|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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](#_Toc54616258)

[1.1 Zweck dieses Dokuments 4](#_Toc54616259)

[1.2 Voraussetzungen 4](#_Toc54616260)

[1.2.1 TIA Portal 4](#_Toc54616261)

[1.2.1.1 PLC 4](#_Toc54616262)

[1.2.1.2 Sigma-7 5](#_Toc54616263)

[1.2.2 SigmaWin+ 6](#_Toc54616264)

[1.2.2.1 Telegramm 6](#_Toc54616265)

[1.2.2.2 Optionale Einstellungen 7](#_Toc54616266)

[2 Hardwarekonfiguration 8](#_Toc54616267)

[3 Software 9](#_Toc54616268)

[3.1 Hierarchie 9](#_Toc54616269)

[3.2 Datenspeicherung 9](#_Toc54616270)

[3.3 FB1 – CallAxis 11](#_Toc54616271)

[3.4 FB2 – Multi-Punkt-Positionierung 12](#_Toc54616272)

[3.4.1 Layout 12](#_Toc54616273)

[3.4.2 Inputs 13](#_Toc54616274)

[3.4.3 Outputs 13](#_Toc54616275)

[3.4.4 Benutzung 14](#_Toc54616276)

[3.5 FBX – 2-Speed-Velocity-Block 15](#_Toc54616277)

[3.5.1 Layout 15](#_Toc54616278)

[3.5.2 Inputs 16](#_Toc54616279)

[3.5.3 Outputs 16](#_Toc54616280)

[3.5.4 Benutzung 17](#_Toc54616281)

[3.5.4.1 Schnelle und langsame Geschwindigkeit 17](#_Toc54616282)

[3.5.4.2 Homing 17](#_Toc54616283)

[3.6 Variablen 18](#_Toc54616284)

Bilderverzeichnis

[Bild 1: Ändern der IP-Adresse](#_Toc54616285)

[Bild 2: Aktivieren des Clock memory bytes](#_Toc54616286)

[Bild 3: Zuweisen eines Gerätenamens](#_Toc54616287)

[Bild 4: USB-Type Mini und Type A](#_Toc54616288)

[Bild 5: Verbindung zum Servopack](#_Toc54616289)

[Bild 6: Unterstützte Telegramme im SIGMA-7](#_Toc54616290)

[Bild 7: Ausgewähltes Telegramm 100](#_Toc54616291)

[Bild 8: Position, Geschwindigkeit und Beschleunigung Numerator (Zähler) und Denominator (Nenner)](#_Toc54616292)

[Bild 9: PROFINET -Kommunikation ist vorhanden](#_Toc54616293)

[Bild 10: Hardwareidentifikation](#_Toc54616294)

[Bild 11: Hierarchie des Programms](#_Toc54616295)

[Bild 12: In- und Outputs von FB1](#_Toc54616296)

[Bild 13: FB1 mit allen verbundenen In- und Outputs](#_Toc54616297)

[Bild 14: Layout von FB2 – Multi-Punkt-Positionierung](#_Toc54616298)

[Bild 15: Layout FB2 – 2-Geschwindigkeiten-Block](#_Toc54616299)

