

Application Note Beispielprogramm Simple Motion PN für 1200/1500

Thema

Application Note

Dokumententyp

Sigma-7

Produkt

Datum	Version	Autor	Änderungen
22.10.2020	Initial	Küster Chris	-
Einschränkungen und Anwendungen			
Sigma-7 200V - PROFINET, Sigma-7 400V - PROFINET, Servopack Firmware 002A oder höher, SigmaWin+ V2.7 oder höher			
SIEMENS PLC 12xx Firmware V4.0 oder höher, SIEMENS PLC 15xx FW V2.5 oder höher, TIA Portal V15.0 oder höher			
Angewandte Dateien			
"Demo_TIA_1x00_V0000.zip"			

Inhaltsverzeichnis

1	Generell	4
1.1	Zweck dieses Dokuments	4
1.2	Voraussetzungen	4
1.2.1	TIA Portal	4
1.2.1.1	PLC	4
1.2.1.2	Sigma-7	5
1.2.2	SigmaWin+	6
1.2.2.1	Telegramm	6
1.2.2.2	Optionale Einstellungen	7
2	Hardwarekonfiguration	8
3	Software	9
3.1	Hierarchie	9
3.2	Datenspeicherung	9
3.3	FB1 – CallAxis	11
3.4	FB2 – Multi-Punkt-Positionierung	12
3.4.1	Layout	12
3.4.2	Inputs	13
3.4.3	Outputs	13
3.4.4	Benutzung	14
3.5	FBX – 2-Speed-Velocity-Block	15
3.5.1	Layout	15
3.5.2	Inputs	16
3.5.3	Outputs	16
3.5.4	Benutzung	17
3.5.4.1	Schnelle und langsame Geschwindigkeit	17
3.5.4.2	Homing	17
3.6	Variablen	18

Bilderverzeichnis

Bild 1: Ändern der IP-Adresse

Bild 2: Aktivieren des Clock memory bytes

Bild 3: Zuweisen eines Gerätenamens

Bild 4: USB-Type Mini und Type A

Bild 5: Verbindung zum Servopack

Bild 6: Unterstützte Telegramme im SIGMA-7

Bild 7: Ausgewähltes Telegramm 100

Bild 8: Position, Geschwindigkeit und Beschleunigung Numerator (Zähler) und Denominator (Nenner)

Bild 9: PROFINET -Kommunikation ist vorhanden

Bild 10: Hardwareidentifikation

Bild 11: Hierarchie des Programms

Bild 12: In- und Outputs von FB1

Bild 13: FB1 mit allen verbundenen In- und Outputs

Bild 14: Layout von FB2 – Multi-Punkt-Positionierung

Bild 15: Layout FB2 – 2-Geschwindigkeiten-Block

1 Generell

1.1 Zweck dieses Dokuments

Dieses Dokument erläutert die Verwendung des Beispielprojektes für die neue YASKAWA Motion Library für die Siemens SPS 1200 und 1500.

Sollten Fehler an dem Antrieb oder Regler auftreten, schauen Sie bitte in der dazugehörigen Dokumentation nach. Dies gilt auch für etwaige Fehler der Bausteinbibliothek Simple Motion. Beide Benutzerhandbücher finden Sie auf der Website <https://www.yaskawa.de/services/dokumenten-download-center>.

1.2 Voraussetzungen

Innerhalb dieses Kapitels werden die Voraussetzungen für die einzelnen PLC-Typen und den Sigma-7 beschrieben.

1.2.1 TIA Portal

1.2.1.1 PLC

IP-Adresse

Unabhängig vom Typ der PLC, welche für das Projekt genutzt wird, muss als erster Schritt die IP-Adresse angepasst und geändert werden, entsprechend ihrem Netzwerk

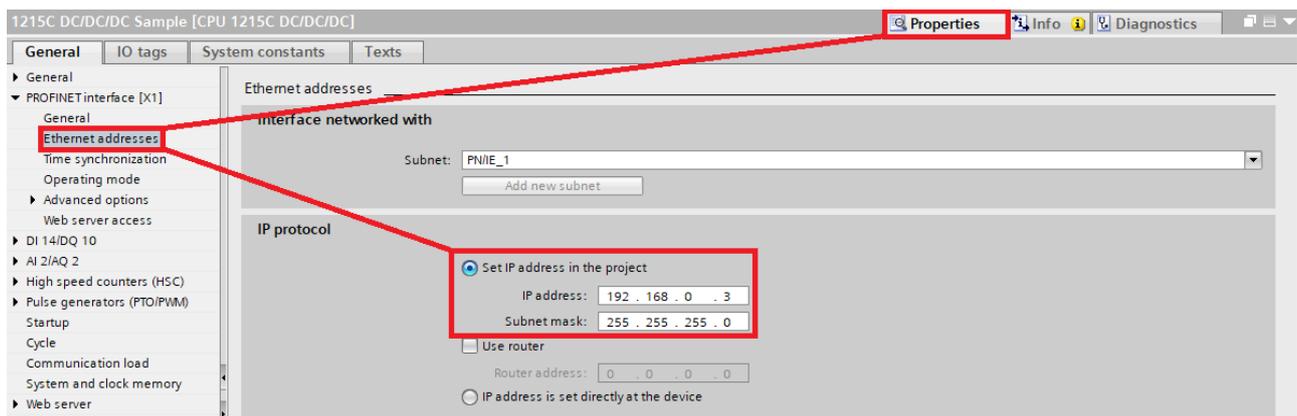


Bild 1: Ändern der IP-Adresse

Uhrzeit und Timer

In diesen Beispiel-Projekten wird ein Uhrzeit-Bit genutzt. Daher muss diese Einstellung in der Hardware-Konfiguration gesetzt werden. Diese Bits wechseln automatisch ihren Status in einem vordefinierten Zeitraum (siehe Bild 2)

