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

Dokument 7100.005404B Version 1.1 4/2017

Inhaltsverzeichnis

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.

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.

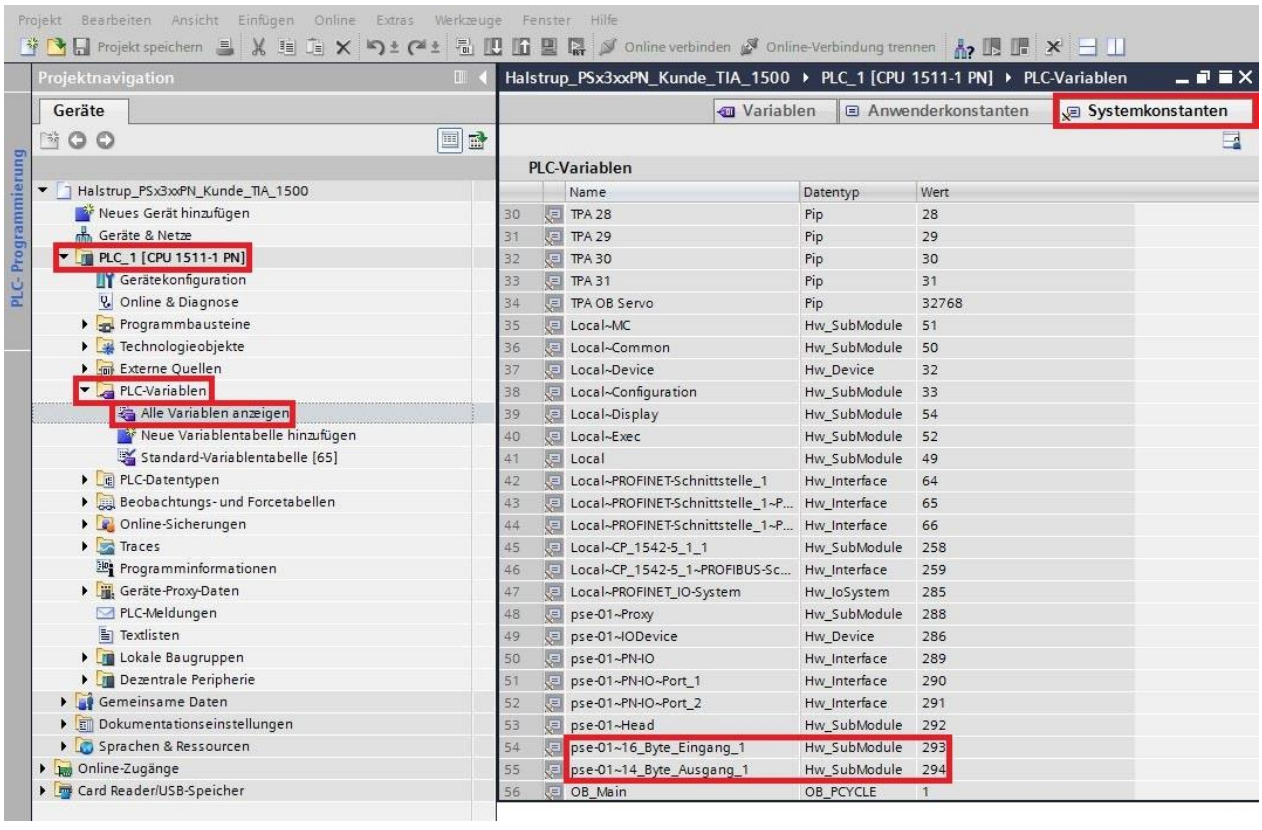
2 Datenstruktur DRIVE_DATA

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	Benutzer	Adresse Prozessdaten Slave → Master
PdAddressOut	INT	Benutzer	Adresse Prozessdaten Master → Slave
Name	STRING[16]	Benutzer (optional)	Name der Achse
Beschreibung	STRING[32]	Benutzer (optional)	Beschreibung (z.B. Funktion, Aufgabe dieser Achse)
State	DINT	Funktionsbausteine	Actual state
Private	STRUCT	Funktionsbausteine	Datenstruktur zur internen Verwendung
Nur in den Versionen für das TIA-Portal:			
DB_LIB_IN/OUT	DT_LIB_IN/OUT	Funktionsbausteine	Datenstruktur zur internen Verwendung

In den Versionen für das TIA-Portal beinhaltet die Struktur DRIVE_DATA zusätzlich einen Parameter der Struktur „DT_LIB_IN/OUT“, daher muss diese Struktur aus der Bibliothek auch in das Projekt übernommen werden.

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

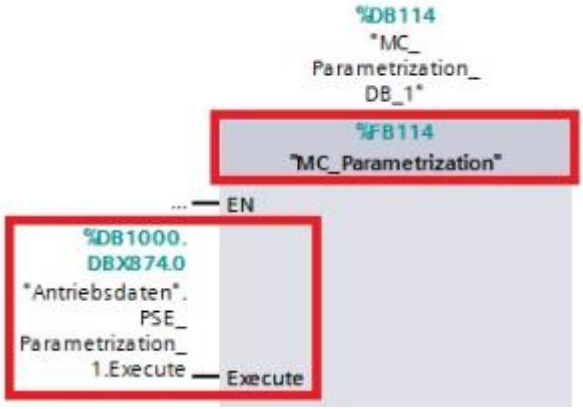


(Adressen der Prozessdaten in Step 7 TIA)

3 Datenbaustein „Antriebsdaten“

In den Versionen für das TIA-Portal dient dieser Datenbaustein als Vorlage für die Anbindung an die Funktionsbausteine. Der Datenbaustein stellt die notwendigen Veriablen 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_PositionParametrization
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
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

