



## **Funktionsbausteine für PSD-4\_\_ und PSC-4\_\_ 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](mailto:info@halstrup-walcher.de)  
Internet: [www.halstrup-walcher.de](http://www.halstrup-walcher.de)

## Inhaltsverzeichnis

<b>1</b>	<b>Sicherheitshinweise .....</b>	<b>4</b>
1.1	Bestimmungsgemäße Verwendung .....	4
1.2	Symbolerklärung .....	4
<b>2</b>	<b>Datenstruktur DRIVE_DATA .....</b>	<b>4</b>
<b>3</b>	<b>Datenbaustein „Antriebsdaten“ .....</b>	<b>6</b>
<b>4</b>	<b>Fehlerbeschreibung (Error ID).....</b>	<b>7</b>
<b>5</b>	<b>Beschreibung und Anwendung der Funktionsbausteine .....</b>	<b>8</b>
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) .....	13
5.9	MC_Move (FB110).....	15
5.10	MC_Error (FB111) .....	19
5.11	MC_Error_ID (FC100).....	22
5.12	MC_ReadParameter (FB112) .....	25
5.13	MC_WriteParameter (FB113).....	29
5.14	MC_Parametrization (FB114).....	32
5.15	MC_PosParametrization (FB115) .....	42

## Bedeutung der Betriebsanleitung

Diese Betriebsanleitung erläutert die Funktionsbausteine für die Positioniersysteme PSD-4\_\_-PN und PSC-4\_\_-PN (mit PROFINET-Schnittstelle).

Die Erläuterungen und Abbildungen dieser Anleitung beziehen sich auf ein Beispielprojekt mit einem PSD-4\_\_-PN-Gerät, sind jedoch ebenso gültig für PSC-4\_\_-PN-Geräte.

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.

© 2021

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 PSD-4\_\_-PN und PSC-4\_\_-PN eignen sich besonders zur automatischen Einstellung von Werkzeugen, Anschlägen oder Spindeln bei Holzverarbeitungsmaschinen, Verpackungsmaschinen, Druckmaschinen, Abfüllanlagen und bei Sondermaschinen.

**Die PSD-4\_\_-PN und PSC-4\_\_-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 shows the Siemens TIA Portal interface. On the left, the 'Geräte & Netze' (Devices & Networks) tree is expanded, showing the configuration of a PLC 1 (CPU 1511-1 PN) and its connection to a PSD411 PSD. The 'Adressübersicht' (Address Overview) table is displayed, showing the assigned addresses for the process data.

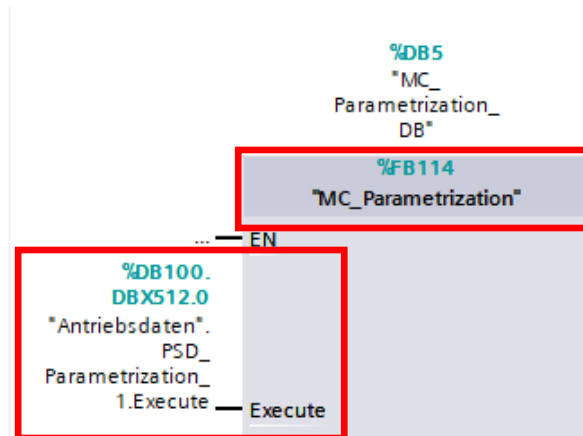
Typ	Adr. von	Adr. bis	Größe	Modul	Baugr...	Steck...	Geräte...	Master-IO-System	TPA	OB
E	0	15	16 Byte	16 Byte Eingang_1	0	1	PSD411 [PSD]	1	PROFINET-System [100]	Automatische Aktua...
A	0	13	14 Byte	14 Byte Ausgang_1	0	2	PSD411 [PSD]	1	PROFINET-System [100]	Automatische Aktua...

(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
<b>16xF000 (mask)</b>	<b>FB</b>
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
<b>16#0F00 (mask)</b>	<b>Internal FB and PD errors</b>
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
<b>16#00F0 (mask)</b>	<b>Parameter errors</b>
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
<b>16#000F (mask)</b>	<b>Drive errors</b>
16#xxx1	Drag error
16#xxx2	Under- or overvoltage motor supply
16#xxx3	Positioning run aborted
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx6	Block or overcurrent error
16#xxx7	Manual displacement
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run
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

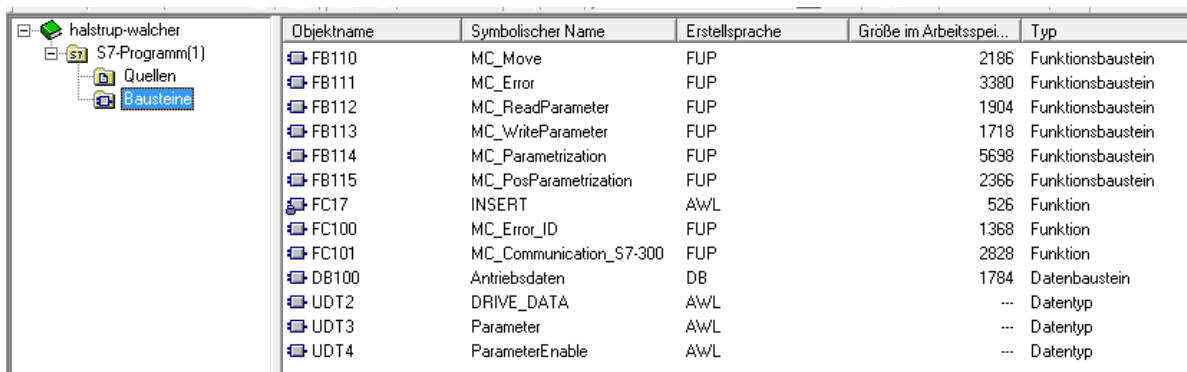
Zunächst müssen die Bausteine in ein eigenes Anwenderprojekt eingebunden werden. Dies geschieht durch Öffnen der gewünschten Bibliothek und Kopieren der gewünschten Funktionsbausteine.

Folgende Bibliotheken stehen in der jeweils aktuellen Version zur Auswahl:

- „library\_Ps4xx\_PN\_S7\_300“  
→ für CPU S7-300 mit Step 7 Classic V5.5
- „library\_Ps4xx\_PN\_TIA13\_300“  
→ für CPU S7-300 im TIA-Portal (V13)
- „library\_Ps4xx\_PN\_TIA13\_1200-1500“  
→ für CPU S7-1200/1500 im TIA-Portal (V13)
- „library\_Ps4xx\_PN\_TIA14\_300“  
→ für CPU S7-300 im TIA-Portal (V14)
- „library\_Ps4xx\_PN\_TIA14\_1200-1500“  
→ für CPU S7-1200/1500 im TIA-Portal (V14)
- „library\_Ps4xx\_PN\_TIA16\_300“  
→ für CPU S7-300 im TIA-Portal (V16)
- „library\_Ps4xx\_PN\_TIA16\_1200-1500“  
→ für CPU S7-1200/1500 im TIA-Portal (V16)

### 5.1 Öffnen der Bibliothek

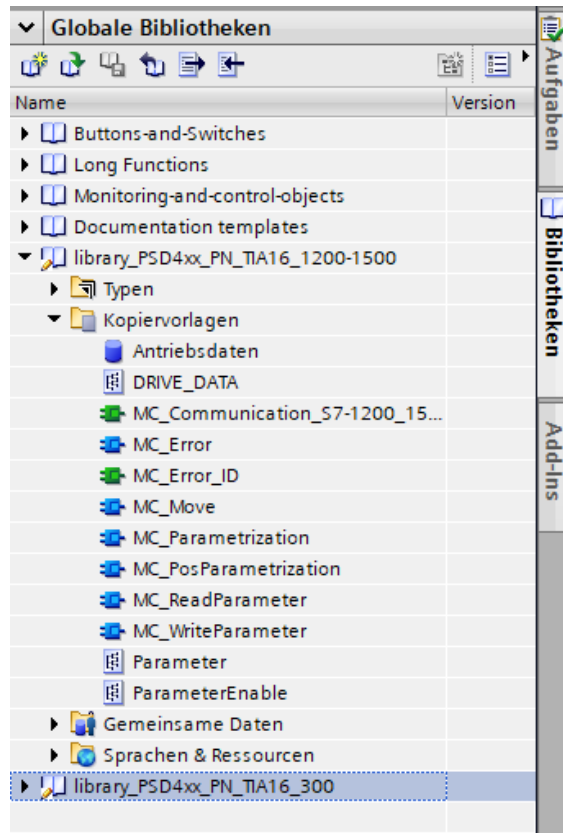
Nach dem Öffnen der Bibliothek präsentieren sich die Funktionsbausteine folgendermaßen:



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)

## 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.

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, 92 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 = „PSD\_1\_FB110“
  - DB1102, symbolischer Name = „PSD\_2\_FB110“
  - DB1103, symbolischer Name = „PSD\_3\_FB110“
- drei Instanzdatenbausteine vom Typ FB111 („MC\_Error“), z.B.
  - DB1111, symbolischer Name = „PSD\_1\_FB111“
  - DB1112, symbolischer Name = „PSD\_2\_FB111“
  - DB1113, symbolischer Name = „PSD\_3\_FB111“
- drei Instanzdatenbausteine vom Typ FB112 („MC\_ReadParameter“), z.B.
  - DB1121, symbolischer Name = „PSD\_1\_FB112“
  - DB1122, symbolischer Name = „PSD\_2\_FB112“
  - DB1123, symbolischer Name = „PSD\_3\_FB112“
- drei Instanzdatenbausteine vom Typ FB113 („MC\_WriteParameter“), z.B.
  - DB1131, symbolischer Name = „PSD\_1\_FB113“
  - DB1132, symbolischer Name = „PSD\_2\_FB113“
  - DB1133, symbolischer Name = „PSD\_3\_FB113“

Danach erzeugen wir gemäß Kap. 5.4 einen weiteren Datenbaustein vom Typ „Global“ (im TIA Portal empfiehlt sich die Vorlage „Antriebsdaten“ zu verwenden) mit Namen DB1000. Darin legen wir drei Variablen vom Typ „DRIVE\_DATA“ an, z.B. mit den folgenden Bezeichnungen:

- „PSD\_1“
- „PSD\_2“
- „PSD\_3“

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

- „PSD\_1\_Move\_DINT\_Sollposition“,
- „PSD\_2\_Move\_DINT\_Sollposition“,
- „PSD\_1\_Read\_BOOL\_Execute“,
- „PSD\_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.

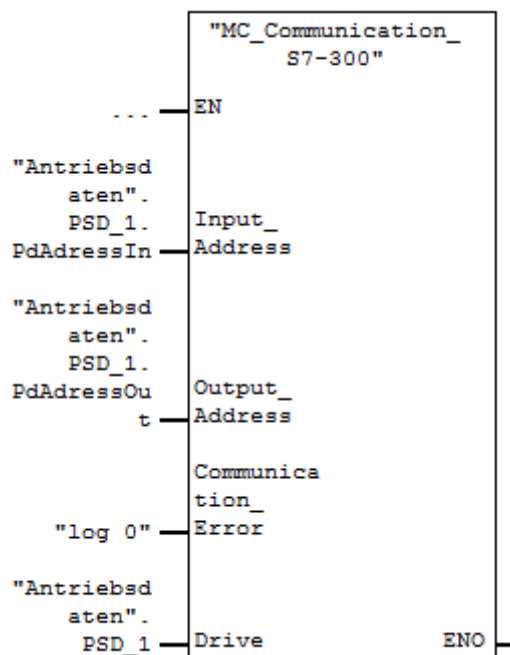
An den Eingang „Drive“ legen wir jeweils die für diese Achse vorgesehene Variable vom Typ „DRIVE\_DATA“ an (also „PSD\_1“, „PSD\_2“ und „PSD\_3“).

Danach verbinden wir die restlichen benötigten Ein- und Ausgänge mit ihren dafür vorgesehenen globalen Variablen (also „PSD\_1\_Move\_DINT\_Sollposition“ usw.).

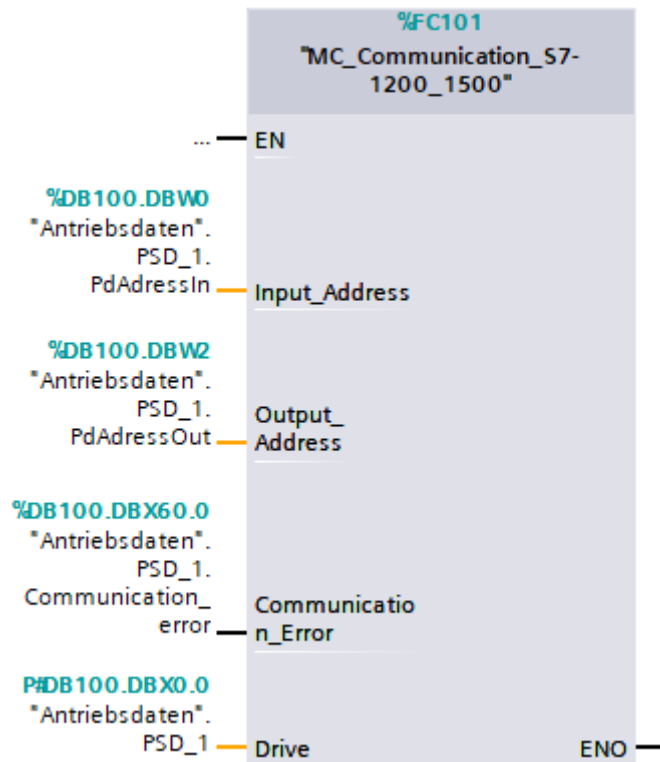
## 5.8 MC\_Communication\_S7-1200\_1500 und \_S7-300 (FC101)

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)

#### 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)

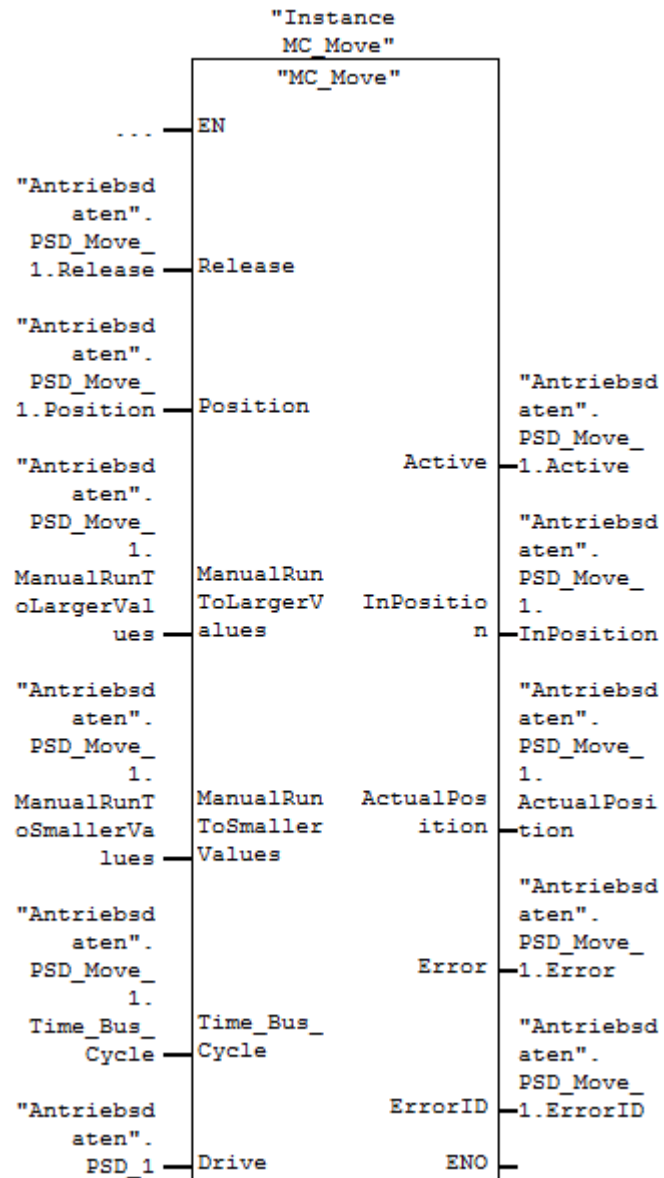
- Typ: BOOL
- Art: INPUT

Beschreibung:

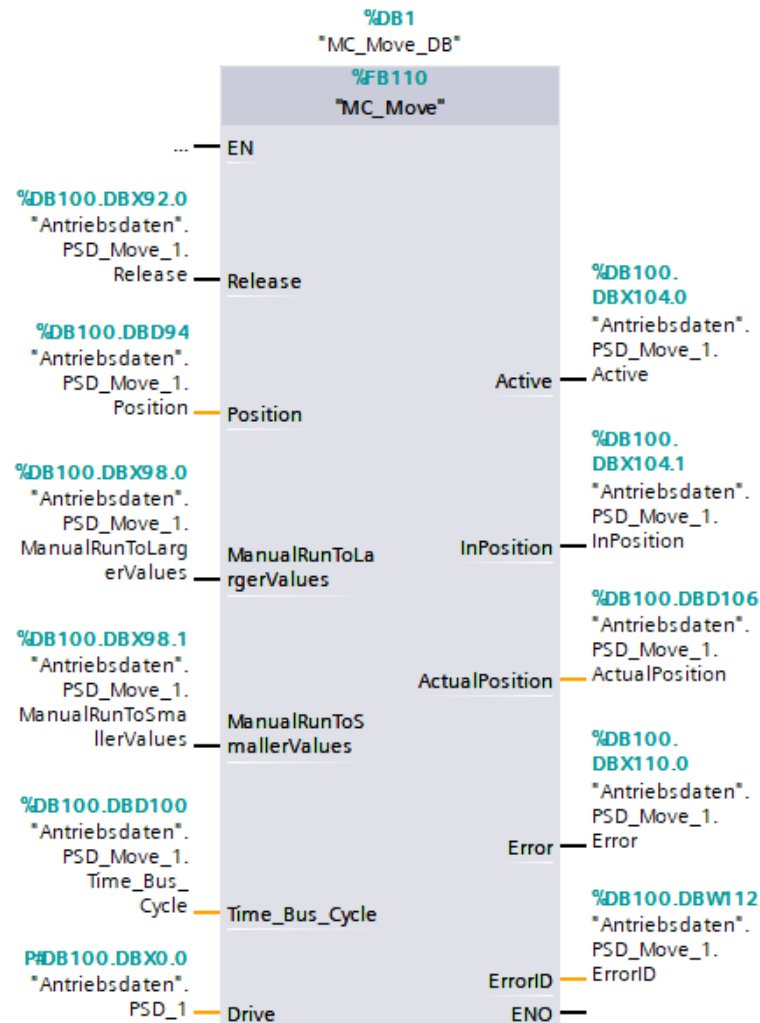
- Die Diagnose der Baugruppen-Zustände der IO-Devices wird in der Regel zentral im SPS-Programm verwaltet, z.B. im OB86 oder mit der Anweisung „DeviceStates“.
- Beim Ausfall des entsprechenden IO-Devices (d.h. hier des Antriebs) ist dieser Eingang mit „TRUE“ zu beschalten. Solange die Kommunikation nicht 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)

### 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.



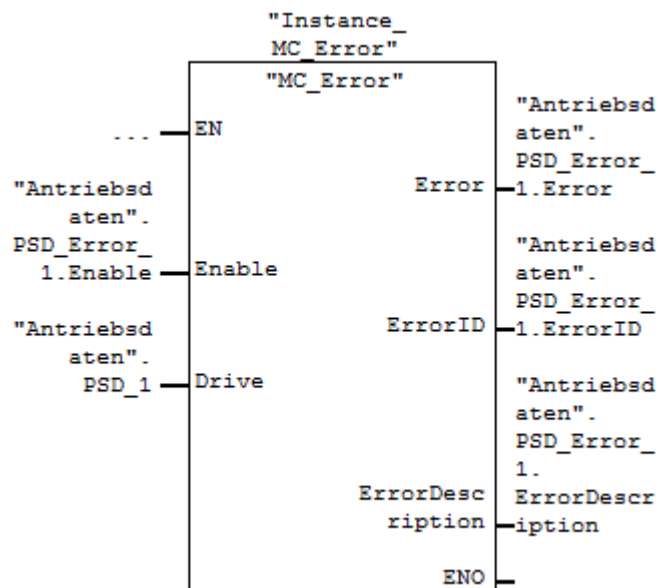
Error und ErrorID von MC\_Move werden stets aktualisiert – auch dann, wenn der Eingang „Release“ nicht gesetzt ist.

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 Priorität hat 16#x1xx):

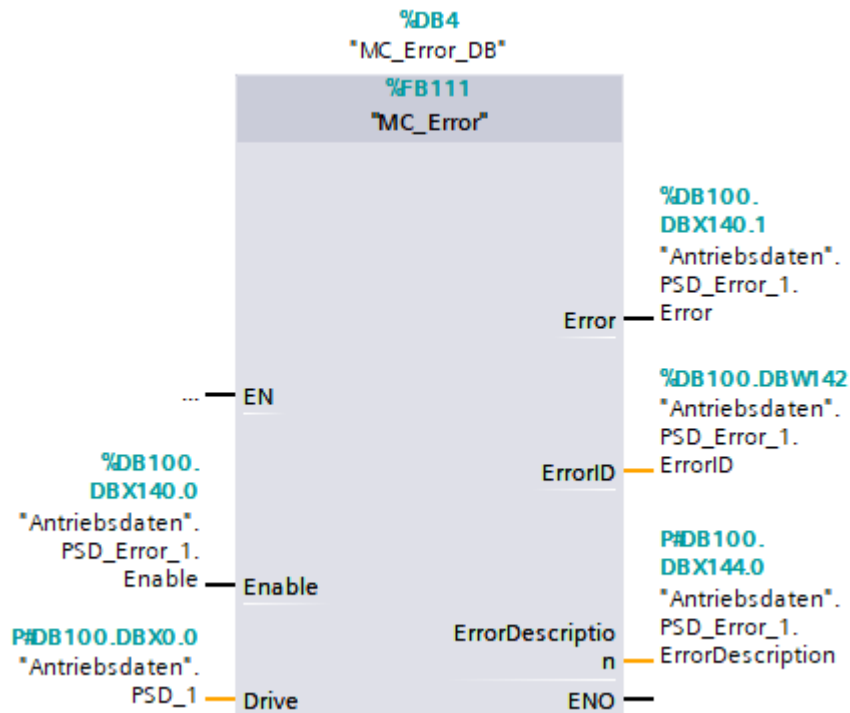
ErrorID	Beschreibung
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x3xx	Invalid PD output address
16#x4xx	Error while reading PD
16#x5xx	Error while writing PD
16#xxx2	Under- or overvoltage motor supply
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run
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)

### 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

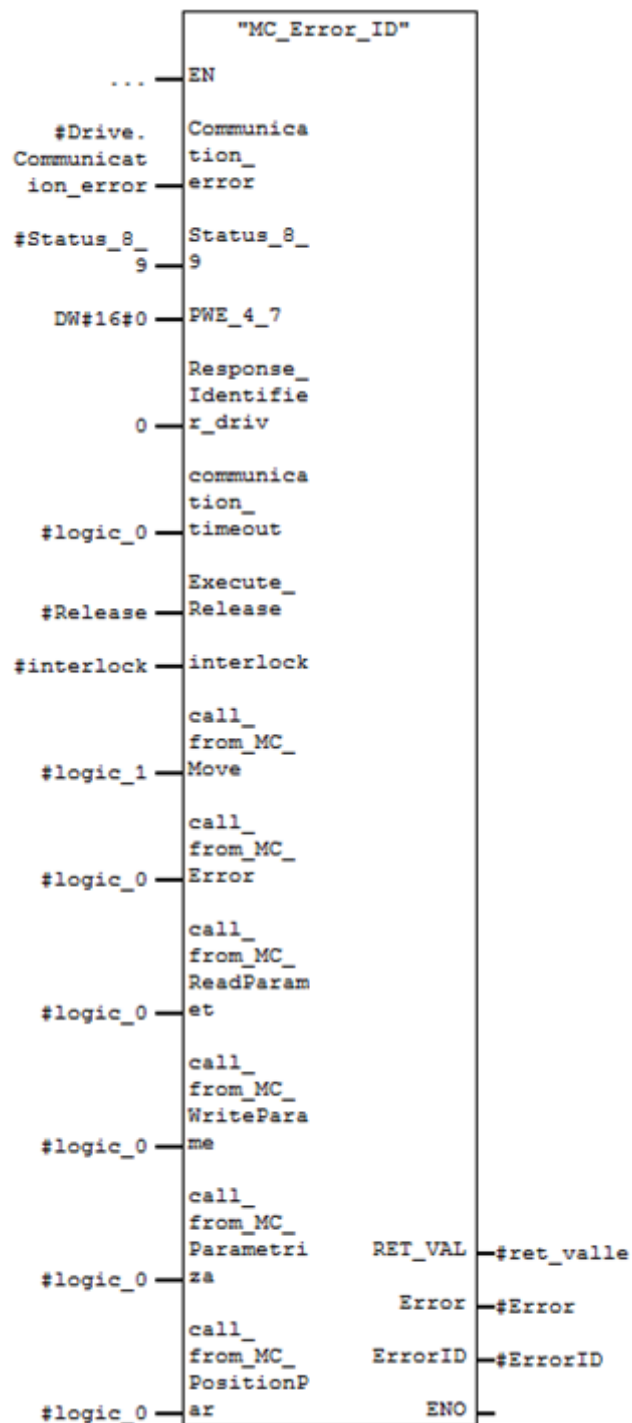
- Typ: STRING
- Defaultwert: „“
- Art: OUTPUT

