoveVelocity_C3	FB50	Endless move
MC_GearIn_C3	FB51	Execute synchronized motion functions (electronic gear), e.g. follow a master drive by encoder-pulses
MC_Home_C3	FB52	Execute homing move
MC_Jog_C3	FB53	Jog+/- move, stops on the software-end-delimiters
C3_Input	FB54	Read out C3-inputs (standard-inputs)
C3_Output	FB55	Write C3- outputs (Standard-outputs)
InDataC3Type	UDT100	Data type for input data CANopen, instanciate once per axis
OutDataC3Type	UDT101	Data type for output data CANopen, instanciate once per axis
SWPosC3Type	UDT102	Data type state word CANopen, INTERNAL USE ONLY
CWPosC3Type	UDT103	Data type control word CANopen, , INTERNAL USE ONLY
AxisRefC3Type	UDT104	Data type axis reference, instanciate once per axis

MC_ReadStatus_C3 (FB)

The MC_ReadStatus_C3 will be used for visualization (State generation) of different drive states. With these information the PLC-programm can watch most activities of the drive.

ⓘ	This block MUST be implemented into the PLC-program, because beside the state generation the complete axis reference will be processed. That's why this block size is larger than at the other MC-blocks. Explanations are made in the chapter axis reference. It is only one instance of this FB allowed and reasonable.

Name	Variables area	Type	Function
Enable	IN	Bool	Activate state generation For the processing of the axis reference the enable-input is not important, but anyway the FB MUST be called.
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
InDataIO	IN_OUT	InDataC3Type (UDT)	Reference to IO-data (input-data CANopen)
OutDataIO	IN_OUT	OutDataC3Type (UDT)	Reference to IO-data (output data CANopen)
Valid	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error
Disabled	OUT	Bool	Axis is disabled
Stopping	OUT	Bool	Axis is stopping
DiscreteMotion	OUT	Bool	Axis is positioning
ContinuousMotion	OUT	Bool	Axis is positioning endless
Homing	OUT	Bool	Axis executes „homing-move“
SynchronizedMotion	OUT	Bool	Axis is positioning synchronized (e.g. via electronic gear)

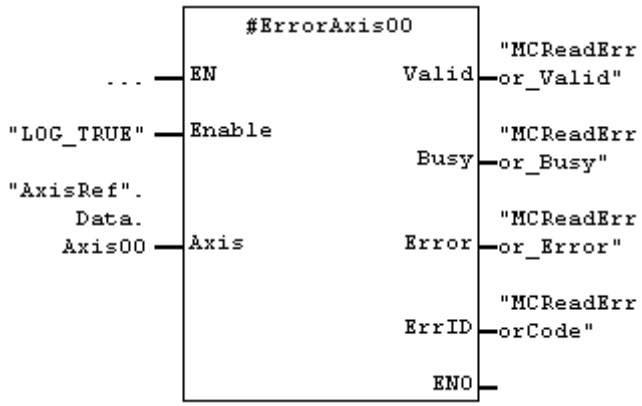


In the STAT-area of a Container-FB instanciated MC_ReadStatus_C3 with the instance name StatusAxis00.

MC_ReadAxisError_C3 (FB)

The MC_ReadAxisError_C3 will be used for visualization of the error code of the axis.
The meaning of the error code is mentioned in the drives help manual.

Name	Variables area	Type	Function
Enable	IN	Bool	Read error code
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Valid	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error
ErrorID	OUT	WORD	Error code of axis (here 16-bit)

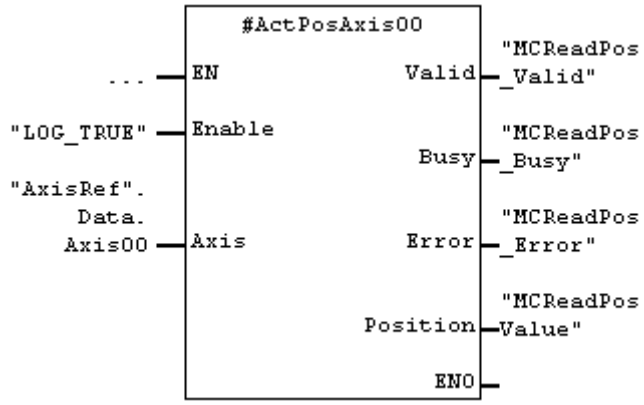


In the STAT-area of a Container-FB instanciated MC_ReadAxisError_C3 with the instance name ErrorAxis00.

MC_ReadActualPosition_C3 (FB)

The MC_ReadActualPosition_C3 provides the absolute position of the axis. This position can be changed by any kind of positioning, jogging, mechanical movments, homing moves or regulating oscillation.