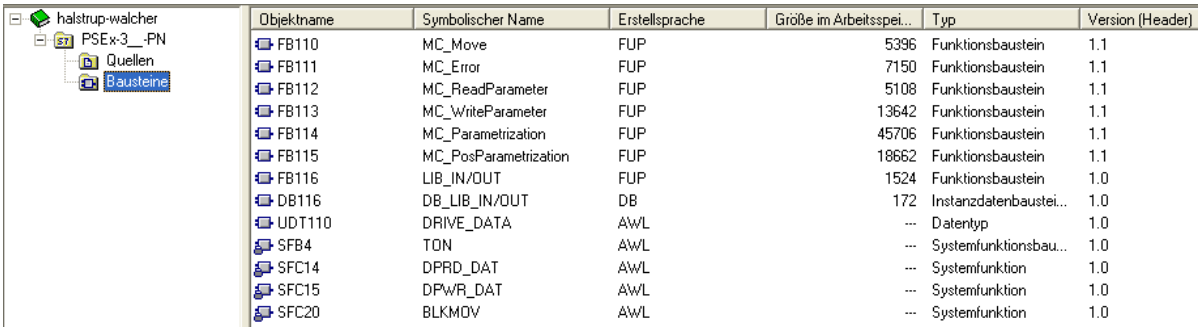
Zunächst müssen die Bausteine in ein eigenes Anwenderprojekt eingebunden werden. Dies geschieht durch Öffnen der gewünschten Bibliothek und Kopieren der gewünschten Funktionsbausteine.

Folgende Bibliotheken stehen zur Auswahl:

- „halstrup-walcher“
  → für Step 7 Classic V5.5
- „library_PSx3xxPN_TIA13SP1_300
  → für CPU S7-300 im TIA-Portal (V13 SP1)
- „library_PSx3xxPN_TIA14_300
  → für CPU S7-300 im TIA-Portal (V14)
- „library_PSx3xxPN_TIA13SP1_1200-1500
  → für CPU S7-1200/1500 im TIA-Portal (V13 SP1)
- „library_PSx3xxPN_TIA14_1200-1500
  → für CPU S7-1200/1500 im TIA-Portal (V14)

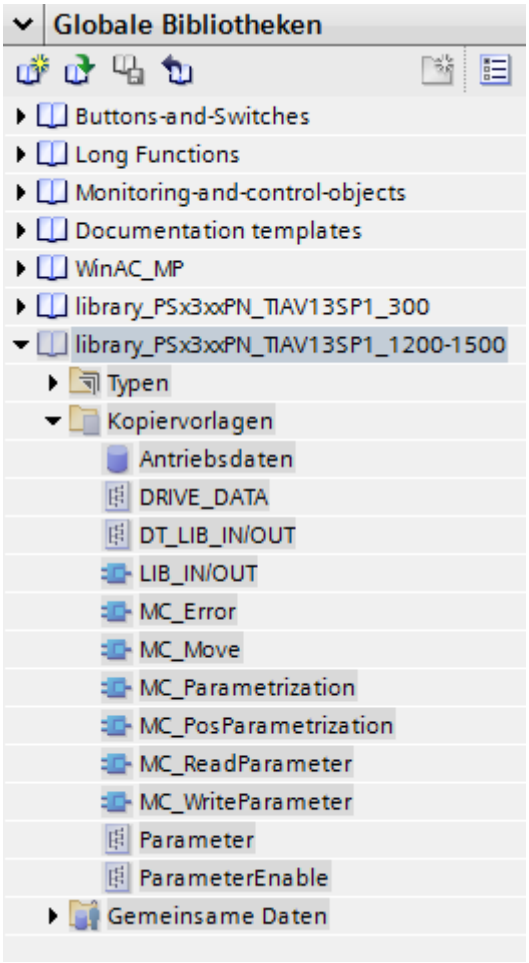
5.1 Öffnen der Bibliothek

Nach dem Öffnen der Bibliothek präsentieren sich die Funktionsbausteine folgendermaßen:



Objektname	Symbolischer Name	Erstellsprache	Größe im Arbeitsspei...	Typ	Version (Header)
FB110	MC_Move	FUP	5396	Funktionsbaustein	1.1
FB111	MC_Error	FUP	7150	Funktionsbaustein	1.1
FB112	MC_ReadParameter	FUP	5108	Funktionsbaustein	1.1
FB113	MC_WriteParameter	FUP	13642	Funktionsbaustein	1.1
FB114	MC_Parametrization	FUP	45706	Funktionsbaustein	1.1
FB115	MC_PosParametrization	FUP	18662	Funktionsbaustein	1.1
FB116	LIB_IN/OUT	FUP	1524	Funktionsbaustein	1.0
DB116	DB_LIB_IN/OUT	DB	172	Instanzdatenbaustei...	1.0
UDT110	DRIVE_DATA	AWL	---	Datentyp	1.0
SFB4	TON	AWL	---	Systemfunktionsbau...	1.0
SFC14	DPRD_DAT	AWL	---	Systemfunktion	1.0
SFC15	DPWR_DAT	AWL	---	Systemfunktion	1.0
SFC20	BLKMOV	AWL	---	Systemfunktion	1.0

(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:

<u>bei Verwendung der Bibliotheksversion für Step 7 Classic V5.5</u>:

In „S7-Programm → Bausteine“ kopieren:

- FB116 („LIB IN/OUT“)
- DB116 („DB_LIB_IN/OUT“)
- UDT110 („DRIVE_DATA“)
- SFB4 („TON“)
- SFC14 („DPRD_DAT“)
- SFC15 („DPWR_DAT“)
- SFC20 („BLK_MOV“)

bei Verwendung einer der Bibliotheksversionen für Step 7 TIA V13 + V14:

In „Programmbausteine“ der gewünschten CPU kopieren:

- Datentyp DRIVE_DATA
- Datentyp DT_LIB_IN/OUT
- Funktionsblock LIB_IN/OUT

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

Bei den Versionen für das TIA-Portal kann zusätzlich 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: 94 Byte in Classic Step 7, 144 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. DB1000), 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:

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

In „S7-Programm → Bausteine“ kopieren:

- FB116 („LIB IN/OUT“)
- DB116 („DB_LIB_IN/OUT“)
- UDT110 („DRIVE_DATA“)
- SFB4 („TON“)
- SFC14 („DPRD_DAT“)
- SFC15 („DPWR_DAT“)
- SFC20 („BLK_MOV“)
- FB110 („MC_Move“)
- FB111 („MC_Error“)
- FB112 („MC_ReadParameter“)
- FB113 („MC_WriteParameter“)

bei Verwendung einer der Bibliotheksversionen für Step 7 TIA V13 + V14:

In „Programmbausteine“ der gewünschten CPU kopieren:

- Datentyp DRIVE_DATA
- Datentyp DT_LIB_IN/OUT
- Funktionsblock LIB_IN/OUT
- FB110 („MC_Move“)
- FB111 („MC_Error“)
- FB112 („MC_ReadParameter“)
- FB113 („MC_WriteParameter“)

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“,
- …

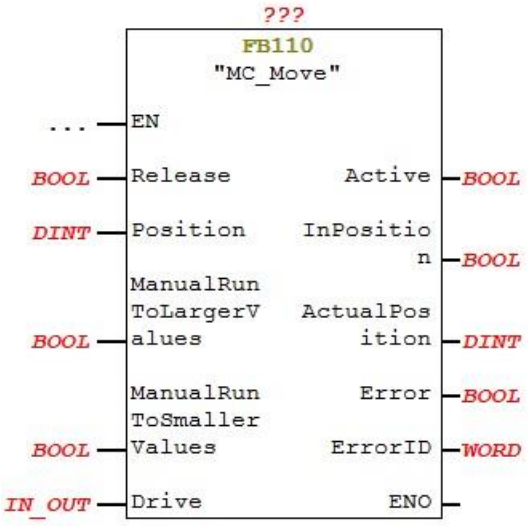
Nun sind alle Vorbereitungen getroffen, um die Bausteine zu verwenden. Im Editor „Bausteine programmieren“ erscheinen die Bausteine „MC_...“ im Abschnitt „FB Bausteine“. In unserem Beispiel werden also in einem Programmteil, der zyklisch aufgerufen wird (z.B. im OB1), die Funktionsblöcke MC_Move, MC_Error, MC_ReadParameter und MC_WriteParameter jeweils drei mal eingefügt und sofort mit ihren jeweiligen Instanzdatenbausteinen verknüpft.

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“).

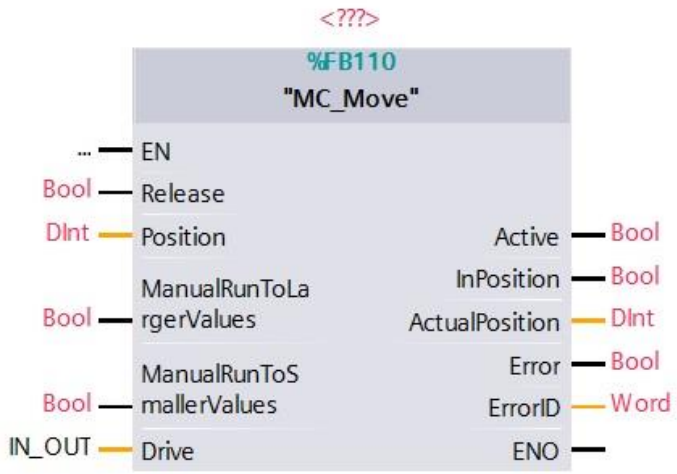
Danach verbinden wir die restlichen benötigten Ein- und Ausgänge mit ihren dafür vorgesehenen globalen Variablen (also „PSE_1_Move_DINT_Sollposition“ usw.).

5.8 MC_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.

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
- ein Kommunikationsfehler auftritt

InPosition
Sollposition erreicht

- Typ: BOOL
- Art: OUTPUT

Dieser Ausgang ist eine Abbildung des Statusbits „Sollposition erreicht“. Falls ein Kommunikationsfehler auftritt, wir 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.

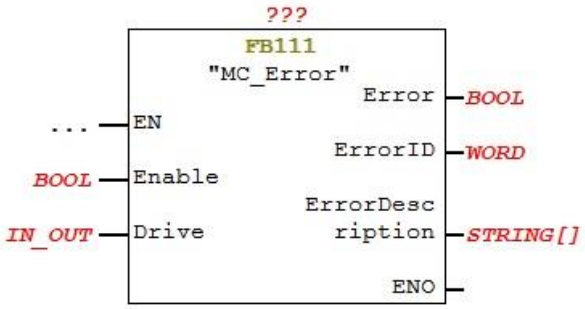
Error und ErrorID von MC_Move werden stets aktualisiert – auch dann, wenn der Eingang „Release“ nicht gesetzt ist.

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

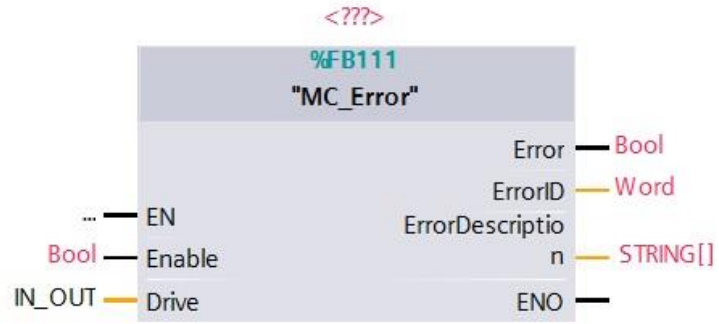
ErrorID	Beschreibung
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x3xx	Invalid PD output address
16#x4xx	Error while reading PD
16#x5xx	Error while writing PD
16#xxx2	Under- or overvoltage motor supply
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx8	Incorrect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

5.9 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

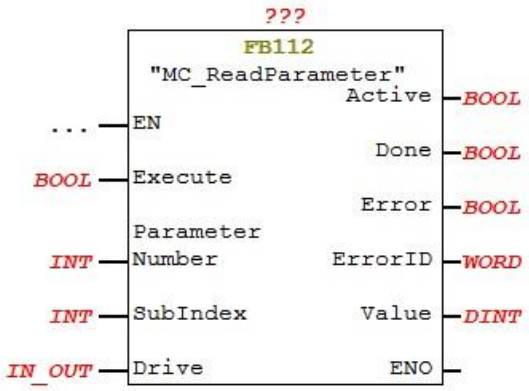
- Typ: STRING
- Defaultwert: „“
- Art: OUTPUT

