

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.....	1
2 Grundprinzipien des Software-Entwurfs.....	2
3 Antriebsfunktionen (MC-Bausteine und -Typen).....	2
3.1 Einstellungen für EPOS2.....	3
3.2 MC_ReadStatus_E2 (FB).....	3
3.3 MC_ReadAxisError_E2 (FB).....	5
3.4 MC_ReadActualPosition_E2 (FB).....	5
3.5 MC_ReadActualVelocity_E2 (FB).....	6
3.6 MC_Reset_E2 (FB).....	6
3.7 MC_Power_E2 (FB).....	7
3.8 Funktionsbausteine im Profile Position Mode.....	7
3.8.1 MC_Stop_E2 (FB).....	8
3.8.2 MC_MoveAbsolute_E2 (FB).....	9
3.8.3 MC_MoveRelative_E2 (FB).....	10
3.9 Funktionsbausteine im Profile Velocity Mode.....	10
3.9.1 MC_MoveVelocity_E2 (FB).....	11
3.10 MC_Jog_E2 (FB).....	12
3.11 Funktionsbausteine im Homing Mode.....	13
3.11.1 MC_Home_E2 (FB).....	14
3.12 Spezial-FB's.....	15
3.13 E2_Input (FB).....	15
3.14 InDataE2Type (UDT).....	16
3.15 OutDataE2Type (UDT).....	16
3.16 AxisRefE2Type.....	16
4 Datenfluss am Beispiel einer MC-Block-Instanz.....	17
5 CANopen-Konfiguration mit dem EPOS-Studio.....	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.....	21
6.6 Mapping R-PDO4.....	21
6.7 Zusätzliche SDO-Übertragung nach PDO-Mapping.....	21
7 S7-Beispiel-Programm.....	21

1 Motivation

Seit Jahren werden von Firmen der Antriebstechnikbranche herstellereigenspezifische 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 angepasstem, monolithischem 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 Maxon EPOS2 24/5 (<http://www.maxonmotor.de>) 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:

EPOS2

Testgerät : EPOS2 24/5
 Software-Version : 0x2122
 Hardware-Version : 0x6220
 EPOS-Studio : 1.44 Revision 1

INSEVIS

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

Es werden aktuell nicht alle verfügbaren Modi unterstützt, wie z.B. der „Master Encoder Mode“, da die HW-Plattformen zu diversitär sind, um mit dem gleichen Encoder-Typ zu arbeiten. Bei Bedarf könnte ein MC_Gear_E2 grundsätzlich nach-implementiert werden.

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_E2“ 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 instanzierbaren 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 „_E2“ 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)

Grundsätzlich wird der Betrieb mit Encoder empfohlen, wenn Positionieraufgaben zu erfüllen sind. Erfolgt die Drehzahlerfassung über Hallensoren, können keine Positionieraufgaben realisiert werden. Lediglich der Drehzahlbetrieb ist sinnvoll. Es wird empfohlen, dann den Motor mit mindestens 1000 U/min zu betreiben.

MC-Baustein/Symbol	Adresse	Funktionalität
MC_ReadStatus_E2	FB40	Visualisierung der Antriebszustände (entstromt, stoppend, profilbasierte Bewegungsfunktionen aktiv, Endlosdrehen aktiv, Referenzfahrt aktiv)
MC_ReadAxisError_E2	FB41	Visualisierung des Fehlercodes des Antriebs

MC-Baustein/Symbol	Adresse	Funktionalität
MC_ReadActualPosition_E2	FB42	Visualisierung der aktuellen Position des Motors
MC_ReadActualVelocity_E2	FB43	Visualisierung der aktuellen Geschwindigkeit des Motors
MC_Reset_E2	FB44	Fehler im Antrieb rücksetzen
MC_Power_E2	FB45	Motor bestromen/entstromen mit Schnellstopprampe
MC_Stop_E2	FB46	Bestromten Motor stoppen mit Schnellstopprampe
MC_MoveAbsolute_E2	FB47	Absolute Position anfahren
MC_MoveRelative_E2	FB48	Relative Distanz abfahren
MC_MoveVelocity_E2	FB50	Endlosbewegung (Drehzahlvorgabe)
MC_Home_E2	FB52	Referenzfahrt ausführen
MC_Jog_E2	FB53	Hand+/- fahren, stoppt an den Software-Endgrenzen
E2_Input	FB54	Eingänge auslesen
InDataE2Type	UDT100	Datentyp für Eingangsdaten CANopen, pro Achse einmal instantiieren
OutDataE2Type	UDT101	Datentyp für Ausgangsdaten CANopen, pro Achse einmal instantiieren
SWPosE2Type	UDT102	Datentyp Statuswort CANopen, NUR IINTERNE VERWENDUNG
CWPosE2Type	UDT103	Datentyp Steuerwort CANopen, NUR IINTERNE VERWENDUNG
AxisRefE2Type	UDT104	Datentyp Achsreferenz, pro Achse einmal instantiieren

3.1 Einstellungen für EPOS2

Da der EPOS2 feste Formate für Positionen (Post-Quad-Inkrement), Geschwindigkeiten (U/min) und Rampen (U/min/s) verwendet, ist eine Umsetzung, die die Vorgabe eher nachvollziehbaren Ingenieurwerten ermöglicht, sinnvoll.

Der Anwender hat daher folgende Parameter in die Achsreferenz zu schreiben, damit direkt in Nutzereinheiten, Nutzereinheiten/s und Nutzereinheiten/s² programmiert werden kann:

```
L      1.000000e+001           // z.B. 10 mm Vorschub/Umdrehung
T      "AxisRef".Data.Axis00.fMechanicPitch
L      4.000000e+003           // Inkremente Enc. nach
                                   // Vervierfachung, z.B. 4000 bei
                                   // einem 1000-Impuls-Encoder
T      "AxisRef".Data.Axis00.fEncPQPerMotRev
```

Zusätzlich ist die CANopen-Knotennummer für z.B. azyklische Abfragen (Fehlernummer) vorzugeben:

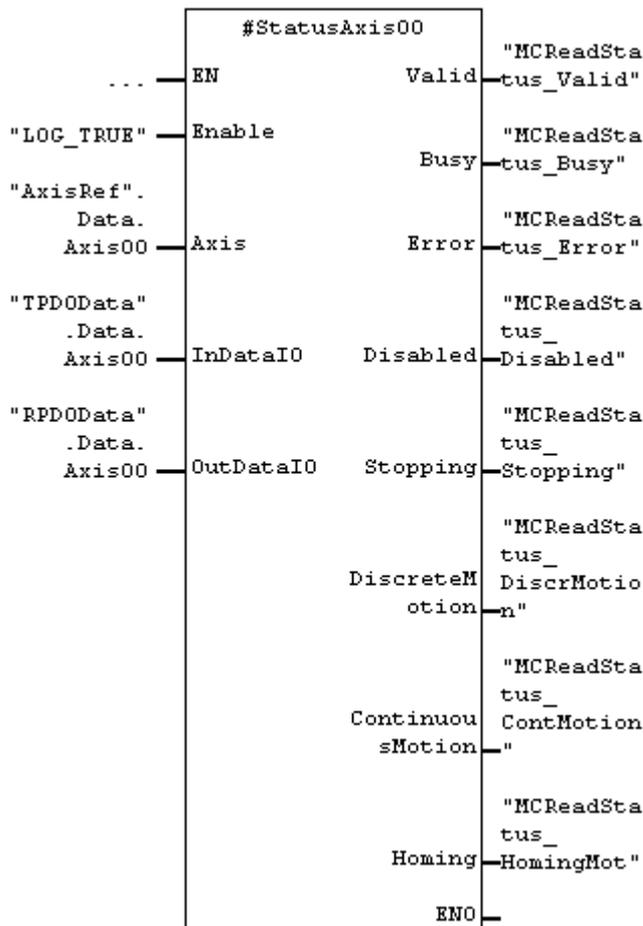
```
L      4
T      "AxisRef".Data.Axis00.iNode
```

3.2 MC_ReadStatus_E2 (FB)

Der MC_ReadStatus_E2 wird zur Visualisierung (Statusbildung) verschiedener Antriebszustände verwendet. Anhand dieser Informationen kann das SPS-Programm Aktivitäten des Antriebs verfolgen.

	<p>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.</p> <p>Es ist nur eine Instanz dieses FB's sinnvoll und erlaubt.</p>
---	--

Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	Statusbildung aktivieren Für die Abarbeitung der Achsreferenz ist der Enable-Eingang unbedeutend, der FB muss einmal pro Zyklus aufgerufen werden.
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
InDataIO	IN_OUT	InDataE2Type (UDT)	Referenz auf IO-Daten (Eingangsdaten CANopen)
OutDataIO	IN_OUT	OutDataE2Type (UDT)	Referenz auf IO-Daten (Ausgangsdaten CANopen)
Valid	OUT	Bool	FB-Daten sind gültig (sobald Enable = True)
Busy	OUT	Bool	FB-Funktion läuft (immer False, da aus PDO-Daten erzeugt)
Error	OUT	Bool	Achse mit Fehler
Disabled	OUT	Bool	Achse stromlos
Stopping	OUT	Bool	Achse stoppt (gerade)
DiscreteMotion	OUT	Bool	Achse positioniert
ContinuousMotion	OUT	Bool	Achse positioniert endlos
Homing	OUT	Bool	Achse führt Homing-Fahrt aus



Im Stat-Bereich eines Container-FB's instanzierter MC_ReadStatus_E2 mit dem Instanznamen StatusAxis00.

3.3 MC_ReadAxisError_E2 (FB)

Der MC_ReadAxisError_E2 wird zur Visualisierung des Fehlercodes der Achse verwendet. **Die Bedeutung des Fehlercodes ist der Hilfeanleitung des EPOS2 zu entnehmen.**

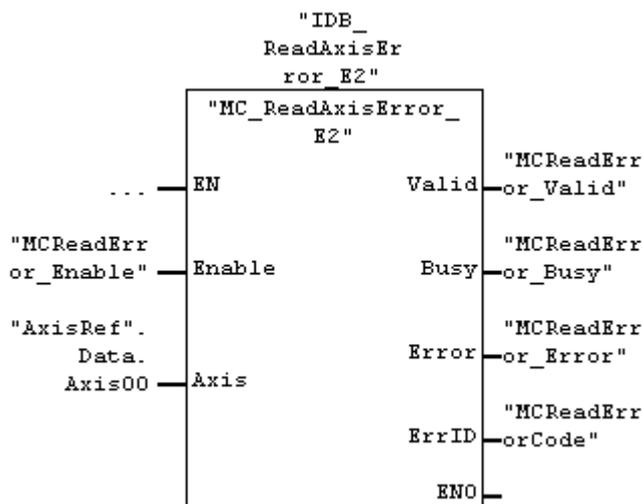
Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	Fehlercode lesen
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Valid	OUT	Bool	FB-Daten sind gültig (sobald Enable = True)
Busy	OUT	Bool	FB-Funktion (SDO-Transfer) läuft (gerade)
Error	OUT	Bool	Achse mit Fehler
ErrorID	OUT	DWORD	Fehlercode Achse (hier 32-Bit)

i Gelesen wird das EPOS2-CANopen-Objekt 0x1003 auf Subindex 1, welches den aktuellen Fehlercode des Gerätes enthält. Das Objekt wird mittels CANopen-SDO gelesen, sobald das Fehlerbit im Status-Wort auf TRUE gesetzt wird (Flanke) oder der Enable-Eingang auf TRUE (Flanke) gesetzt wird.

Erfolgt nach einer TIME-OUT-Zeit von 150ms keine Antwort vom Gerät oder wird beim SDO-Transfer ein Fehler gemeldet, wird ein Fehlercode DW#16#FFFFFFF (SDO-Transfer-Fehler) angezeigt.

Der FB kann in der aktuellen Version nicht in dem STAT-Bereich eines Instanz-DB's eines übergeordneten FB's instanziiert werden! Es muss eine separate Instanz für den MC_ReadAxisError_E2 angelegt werden.

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



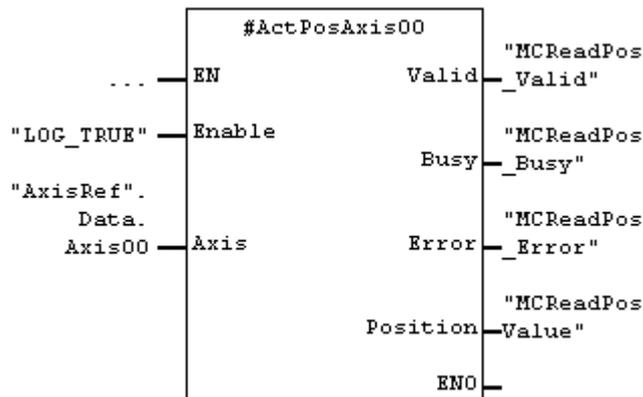
Separat instanziiertes MC_ReadAxisError_E2 mit dem Instanznamen ErrorAxis00.

3.4 MC_ReadActualPosition_E2 (FB)

Der MC_ReadActualPosition_E2 stellt die absolute Istposition der Achse bereit.

Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	Fehlercode lesen
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Valid	OUT	Bool	FB-Daten sind gültig (sobald Enable =

Name	Variablenbereich	Typ	Funktion
			True)
Busy	OUT	Bool	FB-Funktion läuft (immer False, da PDO-Datum)
Error	OUT	Bool	FB-Fehler (immer False, da PDO-Datum)
Position	OUT	REAL	Achsposition in Nutzereinheiten, z.B. „mm“

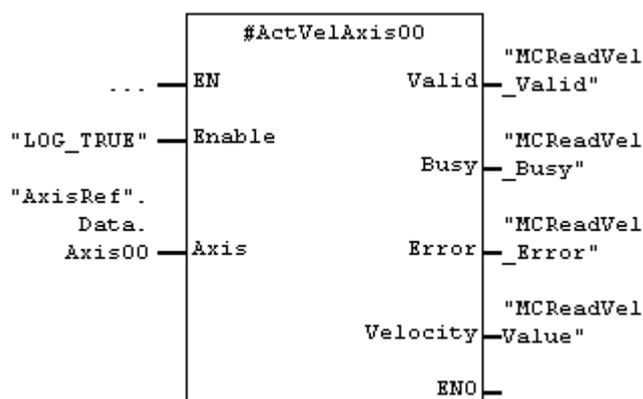


Im Stat-Bereich eines Container-FB's instanziiertes MC_ReadActualPosition_E2 mit dem Instanznamen ActPosAxis00.

3.5 MC_ReadActualVelocity_E2 (FB)

Der MC_ReadActualVelocity_E2 stellt die Istgeschwindigkeit der Achse bereit.

Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	Fehlercode lesen
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Valid	OUT	Bool	FB-Daten sind gültig (sobald Enable = True)
Busy	OUT	Bool	FB-Funktion läuft (immer False, da PDO-Datum)
Error	OUT	Bool	FB-Fehler (immer False, da PDO-Datum)
Velocity	OUT	REAL	Achsgeschwindigkeit in Nutzereinheiten, z.B. „mm/s“



Im Stat-Bereich eines Container-FB's instanziiertes MC_ReadActualVelocity_E2 mit dem Instanznamen ActVelAxis00.

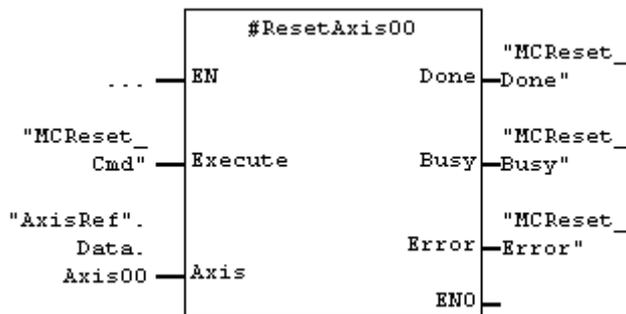
3.6 MC_Reset_E2 (FB)

Mit dem Baustein MC_Reset_E2 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 quitiert Achse
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	FB-Funktion fertig und Achse ohne Fehler
Busy	OUT	Bool	FB-Funktion läuft
Error	OUT	Bool	Achse mit Fehler



Im Stat-Bereich instanzierter MC_Reset_E2 mit dem Instanznamen ResetAxis00.

3.7 MC_Power_E2 (FB)

Mit dem Baustein MC_Power_E2 wird die Achse bestromt (Motor hat Drehmoment bzw. Kraft) oder entstromt (stromlos).



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

Name	Variablenbereich	Typ	Funktion
Enable	IN	Bool	0-1-Flanke bestromt Achse, 1-0-Flanke führt Schnellstopp mit anschließendem Stromlosschalten aus
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Status	OUT	Bool	1 bestromt 0 stromlos
Busy	OUT	Bool	FB-Funktion läuft (Bestromen gerade aktiv)
Error	OUT	Bool	Achse mit Fehler

3.8 Funktionsbausteine im Profile Position Mode

Die Bausteine MC_Stop_E2, MC_MoveAbsolute_E2, MC_MoveRelative_E2 werden im Profile Position Mode ausgeführt. Neben den an den FB-Parametern verfügbaren Vorgaben sind folgende CANopen-Parameter über CANopen-SDO oder einfacher über das EPOS-Studio zu beschreiben.

EPOS-Parameter	Objekt-Index bzw. Subindex	Einstellung EPOS-Studio bzw. SDO-Transfer erforderlich	MC_Stop	MC_MoveAbsolute MC_MoveRelative
Position Window (Positionierfenster für DONE-Meldung)	0x6067 / 0x00 [Inkmente]	Ja		

EPOS-Parameter	Objekt-Index bzw. Subindex	Einstellung EPOS-Studio bzw. SDO-Transfer erforderlich	MC_Stop	MC_MoveAbsolute MC_MoveRelative
Position Window Time (Zeit im Positionierfenster für DONE-Meldung)	0x6067 / 0x00 [ms]	Ja		
Software Position Limit (Software-Endgrenzen)	0x607D / 0x01 Minimal 0x607D / 0x02 Maximal [Inkremente] Die Software-Positionsüberwachung kann mit -2147483648 bzw. +2147483647 deaktiviert werden.	Ja		
Maximal Profile Velocity (Maximale Geschwindigkeit)	0x607F / 0x00 [U/min]	Ja		
QuickStop Deceleration (Schnellstopprampe)	0x6085 / 0x00 [U/min/s]		Ja (Decel) [Units/s ²]	
Max Acceleration (Maximale Beschleunigung/Verzögerung)	0x60C5 / 0x00 [U/min/s]	Ja		
Target Position (Zielposition oder Distanz von der aktuellen Sollposition)	0x607A / 0x00 [Inkremente]			Ja (Position bzw. Distance) [Units]
Profile Velocity (Sollgeschwindigkeit)	0x6081 / 0x00 [U/min]			Ja (Velocity) [Units/s]
Profile Acceleration (Sollbeschleunigung)	0x6083 / 0x00 [U/min/s]			Ja (Accel) [Units/s ²]
Profile Deceleration (Sollverzögerung)	0x6084 / 0x00 [U/min/s]			Ja (Decel) [Units/s ²]
Motion Profile Type (Geschwindigkeitsprofil)	0x6086 / 0x00	Ja 0=lineare Rampen 1=sin ² -Rampen		

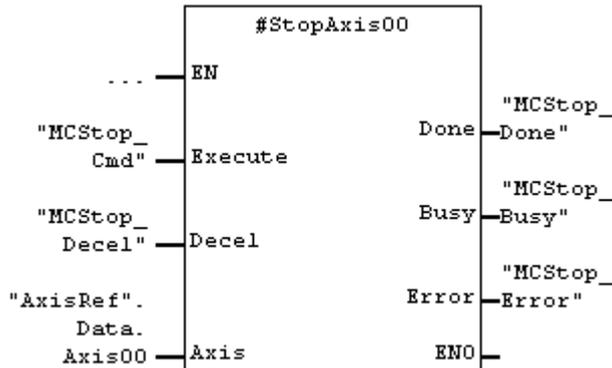
3.8.1 MC_Stop_E2 (FB)