Name	Variables area	Type	Function
Enable	IN	Bool	Read actual position
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Valid	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error
Position	OUT	REAL	Axis position in user units, e.g. „mm“



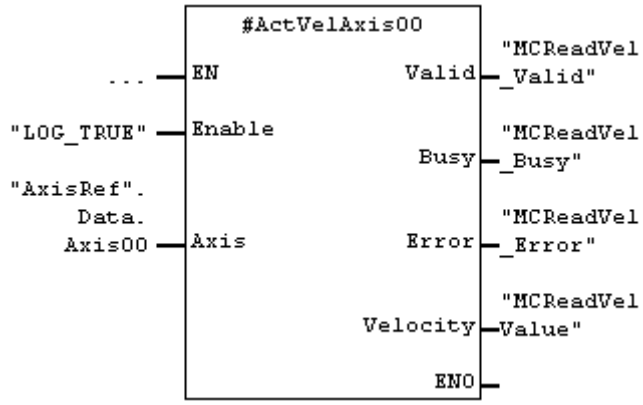
In the STAT-area of a Container-FB instanciated MC_ReadActualPosition_C3 with the instance name ActPosAxis00.

MC_ReadActualVelocity_C3 (FB)

The MC_ReadActualVelocity_C3 provides the actual velocity of the axis. The velocity can be changed by any kind of positioning, jogging, mechanical movments, homing moves or regulating oscillation.

Name	Variables area	Type	Function
Enable	IN	Bool	Read actual velocity
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)

Name	Variables area	Type	Function
Valid	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error
Velocity	OUT	REAL	Axis velocity in user units, e.g. „mm/s“



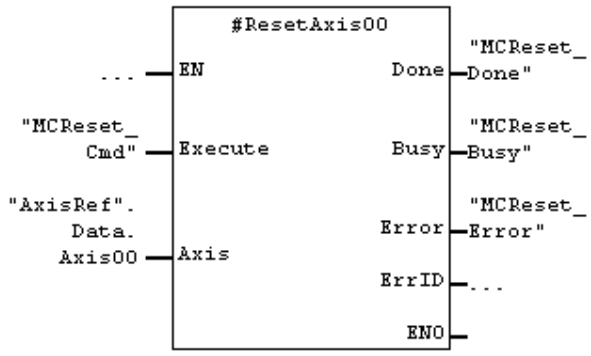
In the STAT-area of a Container-FB instanciated MC_ReadActualVelocity_C3 with the instance name ActVelAxis00.

MC_Reset_C3 (FB)

With the block MC_Reset_C3 the servo axis error will be reset

ⓘ	It is only one instance of this FB allowed and reasonable.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge receipes axis
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error
ErrorID	OUT	WORD	Error code of axis (here 16-bit)



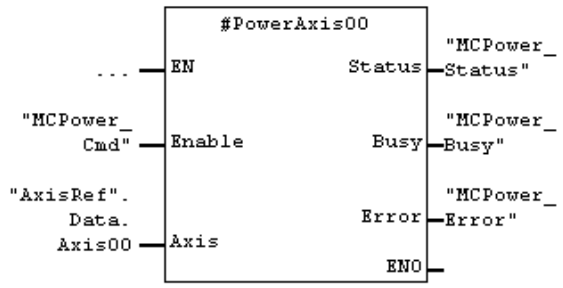
In the STAT-area instanciated MC_Reset_C3 with the instance name ResetAxis00.

MC_Power_C3 (FB)

With the block MC_Power_C3 the axis will be enabled or disabled.

ⓘ	It is only one instance of this FB allowed and reasonable. The immediate stop ramp and the immediate stop jerk will be configuredwerden by the C3-ServoManager.

Name	Variables area	Type	Function
Enable	IN	Bool	0-1-edge enables power stage axis, 1-0-edge executes an immediate stop with following switching to currentless
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Status	OUT	Bool	1 power stage enabled 0 disabled
Busy	OUT	Bool	Function enable power stage even active
Error	OUT	Bool	Axis with error



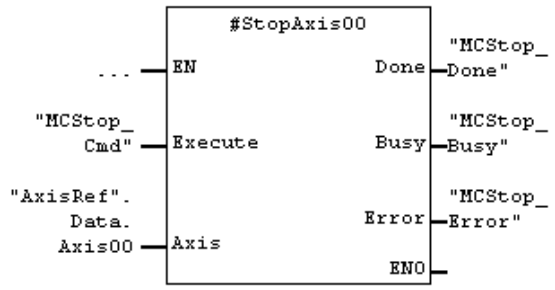
In the STAT-area instanciated MC_Power_C3 with the instance name PowerAxis00.

MC_Stop_C3 (FB)

With the block MC_Stop_C3 the axis will be stopped. Stopping is only possible with a power stage enabled axis.

ⓘ	It is only one instance of this FB allowed and reasonable. The stop ramp and the stop jerk will be configured with the C3 ServoManager. At an 0-1-edge the axis motion is stopped. Axis moves (new requests) will be blocked at activated Stop-Execute (=1) generally.