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

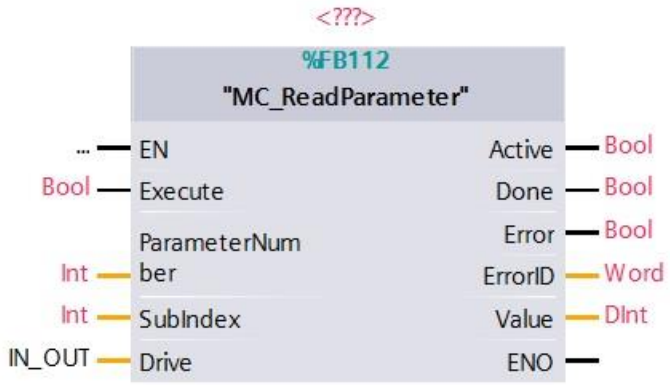
ErrorID	ErrorDescription
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x4xx	Error while reading PD
16#xxx2	Under- or overvoltage motor supply
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run
16#xxx6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

5.10 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.

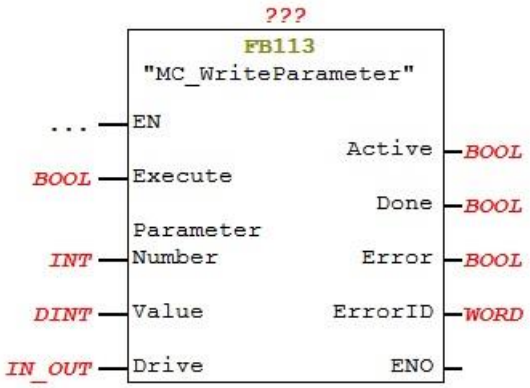
Value
Istwert des ausgelesenen Parameters

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

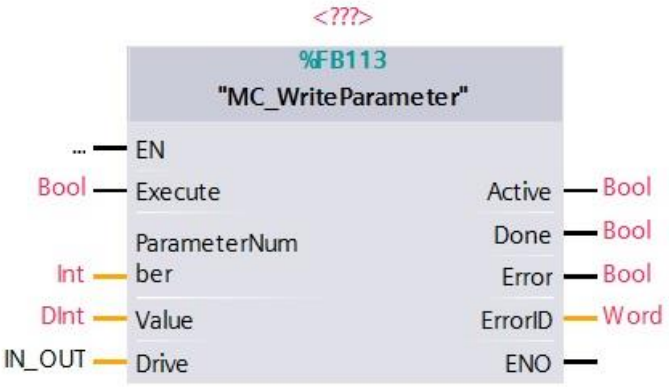
Bei einem Fehler wird 0 ausgegeben.

5.11 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.

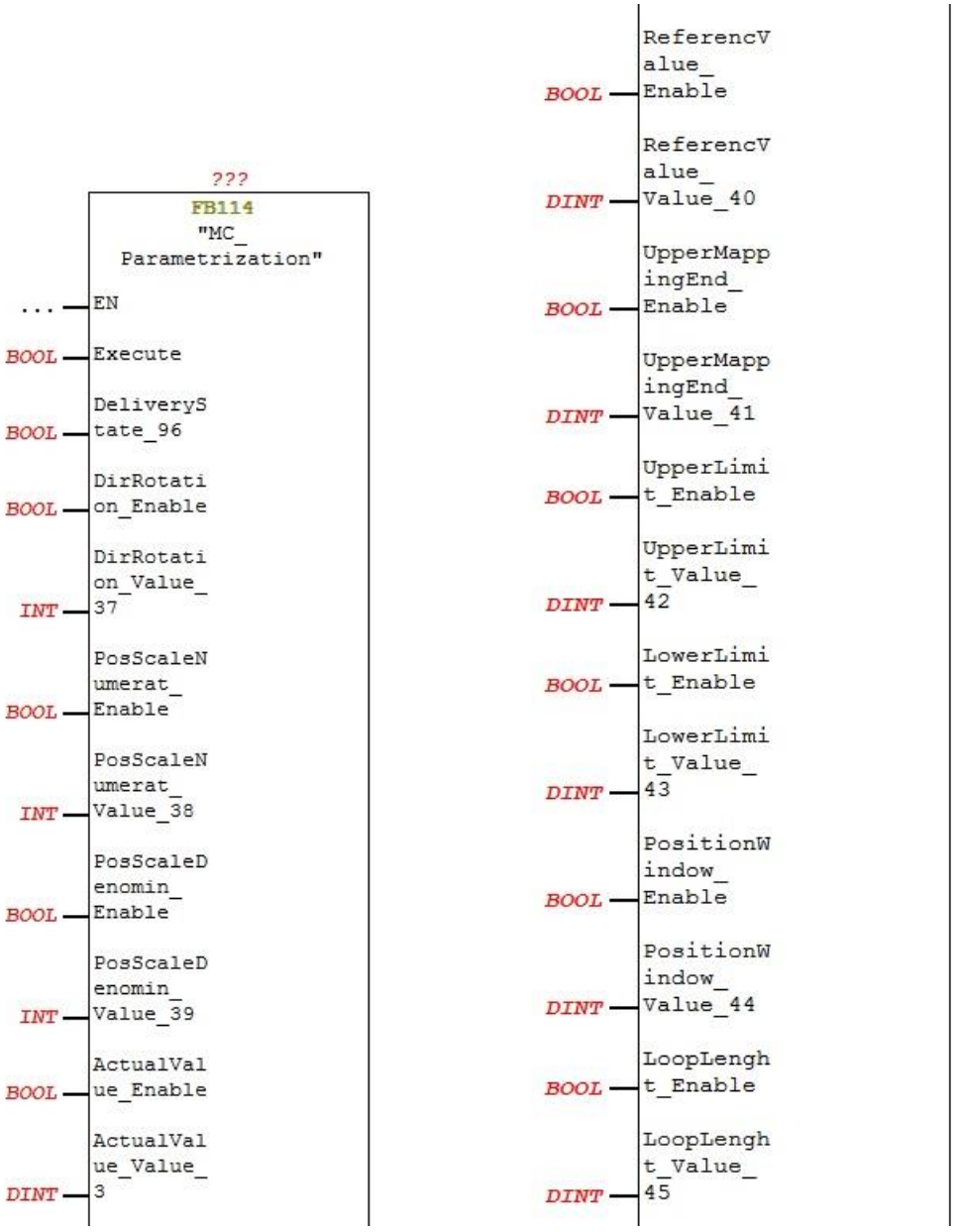
Error
Bit ist gesetzt, wenn während der Ausführung des FBs ein Fehler aufgetreten ist
- Typ: BOOL
- Defaultwert: FALSE
- Art: OUTPUT