Mit dem Baustein MC_Stop_E2 wird die Achse gestoppt. Ein Stoppen ist nur bei bestromter Achse durchführbar.

i	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. Es wird die QuickStopp-Funktion der Achse benutzt! Es ist nur eine Instanz dieses FB's sinnvoll und erlaubt.
----------	---

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	1 Stoppt Achse 0 Bewegungsfreigabe Achse
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Decel	IN	Dint	Stopprampe [Units/s ²]

Name	Variablenbereich	Typ	Funktion
Done	OUT	Bool	Achse gestoppt
Busy	OUT	Bool	FB-Funktion läuft
Error	OUT	Bool	Achse mit Fehler



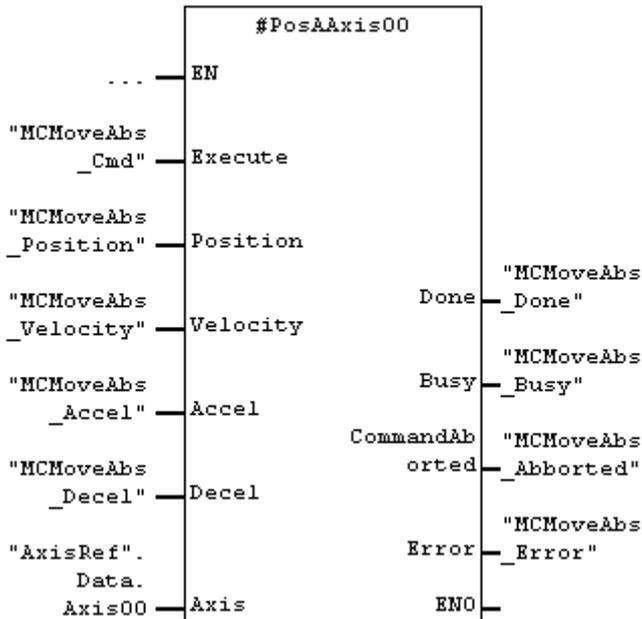
Im Stat-Bereich instanzierter MC_Stop_E2 mit dem Instanznamen StopAxis00.

3.8.2 MC_MoveAbsolute_E2 (FB)

Der Baustein MC_MoveAbsolute_E2 wird für absolute Positionierungen verwendet. Bezugspunkt der absoluten Position ist der durch eine Referenzfahrt 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 Nutzeinheiten, z.B. „mm“
Velocity	IN	Real	Positioniergeschwindigkeit in Nutzeinheiten, z.B. „mm/s“
Accel	IN	Dint	Beschleunigung in Nutzeinheiten, z.B. „mm/s ² “
Decel	IN	Dint	Verzögerung in Nutzeinheiten, z.B. „mm/s ² “
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	Achse hat Zielposition erreicht
Busy	OUT	Bool	FB-Funktion läuft
CommandAborted	OUT	Bool	FB-Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler

Im Stat-Bereich instanziiertes MC_MoveAbsolute_E2 mit dem Instanznamen PosAAxis00.



3.8.3 MC_MoveRelative_E2 (FB)

Der Baustein MC_MoveRelative_E2 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 Nutzeinheiten, z.B. „mm“
Velocity	IN	Real	Positioniergeschwindigkeit in Nutzeinheiten, z.B. „mm/s“
Accel	IN	Dint	Beschleunigung in Nutzeinheiten, z.B. „mm/s ² “
Decel	IN	Dint	Verzögerung in Nutzeinheiten, z.B. „mm/s ² “
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	Achse hat Zielposition erreicht (Distanz abgefahren)
Busy	OUT	Bool	FB-Funktion läuft
CommandAborted	OUT	Bool	FB-Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler

3.9 Funktionsbausteine im Profile Velocity Mode

Die Bausteine MC_Stop_E2, MC_MoveVelocity_E2 und MC_Jog_E2 werden im Profile Velocity Mode ausgeführt. Neben den an den FB-Parametern verfügbaren Vorgaben sind folgende CANopen-Parameter über CANopen-SDO oder einfacher über das EPOS-Studio zu beschreiben.

EPOS-Parameter	Objekt-Index bzw. Subindex	Einstellung EPOS-Studio bzw. SDO-Transfer erforderlich	MC_Stop	MC_MoveVelocity MC_Jog
Velocity Window (Geschwindigkeitsfenst	0x606D / 0x00 [U/min]	Ja		

EPOS-Parameter	Objekt-Index bzw. Subindex	Einstellung EPOS-Studio bzw. SDO-Transfer erforderlich	MC_Stop	MC_MoveVelocity MC_Jog
er für DONE-Meldung)				
Position Window Time (Zeit im Geschwindigkeitsfenster für DONE-Meldung)	0x606E / 0x00 [ms]	Ja		
Maximal Profile Velocity (Maximale Geschwindigkeit)	0x607F / 0x00 [U/min]	Ja		
QuickStop Deceleration (Schnellstopprampe)	0x6085 / 0x00 [U/min/s]		Ja (Decel) [Units/s ²]	
Max Acceleration (Maximale Beschleunigung/Verzögerung)	0x60C5 / 0x00 [U/min/s]	Ja		
Target Velocity (Zielgeschwindigkeit)	0x60FF / 0x00 [U/min]			Ja (Geschwindigkeit) [Units/s]
Profile Acceleration (Sollbeschleunigung)	0x6083 / 0x00 [U/min/s]			Ja (Accel) [Units/s ²]
Profile Deceleration (Sollverzögerung)	0x6084 / 0x00 [U/min/s]			Ja (Decel) [Units/s ²]
Motion Profile Type (Geschwindigkeitsprofil)	0x6086 / 0x00	Ja 0=lineare Rampen 1=sin ² -Rampen		

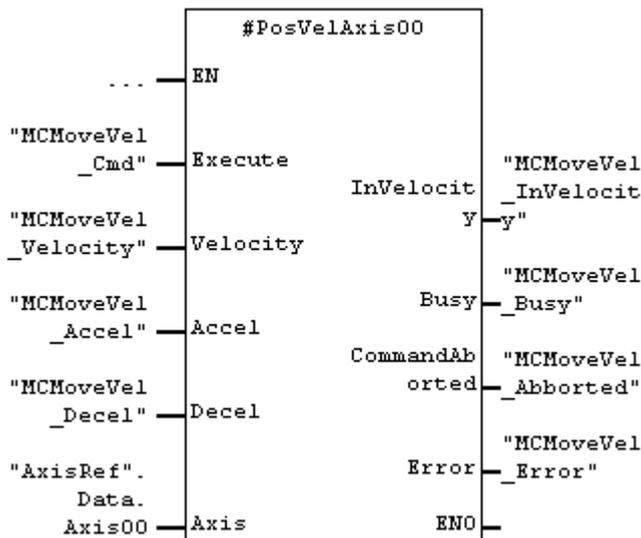
3.9.1 MC_MoveVelocity_E2 (FB)