Name	Variables area	Type	Function
Execute	IN	Bool	1 Stoppes axis 0 Move enable of axis
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Axis stopped
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error

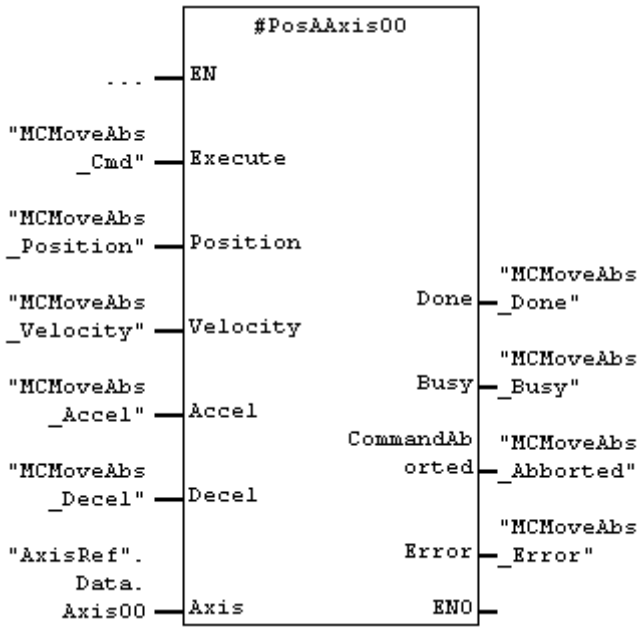


In the STAT-area instanciated MC_Stop_C3 with the instance name StopAxis00.

MC_MoveAbsolute_C3 (FB)

The block MC_MoveAbsolute_C3 will be used for absolute positioning. Reference point of the absolute position (the mathematical zero-pointis is defined by teaching (absolute encoder) or by homing reference travel.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge starts the move
Position	IN	Real	Absolute position in user units, e.g. „mm“
Velocity	IN	Real	Positioning velocity in user units, e.g. „mm/s“
Accel	IN	Dint	Acceleration in user units, e.g. „mm/s^2“
Decel	IN	Dint	Deceleration in user units, e.g. „mm/s^2“
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Axis has reached target position
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging, disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error

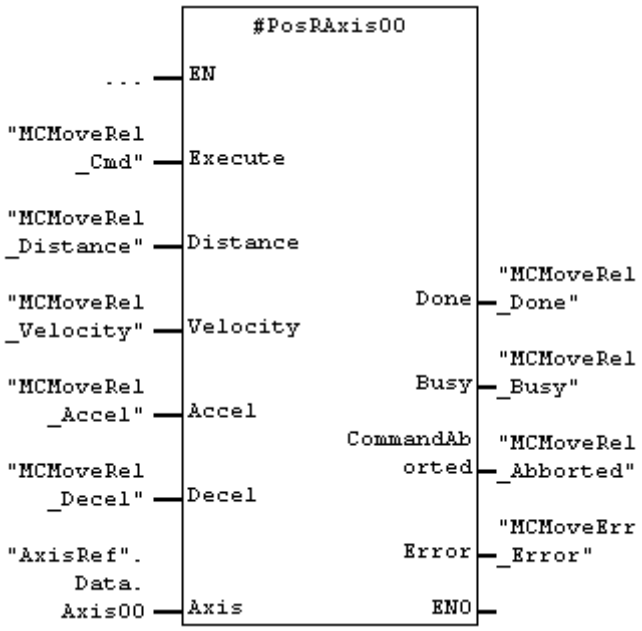


In the STAT-area instanciated MC_MoveAbsolute_C3 with the instance name PosAAxis00.

MC_MoveRelative_C3 (FB)

The block MC_MoveRelative_C3 will be used for relative positioning (for a distance). Reference point for the distance is the actual target position. This kind of positioning is referred as chain positioning.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge starts the move
Distance	IN	Real	Distance in user units e.g. „mm“
Velocity	IN	Real	Positioning velocity in user units e.g. „mm/s“
Accel	IN	Dint	Acceleration in user units, e.g. „mm/s^2“
Decel	IN	Dint	Deceleration in user units, e.g. „mm/s^2“
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Axis has reached target position
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging, disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error

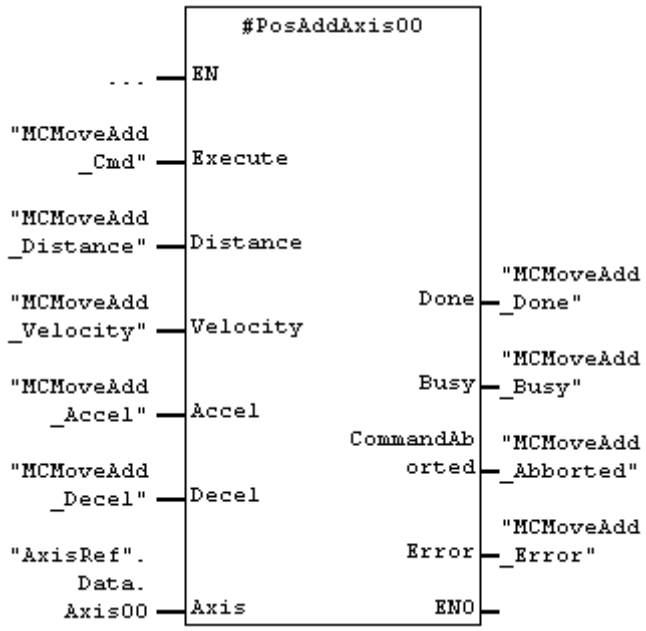


In the STAT-area instanciated MC_MoveRelative_C3 with the instance name PosRAxis00.

