



Funktionsbausteine für PSx-3__ mit PROFINET-Schnittstelle

halstrup-walcher GmbH

Stegener Straße 10
D-79199 Kirchzarten

Phone: +49 (0) 76 61/39 63-0
Fax: +49 (0) 76 61/39 63-99

E-Mail: info@halstrup-walcher.de
Internet: www.halstrup-walcher.de

Inhaltsverzeichnis

1	Sicherheitshinweise	4
1.1	Bestimmungsgemäße Verwendung	4
1.2	Symbolerklärung	4
2	Datenstruktur DRIVE_DATA	4
3	Datenbaustein „Antriebsdaten“	6
4	Fehlerbeschreibung (Error ID).....	7
5	Beschreibung und Anwendung der Funktionsbausteine	8
5.1	Öffnen der Bibliothek	8
5.2	Kopieren von Bausteinen in das Anwenderprojekt	9
5.3	Erzeugen von Instanzdatenbausteinen	10
5.4	Anlegen von globalen Daten	10
5.5	Gemeinsamkeiten aller Funktionsbausteine.....	10
5.6	Verriegelungen zwischen den Funktionsbausteinen.....	11
5.7	Beispiel	11
5.8	MC_Communication_S7-1200_1500 und _S7-300 (FC101)	12
5.9	MC_Move (FB110).....	14
5.10	MC_Error (FB111)	18
5.11	MC_Error_ID (FC100).....	21
5.12	MC_ReadParameter (FB112)	24
5.13	MC_WriteParameter (FB113).....	26
5.14	MC_Parametrization (FB114).....	28
5.15	MC_PosParametrization (FB115)	38

Bedeutung der Betriebsanleitung

Diese Betriebsanleitung erläutert die Funktionsbausteine für die Positioniersysteme PSx-3__-PN (mit PROFINET-Schnittstelle).

Von diesen Geräten können für Personen und Sachwerte Gefahren durch nicht bestimmungsgemäße Verwendung und durch Fehlbedienung ausgehen. Deshalb muss jede Person, die mit der Handhabung der Geräte betraut ist, eingewiesen sein und die Gefahren kennen. Die Betriebsanleitung und insbesondere die darin gegebenen Sicherheitshinweise müssen sorgfältig beachtet werden. **Wenden Sie sich unbedingt an den Hersteller, wenn Sie Teile davon nicht verstehen.**

Der Hersteller behält sich das Recht vor, diese Funktionsbausteine weiterzuentwickeln, ohne dies in jedem Einzelfall zu dokumentieren. Über die Aktualität dieser Betriebsanleitung gibt Ihnen Ihr Hersteller gerne Auskunft.

© 2017

Das Urheberrecht an dieser Betriebsanleitung verbleibt beim Hersteller. Sie darf weder ganz noch in Teilen vervielfältigt oder Dritten zugänglich gemacht werden.

1 Sicherheitshinweise

1.1 Bestimmungsgemäße Verwendung

Die Positioniersysteme PSx-3__-PN eignen sich besonders zur automatischen Einstellung von Werkzeugen, Anschlägen oder Spindeln bei Holzverarbeitungsmaschinen, Verpackungsmaschinen, Druckmaschinen, Abfüllanlagen und bei Sondermaschinen.

Die PSx-3__-PN sind nicht als eigenständige Geräte zu betreiben, sondern dienen ausschließlich zum Anbau an eine Maschine.

1.2 Symbolerklärung

In dieser Betriebsanleitung wird mit folgenden Hervorhebungen auf die darauf folgend beschriebenen Gefahren bei der Handhabung der Anlage hingewiesen:



WARNUNG!

Sie werden auf eine Gefährdung hingewiesen, die zu Körperverletzungen bis hin zum Tod führen kann, wenn Sie die gegebenen Anweisungen missachten.



ACHTUNG!

Sie werden auf eine Gefährdung hingewiesen, die zu einem erheblichen Sachschaden führen kann, wenn Sie die gegebenen Anweisungen missachten.



INFORMATION!

Sie erhalten wichtige Informationen zum sachgemäßen Betrieb.

2 Datenstruktur DRIVE_DATA

Für jeden Antrieb gibt es eine Datenstruktur, in der einige Daten eines Antriebs abgelegt sind. Für jeden Antrieb wird eine globale Instanz dieser Struktur benötigt. Diese Instanz muss jedem FB übergeben werden, der auf den betr. Antrieb wirkt. Hiermit soll z.B. sichergestellt werden, dass nicht gleichzeitig zwei Zugriffe aus unterschiedlichen FBs auf den Parameterkanal durchgeführt werden können. Des Weiteren müssen in dieser Datenstruktur die Adressen zu den Ein-/Ausgangsdaten des jeweiligen Antriebs hinterlegt werden.

Parametername	Datentyp	geschrieben von	Beschreibung
PdAddressIn	INT (S7-1200 + S7-1500) DWORD (S7-300)	Benutzer	Adresse Prozessdaten Device → Controller
PdAddressOut	INT (S7-1200 + S7-1500) DWORD (S7-300)	Benutzer	Adresse Prozessdaten Controller → Device
AxisName	STRING[16]	Benutzer (optional)	Name der Achse
AxisDescription	STRING[32]	Benutzer (optional)	Beschreibung (z.B. Funktion, Aufgabe dieser Achse)
State	DINT	Funktionsbausteine	Actual state
Communication_error	BOOL	Funktionsbausteine	Kommunikationsfehler zum IO-Device
Private	STRUCT	Funktionsbausteine	Datenstruktur zur internen Verwendung

Die folgende Darstellung zeigt, wie im TIA-Portal die vergebenen Adressen der Prozessdaten überprüft werden können:

The screenshot displays the Siemens TIA Portal interface. On the left, the 'Geräte & Netze' (Devices & Networks) tree shows a project named 'Halstrup_PSx3x3PN_571200_1500_2019'. The main workspace shows a network diagram with two devices: 'PLC_4 CPU 1214C' and 'PSE_3 PSE mit 2 Busst...'. They are connected via a 'PN/IE_4' link. Below the workspace, the 'PN/IE_4 [Industrial Ethernet]' properties window is open, showing the 'Allgemein' (General) tab. The 'Adressübersicht' (Address Overview) section is active, displaying a table of addresses for the 'PLC_4 PROFINET IO-System'.

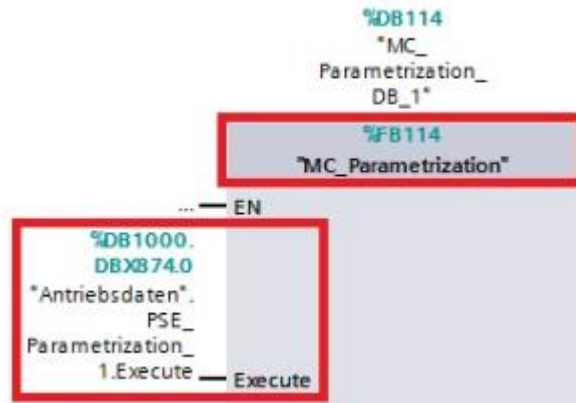
Typ	Adr. von	Adr. bis	Größe	Modul	Baugr...	Steck...	Gerätename	Gerät...	Master-IO-System	TPA
E	68	83	16 Bytes	16 Byte Eingang_1	0	1	PSE_3 [PSE with 2 connectors]	1	PROFINET-System [100]	-
A	64	77	14 Bytes	14 Byte Ausgang_1	0	2	PSE_3 [PSE with 2 connectors]	1	PROFINET-System [100]	-

(Adressen der Prozessdaten in Step 7 TIA)

3 Datenbaustein „Antriebsdaten“

Dieser Datenbaustein dient als Vorlage für die Anbindung an die Funktionsbausteine. Der Datenbaustein stellt die notwendigen Variablen bereit, um alle Ein- und Ausgänge aller zur Verfügung stehender Funktionsbausteine beschalten zu können. Jede Variable ist dabei doppelt vorhanden, so dass zwei Positioniersysteme angesteuert werden können.

Beispiel:



Falls der Datenbaustein in das Projekt übernommen wird, ist empfehlenswert, die nicht verwendeten Variablen zu löschen (damit nicht unnötig Speicher in der SPS belegt wird) und den Baustein auf die Anzahl der tatsächlich vorhandenen Antriebe anzupassen.

Der Datenbaustein benötigt standardmäßig die Datentypen „Parameter“ und „ParameterEnable“, diese müssen also aus der Bibliothek in das Projekt übernommen werden.

4 Fehlerbeschreibung (Error ID)

Nachfolgend die Fehlercodes, die von den Funktionsbausteinen ausgegeben werden:

ErrorID (hex)	Beschreibung
16xF000 (mask)	FB
16#1xxx	Error in MC_Move
16#2xxx	Error in MC_Error
16#3xxx	Error in MC_ReadParameter
16#4xxx	Error in MC_WriteParameter
16#5xxx	Error in MC_Parametrization
16#6xxx	Error in MC_PosParametrization
16#0F00 (mask)	Internal FB and PD errors
16#x1xx	Error in state machine or other FB internal error
16#x2xx	Invalid PD input address
16#x3xx	Invalid PD output address
16#x4xx	Error while reading PD
16#x5xx	Error while writing PD
16#x6xx	Unallowed input data change
16#00F0 (mask)	Parameter errors
16#xx1x	Parameter: communication timeout (1000 ms)
16#xx2x	Parameter: invalid parameter number
16#xx3x	Parameter: value is read only
16#xx4x	Parameter: lower or upper limit exceeded
16#xx5x	Parameter: faulty sub-index
16#xx6x	Parameter: not an array
16#xx7x	Parameter: incorrect data type
16#xx8x	Parameter: setting not allowed (resetting only)
16#xx9x	Parameter: request cannot be processed due to operating state
16#xxAx	Other error
16#000F (mask)	Drive errors
16#xxx1	Drag error
16#xxx2	Under- or overvoltage motor supply / STO-enabling inactive
16#xxx3	Positioning run aborted
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error or STO hardware error
16#xxx6	Block or overcurrent error
16#xxx7	Manual displacement
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run / failure voltage control
16#xxxA	Lower position limit exceeded
16#xxxB	Upper position limit exceeded

Die Fehler „Drive errors“ sind eine Abbildung der Fehlerbits im Statuswort des PSx.

Beispiele:

- Fahrauftrag (MC_Move) mit falschem Sollwert → ErrorID = 16#1008
- Parameter schreiben (MC_WriteParameter) mit ungültiger Parameternummer → ErrorID = 16#4020

5 Beschreibung und Anwendung der Funktionsbausteine

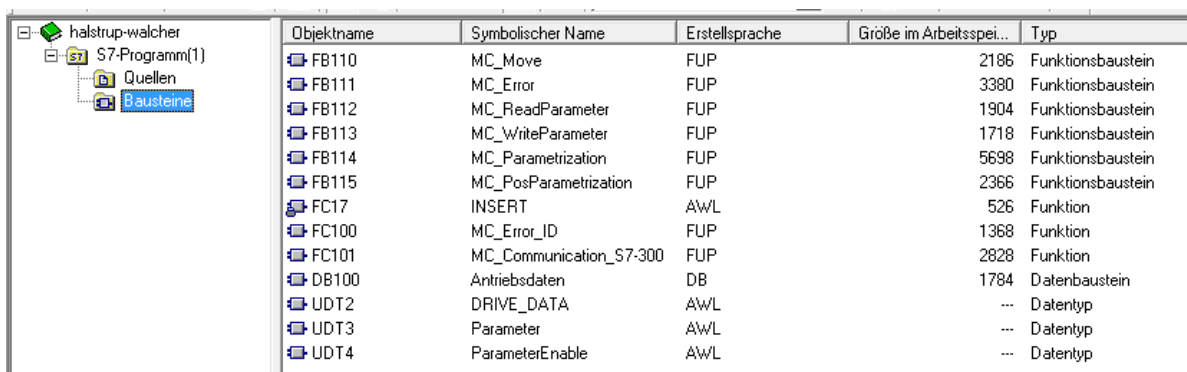
Zunächst müssen die Bausteine in ein eigenes Anwenderprojekt eingebunden werden. Dies geschieht durch Öffnen der gewünschten Bibliothek und Kopieren der gewünschten Funktionsbausteine.