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

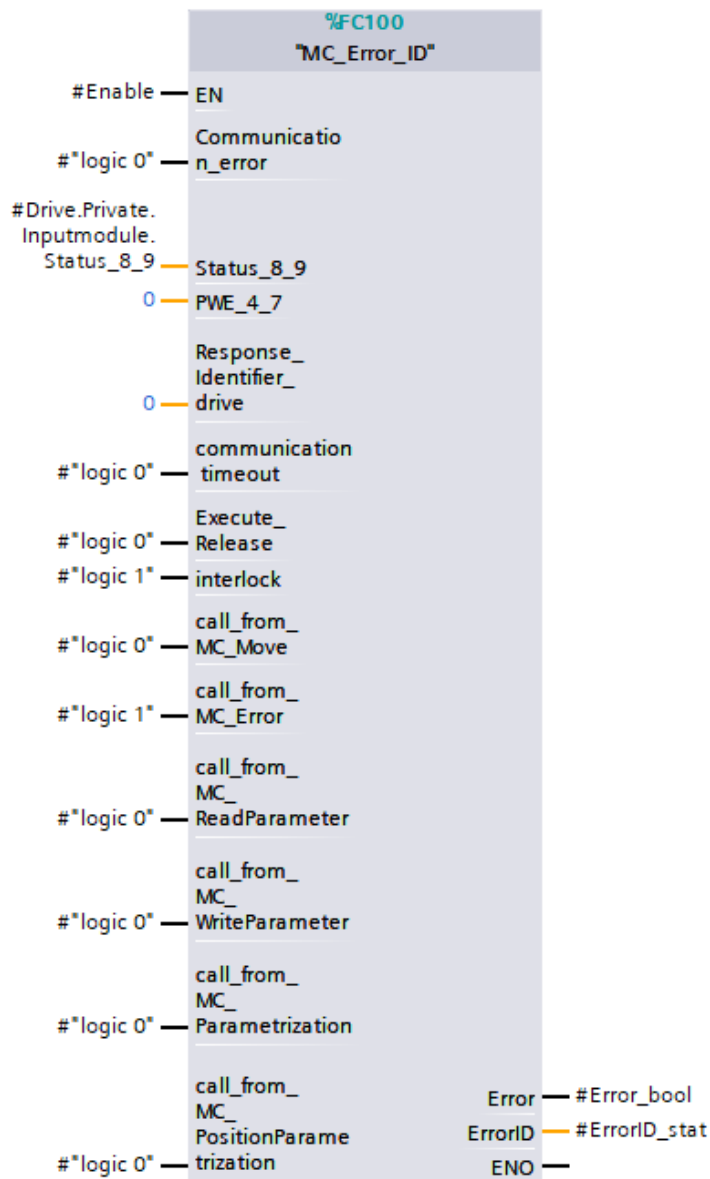
ErrorID	ErrorDescription
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x4xx	Error while reading PD
16#xxx2	Under- or overvoltage motor supply
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run
16#xxx6	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)

### 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

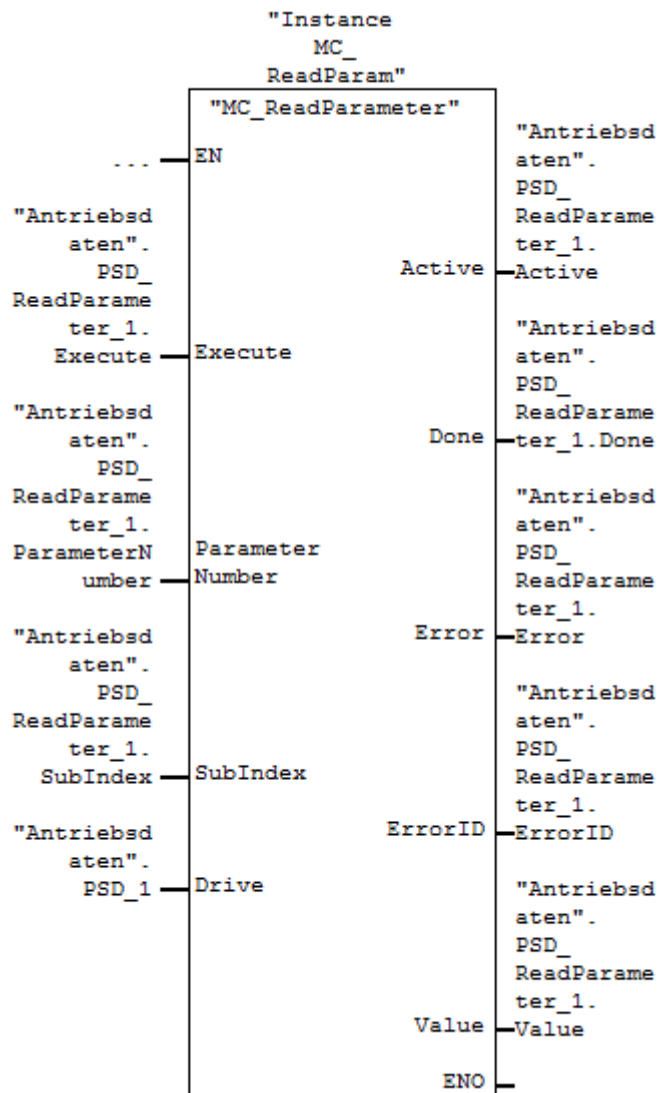
ErrorID

In Abhängigkeit der Eingänge wird „ErrorID“ gesetzt.