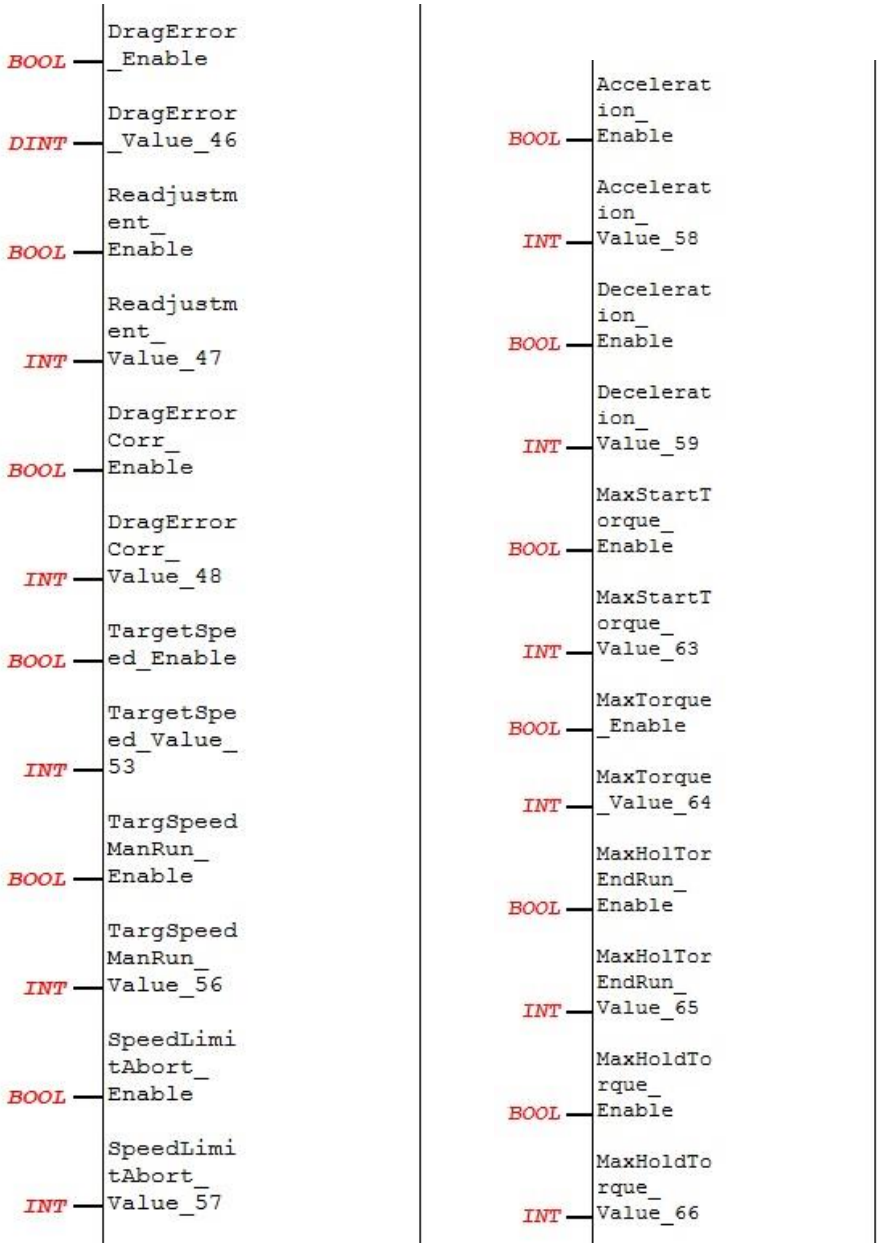
ErrorID
Fehler-ID (siehe Tabelle „ErrorID“ in Kap. 4)
- Typ: WORD
- Defaultwert: 0
- Art: OUTPUT