MC_MoveAdditive_C3 (FB)

The bock MC_MoveAdditive_C3 will be used for additive positioning (for a distance). In difference to the relative positioning the distance will be added here onto the actual target.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge starts the move
Distance	IN	Real	Distance in user units e.g. „mm“
Velocity	IN	Real	Positioning velocity in user units e.g. „mm/s“
Accel	IN	Dint	Acceleration in user units, e.g. „mm/s^2“
Decel	IN	Dint	Deceleration in user units, e.g. „mm/s^2“
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Axis has reached target postion (distance driven)
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging, disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error

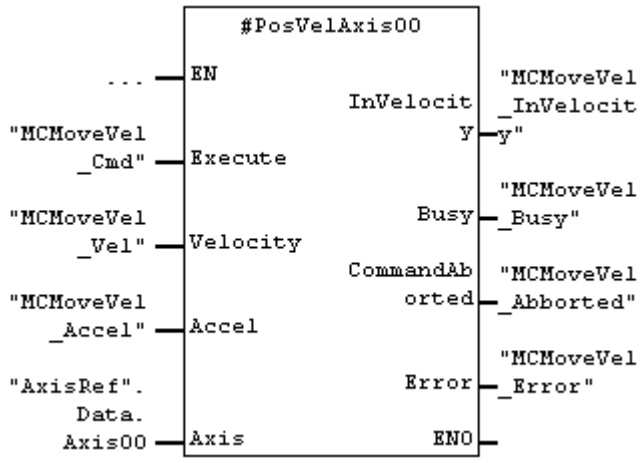


In the STAT-area instanciated MC_MoveAdditive_C3 with the instance name PosAddAxis00.

MC_MoveVelocity_C3 (FB)

The block MC_MoveVelocity_C3 will be used for endless moves, whereat the position controller stays active, so that during position. The move MUST be stopped witht MC_Stop_C3.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge startes the move
Velocity	IN	Real	Positioning velocity in user units e.g. „mm/s“
Accel	IN	Dint	Acceleration and deceleration in user units, e.g. „mm/s^2“
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
InVelocity	OUT	Bool	Axis has reached target velocity
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging, disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error



In the STAT-area instanciated MC_MoveVelocity_C3 with the instance name PosVelAxis00.

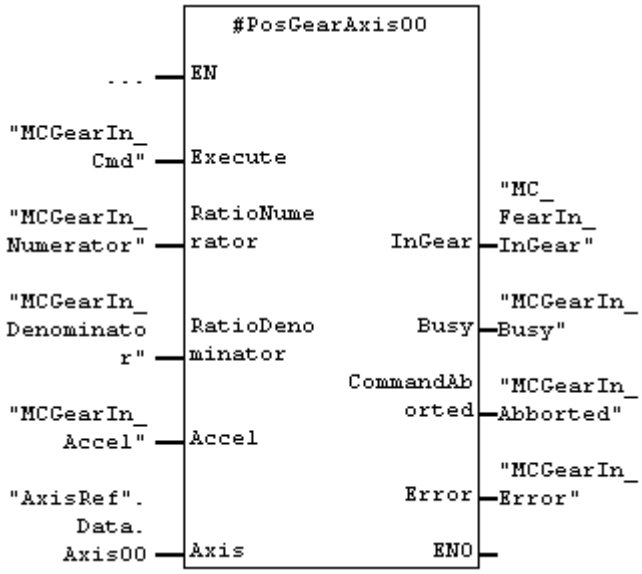
MC_GearIn_C3 (FB)

The block MC_GearIn_C3 will be used, if the axis should follow a master drive (e.g. another axis or an encoder). In that case for a positioning a time based move profile will not be calculated, but a under limitation on a maximale acceleration resp. deceleration a guide value, what can be a RS422-encoder signal or an analog value or a motion-bus-setpoint (using C3 the so called HEDA-Bus). Type and resolution of the master setpoint values will be configured by the C3-ServoManager.

The ratio of numerator/ denominator will be transferred to the servo drive. It is necessary to pay attention that as minimum gear reduction 0.001 can be transferred, also the ratio is not better adjustable than with 0.001.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge startes the move

Name	Variables area	Type	Function
RatioNumerator	IN	Real	Numerator gear factor
RatioDenominator	IN	Dint	Denominator gear factor
Accel	IN	Dint	Maximal acceleration and deceleration in user units e.g. „mm/s²“
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
InVelocity	OUT	Bool	Axis has reached target velocity
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging, disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error



In the STAT-area instanciated MC_GearIn_C3 with the instance name PosGearAxis00.