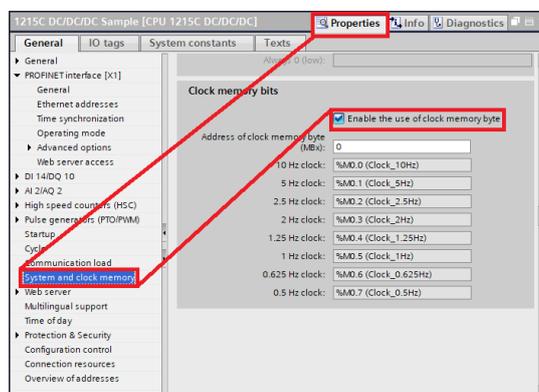


Bild 2: Aktivieren des Clock memory bytes

1.2.1.2 Sigma-7

Um eine korrekte Verbindung zwischen der SPS und dem Sigma-7 Antrieb via PROFINET herzustellen, müssen die Namen der einzelnen Busteilnehmer identisch mit den Namen in der Hardwarekonfiguration sein. Die aktuell vergebenen Namen kann man sich in den online erreichbaren Teilnehmern anzeigen lassen und gegebenen falls ändern wie in Bild 3 beschrieben.

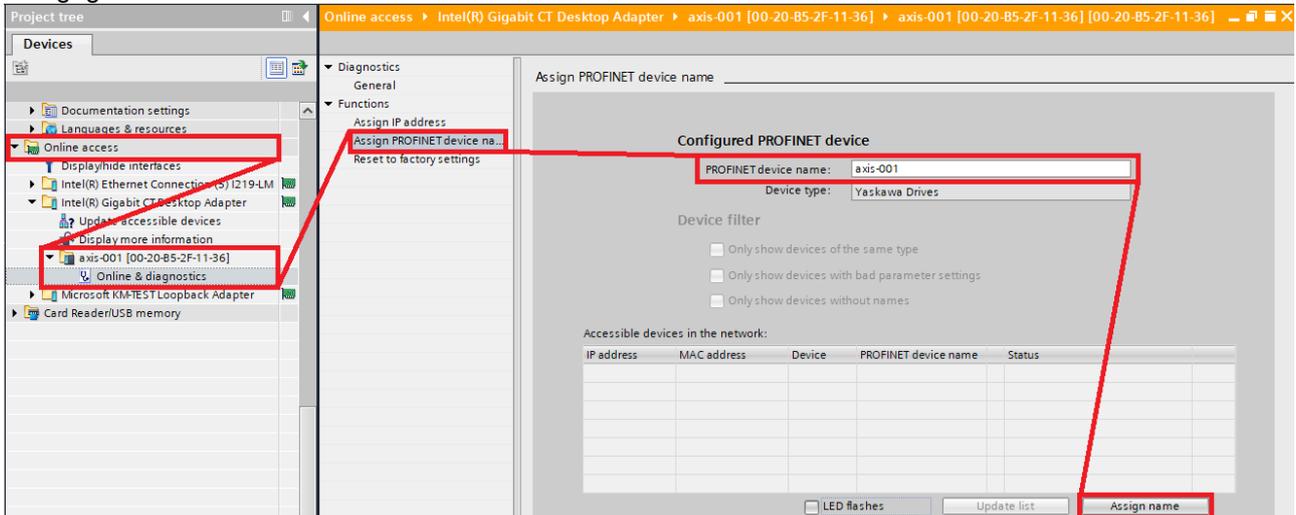


Bild 3: Zuweisen eines Gerätenamens

1.2.2 SigmaWin+

Mit dem YASKAWA-Tool SigmaWin+ müssen einige Bedingungen im Sigma-7 Antrieb eingestellt werden. Die Verbindung zwischen dem Tool und dem Sigma-7 wird mit einem USB-Kabel Typ Mini-B (Antriebsseitig) und USB-Typ A (Computerseitig) hergestellt. Hier wird empfohlen das Verbindungskabel von Yaskawa zu verwenden. Dies hat die Bezeichnung JZSP-CVS06-02-E.



Bild 4: USB-Type Mini und Type A

Öffnen Sie das Tool, wählen Sie die Verbindungsmethode USB und suchen Sie nach Servopacks. Ihr Gerät wird daraufhin im nächsten Bildschirm angezeigt. Wählen Sie dies aus und stellen Sie eine Verbindung her.