# Generell

## 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>.

## Voraussetzungen

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

### TIA Portal

#### 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

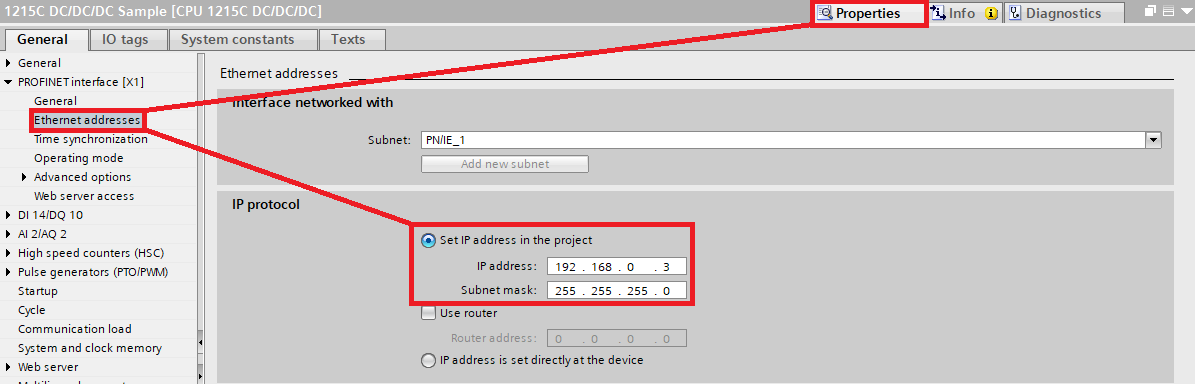


Bild 1: Ändern der IP-Adresse

###### 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)

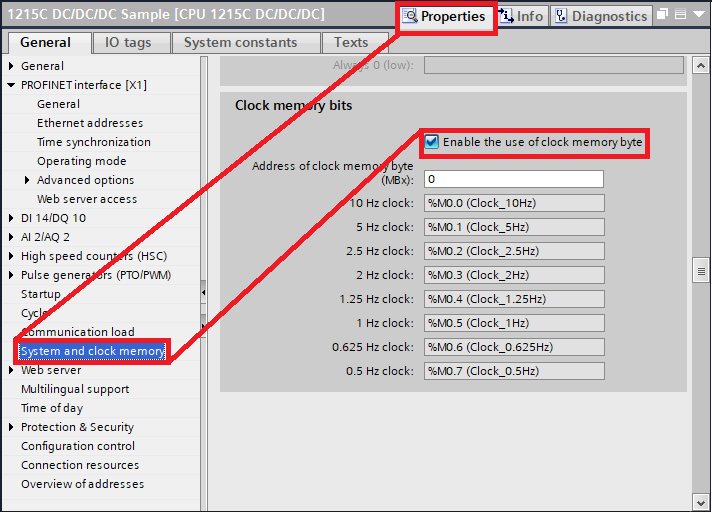


Bild 2: Aktivieren des Clock memory bytes

#### Sigma-7

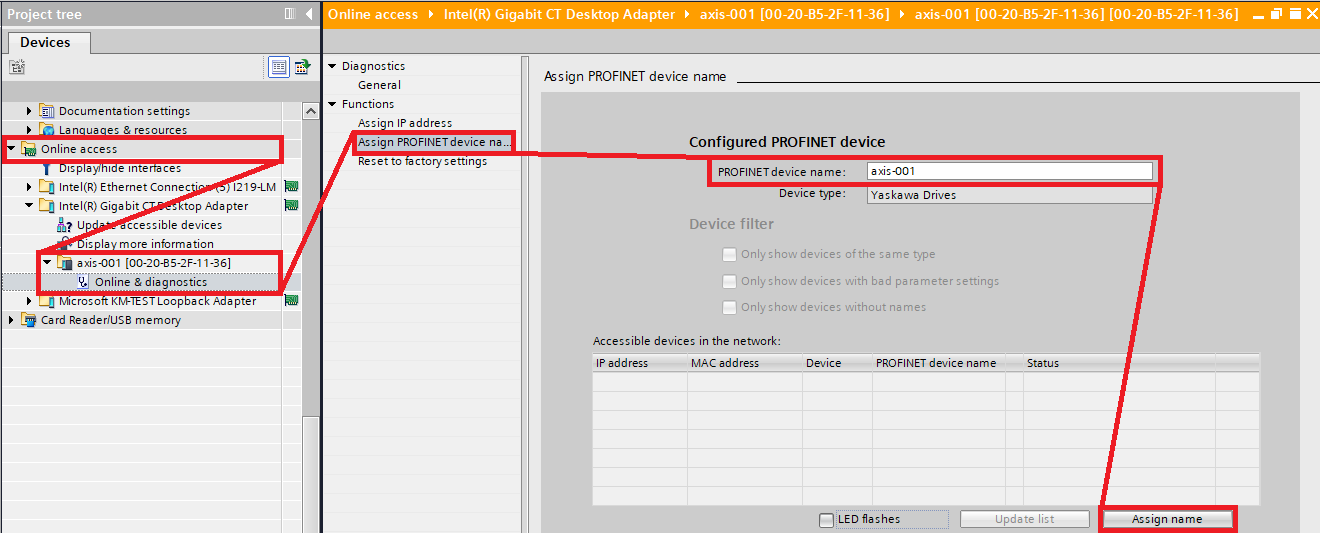
Um eine korrekte Verbindung zwischen der SPS und dem Sigma-7 Antrieb via PROFINET herzustellen, müssen die Namen der einzelnen Busteilnehmer identisch mit den Namen in der Hardwarekonfiguration sein. Die aktuell vergebenen Namen kann man sich in den online erreichbaren Teilnehmern anzeigen lassen und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

Bild 3: Zuweisen eines Gerätenamens

### 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.

Ein Bild, das Kabel, Verbinder, Adapter enthält.

Automatisch generierte Beschreibung

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.