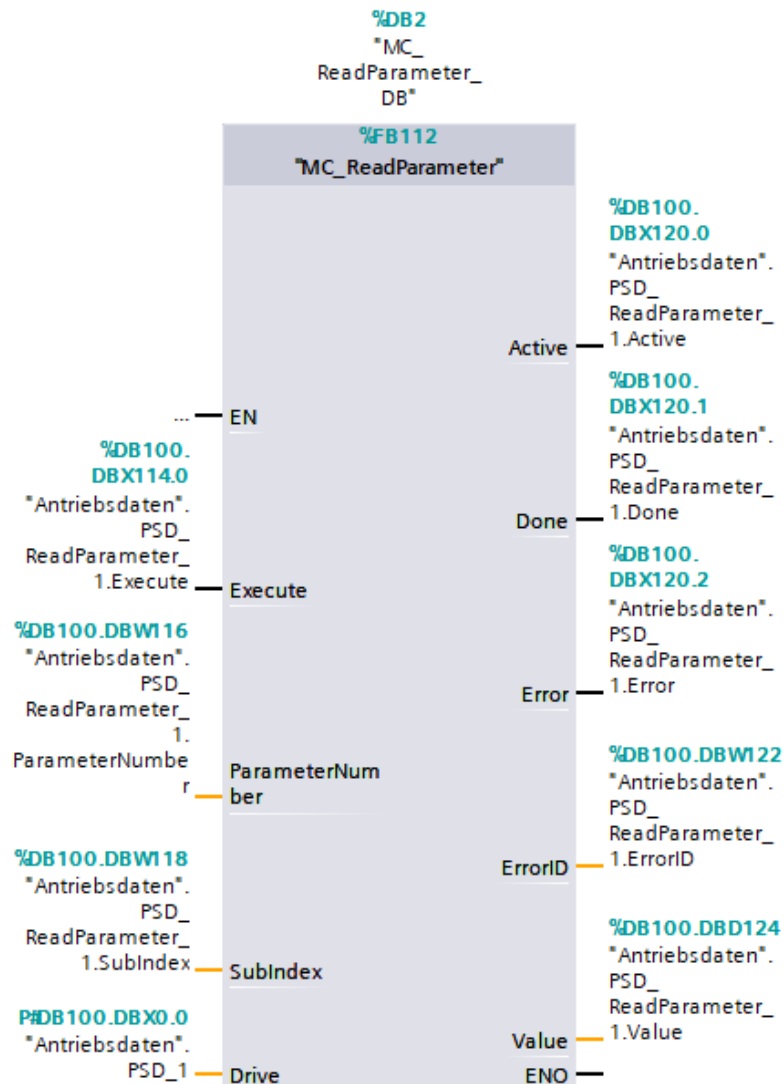
- Typ: WORD
- Art: OUTPUT

**5.12 MC\_ReadParameter (FB112)**

Mit diesem FB können Werte von Parametern aus dem Antrieb ausgelesen werden.



(Ansicht in Step 7 Classic V5.5)



(Ansicht in Step 7 TIA)

### 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

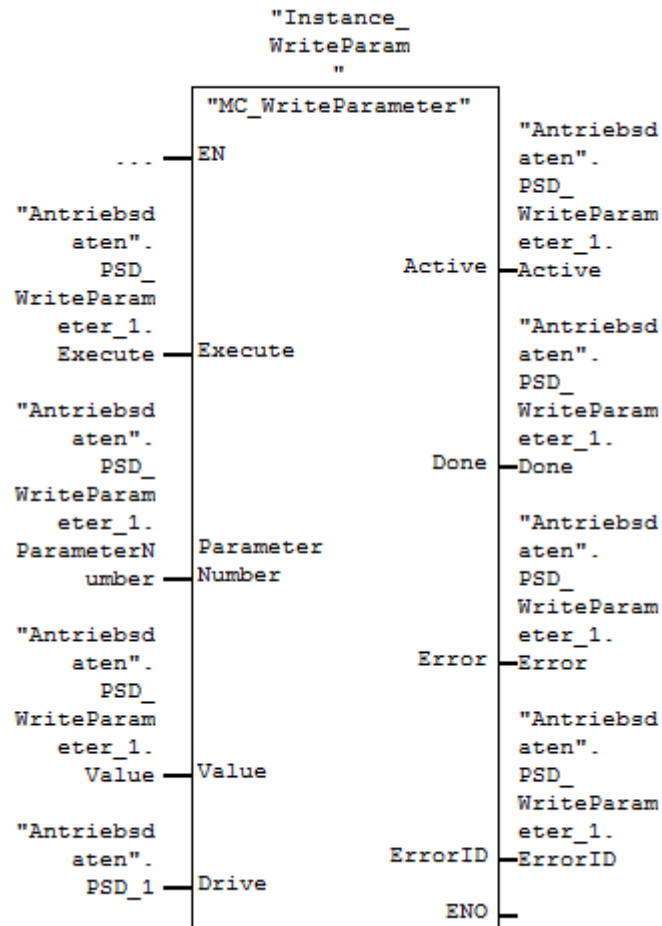
Istwert des ausgelesenen Parameters

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

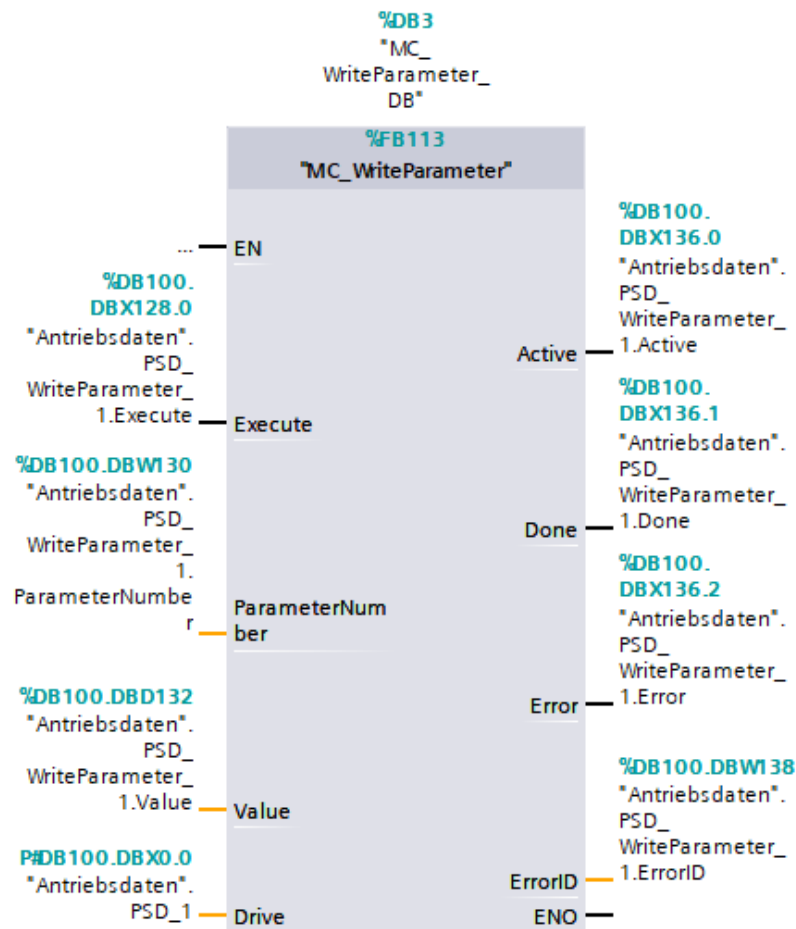
Bei einem Fehler wird 0 ausgegeben.

### 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)

### Execute

Start eines Schreibvorgangs

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

Beschreibung:

Bei einer steigenden Flanke wird ein Schreibvorgang des mit „ParameterNumber“ 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.

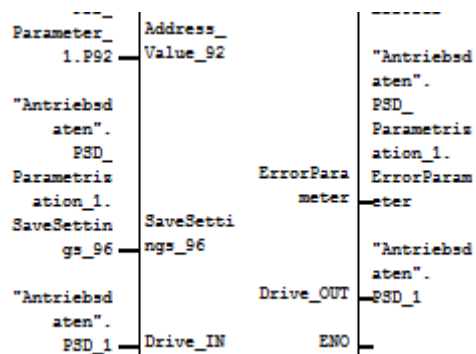
## 5.14 MC\_Parametrization (FB114)

Mit diesem FB können sämtliche Parameter des Antriebs geschrieben werden.