Folgende Bibliotheken stehen zur Auswahl:

- „library_PSx3xxPN_Step7_xxxx“
→ für Step 7 Classic V5.5
- „library_PSx3xxPN_TIA13_300_xxxx“
→ für CPU S7-300 im TIA-Portal (V13 SP1)
- „library_PSx3xxPN_TIA14_300_xxxx“
→ für CPU S7-300 im TIA-Portal (V14)
- „library_PSx3xxPN_TIA16_300_xxxx“
→ für CPU S7-300 im TIA-Portal (V16)
- „library_PSx3xxPN_TIA13_1200-1500_xxxx“
→ für CPU S7-1200/1500 im TIA-Portal (V13 SP1)
- „library_PSx3xxPN_TIA14_1200-1500_xxxx“
→ für CPU S7-1200/1500 im TIA-Portal (V14)
- „library_PSx3xxPN_TIA16_1200-1500_xxxx“
→ für CPU S7-1200/1500 im TIA-Portal (V16)

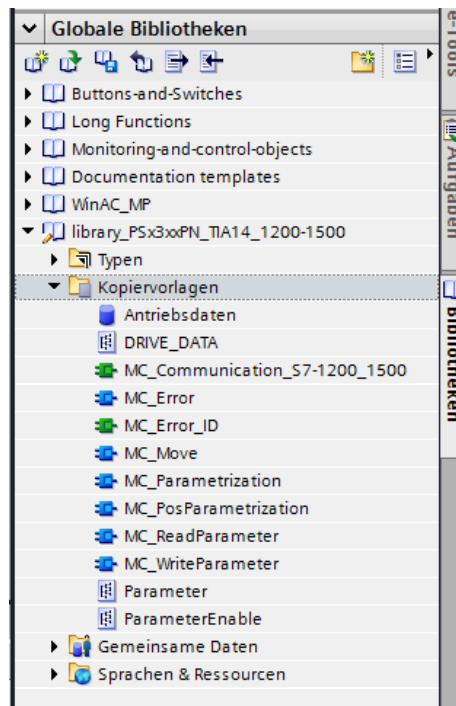
5.1 Öffnen der Bibliothek

Nach dem Öffnen der Bibliothek präsentieren sich die Funktionsbausteine folgendermaßen:



Objektname	Symbolischer Name	Erstelsprache	Größe im Arbeitsspei...	Typ
FB110	MC_Move	FUP	2186	Funktionsbaustein
FB111	MC_Error	FUP	3380	Funktionsbaustein
FB112	MC_ReadParameter	FUP	1904	Funktionsbaustein
FB113	MC_WriteParameter	FUP	1718	Funktionsbaustein
FB114	MC_Parametrization	FUP	5698	Funktionsbaustein
FB115	MC_PosParametrization	FUP	2366	Funktionsbaustein
FC17	INSERT	AWL	526	Funktion
FC100	MC_Error_ID	FUP	1368	Funktion
FC101	MC_Communication_S7-300	FUP	2828	Funktion
DB100	Antriebsdaten	DB	1784	Datenbaustein
UDT2	DRIVE_DATA	AWL	---	Datentyp
UDT3	Parameter	AWL	---	Datentyp
UDT4	ParameterEnable	AWL	---	Datentyp

(Ansicht in Step 7 Classic V5.5)



(Ansicht in Step 7 TIA V13 + V14)

5.2 Kopieren von Bausteinen in das Anwenderprojekt

Folgende Elemente aus der Bibliothek sind (unabh. von den tatsächlich verwendeten Funktionsbausteinen „MC_...“ und der Anzahl der Antriebe) in jedem Fall in das Projekt des Anwenders zu kopieren:

In „Programmbausteine“ der gewünschten CPU kopieren:

- Datentyp DRIVE_DATA („UDT2“ bei Step 7 Classic V5.5)
- für S7-1200 und S7-1500: MC_Communication_S7-1200_1500
alternativ für S7-300: MC_Communication_S7-300
- MC_Error_ID

zusätzlich bei Verwendung der Bibliotheksversion für Step 7 Classic V5.5:

In „S7-Programm → Bausteine“ kopieren:

- FC17 („INSERT“) ... eine Siemens-Systemfunktion

Diese Elemente dienen der Kommunikation mit dem Antrieb. Die Funktionen und Funktionsblöcke aus dieser Gruppe dürfen im Programm NICHT aufgerufen werden.

Daneben sind die gewünschten Funktionsbausteinen „MC_...“ in das Anwenderprojekt zu kopieren, also alle oder eine Auswahl der folgenden Bausteine:

- MC_Move
- MC_Error
- MC_ReadParameter
- MC_WriteParameter
- MC_Parametrization
- MC_PosParametrization
- Datentyp „Parameter“
- Datentyp „ParameterEnable“

Zusätzlich kann der Datenbaustein „Antriebsdaten“ als Vorlage für die Anbindung an die Funktionsbausteine übernommen werden, in dessen Standardausführung müssen dann auch die Datentypen „Parameter“ und „ParameterEnable“ übernommen werden (s. Kap. 3).

5.3 Erzeugen von Instanzdatenbausteinen

Pro Achse und pro gewünschtem Funktionsbaustein muss ein Instanzdatenbaustein erzeugt werden.

5.4 Anlegen von globalen Daten

Pro Achse muss eine globale Variable vom Typ DRIVE_DATA angelegt werden (Größe: 132 Byte in Classic Step 7, 100 Byte im TIA-Portal). Außerdem müssen Variablen erzeugt werden, die an die Ein- und Ausgänge der einzelnen Bausteine angeschlossen werden.

Ggf. ist es sinnvoll, für diese Daten einen weiteren Datenbaustein vom Typ „Global“ anzulegen (z.B. DB100), im Falle des TIA-Portals kann auch die Vorlage „Antriebsdaten“ verwendet werden (s. Kap. 3).

Es müssen nicht alle Ein- und Ausgänge beschaltet werden. Wenn Teile eines Bausteins nicht benötigt werden, können die zugehörigen Eingänge unbeschaltet bleiben, es gilt dann der jeweilige Anfangswert für diesen Eingang. Nicht benötigte Ausgänge können ebenfalls offen bleiben.

5.5 Gemeinsamkeiten aller Funktionsbausteine

Die Bausteine werden in einem Programmteil, der zyklisch aufgerufen wird (z.B. im OB1), eingefügt und sofort mit ihren jeweiligen Instanzdatenbausteinen verknüpft.

An den Eingang „Drive“ (Typ IN_OUT) muss zwingend die für diese Achse vorgesehene Variable vom Typ „DRIVE_DATA“ angeschlossen werden.

Der Eingang EN und der Ausgang ENO jedes Bausteins können unbeschaltet bleiben.

Die Ein-/Ausgänge „Drive“, „EN“ und „ENO“ sind in den folgenden Beschreibungen der einzelnen FBs nicht mehr gesondert aufgeführt.

5.6 Verriegelungen zwischen den Funktionsbausteinen

Die Bausteine sind z.T. gegeneinander verriegelt. Dadurch ist z.B. sichergestellt, dass nicht gleichzeitig zwei Zugriffe aus unterschiedlichen FBs auf den Parameterkanal einer Achse durchgeführt werden können.

Es gelten die folgenden Regeln:

- Wenn der Eingang „Release“ von MC_Move gesetzt ist, können die Bausteine MC_Parametrization und MC_PosParametrization nicht aktiviert werden (Setzen von „Execute“ bringt den Fehler 16#x100).
- Dagegen ist es möglich, während einer Bewegung MC_ReadParameter oder MC_WriteParameter aufzurufen (z.B. um das aktuelle Drehmoment auszulesen oder während der Fahrt die Solldrehzahl zu ändern).
- Wenn der Eingang „Execute“ von MC_Parametrization oder MC_PosParametrization gesetzt ist, kann der Baustein MC_Move nicht aktiviert werden (Setzen von Release bringt den Fehler 16#1100).
- Es kann stets nur einer der Bausteine MC_ReadParameter, MC_WriteParameter, MC_Parametrization oder MC_PosParametrization aktiv sein. Wenn der Eingang „Execute“ einer dieser Bausteine gesetzt ist und der Eingang „Execute“ eines weiteren Bausteins gesetzt wird, gibt dieser den Fehler 16#x100 aus.
- Der Baustein MC_Error kann unabh. von den anderen Bausteinen stets aktiviert sein.

5.7 Beispiel

Folgendes Beispiel soll die Vorgehensweise beim Einbinden der Funktionsbausteine veranschaulichen:

Im Projekt befinden sich drei Antriebe. Jeder Antrieb soll über einen FB MC_Move angesteuert werden, außerdem soll mit MC_Error der Status jedes Antriebs ermittelt werden können und pro Antrieb sollen beliebige Lese- und Schreibzugriffe ausgeführt werden können.

Gemäß Kap. 5.2 kopieren wir dazu die folgenden Bausteine aus der Bibliothek in das Projekt des Anwenders:

In „Programmbausteine“ der gewünschten CPU kopieren:

- Datentyp „DRIVE_DATA“ („UDT2“ bei Step 7 Classic V5.5)
- FC101 („MC_Communication_S7-1200_1500“ oder „MC_Communication_S7-300“)
- FC100 („MC_Error_ID“)
- FB110 („MC_Move“)
- FB111 („MC_Error“)
- FB112 („MC_ReadParameter“)
- FB113 („MC_WriteParameter“)

zusätzlich bei Verwendung der Bibliotheksversion für Step 7 Classic V5.5:

In „S7-Programm → Bausteine“ kopieren:

- FC17 ("INSERT") ... eine Siemens-Systemfunktion

Im nächsten Schritt erzeugen wir gemäß Kap. 5.3 die folgenden Instanzdatenbausteine:

- drei Instanzdatenbausteine vom Typ FB110 („MC_Move“), z.B.
 - DB1101, symbolischer Name = „PSE_1_FB110“
 - DB1102, symbolischer Name = „PSE_2_FB110“
 - DB1103, symbolischer Name = „PSE_3_FB110“
- drei Instanzdatenbausteine vom Typ FB111 („MC_Error“), z.B.
 - DB1111, symbolischer Name = „PSE_1_FB111“
 - DB1112, symbolischer Name = „PSE_2_FB111“
 - DB1113, symbolischer Name = „PSE_3_FB111“
- drei Instanzdatenbausteine vom Typ FB112 („MC_ReadParameter“), z.B.
 - DB1121, symbolischer Name = „PSE_1_FB112“
 - DB1122, symbolischer Name = „PSE_2_FB112“
 - DB1123, symbolischer Name = „PSE_3_FB112“
- drei Instanzdatenbausteine vom Typ FB113 („MC_WriteParameter“), z.B.
 - DB1131, symbolischer Name = „PSE_1_FB113“
 - DB1132, symbolischer Name = „PSE_2_FB113“
 - DB1133, symbolischer Name = „PSE_3_FB113“

Danach erzeugen wir gemäß Kap. 5.4 einen weiteren Datenbaustein vom Typ „Global“ mit Namen DB1000. Darin legen wir drei Variablen vom Typ „DRIVE_DATA“ an, z.B. mit den folgenden Bezeichnungen:

- „PSE_1“
- „PSE_2“
- „PSE_3“

Zusätzlich legen wir dort die Variablen zur Steuerung der einzelnen Bausteine an, z.B.:

- „PSE_1_Move_DINT_Sollposition“,
- „PSE_2_Move_DINT_Sollposition“,
- „PSE_1_Read_BOOL_Execute“,
- „PSE_2_Read_BOOL_Execute“,
- ...