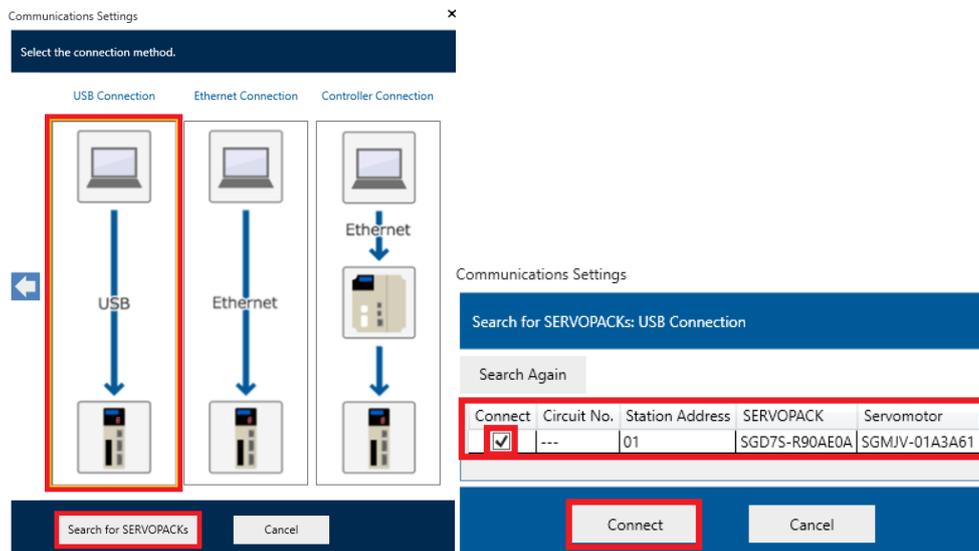


Bild 5: Verbindung zum Servopack

1.2.2.1 Telegramm

Wie in der Hardware-Konfiguration festgelegt, kommuniziert der SIGMA-7 Antrieb über Telegramm 100 mit der PLC. Die Telegramm-Art muss im SIGMA-7 eingestellt werden über den Parameter **PnC90(922)**. Beim Sigma-5 wird diese Einstellung an der Optionskarte vorgenommen mit dem Drehschalter S11. Folgende weitere Telegramme werden ebenfalls vom SIGMA-7 unterstützt:

- 1 : Standard Telegram 1: PROFIDRIVE Velocity Mode
- 2 : Standard Telegram 2: PROFIDRIVE Velocity Mode
- 7 : Standard Telegram 7: PROFIDRIVE Position Mode (Program Submode)
- 9 : Standard Telegram 9: PROFIDRIVE Position Mode (Program + MDI Submode)
- 100 : General Telegram: All Operation Modes**
- 999 : Free Telegram Configuration

Bild 6: Unterstützte Telegramme im SIGMA-7

PnC20(922)	Telegram Selection	-	100 : General Telegram: All Operation Modes
------------	--------------------	---	---

Bild 7: Ausgewähltes Telegramm 100

1.2.2.2 Optionale Einstellungen

In Telegramm 100 ist eine azyklische Kommunikation möglich und auch im Handbuch beschrieben. Diese Funktion ist im Beispiel-Programm aber nicht programmiert. Dies bedeutet, dass die Übersetzungen und Getriebe-Einstellungen händisch über SigmaWin+ in den unten aufgezeigten Parametern eingestellt werden müssen.

No.	Name	Unit	Axis A
PnB02(2301h:00)	Position User Unit : Numerator	–	1
PnB04(2301h:01)	Position User Unit : Denominator	–	1
PnB06(2302h:00)	Velocity User Unit : Numerator	–	1
PnB08(2302h:01)	Velocity User Unit : Denominator	–	1
PnB0A(2303h:00)	Acceleration User Unit : Numerator	–	1
PnB0C(2303h:01)	Acceleration User Unit : Denominator	–	1

Bild 8: Position, Geschwindigkeit und Beschleunigung Numerator (Zähler) und Denominator (Nenner)

2 Hardwarekonfiguration

In diesem Beispielprojekt ist die Hardwarekonfiguration bereits abgeschlossen. Bitte stellen Sie sicher, dass die Konfiguration zu Ihrer vorhandenen Hardware passt.

Die einzelnen Konfigurationen sind identisch bis auf die jeweilige SPS:

- 1215C DC/DC/DC (6ES7 215-1AG40-0XB0)
- 1516TF-3 PN/DP (6ES7 516-3UN00-0AB0)

Bitte passen Sie ihre Hardwarekonfiguration entsprechend ihrem Projekt an und kompilieren diese neu. Sofern keine weiteren Fehler auftreten, können Sie diese direkt in Ihre CPU laden und online testen, ob eine PROFINET-Verbindung aufgebaut ist.

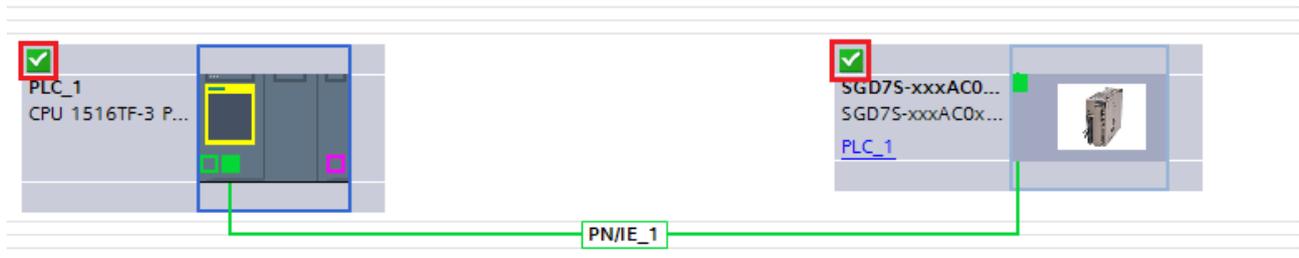


Bild 9: PROFINET -Kommunikation ist vorhanden

Falls ein Fehler auftritt, überprüfen Sie bitte folgende Punkte:

- Name der konfigurierten Geräte
 - o Müssen offline und online identisch sein
- Telegramm der PROFINET-Geräte
 - o Bitte überprüfen sie in SigmaWin+ und in der Hardware-Konfiguration das Telegramm des Sigma-7
 - o Siehe Kapitel Telegramm

Anmerkung: Sobald Sie ihr eigenes Programm verwenden, vergewissern Sie sich bitte, dass Sie die richtige Adressierung benutzen.

