

Application Note sample program Simple Motion PN for 1200/1500

Theme

Application Note

Document type

Sigma-7

Product

Date	Version	Author	Changes
10/22/2020	Initial	Küster Chris	-
Limitations and Applications			
Sigma-7 200V - PROFINET, Sigma-7 400V - PROFINET, Servopack Firmware 002A or higher, SigmaWin + V2.7 or higher			
SIEMENS PLC 12xx firmware V4.0 or higher, SIEMENS PLC 15xx FW V2.5 or higher, TIA Portal V15.0 or higher			
Applied files			
"Demo_TIA_1x00_V0000.zip"			

Contents

1	General	4
1.1	Purpose of this document	4
1.2	Precondition	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	Optional settings	7
2	Hardware configuration	8
3	Software	9
3.1	Hierarchy	9
3.2	Data storage	9
3.3	FB1 – CallAxis	11
3.4	FB2 - multi-point positioning	12
3.4.1	Layout	12
3.4.2	Inputs	13
3.4.3	OUTPUT	13
3.4.4	Usage	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	Usage	17
3.5.4.1	Fast and slow velocity	17
3.5.4.2	Homing	17
3.6	Variables	18

List of figures

Figure 1: Changing the IP address

Figure 2: Activate the clock memory byte s

Figure 3: Assigning a device name

Figure 4: USB-Type Mini and Type A

Figure 5: Connection to the Servopack

Figure 6: Supported telegrams in the SIGMA-7

Figure 7: Selected telegram 100

Figure 8: Position, speed and acceleration numerator and denominator

Figure 9: PROFINET communication is running

Figure 10: Hardware identification

Figure 11: Program hierarchy

Figure 12: Inputs and outputs from FB1

Figure 13: FB1 with all connected inputs and outputs

Figure 14: Layout of FB2 - multi-point positioning

Figure 15: Layout FB2 – 2-Geschwindigkeiten-Block

1 General

1.1 Purpose of this document

This document explains the use of the sample project for the new Simple Motion library for the Siemens PLC 1200 and 1500.

If errors occur on the drive or controller, please refer to the associated documentation. This also applies to any errors in the Simple Motion block library. You can find both user manuals on the website <https://www.yaskawa.de/services/dokumenten-download-center>.

1.2 Precondition

This chapter describes the requirements for the individual PLC types and the Sigma-7.

1.2.1 TIA Portal

1.2.1.1 PLC

IP-Address

Regardless of the type of PLC used for the project, the first step is to adapt and change the IP address according to your network

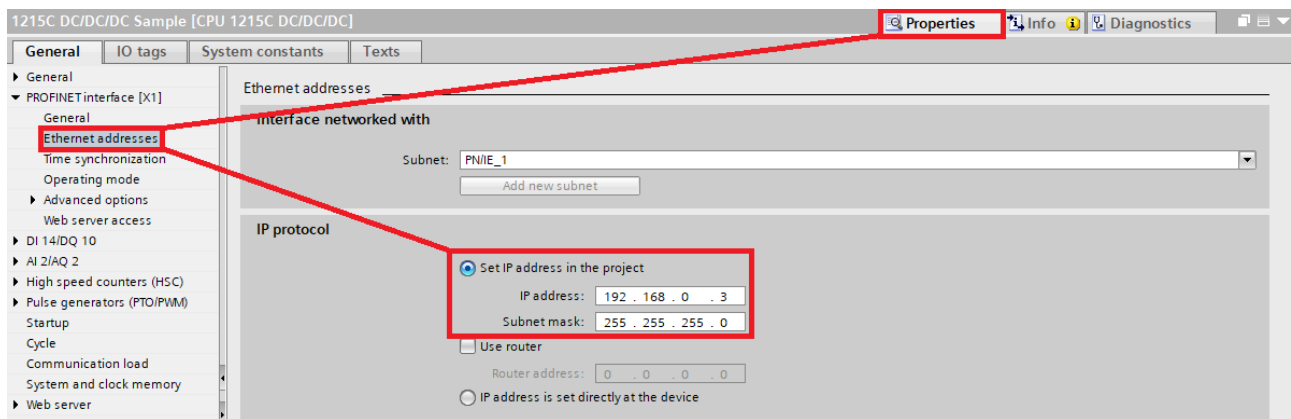


Figure 1: Changing the IP address

Time and timer

A time bit is used in these sample projects. This setting must therefore be made in the hardware configuration. These bits automatically change their status in a predefined period (see Figure 2)