Nun sind alle Vorbereitungen getroffen, um die Bausteine zu verwenden. Im Editor „Bausteine programmieren“ erscheinen die Bausteine „MC_...“ im Abschnitt „FB Bausteine“. In unserem Beispiel werden also in einem Programmteil, der zyklisch aufgerufen wird (z.B. im OB1), die Funktionsblöcke MC_Move, MC_Error, MC_ReadParameter und MC_WriteParameter jeweils drei mal eingefügt und sofort mit ihren jeweiligen Instanzdatenbausteinen verknüpft. Zusätzlich werden MC_Communication_S7-1200_1500 und MC_Error_ID benötigt, wobei MC_Error_ID in den oben genannten FBs aufgerufen wird. MC_Communication_S7-1200_1500 wird pro Antrieb jeweils separat zyklisch aufgerufen.

An den Eingang „Drive“ legen wir jeweils die für diese Achse vorgesehene Variable vom Typ „DRIVE_DATA“ an (also „PSE_1“, „PSE_2“ und „PSE_3“).

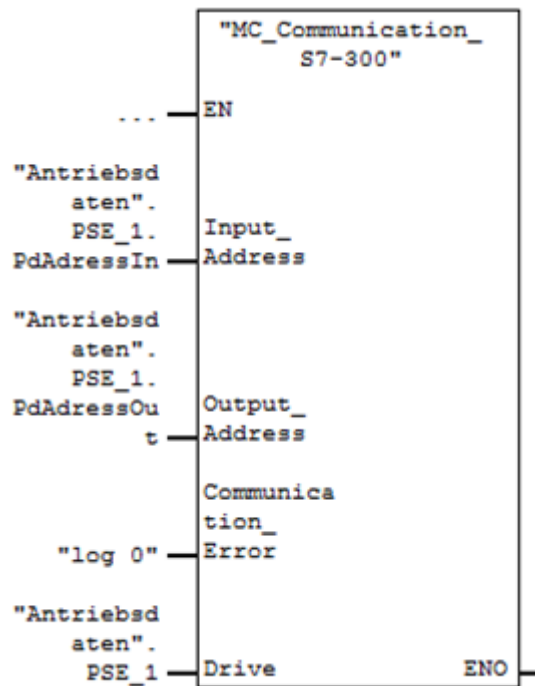
Danach verbinden wir die restlichen benötigten Ein- und Ausgänge mit ihren dafür vorgesehenen globalen Variablen (also „PSE_1_Move_DINT_Sollposition“ usw.).

5.8 MC_Communication_S7-1200_1500, MC_Communication_S7-300 (FC101)

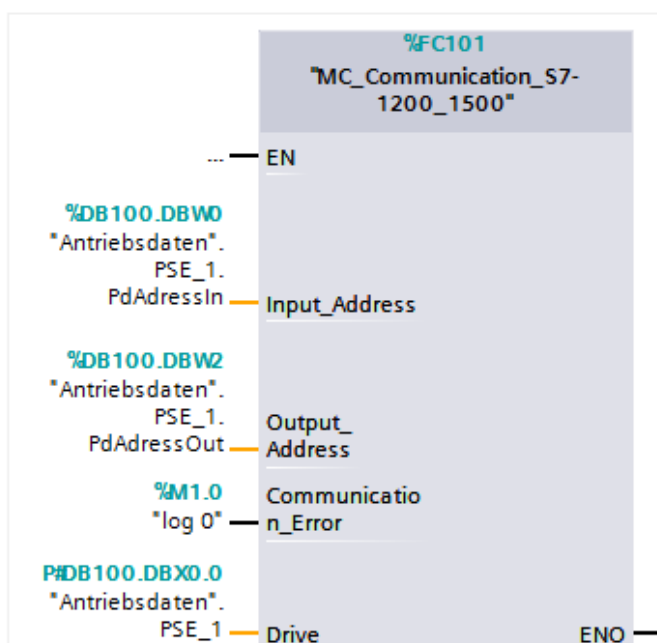
Bei S7-300 Steuerungen heißt der Baustein „MC_Communication_S7-300“. Bei S7-1200- und S7-1500-Steuerungen heißt der Baustein „MC_Communication_S7-1200_1500“. Die

Funktionalität ist in beiden Fällen identisch, im Folgenden wird der Baustein für die S7-1200 und S7-1500 beschrieben.

Dieser FC dient der Kommunikation mit dem Antrieb (IO-Device). Er führt die Kommunikation zentral für alle folgenden Bausteine (MC_Move, MC_ReadParameter, MC_WriteParameter, MC_Error, MC_Parametrization, MC_PosParametrization) durch. Das Ein- und Ausgangsmodul wird im Datentyp DRIVE_DATA abgelegt. Über diese Schnittstelle können die anderen Bausteine darauf zugreifen.



(Ansicht in Step 7 Classic V5.5)



(Ansicht in Step 7 TIA V13 + V14)

Input Address

Adresse des Eingangsmoduls des Antriebs

- Typ: INT (bei S7-1200 und S7-1500), DWORD (bei S7-300)
- Art: INPUT

Output Address

Adresse des Ausgangsmoduls des Antriebs

- Typ: INT (bei S7-1200 und S7-1500), DWORD (bei S7-300)
- Art: INPUT

Communication Error

Kommunikationsfehler zu dem IO-Device (Antrieb)

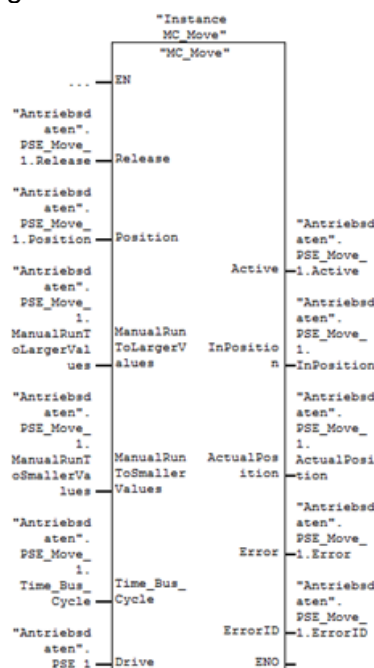
- Typ: BOOL
- Art: INPUT

Beschreibung:

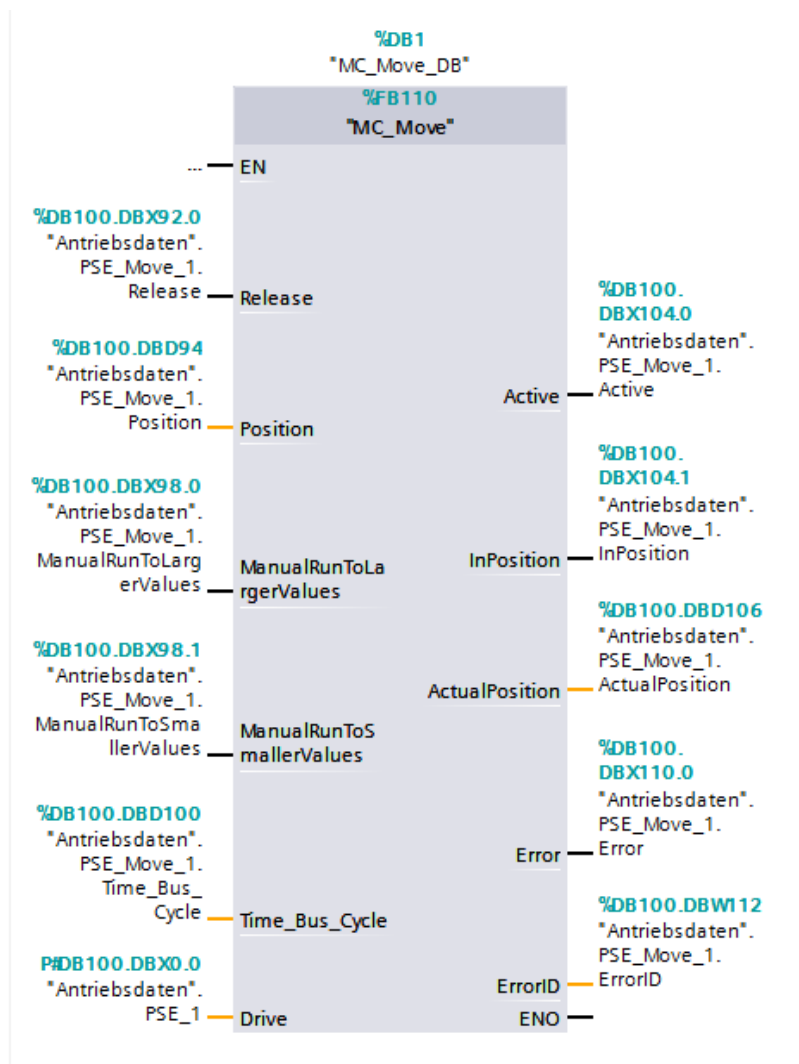
- Die Diagnose der Baugruppen-Zustände der IO-Devices wird in der Regel zentral im SPS-Programm verwaltet, z.B. im OB86 oder mit der Anweisung „DeviceStates“.
- Beim Ausfall des entsprechenden IO-Devices (d.h. hier des Antriebs) ist dieser Eingang mit „TRUE“ zu beschalten. Solange die Kommunikation nicht beeinträchtigt ist, ist der Eingang mit „FALSE“ zu belegen.
- Bei Kommunikationsfehlern werden mit Hilfe von MC_Move gestartete Fahraufträge abgebrochen. Ebenso werden laufende Parameterzugriffe (MC_ReadParameter, MC_WriteParameter) abgebrochen.

5.9 MC_Move (FB110)

Dieser FB dient der Positionierung des Antriebs.



(Ansicht in Step 7 Classic V5.5)



(Ansicht in Step 7 TIA V13 + V14)

Release

Freigabe des Antriebs

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

Beschreibung:

- Ein Sollwert wird erst angefahren, wenn dieser Eingang gesetzt ist.
- Dieser Eingang wirkt direkt auf das Freigabebit (Bit 4) im Steuerwort. Bleibt der Eingang gesetzt und ist z.B. das Nachregeln im Antrieb aktiv, so regelt der Antrieb automatisch nach.
- Ist der Eingang gesetzt und wird der Sollwert geändert, so fährt der Antrieb diesen sofort an. Eine Flanke ist nicht erforderlich.
- Wird der Eingang während der Fahrt zurückgesetzt, stoppt der Antrieb.

Position

Anzufahrender Sollwert

- Typ: DINT
- Anfangswert: 0

- Art: INPUT

Beschreibung:

- Wird während einer Fahrt eine neue Sollposition übertragen, wird diese sofort angefahren.
- Ist nach Fahrtende das Release-Bit noch gesetzt und wird der Sollwert geändert, so fährt der Antrieb diesen sofort an.



INFORMATION!

Um den gleichen Sollwert z.B. nach einem Blockieren anzufahren, muss die Freigabe „Release“ zurückgesetzt und erneut gesetzt werden.

ManualRunToLargerValues

Handfahrt zu größeren Werten

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

Beschreibung:

- Handfahrt zu größeren Werten bis zum oberen Endschalter.
- Der Eingang „Release“ muss zusätzlich gesetzt sein/werden.



ACHTUNG!

Beim Zurücksetzen des Eingangs „ManualRunToLargerValues“ muss auch der Release-Eingang zurückgesetzt werden, da der Antrieb ansonsten den Sollwert (FB-Eingang „Position“) anfährt.

ManualRunToSmallerValues

Handfahrt zu kleineren Werten

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

Beschreibung:

- Handfahrt zu kleineren Werten bis zum unteren Endschalter.
- Der Eingang „Release“ muss zusätzlich gesetzt sein/werden.



ACHTUNG!