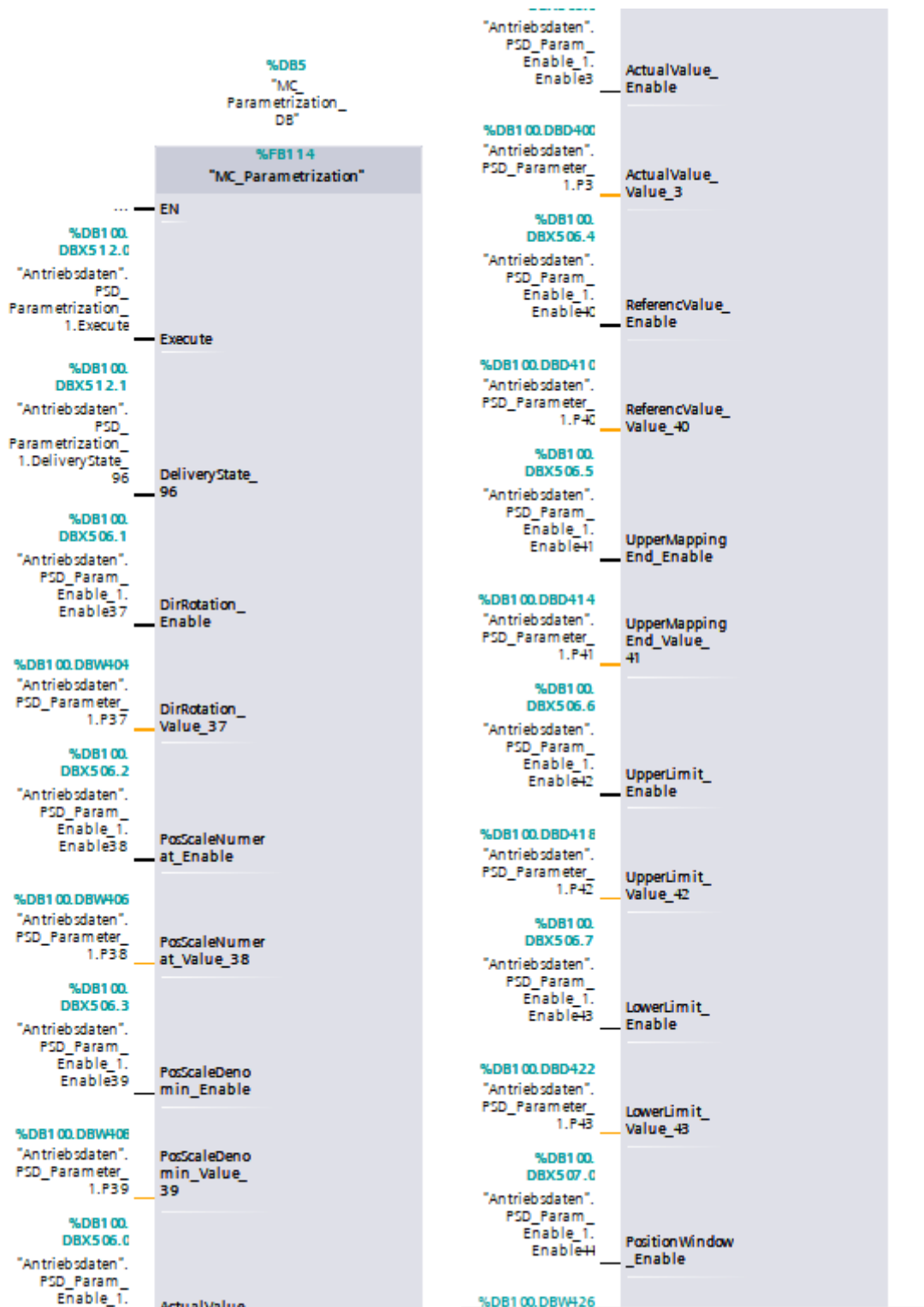
	"Instance_ MC_ Parametris at"	"MC_ Parametrisation"	"Antriebsd aten". PSD_ Parameter_ 1.P40	ReferencV alue_ Value_40	"Antriebsd aten". PSD_Param_ Enable_1. Enable53	TargetSpe ed_Enable
...	EN					
"Antriebsd aten". PSD_ Parametris ation_1. Execute	Execute		"Antriebsd aten". PSD_Param_ Enable_1. Enable41	UpperMapp ingEnd_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P53	TargetSpe ed_Value_ 53
"Antriebsd aten". PSD_ Parametris ation_1. DeliverySt ate_96	DeliveryS tate_96		"Antriebsd aten". PSD_ Parameter_ 1.P41	UpperMapp ingEnd_ Value_41	"Antriebsd aten". PSD_Param_ Enable_1. Enable56	TargSpeed ManRun_ Enable
"Antriebsd aten". PSD_Param_ Enable_1. Enable37	DirRotati on_Enable		"Antriebsd aten". PSD_ Parameter_ 1.P42	UpperLimi t_Enable	"Antriebsd aten". PSD_ Parameter_ 1.P56	TargSpeed ManRun_ Value_56
"Antriebsd aten". PSD_ Parameter_ 1.P37	DirRotati on_Value_ 37		"Antriebsd aten". PSD_Param_ Enable_1. Enable43	UpperLimi t_Value_ 42	"Antriebsd aten". PSD_ Parameter_ 1.P57	SpeedLimi tAbort_ Enable
"Antriebsd aten". PSD_Param_ Enable_1. Enable38	PosScaleN umerat_ Enable		"Antriebsd aten". PSD_ Parameter_ 1.P43	LowerLimi t_Value_ 43	"Antriebsd aten". PSD_ Parameter_ 1.P58	SpeedLimi tAbort_ Value_57
"Antriebsd aten". PSD_ Parameter_ 1.P38	PosScaleN umerat_ Value_38		"Antriebsd aten". PSD_Param_ Enable_1. Enable44	LowerLimi t_Value_ 43	"Antriebsd aten". PSD_ Parameter_ 1.P58	Accelerat ion_ Enable
"Antriebsd aten". PSD_Param_ Enable_1. Enable39	PosScaleD enomin_ Enable		"Antriebsd aten". PSD_ Parameter_ 1.P44	PositionW indow_ Value_44	"Antriebsd aten". PSD_ Parameter_ 1.P59	Accelerat ion_ Value_58
"Antriebsd aten". PSD_ Parameter_ 1.P39	PosScaleD enomin_ Value_39		"Antriebsd aten". PSD_Param_ Enable_1. Enable45	PositionW indow_ Value_44	"Antriebsd aten". PSD_ Parameter_ 1.P59	Decelerat ion_ Value_59
"Antriebsd aten". PSD_Param_ Enable_1. Enable3	ActualVal ue_Enable		"Antriebsd aten". PSD_ Parameter_ 1.P45	LoopLengh t_Enable	"Antriebsd aten". PSD_ Parameter_ 1.P63	Decelerat ion_ Value_59
"Antriebsd aten". PSD_ Parameter_ 1.P3	ActualVal ue_Value_ 3		"Antriebsd aten". PSD_ Parameter_ 1.P45	LoopLengh t_Value_ 45	"Antriebsd aten". PSD_ Parameter_ 1.P63	MaxStartT orque_ Enable
"Antriebsd aten". PSD_Param_ Enable_1. Enable40	ReferencV alue_ Enable		"Antriebsd aten". PSD_ Parameter_ 1.P47	LoopLengh t_Value_ 45	"Antriebsd aten". PSD_ Parameter_ 1.P63	MaxStartT orque_ Value_63
					"Antriebsd aten". PSD_ Parameter_ 1.P64	MaxTorque _Enable



"Antriebsd aten". PSD_ Parameter_ 1.P64	MaxTorque _Value_64	"Antriebsd aten". PSD_Param_ Enable_1. Enable80	GeneralPu rpose1_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P86	GeneralPu rpose7_ Value_86	
"Antriebsd aten". PSD_Param_ Enable_1. Enable65	MaxHoldTor EndRun_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P80	GeneralPu rpose1_ Value_80	"Antriebsd aten". PSD_Param_ Enable_1. Enable87	GeneralPu rpose8_ Enable	
"Antriebsd aten". PSD_ Parameter_ 1.P65	MaxHoldTor EndRun_ Value_65	"Antriebsd aten". PSD_Param_ Enable_1. Enable81	GeneralPu rpose2_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P87	GeneralPu rpose8_ Value_87	
"Antriebsd aten". PSD_Param_ Enable_1. Enable66	MaxHoldTo rque_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P81	GeneralPu rpose2_ Value_81	"Antriebsd aten". PSD_Param_ Enable_1. Enable88	GeneralPu rpose9_ Enable	
"Antriebsd aten". PSD_ Parameter_ 1.P66	MaxHoldTo rque_ Value_66	"Antriebsd aten". PSD_Param_ Enable_1. Enable82	GeneralPu rpose3_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P88	GeneralPu rpose9_ Value_88	
"Antriebsd aten". PSD_Param_ Enable_1. Enable70	AbortRunT ime_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P82	GeneralPu rpose3_ Value_82	"Antriebsd aten". PSD_Param_ Enable_1. Enable89	GeneralPu rpose10_ Enable	
"Antriebsd aten". PSD_ Parameter_ 1.P70	AbortRunT ime_ Value_70	"Antriebsd aten". PSD_Param_ Enable_1. Enable83	GeneralPu rpose4_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P89	GeneralPur pose10_ Value_89	
"Antriebsd aten". PSD_Param_ Enable_1. Enable71	StartTorq ueTime_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P83	GeneralPu rpose4_ Value_83	"Antriebsd aten". PSD_Param_ Enable_1. Enable90	MinVoltag e_Enable	
"Antriebsd aten". PSD_ Parameter_ 1.P71	StartTorq ueTime_ Value_71	"Antriebsd aten". PSD_Param_ Enable_1. Enable84	GeneralPu rpose5_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P90	MinVoltag e_Value_ 90	"Antriebsd aten". PSD_ Parametris ation_1.
"Antriebsd aten". PSD_Param_ Enable_1. Enable72	HoldTorTi EndRun_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P84	GeneralPu rpose5_ Value_84	"Antriebsd aten". PSD_Param_ Enable_1. Enable91	MaxTemper ature_ Enable	Active Active
"Antriebsd aten". PSD_ Parameter_ 1.P72	HoldTorTi EndRun_ Value_72	"Antriebsd aten". PSD_Param_ Enable_1. Enable85	GeneralPu rpose6_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P91	MaxTemper ature_ Value_91	Done Done
"Antriebsd aten". PSD_Param_ Enable_1. Enable76	MotVolFil tTime_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P85	GeneralPu rpose6_ Value_85	"Antriebsd aten". PSD_Param_ Enable_1. Enable92	Address_ Enable	Error Error
"Antriebsd aten". PSD_ Parameter_ 1.P76	MotVolFil tTime_ Value_76	"Antriebsd aten". PSD_Param_ Enable_1. Enable86	GeneralPu rpose7_ Enable	"Antriebsd aten". PSD_ Parameter_ 1.P92	Address_ Value_92	ErrorID ErrorID
						"Antriebsd aten". PSD_ Parametris ation_1.