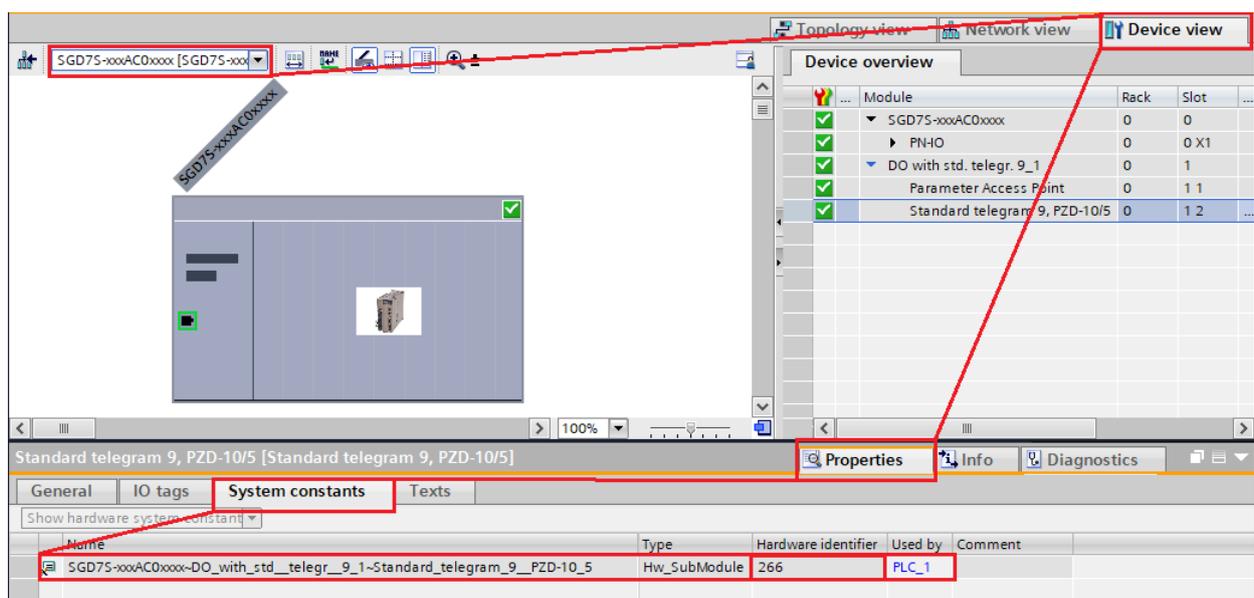


Bild 10: Hardwareidentifikation

3 Software

3.1 Hierarchie

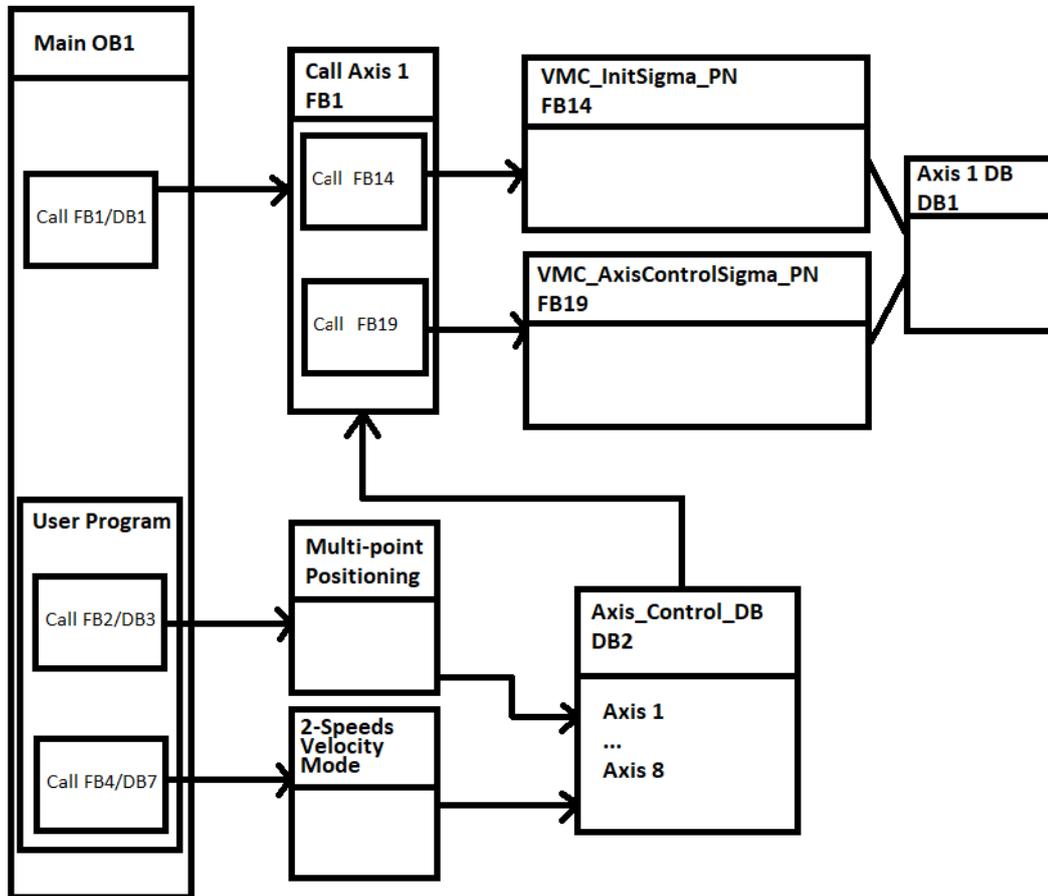


Bild 11: Hierarchie des Programms

Die Hauptstruktur des Programms besteht aus dem OB 1, in dem alle weiteren Bausteine des Programms aufgerufen werden. Zum einen die Bausteine für die Kommunikation in FB1 (FB14 und FB19) und zum anderen das als Beispiel programmierte Anwender-Programm FB2 (Multi-Point-Positioning) und FB4 (2-Speed-Velocity). Die zur Steuerung des Antriebs erforderlichen Daten aus dem Anwender-Programm werden direkt in den DB2 (Axis_Control_DB) geschrieben, der mit dem Achs-Kommunikations-FB (FB19) verbunden ist und hierüber die Achse steuert.

3.2 Datenspeicherung

Der DB2 ist als Array von 8 Achsen aufgebaut. Jede Achse ist dann wiederum unterteilt in die folgenden 6 Abschnitte:

1. Control_In
Dies sind die Steuerregister des Anwenderprogramms.
2. Control_Out
Dies sind die Statusdaten des Anwenderprogramms.
3. Init_In

Die Initial-Werte der Achse werden hierüber angesteuert.

4. Init_Out

Die Initial-Werte der Achse werden hierüber abgebildet.