Beim Zurücksetzen des Eingangs „ManualRunToSmallerValues“ muss auch der Release-Eingang zurückgesetzt werden, da der Antrieb ansonsten den Sollwert (FB-Eingang „Position“) anfährt.

Time_Bus_Cycle

Aktualisierungszeit des Profinet Buszyklus

- Typ: TIME
- Anfangswert: 40 ms
- Art: INPUT

Beschreibung:

- Bei langen Zykluszeiten auf dem Profinet-Bus kann es passieren, dass die SPS keinen Wechsel des Bits „Antrieb läuft“ empfängt. Dies ist dann der Fall, wenn die Fahrtdauer kürzer als der Buszyklus ist.
- Um dem zu begegnen, wurde der Parameter „Time_Bus_Cycle“ eingeführt. Als Defaultwert werden 40 ms für einen Buszyklus angesetzt. Für diese Dauer wird nach

einem Fahrauftrag der Ausgang „Active“ auf jeden Fall gesetzt und der Ausgang „InPosition“ zurückgesetzt. Danach nehmen diese Ausgänge in Abhängigkeit der Rückmeldung des Statusworts (Bit „Antrieb läuft“ und Bit „Sollposition ist erreicht“) den entsprechenden Zustand an.

- Bei längeren Buszykluszeiten empfiehlt es sich, anstatt dem Defaultwert die tatsächliche Dauer des Zyklus multipliziert mit 2 einzugeben.
- Wenn eine kürzere Buszykluszeit garantiert eingehalten wird, die Zeit für die Positionierung sehr kurz ist und nach abgeschlossener Positionierung der übergeordnete Ablauf schnell fortgesetzt werden soll, kann der Wert für Time_Bus_Cycle auch verringert werden.

Active

Fahrauftrag bzw. Fahrt ist aktiv

- Typ: BOOL
- Art: OUTPUT

Dieser Ausgang wird gesetzt, wenn:

- die Freigabe („Release“) von 0 auf 1 gesetzt wird
- die Freigabe („Release“) schon vorhanden ist und sich der Sollwert ändert
- das Bit „Antrieb läuft“ im Status des Antriebs gesetzt ist (z.B. beim Nachregeln des Antriebs)

Dieser Ausgang wird zurückgesetzt, wenn:

- am Ende einer Fahrt das Bit „Antrieb läuft“ im Status des Antriebs nicht mehr gesetzt ist und die Zeit des Parameters „Buszyklus“ (siehe „Time_Bus_Cycle“) abgelaufen ist
- ein Kommunikationsfehler auftritt

InPosition

Sollposition erreicht

- Typ: BOOL
- Art: OUTPUT

Dieser Ausgang ist eine Abbildung des Statusbits „Sollposition erreicht“. Zusätzlich wird der Ausgang für die Dauer des Parameters „Buszyklus“ (siehe „Time_Bus_Cycle“) nach dem Start durch „Release“ auf 0 gesetzt. Falls ein Kommunikationsfehler auftritt, wird er zurückgesetzt.

Actual position

Istwert der Position

- Typ: DINT
- Art: OUTPUT

Dieser Wert ist eine Abbildung der Istposition. Falls ein Kommunikationsfehler auftritt, wird der Wert auf 0 gesetzt.

Error

Fehler bei der Ausführung des FB oder Fehler im Antrieb

- Typ: BOOL
- Art: OUTPUT

Das Fehlerbit kann auch gesetzt sein, während der Antrieb fährt (z.B. Schleppfehler).

ErrorID

Fehler-ID

- Typ: WORD
- Art: OUTPUT

Die ErrorID kann auch gesetzt sein, während der Antrieb fährt (z.B. Schleppfehler). Falls kein Fehler vorliegt, wird 0 ausgegeben.



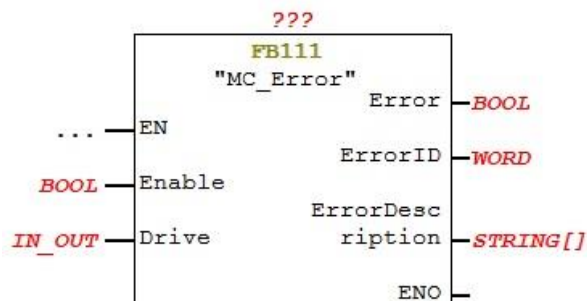
Error und ErrorID von MC_Move werden stets aktualisiert – auch dann, wenn der Eingang „Release“ nicht gesetzt ist.

Falls der Antrieb mehrere Fehler meldet, wird die ErrorID mit der höchsten Priorität ausgegeben. Die Priorität der Ausgabe entspricht der Reihenfolge in der folgenden Tabelle (höchste Prio hat 16#x1xx):

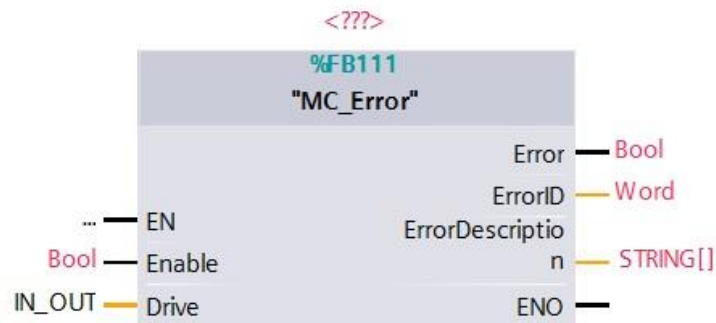
ErrorID	Beschreibung
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x3xx	Invalid PD output address
16#x4xx	Error while reading PD
16#x5xx	Error while writing PD
16#xxx2	Under- or overvoltage motor supply / STO-enabling inactive
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error or STO hardware error
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run / failure voltage control
16#xxx6	Block or overcurrent error
16#xxx7	Manual displacement
16#xxxA	Lower position limit exceeded
16#xxxB	Upper position limit exceeded
16#xxx3	Positioning run aborted
16#xxx1	Drag error

5.10 MC_Error (FB111)

Dieser FB gibt den Status des Antriebs und des FBs als Fehlerbit, Fehler-ID („ErrorID“) und als Text aus. Falls MC_Error aktiviert ist, ist die Fehler-ID stets dieselbe wie diejenige des Bausteins MC_Move.



(Ansicht in Step 7 Classic V5.5)



(Ansicht in Step 7 TIA V13 + V14)

Enable

Die Ausgänge Error, ErrorID und ErrorDescription werden ständig vom Antrieb aktualisiert, solange Enable gesetzt ist. Wird das Enable zurückgesetzt, so nehmen diese Ausgänge die angegebenen Defaultwerte an.

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

Error

Fehler bei der Ausführung des FB oder Fehler im Antrieb

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

ErrorID

Fehler-ID (siehe folgende Tabelle „ErrorID“)

- Typ: WORD
- Defaultwert: 0
- Art: OUTPUT

ErrorDescription

Fehlerbeschreibung als Text

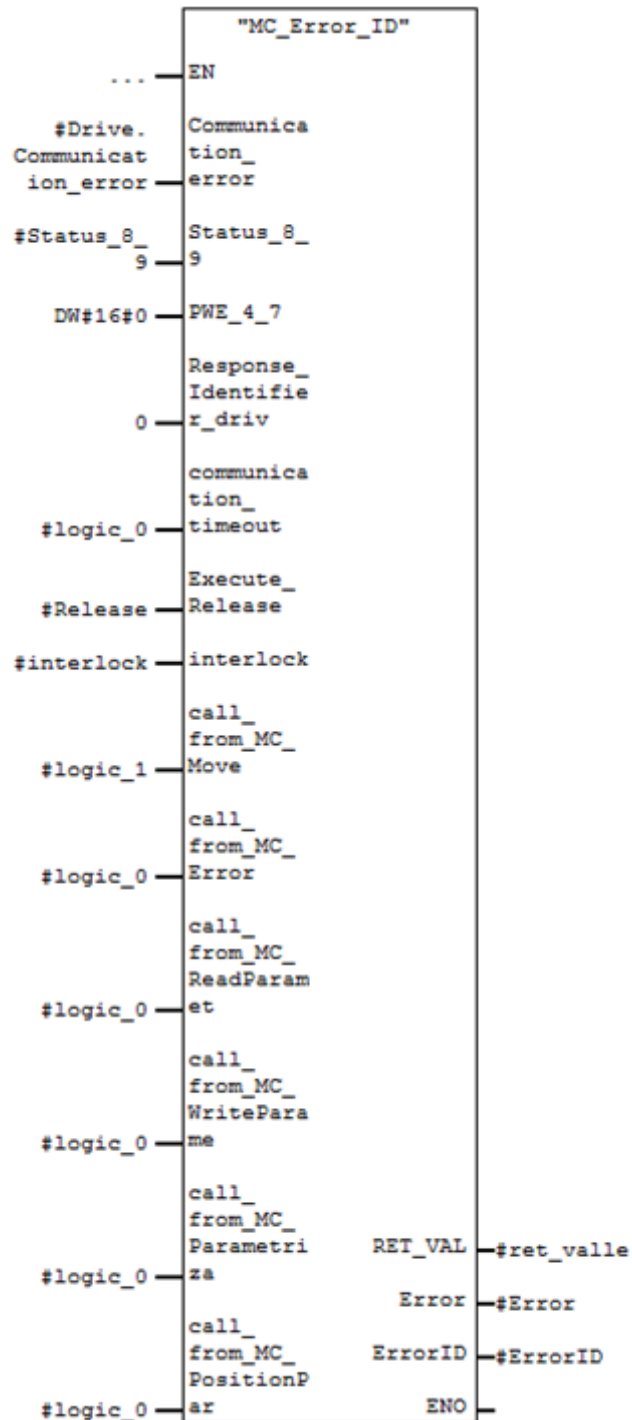
- Typ: STRING
- Defaultwert: „“
- Art: OUTPUT

Die Priorität der Ausgabe entspricht der Reihenfolge in der folgenden Tabelle (höchste Priorität 16#x1xx).

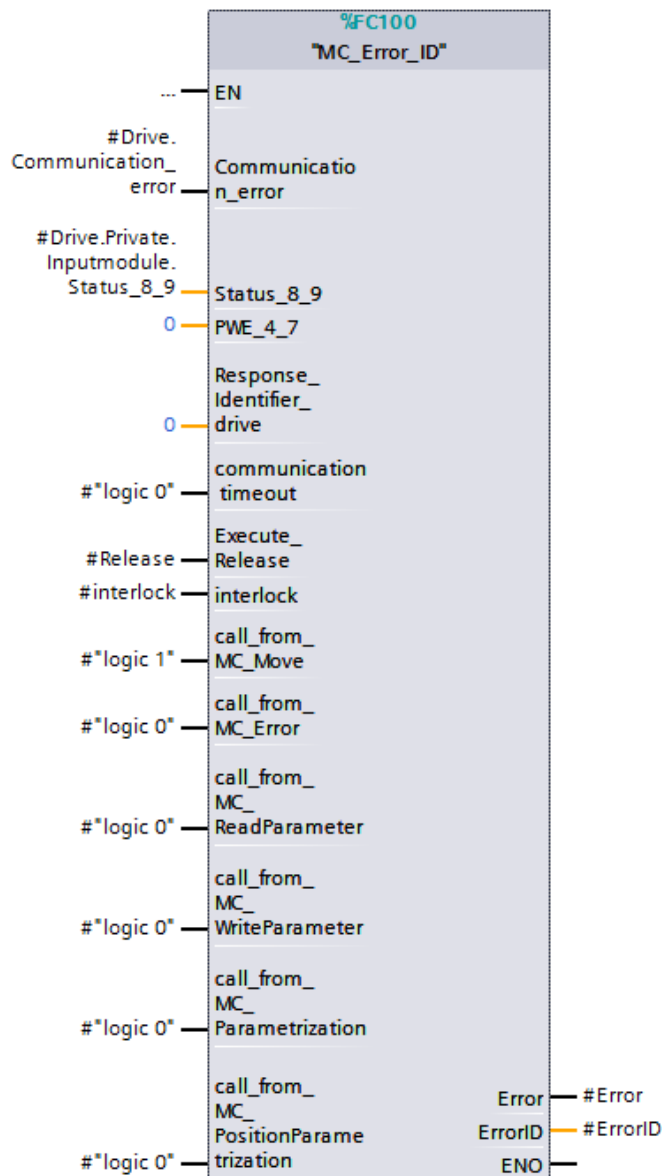
ErrorID	ErrorDescription
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x4xx	Error while reading PD
16#xxx2	Under- or overvoltage motor supply / STO-enabling inactive
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error or STO hardware error
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run / failure voltage control
16#xxx6	Block or overcurrent error
16#xxx7	Manual displacement
16#xxxA	Lower position limit exceeded
16#xxxB	Upper position limit exceeded
16#xxx3	Positioning run aborted
16#xxx1	Drag error

5.11 MC_Error_ID (FC100)

Dieser FC wird intern als Unterfunktion der Bausteine „MC_Move“, „MC_ReadParameter“, „MC_WriteParameter“ und „MC_Error“ eingesetzt. Er ist lediglich aus der Bibliothek in das Anwenderprogramm zu kopieren. Die Beschaltung findet schon intern statt.