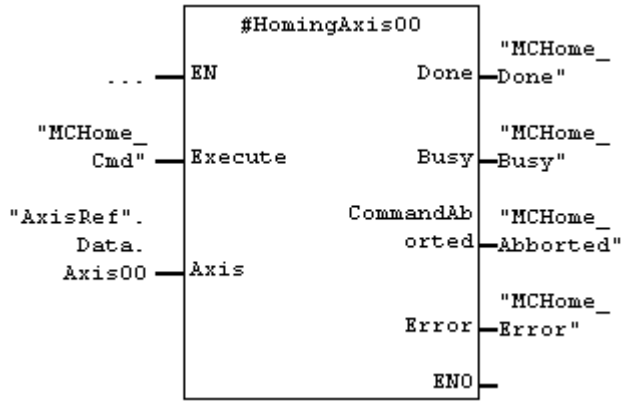
MC_Home_C3 (FB)

the C3-manual. If an absolute encoder or an absolute encoder simulation will be used, the mode must be set to 0 after teaching (no referencing necessary). The profile values (velocity, acceleration, jerk) are part of the C3-configuration.

ⓘ	It is only one instance of this FB allowed and reasonable. (1) Set up the homing-offset in the configuration. After the referencing the axis will be reported as actual position with a value of -1.0 * homing-offset. (2) It is also possible to set up in the C3-ServoManager, if later on during the homing move the mathematical zero point shell be driven too.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge startes the reference move

Name	Variables area	Type	Function
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Reference move finished, the set position was set, the mathematic zero point and if necessary the software end-limits are valid
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging, disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error



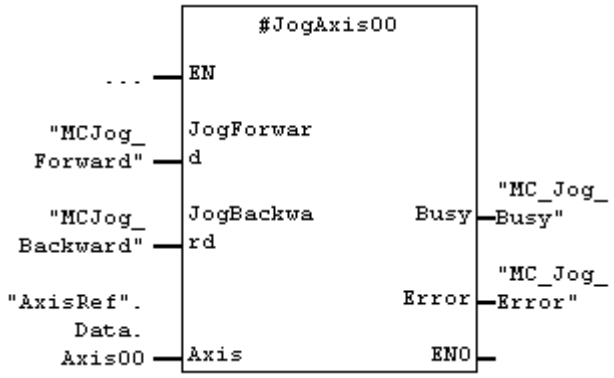
In the STAT-area instanciated MC_Home_C3 with the instance name HomingAxis00.

MC_Jog_C3 (FB)

The block MC_Jog_C3 will be used to move the axis „manually“ (also named „inching“). Both directions are possible, the profile values (acceleration, deceleration, jerk) will be defined by the C3-configuration. If there are desinded software-end-limits, the axis stopps on the defined software-end-limits.

ⓘ	It is only one instance of this FB allowed and useful.

Name	Variables area	Type	Function
JogForward	IN	Bool	0-1-edge starts jogging clockwise, 1-0-edge stops the axis
JogBackward	IN	Bool	0-1-edge starts jogging counterclockwise, 1-0-edge stops the axis
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error

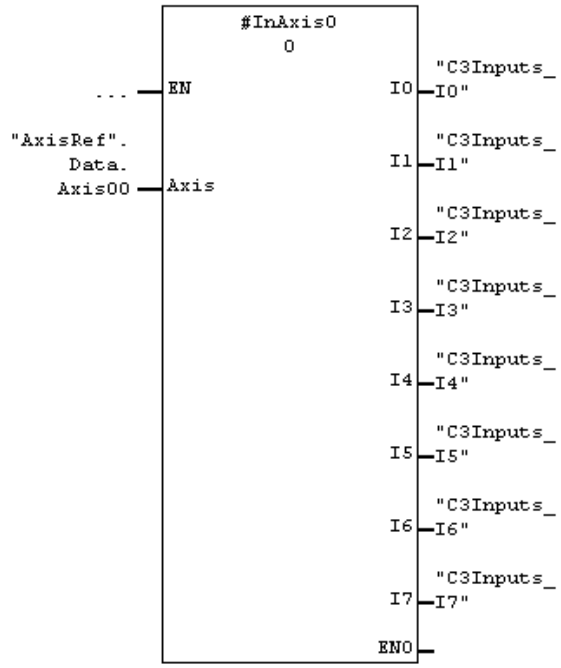


In the STAT-area instanciated MC_Jog_C3 with the instance name JogAxis00.

MC3_Input (FB)

The block C3_Input is a special version for the C3-axis, which provides the „lower“ C3-inputs 0 to 7 as state. Input 5 and 6 are reserved for position switches, so that the polarity can be reverse there by correlative configuration. Input I7 is reserved for the reference/homing switch.

Name	Variables area	Type	Function
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
I0 bis I7	OUT	Bool	Inputs 0 to 7 as state

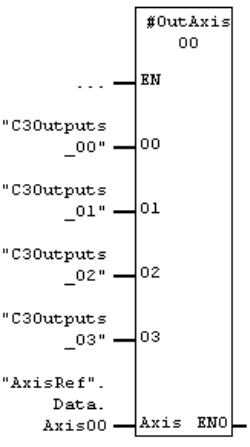


In the STAT-area instanciated C3_Input with the instance name InAxis00.