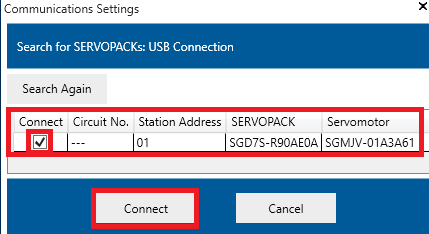
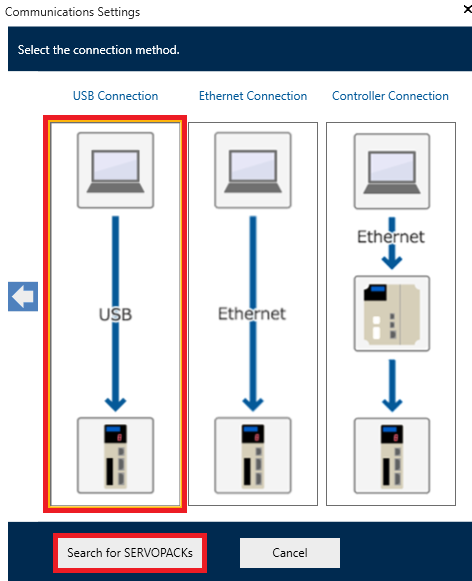


Bild 5: Verbindung zum Servopack

#### 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:

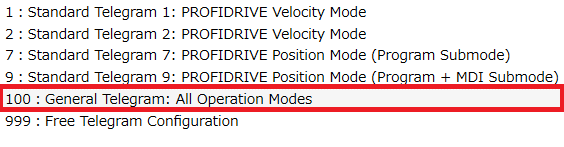


Bild 6: Unterstützte Telegramme im SIGMA-7



Bild 7: Ausgewähltes Telegramm 100

#### 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.

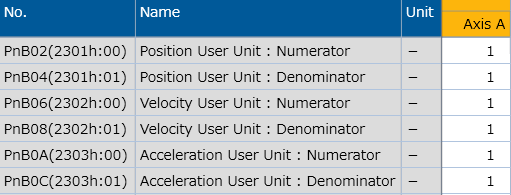


Bild 8: Position, Geschwindigkeit und Beschleunigung Numerator (Zähler) und Denominator (Nenner)

# 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, könne Sie diese direkt in Ihre CPU laden und online testen, ob eine PROFINET-Verbindung aufgebaut ist.

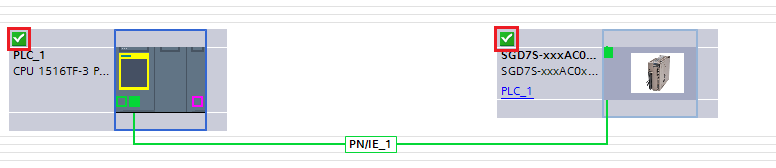


Bild 9: PROFINET -Kommunikation ist vorhanden

Falls ein Fehler auftritt, überprüfen Sie bitte folgende Punkte:

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

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

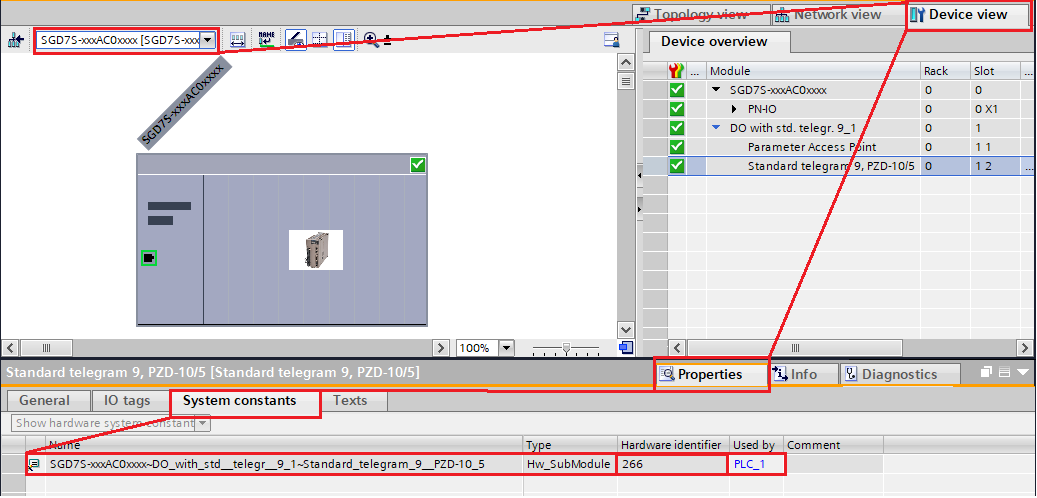


Bild 10: Hardwareidentifikation

# Software

## Hierarchie

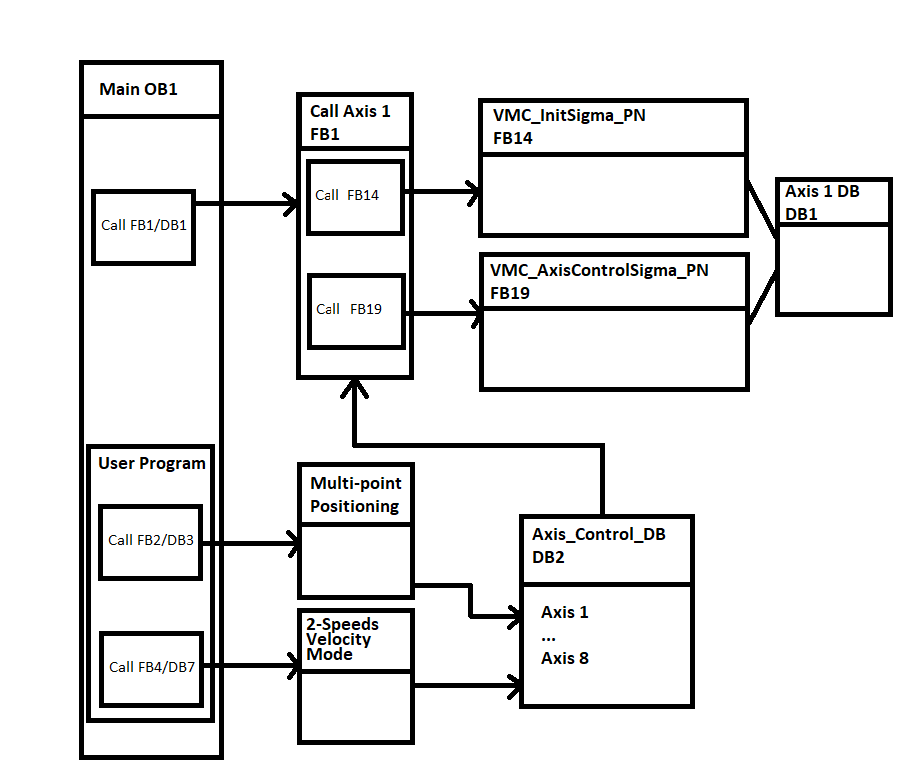


Bild 11: Hierarchie des Programms

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 erfoderlichen 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.

## 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.

1. Control\_Out

Dies sind die Statusdaten des Anwenderprogramms.

1. Init\_In

Die Initial-Werte der Achse werden hierüber angesteuert.

1. Init\_Out

Die Initial-Werte der Achse werden hierüber abgebildet.

1. Config\_Reference

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

1. Axis\_Reference

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

## 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.

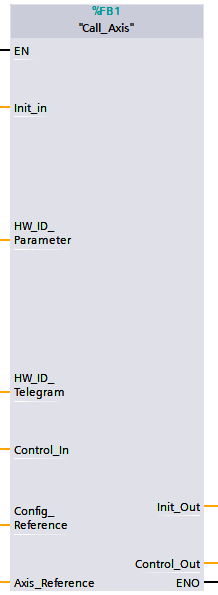


Bild 12: In- und Outputs von FB1

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.