Der Baustein MC_MoveVelocity_E2 wird für Endlosbewegungen verwendet.

i	<p>Die Bewegung wird mit MC_Stop_E2 gestoppt. Wird am MC_MoveVelocity die Geschwindigkeit 0.0 Units/s vorgegeben, „fährt“ der Antrieb mit der Sollgeschwindigkeit 0.0 Units/s, d.h. die interne Sollwertvorgabe ist weiterhin aktiv.</p> <p>Soll sich die Geschwindigkeit fliegend ändern, kann die Geschwindigkeit über eine weitere Instanz von MC_MoveVelocity geändert werden oder das Execute an der „einen“ Instanz kann nachgetriggert werden (erneute 0-1-Flanke).</p>
----------	--

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	0-1-Flanke startet die Bewegung
Velocity	IN	Real	Geschwindigkeit in Nutzereinheiten, z.B. „mm/s“
Accel	IN	Dint	Beschleunigung in Nutzereinheiten, z.B. „mm/s ² “
Decel	IN	Dint	Verzögerung in Nutzereinheiten, z.B. „mm/s ² “
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
InVelocity	OUT	Bool	Achse hat Zielgeschwindigkeit erreicht
Busy	OUT	Bool	FB-Funktion läuft
CommandAborted	OUT	Bool	FB-Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.

Name	Variablenbereich	Typ	Funktion
Error	OUT	Bool	Achse mit Fehler



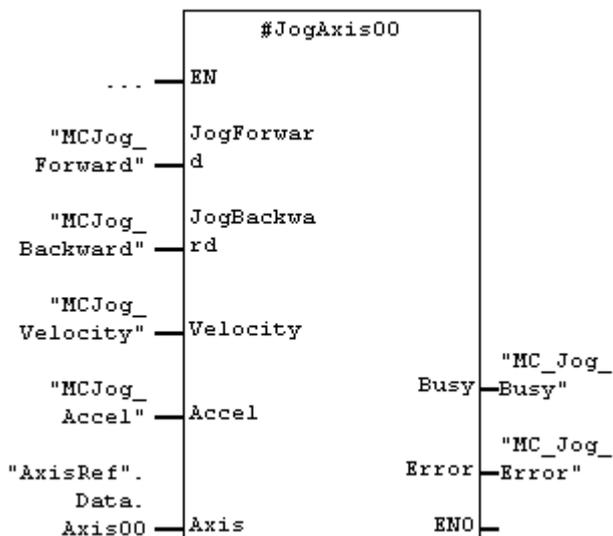
Im Stat-Bereich instanziiertes MC_MoveVelocity_E2 mit dem Instanznamen PosVelAxis00.

3.10 MC_Jog_E2 (FB)

Der Baustein MC_Jog_E2 wird verwendet, um die Achse „manuell“ zu bewegen (auch tippen genannt). Beide Richtungen sind möglich. Der FB funktioniert wie der MC_MoveVelocity, nur das richtungsabhängig gefahren werden kann.

	<p>Wird der Jog-Betrieb beendet (JogForward und JogBackward beide False), wird intern ein QuickStop ausgelöst, wobei jedoch im Unterschied zum „normalen“ Stoppen über MC_Stop_E2 die Accel-Rampe vom FB-Eingang verwendet wird, gegebenenfalls also „weicher“ gestoppt wird.</p> <p>Man beachte auch, dass der Accel-Wert gleichzeitig für die Beschleunigung und Verzögerung gilt.</p>
--	--

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
Velocity	IN	Real	Handgeschwindigkeit in Nutzeinheiten, z.B. „mm/s“
Accel	IN	Dint	Beschleunigung/Verzögerung in Nutzeinheiten, z.B. „mm/s ² “
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Busy	OUT	Bool	FB-Funktion läuft
Error	OUT	Bool	Achse mit Fehler



Im Stat-Bereich instanzierter MC_Jog_E2 mit dem Instanznamen JogAxis00.

3.11 Funktionsbausteine im Homing Mode

Die Bausteine MC_Home_E2 wird im Homing Mode ausgeführt. Neben den an den FB-Parametern verfügbaren Vorgaben sind folgende CANopen-Paramter über CANopen-SDO oder einfacher über das EPOS-Studio zu beschreiben.

EPOS-Parameter	Objekt-Index bzw. Subindex	Einstellung EPOS-Studio bzw. SDO-Transfer erforderlich	MC_Home
Homing Method (Methode des Referenzierens über Endschalter, mechanische Anschläge, Position setzen usw.)	0x6098 / 0x00	Ja	
Homing Speeds (Suchgeschwindigkeit beim Referenzieren)	0x6099 / 0x00 [U/min]	Ja	
Homing Acceleration (Rampen beim Referenzieren)	0x609A / 0x00 [U/min/s]	Ja	
Home Offset (Nach dem eigentlichen Referenzieren wird diese Strecke noch abgefahren, dann der Wert des Parameters Home position als neue Istposition gesetzt)	0x607C / 0x00 [Inkrement]	Ja	
Current Threshold for homing modes -1 bis -4 (Wenn beim Referenzieren gegen den Anschlag gefahren werden soll, Stromgrenze)	0x2080 [mA]	Ja	
Home position (Setzposition nach erfolgreichem Homing und Abfahren der Strecke im Home)	0x2081 / 0x00 [Inkrement]		Ja (Setzposition nach erfolgter Homing-Prozedur) [Units]

EPOS-Parameter	Objekt-Index bzw. Subindex	Einstellung EPOS-Studio bzw. SDO-Transfer erforderlich	MC_Home
Offset)			
Motion Profile Type (Geschwindigkeitsprofil)	0x6086 / 0x00	Ja 0=lineare Rampen 1=sin ² -Rampen	

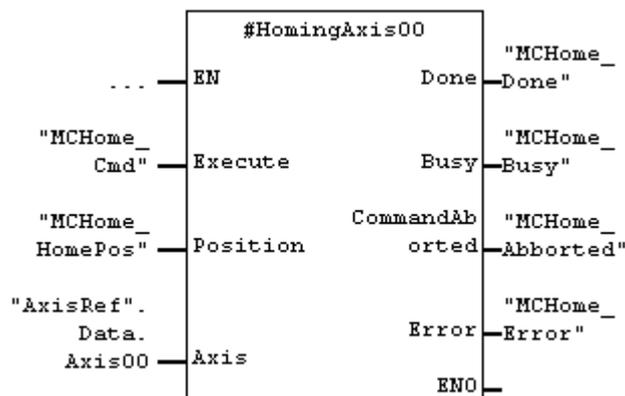
3.11.1 MC_Home_E2 (FB)

Der Baustein MC_Home_E2 wird verwendet, um den mathematischen Bezugspunkt für folgende Positionierungen der Achse zu definieren. Die Referenzierarten (Modi) sind der EPOS2-Beschreibung zu entnehmen.