5. Config_Reference

Die Konfigurations-Werte der Achse werden hier gespeichert und abgerufen durch das Programm.

6. Axis_Reference

Der aktuelle Status der Achse wird hier gespeichert und abgerufen durch das Programm.

3.3 FB1 – CallAxis

In diesem Baustein werden die zur Kommunikation erforderlichen Bausteine VMC_InitSigma_PN und VMC_ControlAxisSigma_PN aufgerufen und angesteuert. Die Übergabe der Variablen findet über den DB2 statt, der genauer in Kapitel 3.2 beschrieben ist.



Bild 12: In- und Outputs von FB1

Die Eingänge "HW_ID_Parameter" und "HW_ID_Telegram" werden durch die Hardware-Adressen des PROFINET-Antriebs belegt. Alle anderen Ein- und Ausgänge sind an den DB2 angeschlossen, der vom Beispielprogramm gesteuert wird.

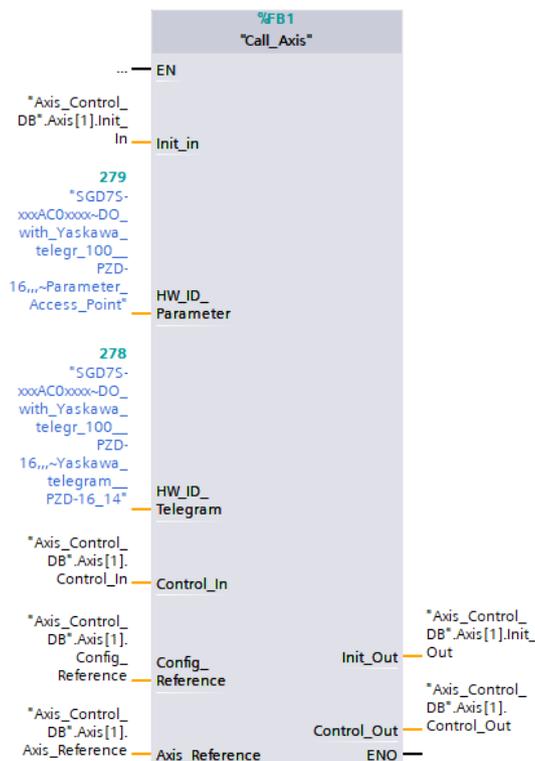


Bild 13: FB1 mit allen verbundenen In- und Outputs

3.4 FB2 – Multi-Punkt-Positionierung

Dieser Funktionsbaustein lässt den Antrieb zu 3 vordefinierten Positionen fahren und wiederholt den Vorgang bis der Eingang "Stop" aktiviert wird. Ein Homing mit diesem Block ist ebenfalls möglich. Die Programmierung des Blocks erfolgte komplett in der Sprache SCL.