(Ansicht in Step 7 Classic V5.5)



(Ansicht in Step 7 TIA V13 + V14)

Communication error

Bei Kommunikationsfehlern werden die Ausgänge „Error“ und die entsprechende „Error-ID“ gesetzt.

- Typ: BOOL
- Art: INPUT

Status 8_9

Auswertung des Statuswort des Antriebs

- Typ: STRUCT
- Art: INPUT

PWE 4_7

Auswertung des Parameterwerts

- Typ: DWORD
- Art: INPUT

Response Identifier drive

Auswertung der Antwortkennung des Antriebs

- Typ: INT
- Art: INPUT

communication timeout

Auswertung eines Kommunikations-Timeout

- Typ: BOOL
- Art: INPUT

Execute Release

Wird für die Auswertung von internen Verriegelungen benötigt. Je nachdem, in welchem FB MC_Error_ID aufgerufen wird, ist hier „Execute“ oder „Release“ beschaltet.

- Typ: BOOL
- Art: INPUT

interlock

Wird für die Auswertung von internen Verriegelungen benötigt.

- Typ: BOOL
- Art: INPUT

Call from MC Move

FC (Unterfunktion) wird in Baustein „MC_Move“ aufgerufen. Je nachdem, in welchem FB MC_Error_ID aufgerufen wird, wird entsprechend die erste Ziffer der Error-ID gesetzt.

- Typ: BOOL
- Art: INPUT

Call from MC Error

FC (Unterfunktion) wird in Baustein „MC_Error“ aufgerufen. Je nachdem, in welchem FB MC_Error_ID aufgerufen wird, wird entsprechend die erste Ziffer der Error-ID gesetzt.

- Typ: BOOL
- Art: INPUT

Call from MC ReadParameter

FC (Unterfunktion) wird in Baustein „MC_ReadParameter“ aufgerufen. Je nachdem, in welchem FB MC_Error_ID aufgerufen wird, wird entsprechend die erste Ziffer der Error-ID gesetzt.

- Typ: BOOL
- Art: INPUT

Call from MC_WriteParameter

FC (Unterfunktion) wird in Baustein „MC_WriteParameter“ aufgerufen. Je nachdem, in welchem FB MC_Error_ID aufgerufen wird, wird entsprechend die erste Ziffer der Error-ID gesetzt.

- Typ: BOOL
- Art: INPUT

Call from MC Parametrization

FC (Unterfunktion) wird in Baustein „MC_Parametrization“ aufgerufen. Je nachdem, in welchem FB MC_Error_ID aufgerufen wird, wird entsprechend die erste Ziffer der Error-ID gesetzt.

- Typ: BOOL
- Art: INPUT

Call from MC_PosParametrization

FC (Unterfunktion) wird in Baustein „MC_PosParametrization“ aufgerufen. Je nachdem, in welchem FB MC_Error_ID aufgerufen wird, wird entsprechend die erste Ziffer der Error-ID gesetzt.

- Typ: BOOL
- Art: INPUT

Error

Wenn ein Fehler auftritt, wird dieses Bit gesetzt.

- Typ: BOOL
- Art: OUTPUT

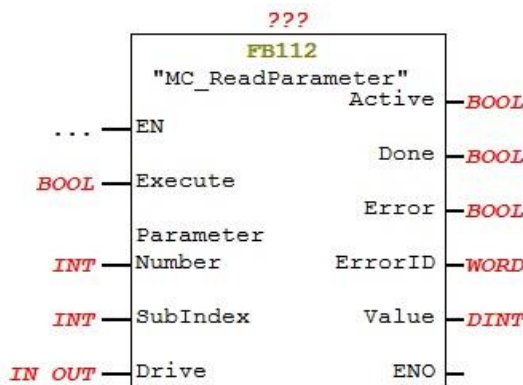
ErrorID

In Abhängigkeit der Eingänge wird „ErrorID“ gesetzt.

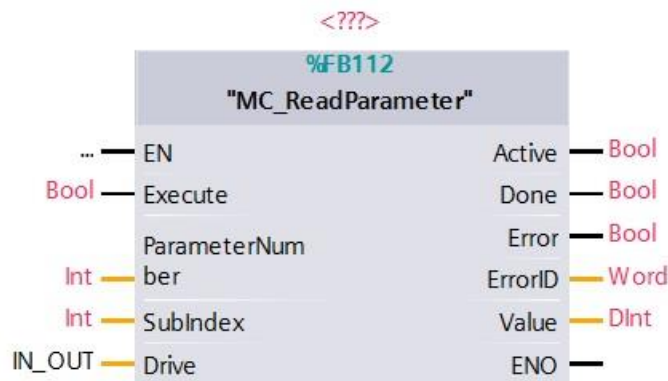
- Typ: WORD
- Art: OUTPUT

5.12 MC_ReadParameter (FB112)

Mit diesem FB können Werte von Parametern aus dem Antrieb ausgelesen werden. Alle Parameter außer Par. 23 („Gerätetyp als String“) können gelesen werden.



(Ansicht in Step 7 Classic V5.5)



(Ansicht in Step 7 TIA V13 + V14)

Execute

Start eines Lesevorgangs

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

Beschreibung:

Bei einer steigenden Flanke wird ein Lesevorgang des mit „ParameterNumber“ und „Subindex“ spezifizierten Parameters gestartet. Für einen erneuten Lesevorgang muss erneut eine steigende Flanke generiert werden. Wird das Bit zurückgesetzt, so nehmen die Ausgänge die angegebenen Defaultwerte an.

ParameterNumber

Parameternummer des auszulesenden Parameters

- Typ: INT
- Anfangswert: 0
- Art: INPUT

SubIndex

Subindex des Parameters

- Typ: INT
- Anfangswert: 0
- Art: INPUT

Active

Bit ist gesetzt, solange der Lesevorgang läuft

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

Das Bit wird zurückgesetzt, sobald der Wert gelesen wurde oder ein Fehler aufgetreten ist.

Done

Bit ist gesetzt, sobald der Parameter erfolgreich gelesen wurde und an „Value“ anliegt

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

Das Bit wird beim Start eines Lesevorgangs zurückgesetzt.

Error

Bit ist gesetzt, wenn während der Ausführung des FBs ein Fehler aufgetreten ist

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

ErrorID

Fehler-ID (siehe Tabelle „ErrorID“ in Kap. 4)

- Typ: WORD
- Defaultwert: 0
- Art: OUTPUT

Antriebsfehler („Drive errors“) werden beim Lesen eines Parameters nicht beachtet.

Value

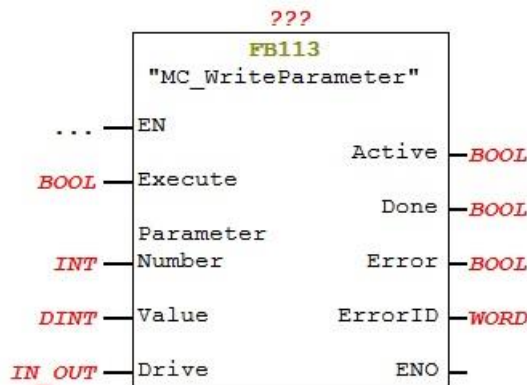
Istwert des ausgelesenen Parameters

- Typ: DINT
- Defaultwert: 0
- Art: OUTPUT

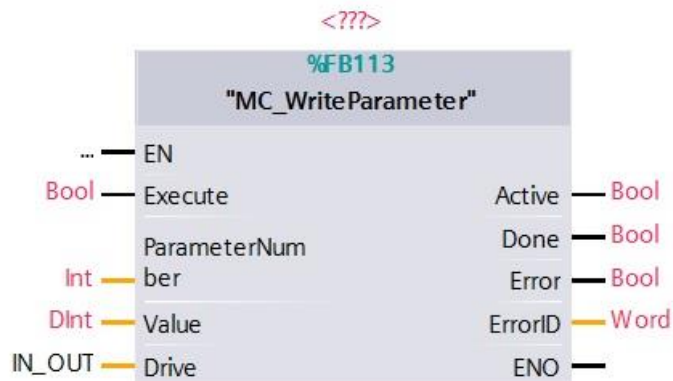
Bei einem Fehler wird 0 ausgegeben.

5.13 MC_WriteParameter (FB113)

Mit diesem FB können Parameterwerte in den Antrieb geschrieben werden.



(Ansicht in Step 7 Classic V5.5)



(Ansicht in Step 7 TIA V13 + V14)

Execute

Start eines Schreibvorgangs

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

Beschreibung:

Bei einer steigenden Flanke wird ein Schreibvorgang des mit „ParameterNumber“ und „Subindex“ spezifizierten Parameters mit dem Wert aus dem Eingang „Value“ gestartet. Für einen erneuten Schreibvorgang muss erneut eine steigende Flanke generiert werden. Wird das Bit zurückgesetzt, so nehmen die Ausgänge die angegebenen Defaultwerte an.

ParameterNumber

Parameternummer des zu schreibenden Parameters

- Typ: INT
- Anfangswert: 0
- Art: INPUT

Value

zu schreibender Wert des betr. Parameters

- Typ: DINT
- Anfangswert: 0
- Art: INPUT

Active

Bit ist gesetzt, solange der Schreibvorgang läuft

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

Das Bit wird zurückgesetzt, sobald der Wert geschrieben wurde oder ein Fehler aufgetreten ist.

Done

Bit ist gesetzt, sobald der Parameter erfolgreich geschrieben wurde

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

Das Bit wird beim Start eines Schreibvorgangs zurückgesetzt.