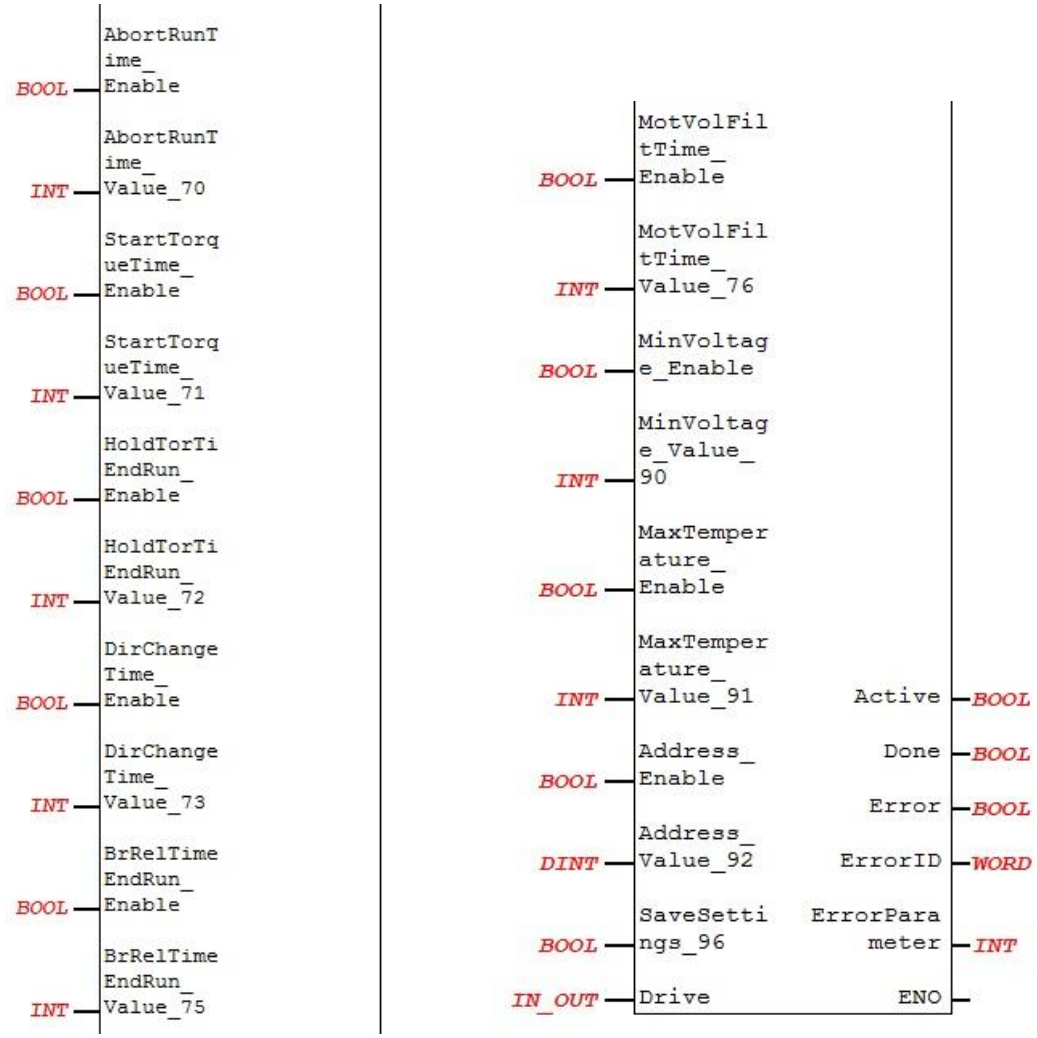
Antriebsfehler („Drive errors“) werden beim Schreiben eines Parameters nicht beachtet.

5.12 MC_Parametrization (FB114)

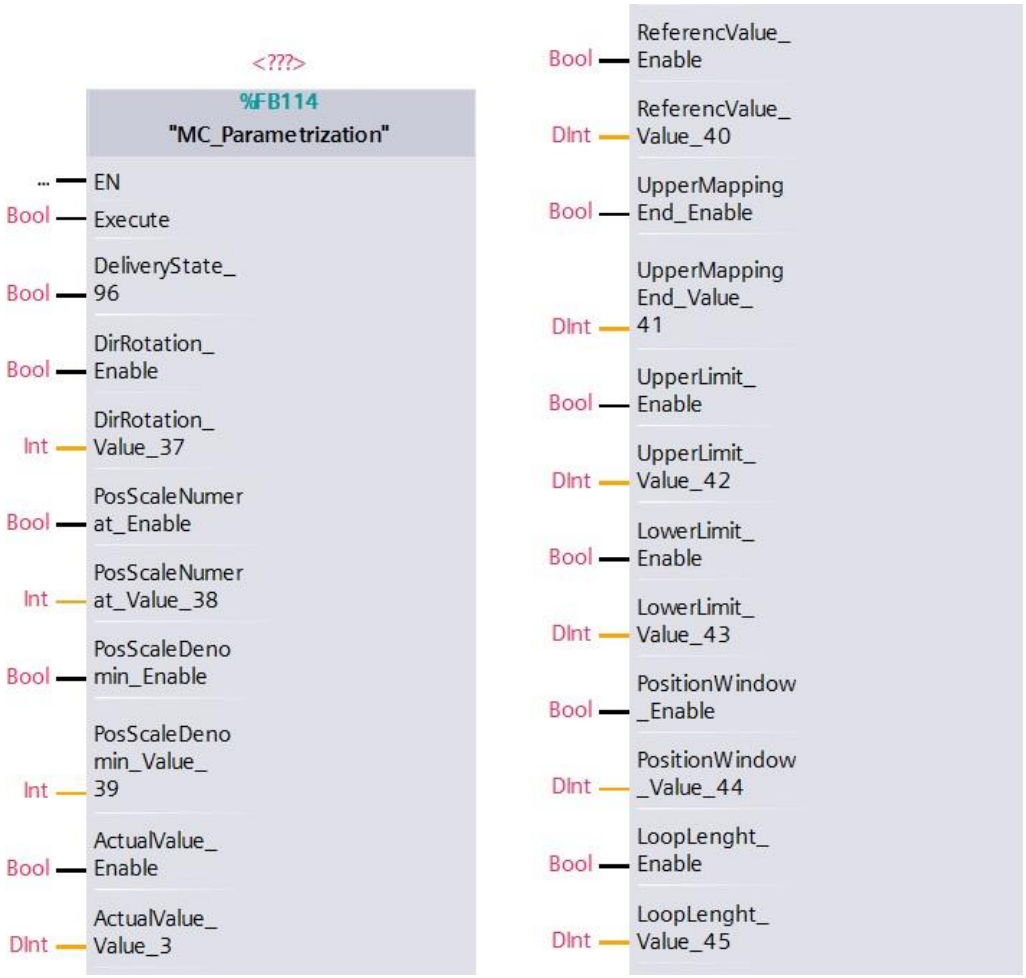
Mit diesem FB können sämtliche Parameter des Antriebs geschrieben werden.