Im Laboraufbau wurden die Modi „35“ und „-3“ getestet.

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

Name	Variablenbereich	Typ	Funktion
Execute	IN	Bool	0-1-Flanke startet die Referenzfahrt
Position	IN	REAL	Setzposition (neue Sollposition) in Nutzereinheiten, z.B. „mm“, der Wert des Parameter 0x2081/0x00 wird beschrieben
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
Done	OUT	Bool	Referenzfahrt (inkl. Abfahren der Distanz im Parameter 0x607C/0x00 Home Offset) beendet, Setzposition gesetzt, der mathematische Nullpunkt und gegebenenfalls die Software-Endgrenzen sind gültig
Busy	OUT	Bool	FB-Funktion läuft
CommandAborted	OUT	Bool	FB-Kommando wurde abgebrochen durch neue Positionierung, Handfahren, Stromlosschalten, Stoppen etc.
Error	OUT	Bool	Achse mit Fehler oder Referenzierung (Homing) mit Fehler beendet



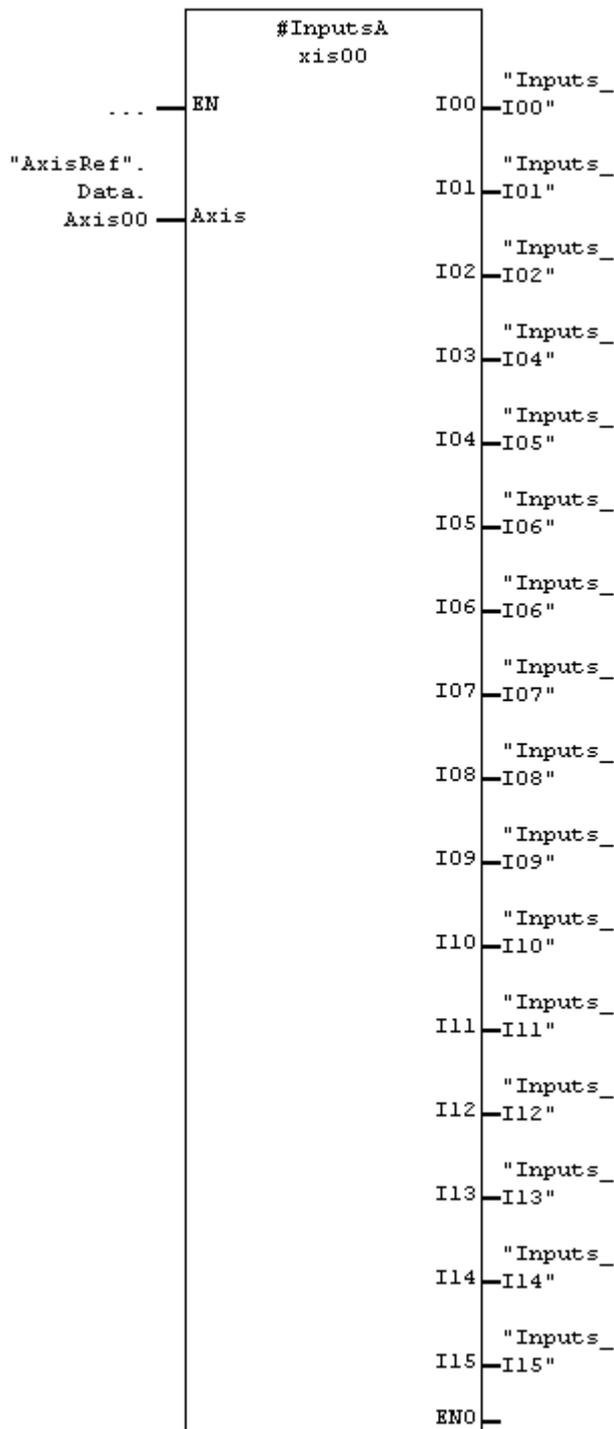
Im Stat-Bereich instanzierter MC_Home_E2 mit dem Instanznamen HomingAxis00.

3.12 Spezial-FB's

3.13 E2_Input (FB)

Der Baustein E2_Input ist eine Spezialversion für die EPOS2-Achse, man beachte unterschiedliche Versionen der EPOS2-Familie.

Name	Variablenbereich	Typ	Funktion
Axis	IN_OUT	AxisRefE2Type (UDT)	Achsreferenz (Achsverweis)
I00 bis I15	OUT	Bool	Eingänge 0 bis 15 als Status



Im Stat-Bereich instanzierter E2_Input mit dem Instanznamen InAxis00.

3.14 InDataE2Type (UDT)

Dieser Datentyp ist bei Verwendung in einem Datenbaustein mit einem Namen, z.B. Axis00 zu instantiiieren. Pro Achse wird genau 1 Instanz benötigt. Die Instanz-Daten entsprechen T-PDO-Daten der EPOS2-Achse.



Am besten werden alle Instanzen (der verschiedenen Achsen) von InDataE2Type in einem separaten DB (z.B. „TPDODATA“) angelegt.

```
wStatusWord      : WORD;          // TPD01, async, 0x6041 + 0x00, Status word
wDigInWord       : WORD;          // TPD01, async, 0x2071 + 0x01, Input state
byActModeOfOp    : BYTE;         // TPD01, async, 0x6061 + 0x00, Actual mode of operation
diActPosition    : DINT;         // TPD02, async, 0x6064 + 0x00, Actual position [pq Enc.]
diActVelocity    : DINT;         // TPD02, async, 0x606C + 0x00, Actual velocity [rev/min]
```

3.15 OutDataE2Type (UDT)

Dieser Datentyp ist bei Verwendung in einem Datenbaustein mit einem Namen, z.B. Axis00 zu instantiiieren. Pro Achse wird genau 1 Instanz benötigt. Die Instanz-Daten entsprechen R-PDO-Daten der EPOS2-Achse.