Error

Bit ist gesetzt, wenn während der Ausführung des FBs ein Fehler aufgetreten ist

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

ErrorID

Fehler-ID (siehe Tabelle „ErrorID“ in Kap. 4)

- Typ: WORD
- Defaultwert: 0
- Art: OUTPUT

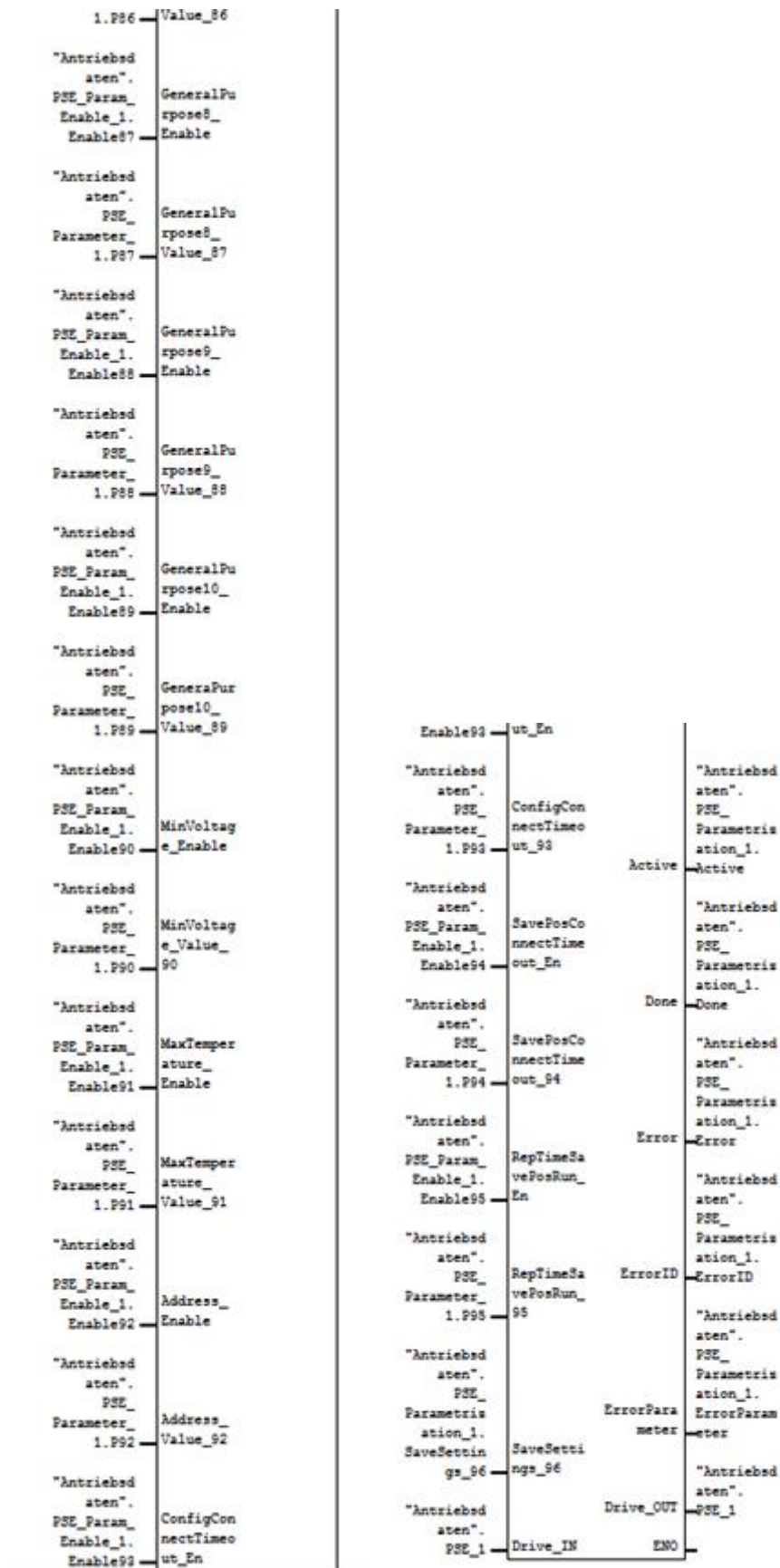
Antriebsfehler („Drive errors“) werden beim Schreiben eines Parameters nicht beachtet.

5.14 MC_Parametrization (FB114)

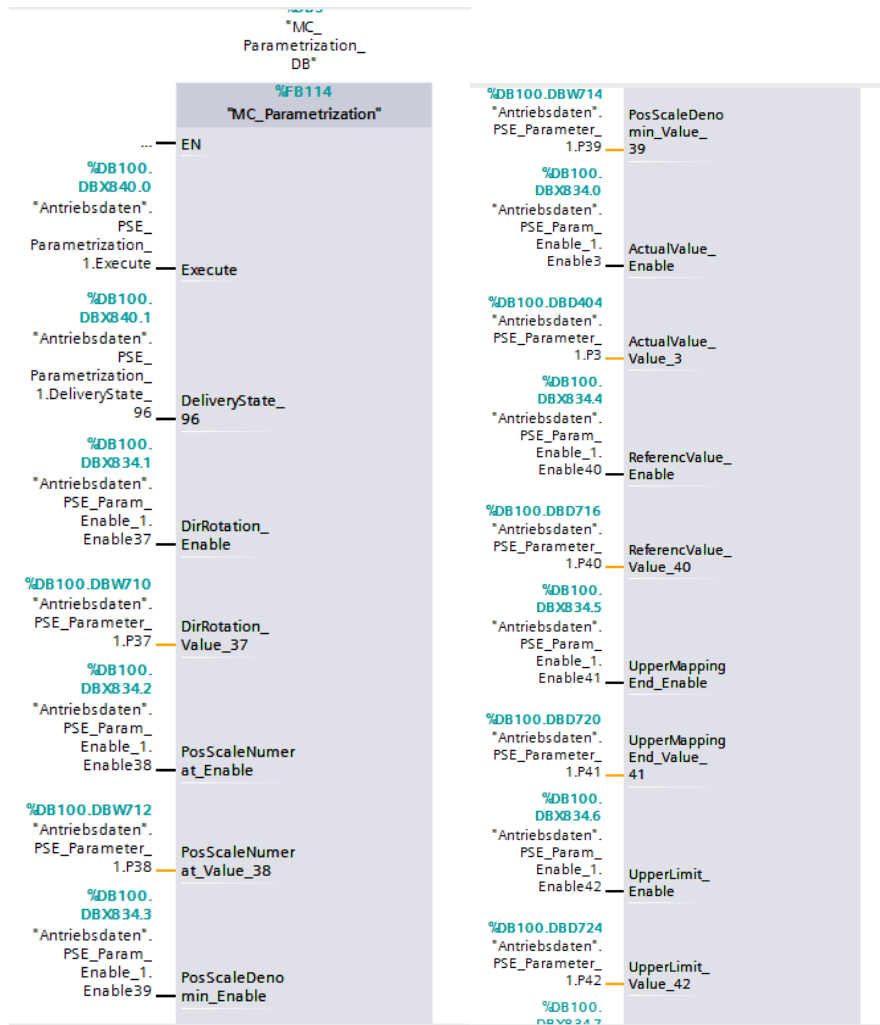
Mit diesem FB können sämtliche Parameter des Antriebs geschrieben werden.

"Instance_		MC_	Enable40	Enable	1.P46	Value_46
Parametris		at"	"Antriebs		"Antriebs	
"MC_			aten".		aten".	
Parametrisation"		PSE_	PSE_	ReferencV	PSE_Param_	Readjustm
...		Parameter_	Parameter_	alue_	Enable_1.	ent_
EN		1.P40	Value_40		Enable47	Enable
"Antriebs		"Antriebs			"Antriebs	
aten".		aten".			aten".	
PSE_		PSE_Param_	UpperMapp		PSE_	Readjustm
Parametris		Enable_1.	ingEnd_		Parameter_	ent_
ation_1.		Enable41	Enable		1.P47	Value_47
Execute	Execute					
"Antriebs		"Antriebs			"Antriebs	
aten".		aten".			aten".	
PSE_		PSE_Param_	UpperMapp		PSE_Param_	DragError
Parametris		Enable_1.	ingEnd_		Enable_1.	Corr_
ation_1.		Enable41	Value_41		Enable48	Enable
DeliverySt	DeliveryS	"Antriebs			"Antriebs	
ate_96	tate_96	aten".			aten".	
"Antriebs		PSE_Param_	UpperLimi		PSE_	DragError
aten".		Enable_1.	t_Enable		Parameter_	Corr_
PSE_Param_		Enable42			1.P48	Value_48
Enable_1.	DirRotati	"Antriebs			"Antriebs	
Enable37	on_Enable	aten".			aten".	
"Antriebs		PSE_	UpperLimi		PSE_Param_	sizeIndiI
aten".		Parameter_	t_Value_		Enable_1.	crement_
PSE_		1.P42	42		Enable49	Enable
Parameter_	DirRotati	"Antriebs			"Antriebs	
1.P37	on_Value_	aten".			aten".	
37		PSE_Param_	LowerLimi		PSE_	sizeInInc
"Antriebs		Enable_1.	t_Enable		Parameter_	rement_
aten".		Enable43			1.P49	Value_49
PSE_Param_	PosScaleN	"Antriebs			"Antriebs	
Enable_1.	umerat_	aten".			aten".	
Enable38	Enable	PSE_	LowerLimi		PSE_Param_	TargetSpe
"Antriebs		Parameter_	t_Value_		Enable_1.	ed_Enable
aten".		1.P43	43		Enable53	
PSE_	PosScaleN	"Antriebs			"Antriebs	
Parameter_	umerat_	aten".			aten".	
1.P38	Value_38	PSE_Param_	PositionN		PSE_	TargetSpe
Value_38		Enable_1.	indow_		Parameter_	ed_Value_
"Antriebs		Enable44	Enable		1.P53	53
aten".		"Antriebs			"Antriebs	
PSE_Param_	PosScaleD	aten".			aten".	
Enable_1.	enomin_	PSE_	PositionN		PSE_Param_	TargSpeed
Enable39	Enable	Parameter_	indow_		Enable_1.	ManRun_
"Antriebs		1.P44	Value_44		Enable56	Enable
aten".		"Antriebs			"Antriebs	
PSE_	PosScaleD	aten".			aten".	
Parameter_	enomin_	PSE_Param_	LoopLength		PSE_	TargSpeed
1.P39	Value_39	Enable_1.	t_Enable		Parameter_	ManRun_
Value_39		Enable45			1.P56	Value_56
"Antriebs		"Antriebs			"Antriebs	
aten".		aten".			aten".	
PSE_Param_	ActualVal	PSE_	LoopLength		PSE_Param_	SpeedLimi
Enable_1.	ue_Enable	Parameter_	t_Value_		Enable_1.	tAbort_
Enable43		1.P45	45		Enable57	Enable
"Antriebs		"Antriebs			"Antriebs	
aten".		aten".			aten".	
PSE_	ActualVal	PSE_Param_	DragError		PSE_	SpeedLimi
Parameter_	ue_Value_	Enable_1.	_Enable		Parameter_	tAbort_
1.P3	3	Enable46			1.P57	Value_57
"Antriebs		"Antriebs			"Antriebs	
aten".		aten".			aten".	
PSE_Param_	ReferencV	PSE_	DragError		PSE_Param_	Accelerat
Enable_1.	alue_	Parameter_	Value_46		Enable_1.	ion_
Enable40	Enable	1.P46	Value_46		Enable58	Enable