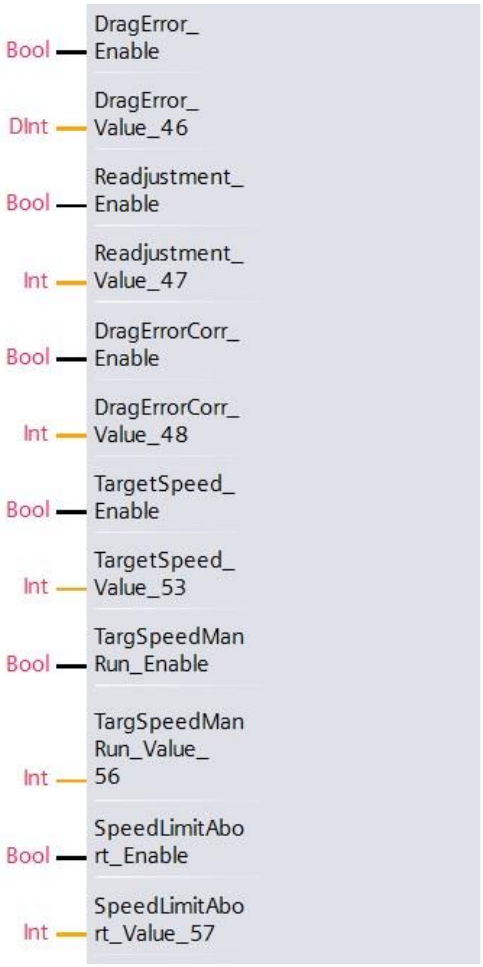


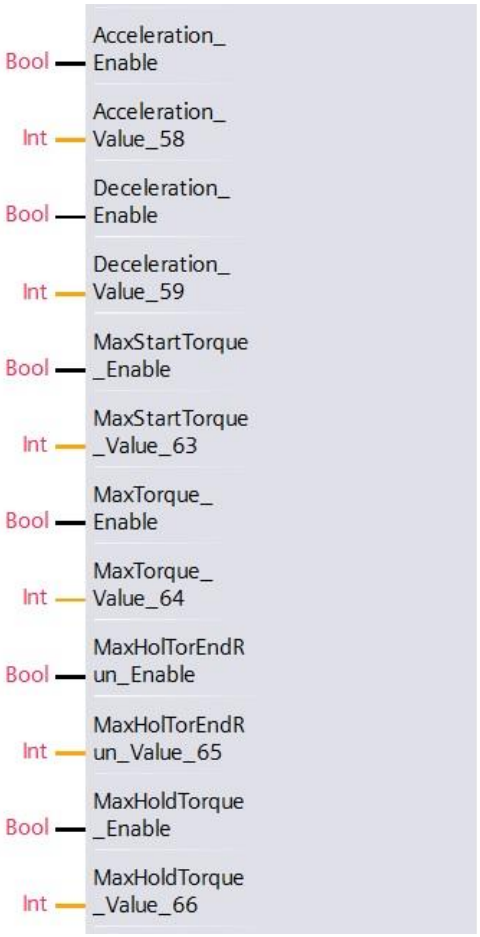

BOOL DragError_Enable
DINT DragError_Value_46
BOOL Readjustment_Enable
INT Readjustment_Value_47
BOOL DragErrorCorr_Enable
INT DragErrorCorr_Value_48
BOOL TargetSpeed_Enable
INT TargetSpeed_Value_53
BOOL TargSpeedManRun_Enable
INT TargSpeedManRun_Value_56
BOOL SpeedLimitAbort_Enable
INT SpeedLimitAbort_Value_57
BOOL Acceleration_Enable
INT Acceleration_Value_58
BOOL Deceleration_Enable
INT Deceleration_Value_59
BOOL MaxStartTorque_Enable
INT MaxStartTorque_Value_63
BOOL MaxTorque_Enable
INT MaxTorque_Value_64
BOOL MaxHolTorEndRun_Enable
INT MaxHolTorEndRun_Value_65
BOOL MaxHoldTorque_Enable
INT MaxHoldTorque_Value_66

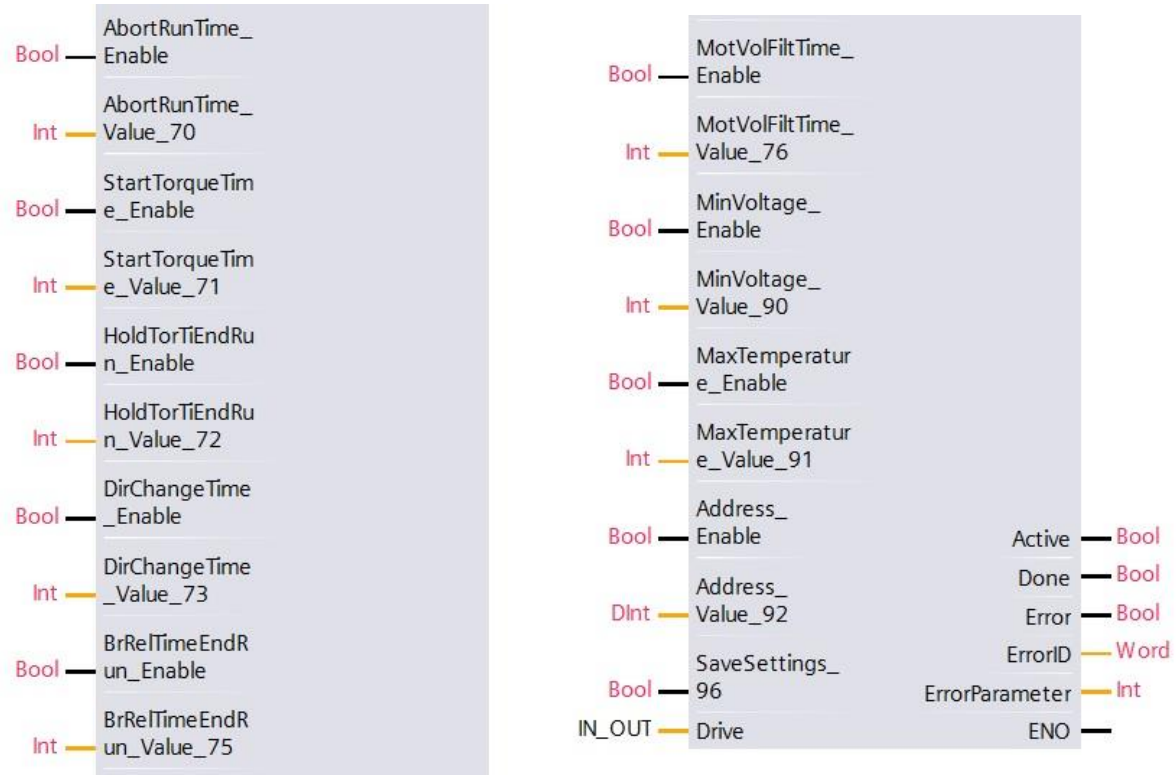


(Ansicht in Step 7 Classic V5.5)