3.4.1 Layout

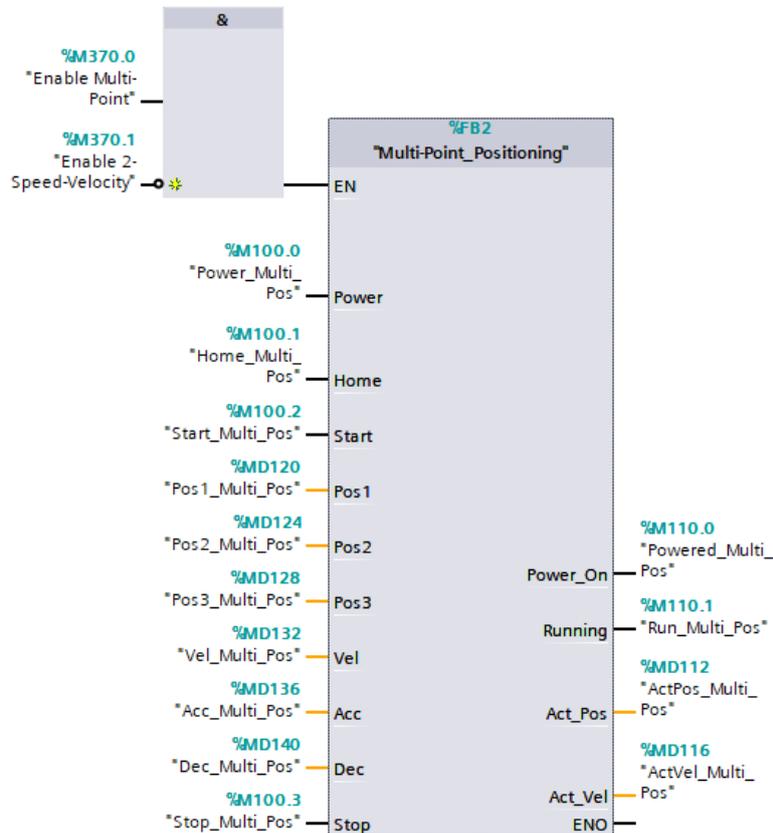


Bild 14: Layout von FB2 – Multi-Punkt-Positionierung

3.4.2 Inputs

Name	Datentyp	Default	Kommentar
Input			
Power	Bool	FALSE	1 = Schaltet den Antrieb ein
Home	Bool	FALSE	Startet die Homing-Prozedur
Start	Bool	0.0	Startet die Positionierfahrt
Pos1	Real	0.0	Ziel-Position 1
Pos2	Real	0.0	Ziel-Position 2
Pos3	Real	0.0	Ziel-Position 3
Vel	Real	0.0	Bewegungsgeschwindigkeit während der Fahrten
Acc	Real	0.0	Benutzerwert Beschleunigung. Wenn 0, dann automatisch auf 100 gesetzt.
Dec	Real	0.0	Benutzerwert negative Beschleunigung (Bremsung). Wenn 0, dann automatisch auf 100 gesetzt.
Stop	Bool	FALSE	Stoppt den Antrieb

3.4.3 Outputs

Name	Datentyp	Default	Kommentar
Output			
Power_On	Bool	FALSE	1 = Servo ist eingeschaltet
Running	Bool	FALSE	1 = Positionierfahrt ist gestartet
Act_Pos	DWord	16#0	Zeigt die aktuelle Position
Act_Vel	DWord	16#0	Zeigt die aktuelle Geschwindigkeit an

3.4.4 Benutzung

Die folgende Schrittkette beschreibt, wie der FB funktioniert und in welcher Reihenfolge welche Bits geschaltet werden müssen.

Anmerkung: Bitte stellen Sie zur eigenen Sicherheit fest, dass der Motor freisteht und sich drehen kann.

1. Aktivieren des Bit MB370.0 ("Enable_Multi-Point") in OB1
2. Deaktivieren des Bit MB370.1 ("Enable_2_Speed-Velocity") in OB1
 - a. Gibt den Funktionsblock frei
3. Benutzerwerte eingeben:
 - a. "Pos1_Multi_Pos"
 - b. "Pos2_Multi_Pos"
 - c. "Pos3_Multi_Pos"
 - d. "Vel_Multi_Pos"
 - e. "Acc_Multi_Pos"
 - f. "Dec_Multi_Pos"
4. Aktivieren des Eingangs "Power" mit dem Bit M100.0 ("Power_Multi_Pos")
5. Aktivieren des Homing-Bits: "Home" (wenn benötigt)
 - a. Startet die Homing-Prozedur des Reglers
 - b. Homing ist abgeschlossen, wenn der Ausgang "ActPos_Multi_Pos" 0 zeigt
6. Deaktivieren des Eingangs "Home"
7. Aktivieren von "Start" mit dem Bit MB100.2 ("Start_Multi_Pos")
 - a. Der Motor startet und fährt zur ersten Position
 - b. Dort angekommen, fährt er direkt zur zweiten Position
 - c. Dort angekommen, fährt er direkt zur dritten Position
8. Ausgang "ActPos_Multi_Pos" zeigt die aktuelle Position
9. Um die Fahrt zu stoppen, aktivieren Sie den Eingang "Stop" mit dem Bit Stop_Multi_Pos (MB 100.3)

3.5 FBX – 2-Speed-Velocity-Block

Bei diesem Funktionsblock werden 2 Geschwindigkeiten vorgegeben, die beim Setzen des jeweiligen Bits aktiviert werden. Vorrang hat hier der Eingang mit der geringeren Geschwindigkeit. Nähere Informationen zu den Ein- und Ausgängen finden Sie weiter unten.