Enable58	Enable	1.P70	Value_70	Enable80	Enable
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	Acceleration_	PSE_Param_	StartTorqueTime_	PSE_Param_	GeneralPurpose1_
Parameter_		Enable1_	Enable71	Parameter_	Value_80
1.P58	Value_58			1.P80	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	Deceleration_	PSE_Param_	StartTorqueTime_	PSE_Param_	GeneralPurpose2_
Enable1_		Parameter_	Value_71	Enable1_	Enable
Enable59	Enable	1.P71		Enable81	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	Deceleration_	PSE_Param_	HoldTorqueTime_	PSE_Param_	GeneralPurpose2_
Parameter_		Enable1_	EndRun_	Parameter_	Value_81
1.P59	Value_59	Enable72	Enable	1.P81	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	MaxStartTorque_	PSE_Param_	HoldTorqueTime_	PSE_Param_	GeneralPurpose3_
Enable1_		Parameter_	EndRun_	Enable1_	Enable
Enable63	Enable	1.P72	Value_72	Enable82	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	MaxStartTorque_	PSE_Param_	DirChangeTime_	PSE_Param_	GeneralPurpose3_
Parameter_		Enable1_	Enable73	Parameter_	Value_82
1.P63	Value_63			1.P82	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	MaxTorque_	PSE_Param_	DirChangeTime_	PSE_Param_	GeneralPurpose4_
Enable1_		Parameter_	Value_73	Enable1_	Enable
Enable64	Enable	1.P73		Enable83	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	MaxTorque_	PSE_Param_	IdlePeriodManualRun_	PSE_Param_	GeneralPurpose4_
Parameter_		Enable1_	Enable74	Parameter_	Value_83
1.P64	Value_64			1.P83	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	MaxHoldTorque_	PSE_Param_	IdlePeriodManualRun_	PSE_Param_	GeneralPurpose5_
Enable1_		Parameter_	Value_74	Enable1_	Enable
Enable65	Enable	1.P74		Enable84	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	MaxHoldTorque_	PSE_Param_	BrakeRelTime_	PSE_Param_	GeneralPurpose5_
Parameter_		Enable1_	EndRun_	Parameter_	Value_84
1.P65	Value_65	Enable75	Enable	1.P84	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	MaxHoldTorque_	PSE_Param_	BrakeRelTime_	PSE_Param_	GeneralPurpose6_
Enable1_		Parameter_	Value_75	Enable1_	Enable
Enable66	Enable	1.P75		Enable85	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	MaxHoldTorque_	PSE_Param_	MotorVoltageFilterTime_	PSE_Param_	GeneralPurpose6_
Parameter_		Enable1_	Enable76	Parameter_	Value_85
1.P66	Value_66			1.P85	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	AbortRunTime_	PSE_Param_	MotorVoltageFilterTime_	PSE_Param_	GeneralPurpose7_
Enable1_		Parameter_	Value_76	Enable1_	Enable
Enable70	Enable	1.P76		Enable86	
"Antriebsdaten".		"Antriebsdaten".		"Antriebsdaten".	
PSE_Param_	AbortRunTime_	PSE_Param_	GeneralPurpose1_	PSE_Param_	GeneralPurpose7_
Parameter_		Enable1_	Enable80	Parameter_	Value_86
1.P70	Value_70			1.P86	



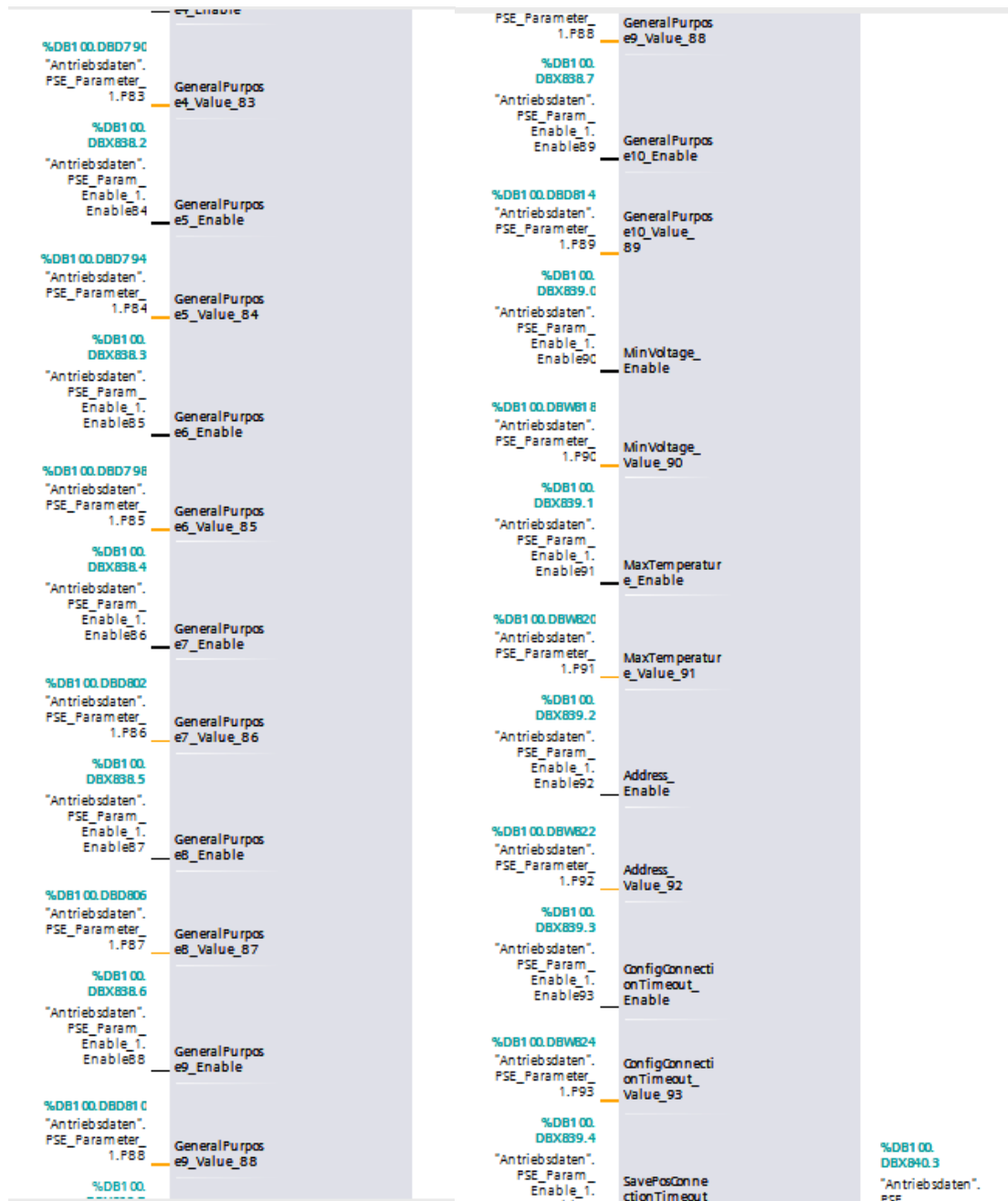
(Ansicht in Step 7 Classic V5.5)

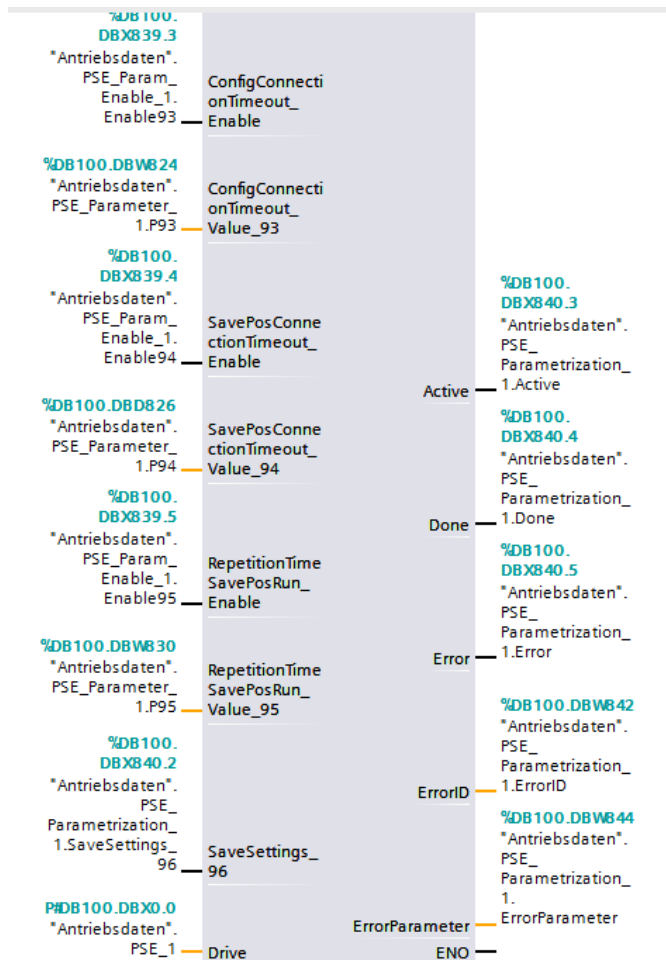






<p>Antriebsdaten . PSE_Parameter_ 1.P71</p> <p>%DB100. DBX837.1 "Antriebsdaten". PSE_Param_ Enable_1. Enable72</p> <p>%DB100.DBW768 "Antriebsdaten". PSE_Parameter_ 1.P72</p> <p>%DB100. DBX837.2 "Antriebsdaten". PSE_Param_ Enable_1. Enable73</p> <p>%DB100.DBW770 "Antriebsdaten". PSE_Parameter_ 1.P73</p> <p>%DB100. DBX837.3 "Antriebsdaten". PSE_Param_ Enable_1. Enable74</p> <p>%DB100.DBW772 "Antriebsdaten". PSE_Parameter_ 1.P74</p> <p>%DB100. DBX837.4 "Antriebsdaten". PSE_Param_ Enable_1. Enable75</p> <p>%DB100.DBW774 "Antriebsdaten". PSE_Parameter_ 1.P75</p> <p>%DB100. DBX837.5 "Antriebsdaten".</p>	<p>StartTorqueTime_ e_Value_71</p> <p>HoldTorTimeEndRun_ Enable</p> <p>HoldTorTimeEndRun_ Value_72</p> <p>DirChangeTime_ Enable</p> <p>DirChangeTime_ Value_73</p> <p>IdlePeriodManualRun_ Enable</p> <p>IdlePeriodManualRun_ Value_74</p> <p>BrRelTimeEndRun_ Enable</p> <p>BrRelTimeEndRun_ Value_75</p>	<p>PSE_Parameter_ Enable_1. Enable76</p> <p>%DB100.DBW776 "Antriebsdaten". PSE_Parameter_ 1.P76</p> <p>%DB100. DBX837.6 "Antriebsdaten". PSE_Param_ Enable_1. Enable80</p> <p>%DB100.DBD778 "Antriebsdaten". PSE_Parameter_ 1.P80</p> <p>%DB100. DBX837.7 "Antriebsdaten". PSE_Param_ Enable_1. Enable81</p> <p>%DB100.DBD782 "Antriebsdaten". PSE_Parameter_ 1.P81</p> <p>%DB100. DBX838.0 "Antriebsdaten". PSE_Param_ Enable_1. Enable82</p> <p>%DB100.DBD786 "Antriebsdaten". PSE_Parameter_ 1.P82</p> <p>%DB100. DBX838.1 "Antriebsdaten". PSE_Param_ Enable_1. Enable83</p> <p>%DB100.DBD790 "Antriebsdaten".</p>	<p>MotVolFiltTime_ Enable</p> <p>MotVolFiltTime_ Value_76</p> <p>GeneralPurpose1_ Enable</p> <p>GeneralPurpose1_ Value_80</p> <p>GeneralPurpose2_ Enable</p> <p>GeneralPurpose2_ Value_81</p> <p>GeneralPurpose3_ Enable</p> <p>GeneralPurpose3_ Value_82</p> <p>GeneralPurpose4_ Enable</p>
--	--	--	--





(Ansicht in Step 7 TIA V13 + V14)

Folgendes ist bei der Nutzung des FBs zu beachten:

- Zu jedem Parameterwert gibt es ein Enable, um festzulegen, ob der Parameter geschrieben werden soll.
Bsp.: DirRotation_Enable = 1 → DirRotation_Value wird gesetzt
- Die Reihenfolge der Schreibzugriffe erfolgt wie im FB dargestellt („DeliveryState“ → „DirRotation“ → ...).
- Wahlweise kann vor dem Setzen einzelner Parameter ein Auslieferungszustand angefordert werden. Dazu muss vor der Ausführung des FBs der Eingang „DeliveryState_96“ auf TRUE gesetzt werden. Dadurch werden die Werte aller Parameter auf den Auslieferungszustand gesetzt (zunächst ohne zu speichern).
- Wahlweise können die geschriebenen Werte am Ende auch gespeichert werden. Dazu muss vor der Ausführung des FBs der Eingang „SaveSettings_96“ auf TRUE gesetzt werden.
- Bei einem Schreibfehler eines Parameters werden die nachfolgenden Parameter nicht mehr geschrieben und es erfolgt auch kein Speichern der Werte, falls der Eingang „SaveSettings“ gesetzt ist.