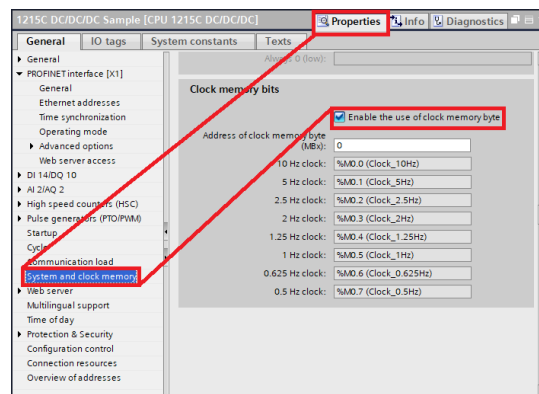


Figure 2: Activate the clock memory byte s

1.2.1.2 Sigma-7

In order to establish a correct connection between the PLC and the Sigma-7 drive via PROFINET, the names of the individual bus participants must be identical to the names in the hardware configuration. The names currently assigned can be shown online in the list of accessible devices and if necessary, changed as described in Figure 3.

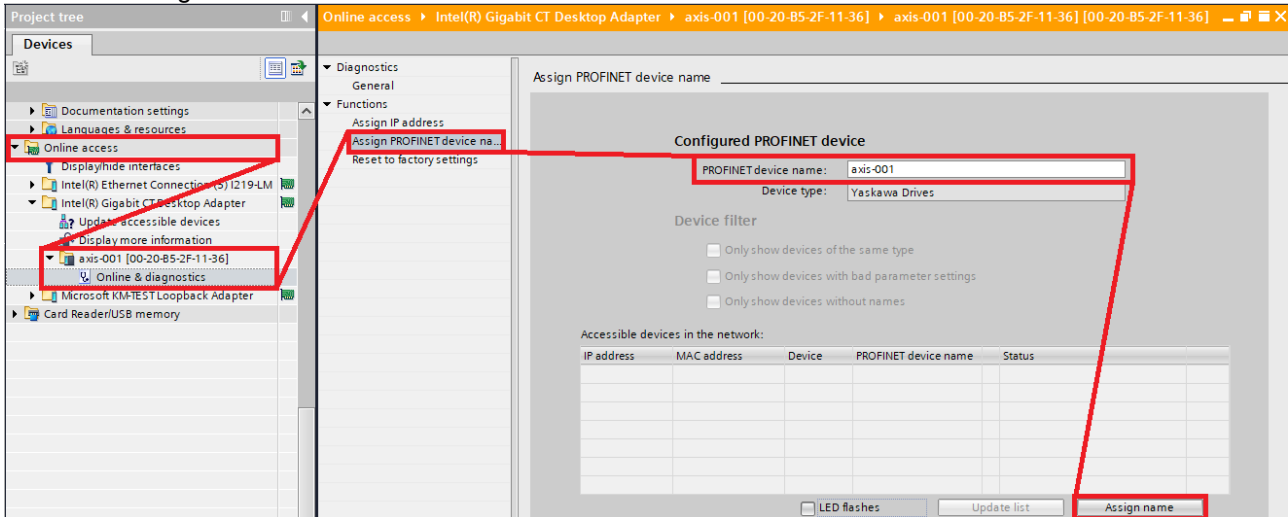


Figure 3: Assigning a device name

1.2.2 SigmaWin+

With the YASKAWA tool SigmaWin+, some conditions must be set in the Sigma-7 drive. The connection between the tool and the Sigma-7 is established with an USB type Mini-B cable (drive side) and USB type A (computer side). It is recommended to use the Yaskawa connection cable here. This has the designation JZSP-CVS06-02-E.



Figure 4: USB-Type Mini and Type A

Open the tool, select the USB connection method and search for servo packs. Your device will then be listed on the next screen. Select this and establish a connection.