C3_Output (FB)

The block C3_Output is a special version for the C3-axis, which provides the „lower“ C3-outputs0 to 3 to set/reset.

Name	Variables area	Type	Function
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis reference (axis pointer)
O0 bis O3	IN	Bool	Outputs 0 to 3 to set/reset



In the STAT-area instanciated C3_Output with the instance name OutAxis00.

InDataC3Type (UDT)

This data type is to instanciate while using in a data block with a name, e.g.. Axis00. Exactly 1 instance per axis will be needed. The instance data correlate to the T-PDO-data of the C3-axis.

ⓘ	The best solution is to set up all instances (of the different axis´) of InDataC3Type in a separate DB (e.g. „TPDODATA“).

```
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]
```

OutDataC3Type (UDT)

This data type is to instanciate while using in a data block with a name, e.g.. Axis00. Exactly 1 instance per axis will be needed. The instance data correlate to the R-PDO-data of the C3-axis.

ⓘ	The best solution is to set up all instances (of the different axis´) of OutDataC3Type in a separate DB (e.g. „RPDODATA“).

```
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. acceleration [units/s2]
diProfDecel    :   DINT;      // RPDO3, async, 0x6084 + 0x00, Prof. deceleration [units/s2]
```

AxisRefC3Type

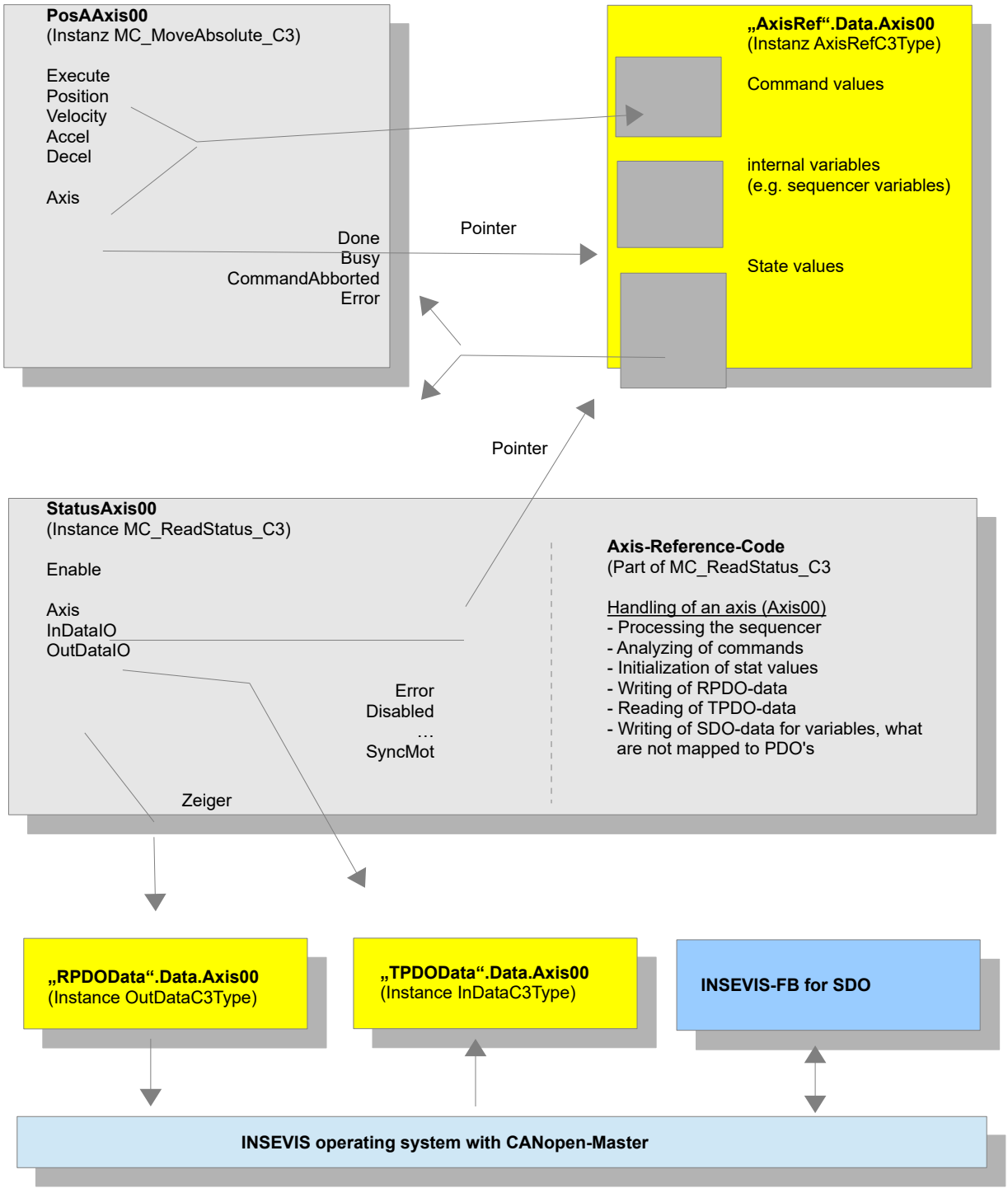
This data type will be used internally as axis working data of the axis. Because the instanciated variables were handled over as IN_OUT, the efford for copying is low.