Execute

Start eines Parametriervorgangs

- Typ: BOOL
- Anfangswert: FALSE

- Art: INPUT

Beschreibung:

Bei einer steigenden Flanke wird ein Parametriervorgang mit den angegebenen Werten gestartet. Für einen erneuten Parametriervorgang muss erneut eine steigende Flanke generiert werden. Wird das Bit zurückgesetzt, so nehmen die Ausgänge die angegebenen Defaultwerte an.

DeliveryState

Laden der Werkseinstellungen (zunächst ohne Speichern)

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

Stationsname und IP-Adresse bleiben jedoch unbeeinflusst.

x_Enable

Falls gesetzt, wird der betr. Parameter geschrieben

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

x_Value

Sollwert des Parameters

- Anfangswert: 0
- Art: INPUT

Die Parameternummer ist hinter dem Parameternamen angegeben. Der Datentyp, eine Beschreibung sowie der Wertebereich kann der Betriebsanleitung des PSx-3__-PN entnommen werden.

SaveSettings

Speichern der Einstellungen

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

Active

Bit ist gesetzt, solange die Parametrierung läuft

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

Das Bit wird zurückgesetzt, sobald die Parameterisierung erfolgreich beendet wurde oder ein Fehler aufgetreten ist.

Done

Bit ist gesetzt, sobald die Parameterisierung erfolgreich beendet wurde

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

Das Bit wird beim Start einer Parametrierung zurückgesetzt.

Error

Bit ist gesetzt, wenn während der Ausführung des FBs ein Fehler aufgetreten ist

- Typ: BOOL

- Defaultwert: FALSE
- Art: OUTPUT

ErrorID

Fehler-ID (siehe Tabelle „ErrorID“ in Kap. 4)

- Typ: WORD
- Defaultwert: 0
- Art: OUTPUT

Antriebsfehler („Drive errors“) werden bei einer Parametrierung nicht beachtet.

ErrorParameter

Parameternummer, bei dem im Fall eines Fehlers der Fehler aufgetreten ist

- Typ: INT
- Defaultwert: 0
- Art: OUTPUT

Falls es keinen Fehler gab, wird 0 ausgegeben.

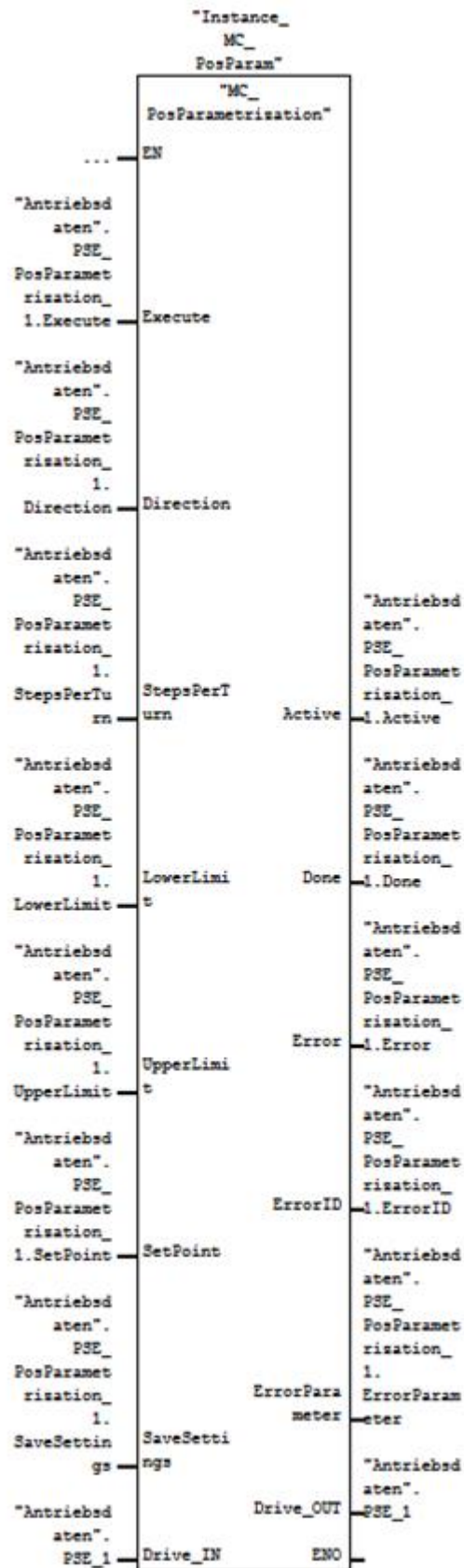


INFORMATION!

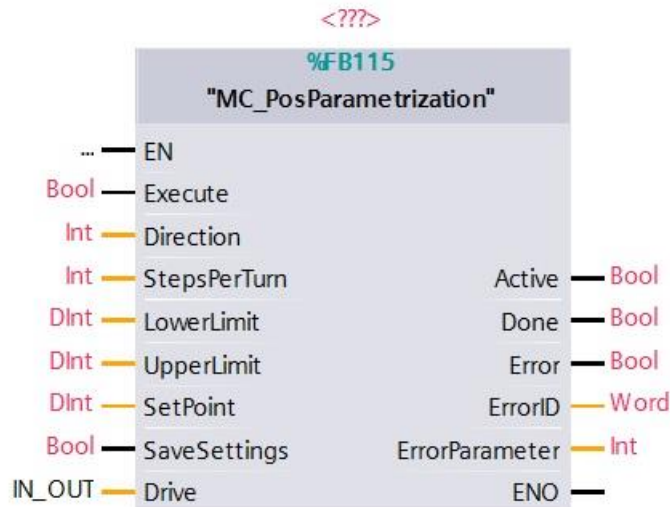
Im Falle der S7-300 muss bei den Bausteinen „MC_Parametrization“ und „MC_PosParametrization“ die Variable vom Typ „DRIVE_DATA“ an den Eingang „Drive_IN“ und an den Ausgang „Drive_OUT“ angeschlossen werden (bei den S7-1200 und S7-1500 im TIA-Portal ist die Variable nur einmal an den Ein-/Ausgang „Drive“ angeschlossen).

5.15 MC_PosParametrization (FB115)

Mit diesem FB kann die Parametrierung der Positionsdaten vorgenommen werden (Parameter, die die angezeigte Istposition beeinflussen).



(Ansicht in Step 7 Classic V5.5)



(Ansicht in Step 7 TIA V13 + V14)

Folgendes ist bei der Nutzung des FBs zu beachten:

- Es müssen alle Werte gesetzt werden und die Werte müssen in einem sinnvollen Bezug zueinander stehen. Alle Werte werden verarbeitet, danach werden die folgenden Parameter in der angeg. Reihenfolge geschrieben:
 - Drehsinn (Par. 37) = Direction
 - Istwertbewertung Zähler (Par. 38) = 400
 - Istwertbewertung Nenner (Par. 39) = StepsPerTurn
 - Istwert (Par. 3) = SetPoint
 - Falls (SetPoint > UpperLimit):
 - oberes Mapping-Ende (Par. 41) = SetPoint + (3 x StepsPerTurn)
 - sonst:
 - oberes Mapping-Ende (Par. 41) = UpperLimit + (3 x StepsPerTurn)
 - obere Endbegrenzung (Par. 42) = UpperLimit
 - untere Endbegrenzung (Par. 43) = LowerLimit
 - Die Anzahl der Schritte pro Umdrehung „StepsPerTurn“ ergibt unmittelbar den Wert des Parameters „Istwertbewertung Nenner“ (Par. 39). Dabei wird angenommen, dass der Wert von „Istwertbewertung Zähler“ (Par. 38) im Auslieferungszustand ist, also auf 400.
 - Vor dem Schreiben der Parameter werden die eingegebenen Werte auf Gültigkeit geprüft.

Nachfolgend die Bedingungen und die Fehlermeldungen, die bei nicht erfüllter Bedingung ausgegeben werden.

Bedingung	ErrorID	ErrorParameter
StepsPerTurn \geq 1	16#6140	39
StepsPerTurn \leq 10000	16#6140	39
LowerLimit \leq UpperLimit	16#6140	42
(UpperLimit - LowerLimit) / StepsPerTurn \leq 250	16#6140	43
Falls SetPoint < LowerLimit: (UpperLimit - SetPoint) / StepsPerTurn \leq 250	16#6140	3
Falls SetPoint > UpperLimit: (SetPoint - LowerLimit) / StepsPerTurn \leq 250	16#6140	3

- Wahlweise können die geschriebenen Werte am Ende auch gespeichert werden. Dazu muss vor der Ausführung des FBs der Eingang „SaveSettings“ auf TRUE gesetzt werden.
- Bei einem Schreibfehler eines Parameters werden die nachfolgenden Parameter nicht mehr geschrieben und es erfolgt auch kein Speichern der Werte, falls der Eingang „SaveSettings“ gesetzt ist.

Execute

Start eines Parametriervorgangs

- Typ: BOOL
- Anfangswert: FALSE
- Art: INPUT

Beschreibung:

Bei einer steigenden Flanke wird ein Parametriervorgang mit den angegebenen Werten gestartet. Für einen erneuten Parametriervorgang muss erneut eine steigende Flanke generiert werden. Wird das Bit zurückgesetzt, so nehmen die Ausgänge die angegebenen Defaultwerte an.

Direction

Richtung, in der der Antrieb bei größeren Werten drehen soll (bei Sicht auf die Abtriebswelle):

0 → CW, 1 → CCW

- Typ: INT
- Anfangswert: 0
- Art: INPUT

StepsPerTurn

Schritte pro Umdrehung an der Abtriebswelle (Auflösung)

- Typ: INT
- Anfangswert: 0
- Art: INPUT

LowerLimit

Untere Endbegrenzung

- Typ: DINT
- Anfangswert: 0
- Art: INPUT

UpperLimit

Obere Endbegrenzung

- Typ: DINT
- Anfangswert: 0
- Art: INPUT

SetPoint

Wert, auf den das Messsystem referenziert wird (neuer Istwert an der aktuellen Position)

- Typ: DINT
- Anfangswert: 0
- Art: INPUT

SaveSettings

Speichern der Einstellungen

- Typ: BOOL

- Anfangswert: FALSE
- Art: INPUT

Active

Bit ist gesetzt, solange die Parametrierung läuft

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

Das Bit wird zurückgesetzt, sobald die Parameterisierung erfolgreich beendet wurde oder ein Fehler aufgetreten ist.

Done

Bit ist gesetzt, sobald die Parameterisierung erfolgreich beendet wurde

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

Das Bit wird beim Start einer Parametrierung zurückgesetzt.

Error

Bit ist gesetzt, wenn während der Ausführung des FBs ein Fehler aufgetreten ist

- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

ErrorID

Fehler-ID (siehe Tabelle „ErrorID“ in Kap. 4)

- Typ: WORD
- Defaultwert: 0
- Art: OUTPUT

Antriebsfehler („Drive errors“) werden bei einer Parametrierung nicht beachtet.

ErrorParameter

Parameternummer, bei dem im Fall eines Fehlers der Fehler aufgetreten ist

- Typ: INT
- Defaultwert: 0
- Art: OUTPUT

Falls es keinen Fehler gab, wird 0 ausgegeben.



INFORMATION!

Im Falle der S7-300 muss bei den Bausteinen „MC_Parametrization“ und „MC_PosParametrization“ die Variable vom Typ „DRIVE_DATA“ an den Eingang „Drive_IN“ und an den Ausgang „Drive_OUT“ angeschlossen werden (bei den S7-1200 und S7-1500 im TIA-Portal ist die Variable nur einmal an den Ein-/Ausgang „Drive“ angeschlossen).