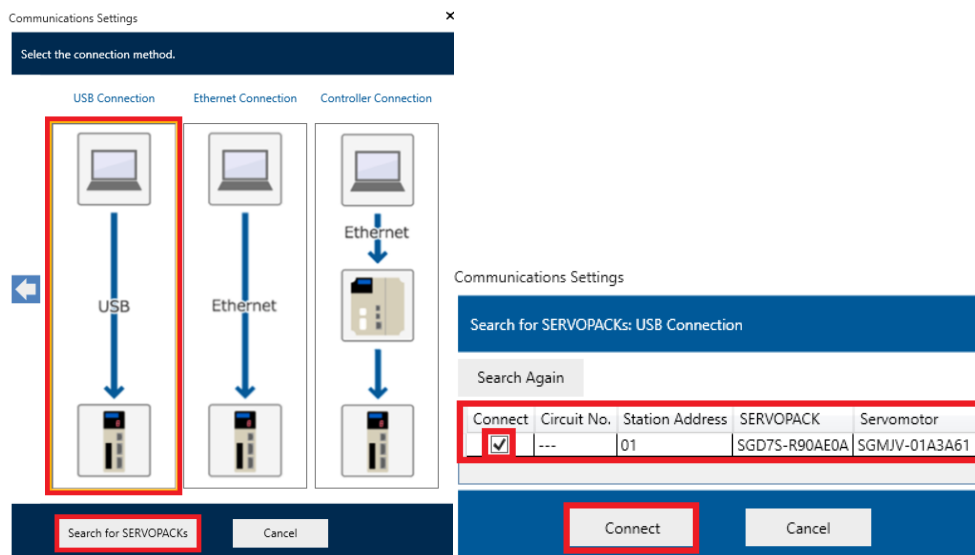


Figure 5: Connection to the Servopack

1.2.2.1 Telegramm

As specified in the hardware configuration, the SIGMA-7 drive communicates with the PLC via telegram 100. The telegram type must be set in the SIGMA-7 via the parameter **PnC90(922)**. With the Sigma-5 this setting is made on the option card with the rotary switch S11. The following additional telegrams are also supported by the SIGMA-7:

- 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

Figure 6: Supported telegrams in the SIGMA-7

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

Figure 7: Selected telegram 100

1.2.2.2 Optional settings

Acyclic communication is possible in telegram 100 and is also described in the manual. This function is not programmed in the example program. This means that the gear ratios and gear settings must be set manually using SigmaWin + in the parameters shown below.

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

Figure 8: Position, speed and acceleration numerator and denominator

2 Hardware configuration

In this sample project, the hardware configuration has already been completed. Please make sure that the configuration matches your existing hardware.

The individual configurations are identical except the respective PLC:

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

Please adapt your hardware configuration according to your project and recompile it. If no further errors occur, you can upload it directly into your CPU and test online whether a PROFINET connection was established.

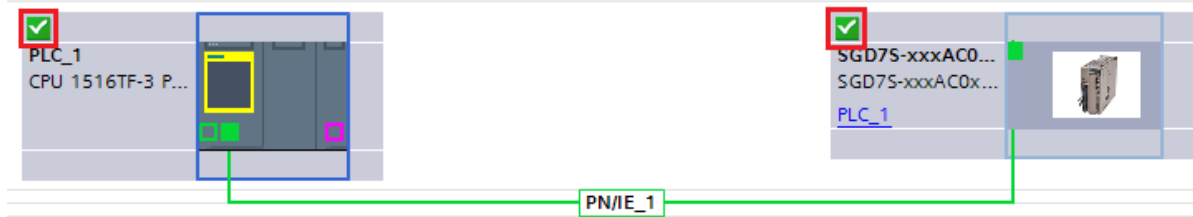


Figure 9: PROFINET communication is running

If an error occurs, please check the following points:

- Name of the configured devices
 - o Must be the same offline and online
- Telegram of the PROFINET devices
 - o Please check the Sigma-7 telegram in SigmaWin + and in the hardware configuration
 - o Refer to chapter Telegramm

Annotation: Once you are using your own program, please make sure that you are using the correct addressing.