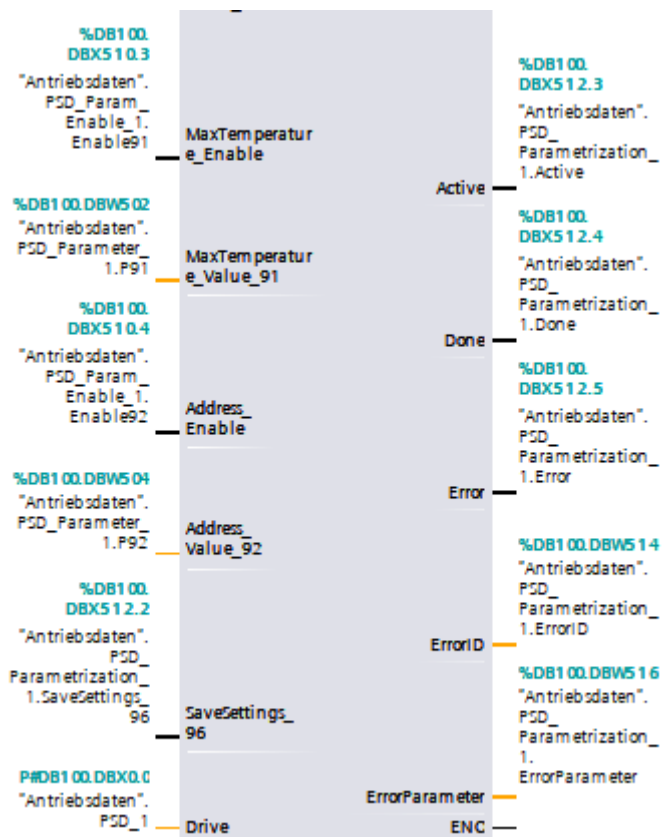
(Ansicht in Step 7 Classic V5.5)



Enable44	PositionWindow_Enable	%DB1 00.DBW426 "Antriebsdaten". PSD_Parameter_1.P44	PositionWindow_Value_44	%DB1 00.DBX5 07.6 "Antriebsdaten". PSD_Parameter_1.Enable58	Acceleration_Enable
%DB1 00.DBX5 07.1 "Antriebsdaten". PSD_Parameter_1.Enable45	LoopLength_Enable	%DB1 00.DBX5 07.2 "Antriebsdaten". PSD_Parameter_1.Enable47	Readjustment_Enable	%DB1 00.DBW440 "Antriebsdaten". PSD_Parameter_1.P58	Acceleration_Value_58
%DB1 00.DBX5 07.1 "Antriebsdaten". PSD_Parameter_1.P45	LoopLength_Value_45	%DB1 00.DBX5 08.0 "Antriebsdaten". PSD_Parameter_1.Enable63	MaxStartTorque_Enable	%DB1 00.DBX5 07.7 "Antriebsdaten". PSD_Parameter_1.Enable59	Deceleration_Enable
%DB1 00.DBW432 "Antriebsdaten". PSD_Parameter_1.P47	Readjustment_Value_47	%DB1 00.DBW442 "Antriebsdaten". PSD_Parameter_1.P59	Deceleration_Value_59	%DB1 00.DBX5 08.1 "Antriebsdaten". PSD_Parameter_1.Enable64	MaxTorque_Enable
%DB1 00.DBX5 07.3 "Antriebsdaten". PSD_Parameter_1.Enable53	TargetSpeed_Enable	%DB1 00.DBW444 "Antriebsdaten". PSD_Parameter_1.P63	MaxStartTorque_Value_63	%DB1 00.DBX5 08.2 "Antriebsdaten". PSD_Parameter_1.Enable65	MaxHoldTorEndRun_Enable
%DB1 00.DBW434 "Antriebsdaten". PSD_Parameter_1.P53	TargetSpeed_Value_53	%DB1 00.DBW446 "Antriebsdaten". PSD_Parameter_1.P64	MaxTorque_Value_64	%DB1 00.DBX5 08.3 "Antriebsdaten". PSD_Parameter_1.Enable66	MaxHoldTorque_Enable
%DB1 00.DBX5 07.4 "Antriebsdaten". PSD_Parameter_1.Enable56	TargSpeedManRun_Enable				
%DB1 00.DBW436 "Antriebsdaten". PSD_Parameter_1.P56	TargSpeedManRun_Value_56				
%DB1 00.DBX5 07.5 "Antriebsdaten". PSD_Parameter_1.Enable57	SpeedLimitAbort_Enable				
%DB1 00.DBW438 "Antriebsdaten". PSD_Parameter_1.P57	SpeedLimitAbort_Value_57				

Enable_1. Enable66	MaxHoldTorque _Enable	%DB1 00.DBD460 "Antriebsdaten". PSD_Parameter_ 1.P80	GeneralPurpos e1_Value_80
%DB1 00.DBW450 "Antriebsdaten". PSD_Parameter_ 1.P66	MaxHoldTorque _Value_66	%DB1 00. DBX509.1 "Antriebsdaten". PSD_Param_ Enable_1. Enable81	GeneralPurpos e2_Enable
%DB1 00. DBX508.4 "Antriebsdaten". PSD_Param_ Enable_1. Enable70	AbortRunTime_ Enable	%DB1 00.DBD464 "Antriebsdaten". PSD_Parameter_ 1.P81	GeneralPurpos e2_Value_81
%DB1 00.DBW452 "Antriebsdaten". PSD_Parameter_ 1.P70	AbortRunTime_ Value_70	%DB1 00. DBX509.2 "Antriebsdaten". PSD_Param_ Enable_1. Enable82	GeneralPurpos e3_Enable
%DB1 00. DBX508.5 "Antriebsdaten". PSD_Param_ Enable_1. Enable71	StartTorqueTim e_Enable	%DB1 00.DBD468 "Antriebsdaten". PSD_Parameter_ 1.P82	GeneralPurpos e3_Value_82
%DB1 00.DBW454 "Antriebsdaten". PSD_Parameter_ 1.P71	StartTorqueTim e_Value_71	%DB1 00. DBX509.3 "Antriebsdaten". PSD_Param_ Enable_1. Enable83	GeneralPurpos e4_Enable
%DB1 00. DBX508.6 "Antriebsdaten". PSD_Param_ Enable_1. Enable72	HoldTorTiEndRu n_Enable	%DB1 00.DBD472 "Antriebsdaten". PSD_Parameter_ 1.P83	GeneralPurpos e4_Value_83
%DB1 00.DBW456 "Antriebsdaten". PSD_Parameter_ 1.P72	HoldTorTiEndRu n_Value_72	%DB1 00. DBX509.4 "Antriebsdaten". PSD_Param_ Enable_1. Enable84	GeneralPurpos e5_Enable
%DB1 00. DBX508.7 "Antriebsdaten". PSD_Param_ Enable_1. Enable76	MotVolFiltTime_ Enable	%DB1 00.DBD476 "Antriebsdaten". PSD_Parameter_ 1.P84	GeneralPurpos e5_Value_84
%DB1 00.DBW458 "Antriebsdaten". PSD_Parameter_ 1.P76	MotVolFiltTime_ Value_76	%DB1 00. DBX509.5 "Antriebsdaten". PSD_Param_ Enable_1. Enable85	GeneralPurpos e6_Enable
%DB1 00. DBX509.0 "Antriebsdaten". PSD_Param_ Enable_1. Enable80	GeneralPurpos e1_Enable	%DB1 00.DBD480 "Antriebsdaten". PSD_Parameter_ 1.P85	GeneralPurpos e6_Value_85
%DB1 00.DBD460 "Antriebsdaten". PSD_Parameter_ 1.P80	GeneralPurpos e1_Value_80	%DB1 00. DBX509.6 "Antriebsdaten". PSD_Param_ Enable_1. Enable86	GeneralPurpos e7_Enable