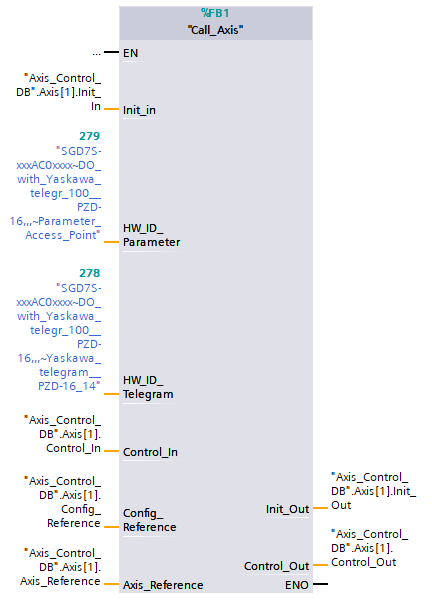


Bild 13: FB1 mit allen verbundenen In- und Outputs

## 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.

### Layout

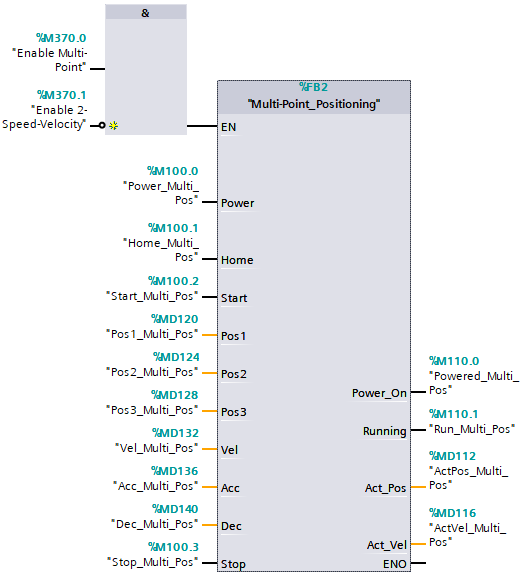


Bild 14: Layout von FB2 – Multi-Punkt-Positionierung

### 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 |

### 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 |

### Benutzung

Die folgende Schrittkette beschreibt, wie der FB funktioniert und in welcher Reihenfolge welche Bits geschalten 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
   1. Gibt den Funktionsblock frei
3. Benutzerwerte eingeben:
   1. "Pos1\_Multi\_Pos"
   2. "Pos2\_Multi\_Pos"
   3. "Pos3\_Multi\_Pos"
   4. "Vel\_Multi\_Pos"
   5. "Acc\_Multi\_Pos"
   6. "Dec\_Multi\_Pos"
4. Aktivieren des Eingangs "Power” mit dem Bit M100.0 ("Power\_Multi\_Pos")
5. Aktivieren des Homing-Bits: "Home” (wenn benötigt)
   1. Startet die Homing-Prozedur des Reglers
   2. 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")
   1. Der Motor startet und fährt zur ersten Position
   2. Dort angekommen, fährt er direkt zur zweiten Position
   3. 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)

## 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.

### Layout

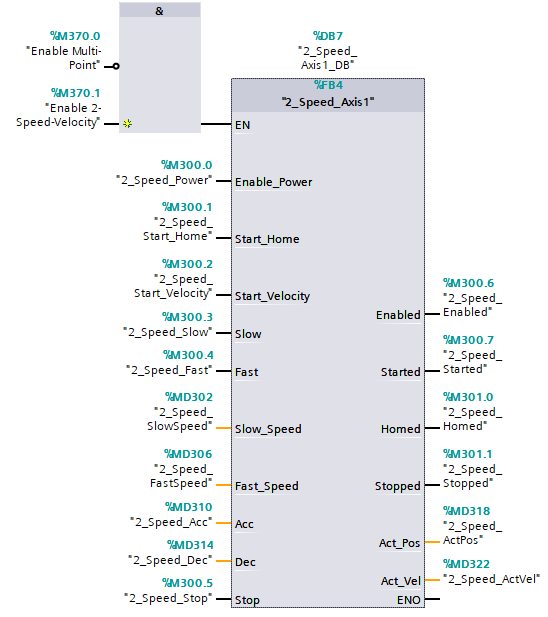


Bild 15: Layout FB2 – 2-Geschwindigkeiten-Block

### 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 |

### 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 |

### Benutzung

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

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

#### 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
   1. Gibt den Funktionsblock frei
3. Aktivierung von Eingang "Enable\_Power”
4. Benutzerwerte eingeben:
   1. "Slow\_Speed”
   2. "Fast\_Speed”
   3. "Acc”
   4. "Dec”
5. Eingang "Slow" oder "Fast" aktivieren
   1. "Slow" aktiviert die langsame Geschwindigkeit
   2. "Fast" aktiviert die schnelle Geschwindigkeit
   3. Sind beide Eingänge aktiviert, hat die langsame Geschwindigkeit Vorrang!
6. Aktivierung von Eingang "Start\_Velocity”
   1. 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

#### Homing

1. Aktivierung von Bit MB370.1 ("Enable 2-Speed-Velocity ”) in OB1
2. Deaktivierung von Bit MB370.0 ("Enable Multi-Point ”) in OB1
   1. Gibt den Funktionsblock frei
3. Aktivierung von Eingang "Enable\_Power”
4. Aktivierung von Eingang "Start\_Home”
   1. Homing ist abgeschlossen, wenn der Ausgang "Act\_Pos” 0 zeigt
   2. Der Ausgang "Homed" den Status TRUE hat

## 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 schnellle 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 |