<???>
%FB114
"MC_Parametrization"
... EN
Bool Execute
Bool DeliveryState_96
Bool DirRotation_Enable
Int DirRotation_Value_37
Bool PosScaleNumerat_Enable
Int PosScaleNumerat_Value_38
Bool PosScaleDenomin_Enable
Int PosScaleDenomin_Value_39
Bool ActualValue_Enable
DInt ActualValue_Value_3
Bool ReferencValue_Enable
DInt ReferencValue_Value_40
Bool UpperMappingEnd_Enable
DInt UpperMappingEnd_Value_41
Bool UpperLimit_Enable
DInt UpperLimit_Value_42
Bool LowerLimit_Enable
DInt LowerLimit_Value_43
Bool PositionWindow_Enable
DInt PositionWindow_Value_44
Bool LoopLenght_Enable
DInt LoopLenght_Value_45


Bool
DragError_ Enable
DInt
DragError_ Value_46
Bool
Readjustment_ Enable
Int
Readjustment_ Value_47
Bool
DragErrorCorr_ Enable
Int
DragErrorCorr_ Value_48
Bool
TargetSpeed_ Enable
Int
TargetSpeed_ Value_53
Bool
TargSpeedMan Run_Enable
Int
TargSpeedMan Run_Value_ 56
Bool
SpeedLimitAbo rt_Enable
Int
SpeedLimitAbo rt_Value_57


Bool
Acceleration_ Enable
Int
Acceleration_ Value_58
Bool
Deceleration_ Enable
Int
Deceleration_ Value_59
Bool
MaxStartTorque _Enable
Int
MaxStartTorque _Value_63
Bool
MaxTorque_ Enable
Int
MaxTorque_ Value_64
Bool
MaxHolTorEndR un_Enable
Int
MaxHolTorEndR un_Value_65
Bool
MaxHoldTorque _Enable
Int
MaxHoldTorque _Value_66



(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 Parameterierung erfolgreich beendet wurde oder ein Fehler aufgetreten ist.

Done
Bit ist gesetzt, sobald die Parameterierung 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.

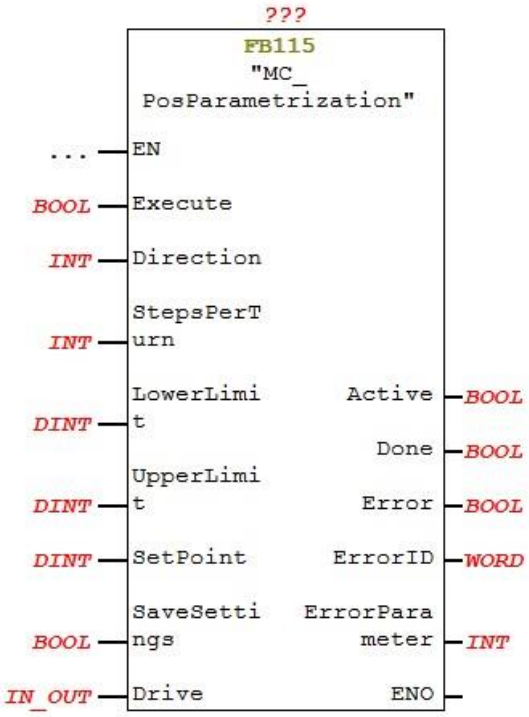
ErrorParameter
Parameternummer, bei dem im Fall eines Fehlers der Fehler aufgetreten ist

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

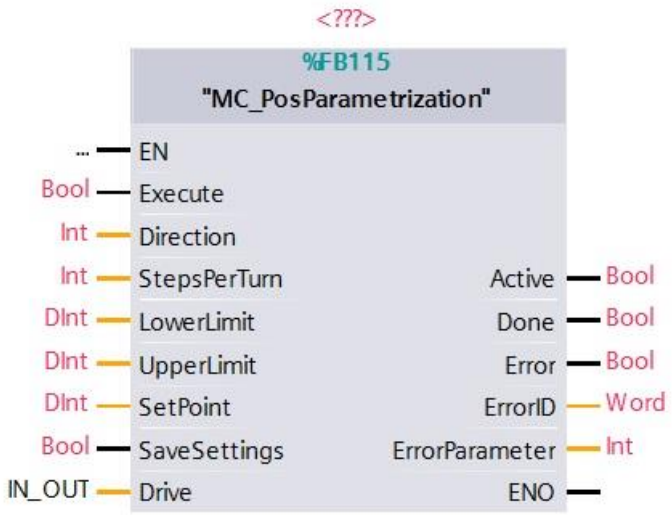
Falls es keinen Fehler gab, wird 0 ausgegeben.

5.13 MC_PositionParametrization (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 ≥ 1	16#6140	39
StepsPerTurn ≤ 10000	16#6140	39
LowerLimit ≤ UpperLimit	16#6140	42
(UpperLimit - LowerLimit) / StepsPerTurn ≤ 250	16#6140	43
Falls SetPoint < LowerLimit: (UpperLimit - SetPoint) / StepsPerTurn ≤ 250	16#6140	3
Falls SetPoint > UpperLimit: (SetPoint - LowerLimit) / StepsPerTurn ≤ 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 Parameterierung erfolgreich beendet wurde oder ein Fehler aufgetreten ist.

Done
Bit ist gesetzt, sobald die Parameterierung 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.