(Ansicht in Step 7 TIA)

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 PSD-4\_\_-PN bzw. PSC-4\_\_-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

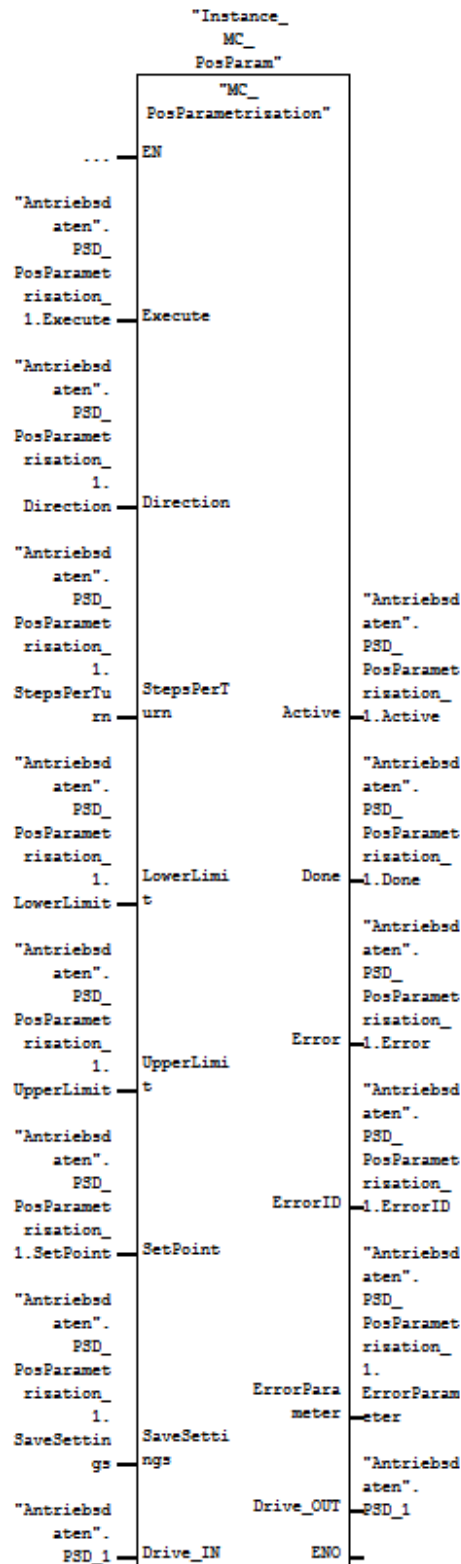
Falls es keinen Fehler gab, wird 0 ausgegeben.

**INFORMATION!**

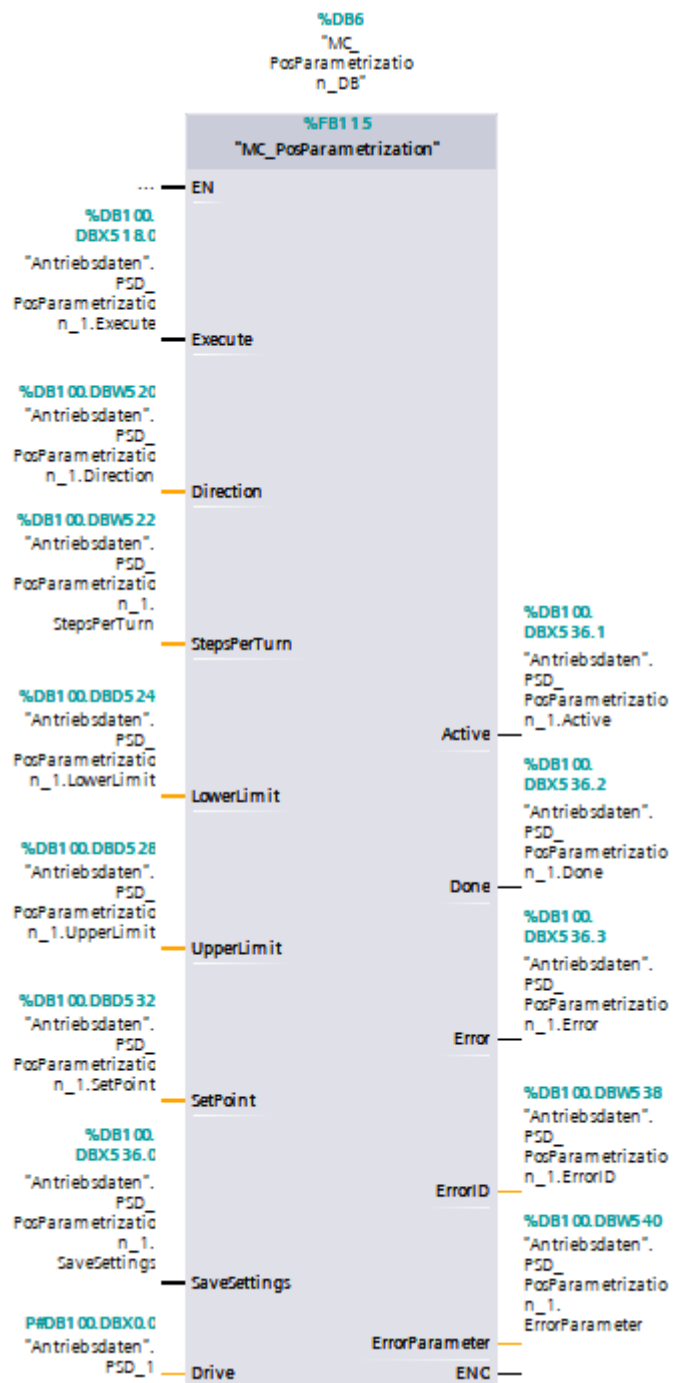
Im Falle der S7-300 muss bei den Bausteinen „MC\_Parametrization“ und „MC\_PosParametrization“ die Variable vom Typ „DRIVE\_DATA“ an den Eingang „Drive\_IN“ und an den Ausgang „Drive\_OUT“ angeschlossen werden (bei den S7-1200 und S7-1500 im TIA-Portal ist die Variable nur einmal an den Ein-/Ausgang „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)

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) =  $\text{SetPoint} + (3 \times \text{StepsPerTurn})$

sonst:

oberes Mapping-Ende (Par. 41) =  $\text{UpperLimit} + (3 \times \text{StepsPerTurn})$

- 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
$\text{StepsPerTurn} \geq 1$	16#6140	39
$\text{StepsPerTurn} \leq 10000$	16#6140	39
$\text{LowerLimit} \leq \text{UpperLimit}$	16#6140	42
$(\text{UpperLimit} - \text{LowerLimit}) / \text{StepsPerTurn} \leq a$	16#6140	43
Falls $\text{SetPoint} < \text{LowerLimit}$ : $(\text{UpperLimit} - \text{SetPoint}) / \text{StepsPerTurn} \leq a$	16#6140	3
Falls $\text{SetPoint} > \text{UpperLimit}$ : $(\text{SetPoint} - \text{LowerLimit}) / \text{StepsPerTurn} \leq a$	16#6140	3

Der Wert „a“ hängt dabei von der verwendeten Getriebevariante ab:

Getriebevariante	PSD401/411, PSD422/432	PSD403/413	PSD426/436	PSD428/438
Getriebevariante	PSC401/411, PSC422/432	PSC402/412	PSC426/436	PSC428/438
Wert a	4026	986	1274	977

- 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).