3.5.1 Layout

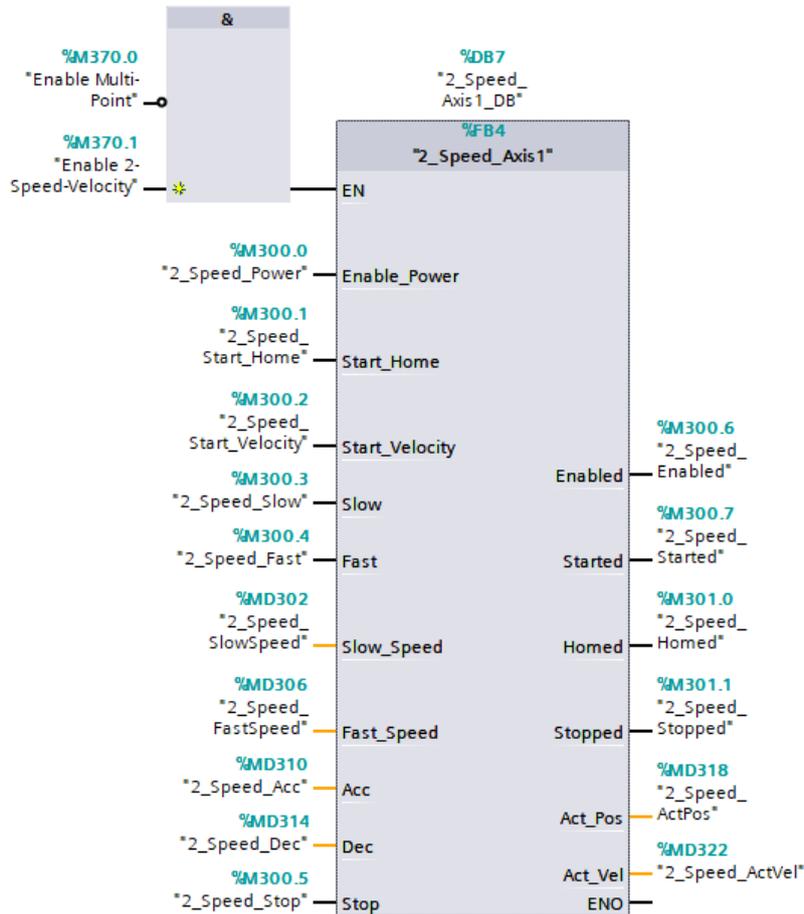


Bild 15: Layout FB2 – 2-Geschwindigkeiten-Block

3.5.2 Inputs

Name	Datentyp	Default	Kommentar
Input			
Enable_Power	Bool	FALSE	Schaltet den Antrieb ein
Start_Home	Bool	FALSE	Aktivieren der Homing-Funktion
Start_Velocity	Bool	FALSE	Startet den Servo-Antrieb
Slow	Bool	FALSE	Aktiviert die langsame Geschwindigkeit
Fast	Bool	FALSE	Aktiviert die schnelle Geschwindigkeit
Slow_Speed	Real	0.0	Benutzerwert der langsamen Geschwindigkeit
Fast_Speed	Real	0.0	Benutzerwert der schnellen Geschwindigkeit
Acc	Real	0.0	Benutzerwert Beschleunigung.
Dec	Real	0.0	Benutzerwert negative Beschleunigung (Bremsung).
Stop	Bool	FALSE	Aktiviert die Positionierung

3.5.3 Outputs

Name	Datentyp	Default	Kommentar
Output			
Enabled	Bool	FALSE	1 = Antrieb ist eingeschalten
Started	Bool	FALSE	1 = Antrieb ist gestartet
Homed	Bool	FALSE	1 = Antrieb ist gehomed. Aktuelle Position = 0.0
Stopped	Bool	FALSE	1 = Motor steht
Act_Pos	Real	0.0	Zeigt die aktuelle Position
Act_Vel	Real	0.0	Zeigt die aktuelle Geschwindigkeit

3.5.4 Benutzung

Die folgende Schrittkette beschreibt wie der FB funktioniert und in welcher Reihenfolge welche Bits geschaltet werden müssen.

Anmerkung: Bitte stellen Sie zur eigenen Sicherheit fest, dass der Motor freisteht und sich drehen kann

3.5.4.1 Schnelle und langsame Geschwindigkeit

1. Aktivierung von Bit MB370.1 ("Enable 2-Speed-Velocity ") in OB1
2. Deaktivierung von Bit MB370.0 ("Enable Multi-Point ") in OB1
 - a. Gibt den Funktionsblock frei
3. Aktivierung von Eingang "Enable_Power"
4. Benutzerwerte eingeben:
 - a. "Slow_Speed"
 - b. "Fast_Speed"
 - c. "Acc"
 - d. "Dec"
5. Eingang "Slow" oder "Fast" aktivieren
 - a. "Slow" aktiviert die langsame Geschwindigkeit
 - b. "Fast" aktiviert die schnelle Geschwindigkeit
 - c. Sind beide Eingänge aktiviert, hat die langsame Geschwindigkeit Vorrang!
6. Aktivierung von Eingang "Start_Velocity"
 - a. Startet den Antrieb mit der eingestellten Geschwindigkeit (siehe Punkt 5)
7. Während der Fahrt kann nun mit den Eingängen "Slow" und "Fast" die zu fahrende Geschwindigkeit aktiviert/deaktiviert werden

3.5.4.2 Homing

1. Aktivierung von Bit MB370.1 ("Enable 2-Speed-Velocity ") in OB1
2. Deaktivierung von Bit MB370.0 ("Enable Multi-Point ") in OB1
 - a. Gibt den Funktionsblock frei
3. Aktivierung von Eingang "Enable_Power"
4. Aktivierung von Eingang "Start_Home"
 - a. Homing ist abgeschlossen, wenn der Ausgang "Act_Pos" 0 zeigt
 - b. Der Ausgang "Homed" den Status TRUE hat

3.6 Variablen

Alle verwendeten Variablen in diesem Beispielprogramm sind unten aufgeführt und kommentiert.