Am besten werden alle Instanzen (der verschiedenen Achsen) von OutDataE2Type in einem separaten DB (z.B. „RPDODATA“) angelegt.

```
wControlWord     : WORD;          // RPD01, async, 0x6040 + 0x00, Control word
diTarget         : DINT;          // RPD01, async, 0x607a + 0x00, Target [pq Enc.]
byModeOfOp       : BYTE;         // RPD01, async, 0x6060 + 0x00, Mode of operation
diProfVelocity   : DINT;         // RPD02, async, 0x6081 + 0x00, Profile velocity [rev/min]
diHomePosition   : DINT;         // RPD02, async, 0x2081 + 0x00, Home position [pq Enc.]
diProfAccel      : DINT;         // RPD03, async, 0x6083 + 0x00, Prof. acceleration [rev/min/s]
diProfDecel      : DINT;         // RPD03, async, 0x6084 + 0x00, Prof. deceleration [rev/min/s]
diQuickStopDecel : DINT;         // RPD04, async, 0x6085 + 0x00, Prof. deceleration [rev/min/s]
diSpeed          : DINT;         // RPD04, async, 0x60FF + 0x00, Prof. deceleration [rev/min]
```

3.16 AxisRefE2Type

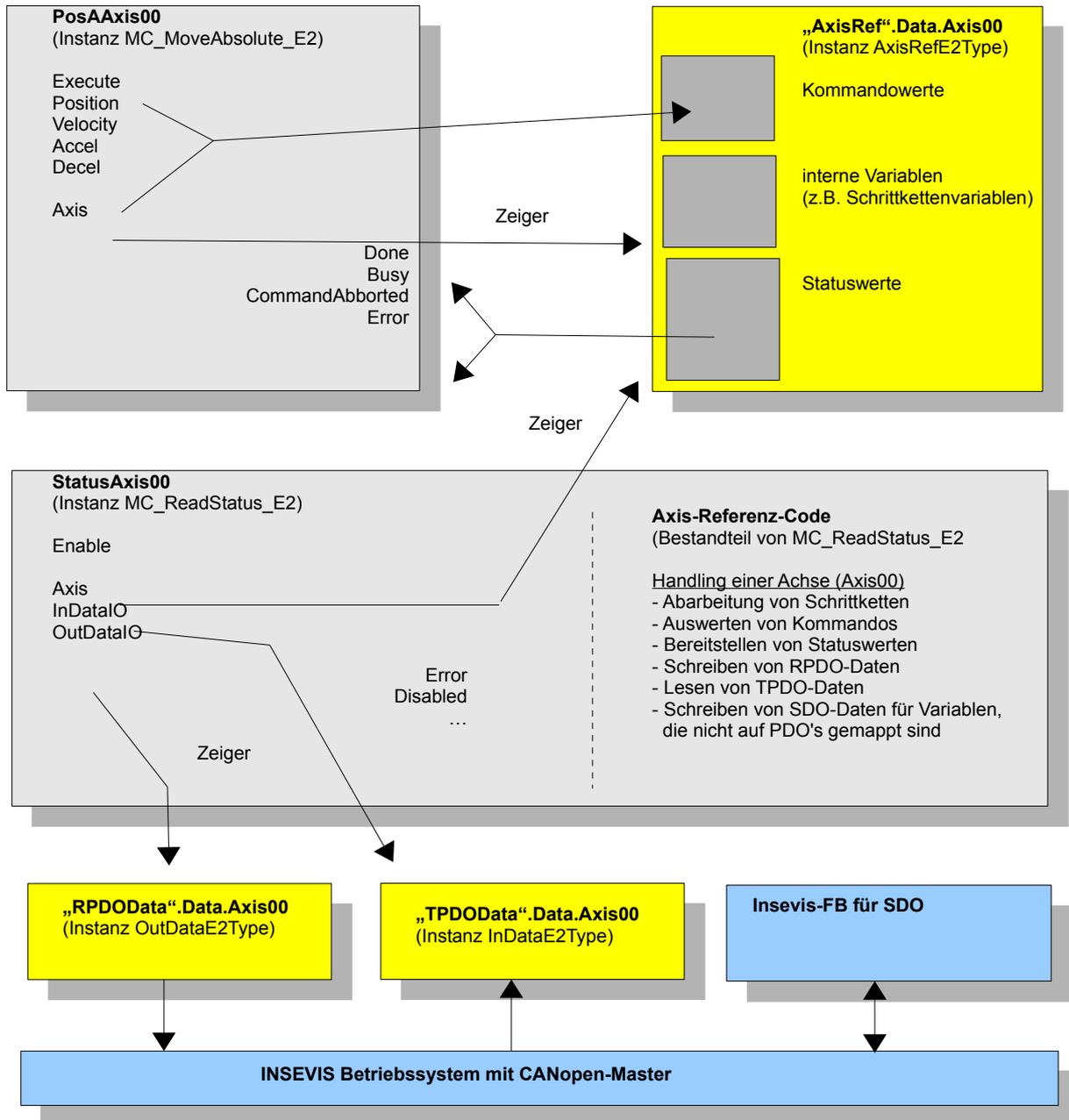
Dieser Datentyp wird intern zur Referenzierung der Achse benötigt. Alle Instanzen von MC-Bausteinen bestimmen damit die verbundene Achse. Da die instantiierten Variablen als IN_OUT übergeben werden, ist der Kopieraufwand gering. Der Baustein MC_ReadStatus_E2 verwendet die mit der Achsreferenz übergebenen Werte zur Abarbeitung der Schrittketten.



Am besten werden alle Instanzen (der verschiedenen Achsen) von AxisrefE2Type 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.



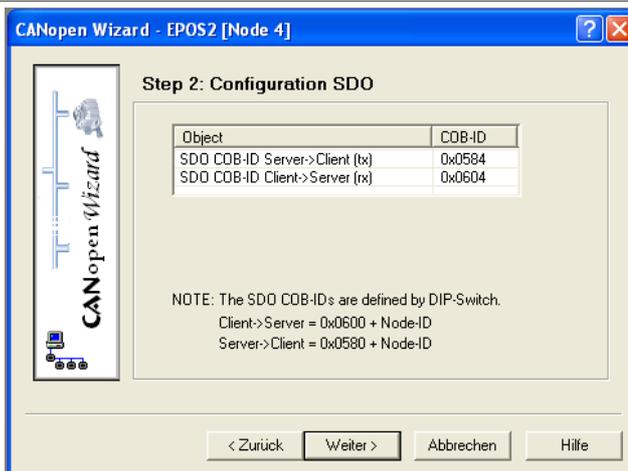
5 CANopen-Konfiguration mit dem EPOS-Studio

Grundsätzlich kann hier nicht auf die allgemeine Konfiguration eines EPOS2-Servoantriebs eingegangen werden. Wichtig für den CANopen-Teil ist hier lediglich, dass über den CANopen-Wizard die eigentliche Konfiguration geprüft (nach Konfiguration mit der ConfigStage und Anlauf der INSEVIS-SPS) werden sollte, da die Einstellungen der CANopen-Slaves durch die INSEVIS-SPS selbst erfolgen.

Zudem ist die CANopen-Node-ID am entsprechenden Jumper (CAN-ID) einzustellen. Die CAN-Bitrate (0x2001) steht per default auf dem Wert „9“ (Automatische Erkennung), diesen Wert sollte man der Einfachheit halber unverändert lassen.