The block MC_ReadStatus_C3 uses these variables, definded by the axis reference, for processing the sequencers.

ⓘ	The best solution is to set up all instances (of the different axis´) of AxisRefC2Type in a separate DB (e.g. „AXISREF“)

Sample of a MC-block-instance

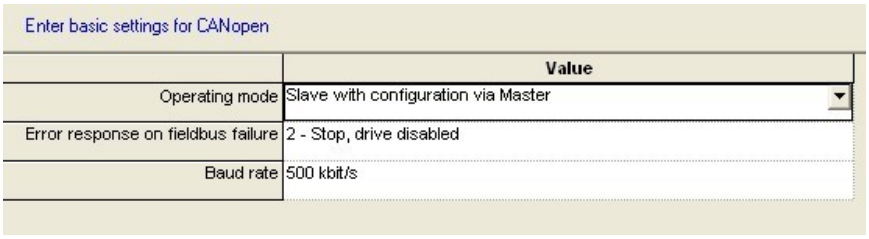
Following figure shows the use and the data flow of a MC-block for one axis with the name Axis00.



CANopen-configuration with the C3-ServoManager

Configure communication

Gererally there can not payed attention for the common configuration of a C3-servo drive. Important-for the CANopen-part is only, that C3I21 can be configured with up to 4 PDO's in each direction and that it supports SDO's. Following settings have to be done by the C3-Wizard for CAN.

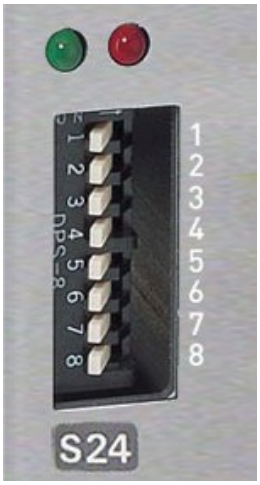


operation mode
Slave with configuration via master
→ The PDO's will be configured and allocated from CANopen-Master

Error reaction while bus failure
→ Here was choosen an alternative, where the servo drive stopps at bus failure and switches currentless.

Baud rate
Adjustable in steps from 20 kBit/s up to 1 MBits/s

On the C3-device the Node-ID must be set up, optional the baud rate can be set, what is only useful, if the device would be configured completely via CAN. The efford for this is really high (see the manufacturers instruction for more).

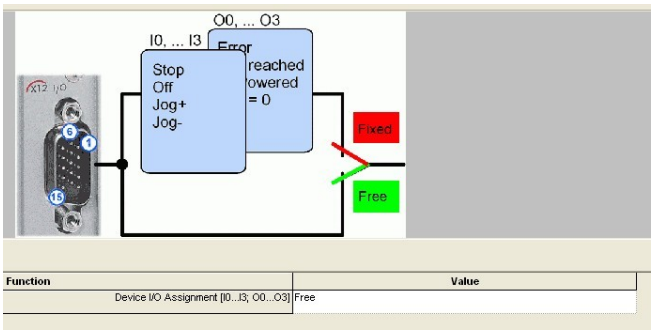


Set up the Node-ID
→ DIP-switch 1..7: binary setting (OFF/ON) of the
Node-ID from 1 bis 127, DIP-switch 8: OFF

Configure C3-device

The device is to be configured referring to the requrements of your hardware (device, motor) and application. Parameters, like e.g. the jog-profile are to provide, because not every parameter can be provided via the MC-blocks.

Note following setting for the in-/outputs on the C3:



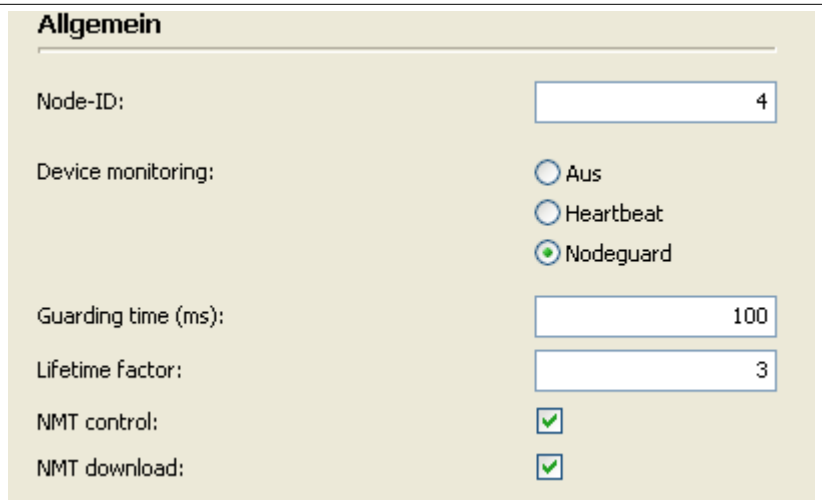
The 24V-inputs I0...I7 can be used with this setting on the C3 „free“ and can be used as additional resources for the PLC. If position switches will be used I5 resp. I6 are designed for this purpose. If a switch is used on a reference move, I7 must be used for it.