Name	Datentyp	Adresse	Kommentar
General			
System_Byte	Byte	%MB0	
FirstScan	Bool	%M0.0	Ändert den Status beim ersten Scan nach dem Einschalten der CPU
DiagStatusUpdate	Bool	%M0.1	Ändert den Status bei einem Diagnose Status Update
AlwaysTRUE	Bool	%M0.2	Ist immer auf "TRUE"
AlwaysFALSE	Bool	%M0.3	Ist immer auf "FALSE"
Clock_Byte	Byte	%MB1	
Clock_10Hz	Bool	%M1.0	Ändert den Status im Takt von 10Hz
Clock_5Hz	Bool	%M1.1	Ändert den Status im Takt von 5Hz
Clock_2.5Hz	Bool	%M1.2	Ändert den Status im Takt von 2.5Hz
Clock_2Hz	Bool	%M1.3	Ändert den Status im Takt von 2Hz
Clock_1.25Hz	Bool	%M1.4	Ändert den Status im Takt von 1.25Hz
Clock_1Hz	Bool	%M1.5	Ändert den Status im Takt von 1Hz
Clock_0.625Hz	Bool	%M1.6	Ändert den Status im Takt von 0.625Hz
Clock_0.5Hz	Bool	%M1.7	Ändert den Status im Takt von 0.5Hz
Enable			
Enable Multi-Point	Bool	%M370.0	Aktiviert die Multi-Punkt-Positionierung und sperrt die 2-Geschwindigkeiten-Funktion
Enable 2-Speed-Velocity	Bool	%M370.1	Aktiviert die 2-Geschwindigkeiten-Funktion und sperrt die Multi-Punkt-Positionierung
Multi-Point-Positioning			
Power_Multi_Pos	%M100.0	Bool	Schaltet den Antrieb ein
Home_Multi_Pos	%M100.1	Bool	Startet die Homing-Funktion
Start_Multi_Pos	%M100.2	Bool	Startet die Positionier-Fahrt
Stop_Multi_Pos	%M100.3	Bool	Stoppt die Positionier-Fahrt
Powered_Multi_Pos	%M110.0	Bool	Speichert den aktuellen Status des Ausgangs "Power_On"
Run_Multi_Pos	%M110.1	Bool	Antrieb ist in der aktiven Positionier-Fahrt
ActPos_Multi_Pos	%MD112	Real	Zeigt die aktuelle Position an
ActVel_Multi_Pos	%MD116	Real	Zeigt die aktuelle Geschwindigkeit an
Pos1_Multi_Pos	%MD120	Real	Ziel-Position 1
Pos2_Multi_Pos	%MD124	Real	Ziel-Position 2
Pos3_Multi_Pos	%MD128	Real	Ziel-Position 3
Vel_Multi_Pos	%MD132	Real	Geschwindigkeit während der Fahrt Kann während der Fahrt angepasst und geändert werden
Acc_Multi_Pos	%MD136	Real	Beschleunigung während der Fahrt Kann während der Fahrt angepasst werden
Dec_Multi_Pos	%MD140	Real	Negative Beschleunigung während der Fahrt Kann während der Fahrt angepasst werden

Name	Datentyp	Adresse	Kommentar
2-Speed-Velocity			
2_Speed_Vel_Power	%M300.0	Bool	Schaltet den Antrieb ein
2_Speed_Vel_Start_Home	%M300.1	Bool	Startet die Homing-Funktion des Antriebs
2_Speed_Vel_Start	%M300.2	Bool	Startet den Antrieb
2_Speed_Vel_Slow	%M300.3	Bool	Aktiviert die langsame Geschwindigkeit
2_Speed_Vel_Fast	%M300.4	Bool	Aktiviert die schnelle Geschwindigkeit
2_Speed_Vel_SlowSpeed	%MD302	Real	Langsame Geschwindigkeit Kann während der Fahrt angepasst werden
2_Speed_Vel_FastSpeed	%MD306	Real	Schnelle Geschwindigkeit Kann während der Fahrt angepasst werden
2_Speed_Vel_Acc	%MD310	Real	Beschleunigung während der Fahrt Kann während der Fahrt angepasst werden
2_Speed_Vel_Dec	%MD314	Real	Negative Beschleunigung während der Fahrt Kann während der Fahrt angepasst werden
2_Speed_Vel_Stop	%M300.5	Bool	Stoppt den Antrieb
2_Speed_Vel_Enabled	%M300.6	Bool	Antrieb ist angeschalten
2_Speed_Vel_Started	%M300.7	Bool	Antrieb ist gestartet
2_Speed_Vel_Homed	%M301.0	Bool	Antrieb ist gehomed
2_Speed_Vel_Stopped	%M301.1	Bool	Antrieb ist gestoppt
2_Speed_Vel_ActPos	%MD318	Real	Zeigt die aktuelle Position
2_Speed_Vel_ActVel	%MD322	Real	Zeigt die aktuelle Geschwindigkeit