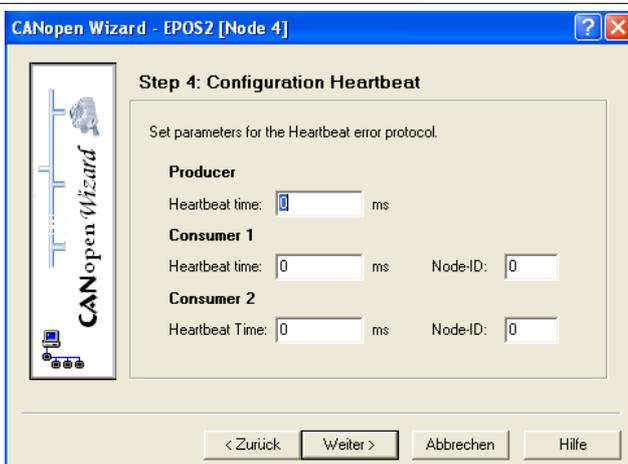
Verwenden Sie das „Object Dictionary“, um aktuelle Werte zu prüfen bzw. zu verändern.



Überprüfen der COB-ID's für die SDO's, i.d.R. Keine Veränderungen notwendig.



Die PDO-Konfiguration kann komplett übersprungen werden.



Eine Heartbeat-Überwachung wird nicht verwendet.

	<p>Nodeguarding wird durch die INSEVIS-SPS konfiguriert und aktiviert.</p>
--	--

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.

<p>Allgemein</p> <p>Node-ID: <input type="text" value="4"/></p> <p>Device monitoring: <input type="radio"/> Aus <input type="radio"/> Heartbeat <input checked="" type="radio"/> Nodeguard</p> <p>Guarding time (ms): <input type="text" value="100"/></p> <p>Lifetime factor: <input type="text" value="3"/></p> <p>NMT control: <input checked="" type="checkbox"/></p> <p>NMT download: <input checked="" type="checkbox"/></p>	<p><u>Festlegen der Node-ID und des Guardings</u> EPOS2 unterstützt Nodeguarding CANopen-Einstellungen (wie COB-ID's) zum EPOS2 laden</p>
<p>Tx PDO</p> <p><input checked="" type="checkbox"/> TxPDO1 <input type="text" value="TxPDO1"/></p> <p><input checked="" type="checkbox"/> TxPDO2 <input type="text" value="TxPDO2"/></p> <p><input type="checkbox"/> TxPDO3 <input type="text" value="TxPDO3"/></p> <p><input type="checkbox"/> TxPDO4 <input type="text" value="TxPDO4"/></p>	<p><u>TPDO (EPOS2→ CANopen-Master)</u> Es werden für die Empfangsrichtung 2 T-PDO's benötigt. Download Kommunikationsparameter und des Mappings sind zu aktivieren.</p> <p>Übertragungsverhalten TPDO1 Typ: 254 keine Sperrzeit</p> <p>Übertragungsverhalten TPDO2 Typ: 254 Sperrzeit von z.B. 100ms definieren!</p>

<div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>Rx PDO</p> <div style="display: flex; flex-direction: column; gap: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <input checked="" type="checkbox"/> RxPDO1 <div style="border: 1px solid #ccc; padding: 2px 10px; border-radius: 3px;">RxPDO1</div> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <input checked="" type="checkbox"/> RxPDO2 <div style="border: 1px solid #ccc; padding: 2px 10px; border-radius: 3px;">RxPDO2</div> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <input checked="" type="checkbox"/> RxPDO3 <div style="border: 1px solid #ccc; padding: 2px 10px; border-radius: 3px;">RxPDO3</div> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <input checked="" type="checkbox"/> RxPDO4 <div style="border: 1px solid #ccc; padding: 2px 10px; border-radius: 3px;">RxPDO4</div> </div> </div> </div>	<p><u>RPDO (CANopen-Master → EPOS2)</u></p> <p>Es werden für die Senderichtung 4 R-PDO's benötigt. Download Kommunikationsparameter und des Mappings sind zu aktivieren.</p> <p>Übertragungsverhalten RPDO1 Typ: 254 keine Sperrzeit</p> <p>Übertragungsverhalten RPDO2 Typ: 254 keine Sperrzeit</p> <p>Übertragungsverhalten RPDO3 Typ: 254 keine Sperrzeit</p> <p>Übertragungsverhalten RPDO4 Typ: 254 keine Sperrzeit</p>
<div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>SDO</p> <div style="text-align: right; margin-top: 10px;"> <div style="border: 1px solid #ccc; padding: 2px 10px; border-radius: 3px;">SDOs</div> </div> </div>	<p><u>Zusätzliche Konfiguration über SDO</u></p> <p>Es werden keine zusätzlichen SDO's benötigt. Gegebenenfalls kann man hier CANopen-Daten übertragen, die man ansonsten über das EPOS-Studio übertragen müsste.</p>

6.1 Mapping T-PDO1

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „InDataE2Type“: **0** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6041	0x00	16 Bit/Word	Statuswort DS402
2	0x2071	0x01	16 Bit /Word	Eingangswort der DE
3	0x6061	0x00	16 Bit/Word	Aktuelle Betriebsart DS402

6.2 Mapping T-PDO2

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „InDataE2Type“: **6** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6064	0x00	32 Bit/DWord	Aktuelle Position [Post quadrature encoder]
2	0x606C	0x00	32 Bit/DWord	Aktuelle Geschwindigkeit [U/min]

6.3 Mapping R-PDO1

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „OutDataE2Type“: **0** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6040	0x00	16 Bit/Word	Steuerwort DS402
2	0x607A	0x00	32Bit/DWord	Target [Post quadrature encoder]
3	0x6060	0x00	16 Bit/Word	Betriebsart DS402

6.4 Mapping R-PDO2

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „OutDataE2Type“: **8** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6081	0x00	32 Bit/DWord	Profilgeschwindigkeit [U/min]
2	0x2081	0x00	32 Bit/DWord	Home-Position [Post quadrature encoder]

6.5 Mapping R-PDO3

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „OutDataE2Type“: **16** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6083	0	32 Bit/DWord	Profilbeschleunigung [U/min/s]
2	0x6084	0	32 Bit/DWord	Profilverzögerung [U/min/s]

6.6 Mapping R-PDO4

Offset im Datenbereich (z.B. Datenbaustein) einer Instanz vom Typ „OutDataE2Type“: **24** (Byte-Offset)

Nummer	Index	Subindex	Größe	Erklärung
1	0x6085	0	32 Bit/DWord	Schnellstopprampe [U/min/s]
2	0x60FF	0	32 Bit/DWord	Geschwindigkeit im Profile Velocity Mode [U/min]

6.7 Zusätzliche SDO-Übertragung nach PDO-Mapping

Nummer	Index	Subindex	Größe	Wert	Erklärung

7 S7-Beispiel-Programm

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