The 24V-outputs O0...O3 are available as additional resources for the PLC. Note the boundary conditions like (current capability, switching of inductive loads, and so on).

Slave-Configuration with ConfigStage

With the ConfigStage-software will amongst others configurated the CANopen-Master and each CANopen-Slave. Also the connection from PLC-data (e.g. DataBlock and offset in the DataBlock) to the CANopen-data (R-PDO's, T-PDO's) will bedefined.

The axis can be taken over as type into the library of the ConfigStage!



Definition of Node-ID and Guardings

C3 supports Nodeguarding

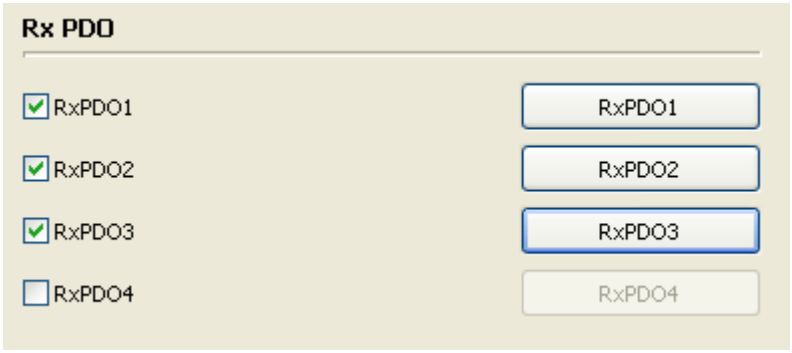
CANopen-settings (like COB-ID's) to load to C3

TPDO (C3 → CANopen-Master)

2 T-PDO's are necessary for the receive direction.
activate the download of the communication parameter and mapping

Response characteristics TPDO1
Typ: 254, no blocking time

Response characteristics TPDO2
Typ: 254, define a blocking time of e.g. 100ms !

	RPDO (CANopen-Master → C3) 3 R-PDO's are necessary for the send direction. activate the download of the communication parameter and mapping Response characteristics RPDO1 Typ: 254, no blocking time Response characteristics RPDO2 Typ: 254, no blocking time Response characteristics RPDO3 Typ: 254, no blocking time
	additional configuration via SDO After the download of the mapping parameters (initiated by PLC-firmware) further settings on the C3, what can not be made by the C3-ServoManager-configuration, will be transfererd by SDO

Mapping T-PDO1

Offset in data area (e.g. data block) of an instance from type „InDataC3Type“: **0** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6041	0	16 Bit/Word	State word DS402
2	0x6061	0	16 Bit/Word	Actual operation mode DS402
3	0x6100	1	16Bit/Word	Basic inputs C3
4	0x603F	0	16Bit/Word	Error code C3

Mapping T-PDO2

Offset in data area (e.g. data block) of an instance from type „InDataC3Type“: **8** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6064	0	32 Bit/DWord	Actual position {units * 1000]
2	0x606C	0	32 Bit/DWord	Actual velocity [units/s * 1000]

Mapping R-PDO1

Offset in data area (e.g. data block) of an instance from type „OutDataC3Type“: **0** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6040	0	16 Bit/Word	Control word DS402

Number	Index	Subindex	Size	Explanation
2	0x6060	0	16 Bit/Word	Operation mode DS402
3	0x607A	0	32Bit/DWord	Desired value 1 (variable), [e.g. units * 1000]

Mapping R-PDO2

Offset in data area (e.g. data block) of an instance from type „OutDataC3Type“: **8** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6081	0	32 Bit/DWord	Profile velocity [units/s * 1000]
2	0x6300	1	16Bit/Word	Basic outputs C3

Mapping R-PDO3

Offset in data area (e.g. data block) of an instance from type „OutDataC3Type“: **12** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6083	0	32 Bit/DWord	Profile acceleration [units/s^2 * 1000]
2	0x6084	0	32 Bit/DWord	Profile deceleration [units/s^2 * 1000]

Additional SDO-transfers after PDO-mapping

Number	Index	Subindex	Size	Value	Explanation
1	0x605A	0	16 Bit/Word	6	„Quick stop mode“ set up so, that a stopp causes a reject of the position

S7-Sample-program

The sample project consists of an S7-program, what demonstrates the application of the MC-blocks.

INSEVIS

INSEVIS Vertriebs GmbH

Am Weichselgarten 7
D - 91058 Erlangen

Fon: +49(0)9131-691-440
Fax: +49(0)9131-691-444
Web: www.insevis.de
E-Mail: info@insevis.de