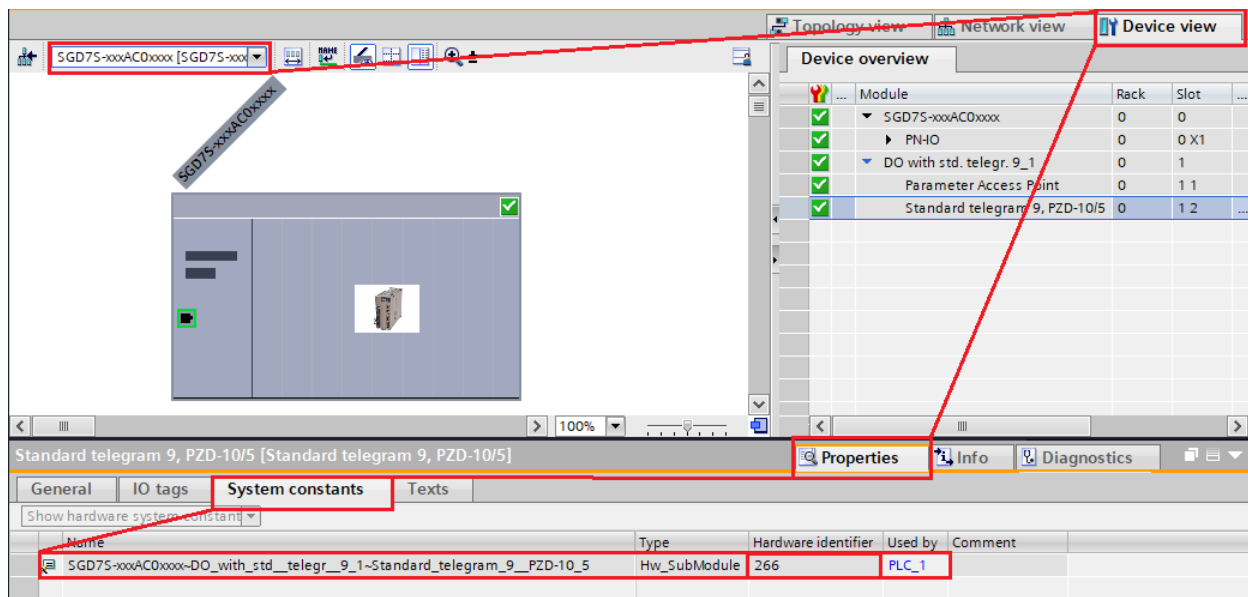


Figure 10: Hardware identification

3 Software

3.1 Hierarchy

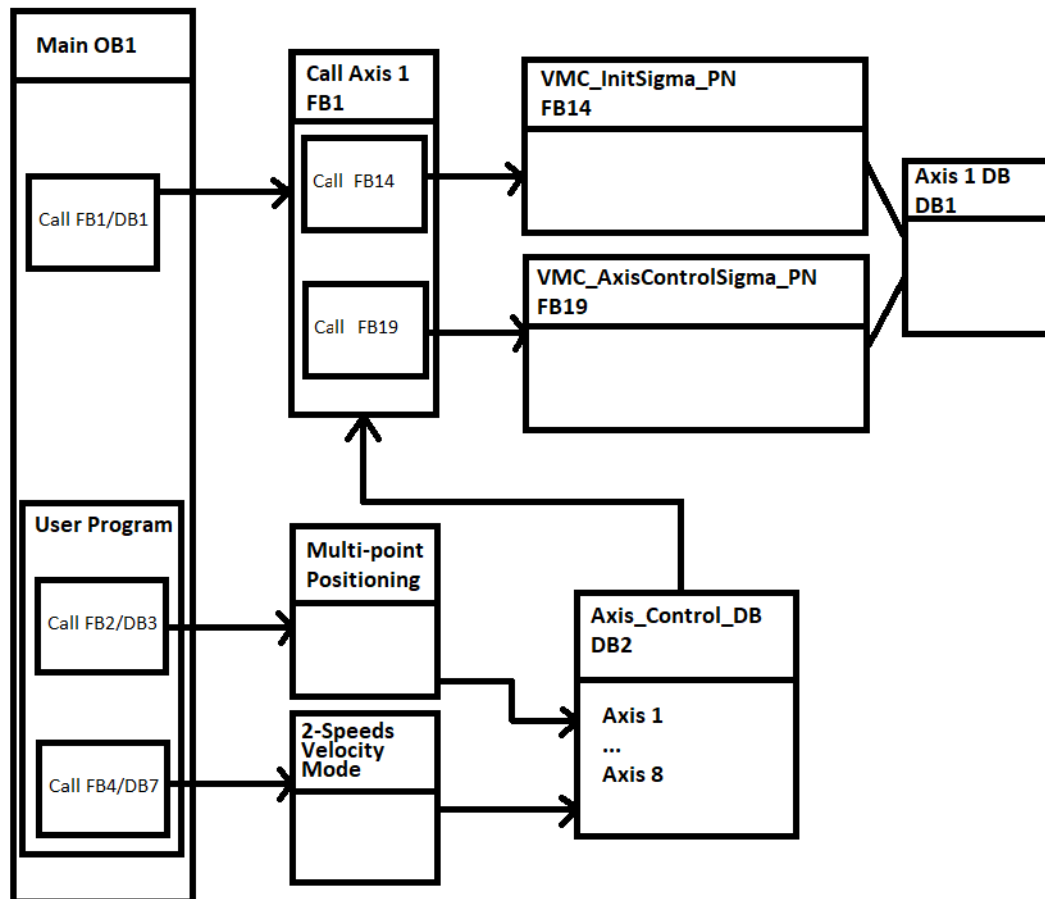


Figure 11: Program hierarchy

The main structure of the program consists of OB 1, in which all other blocks of the program are called. On the one hand the blocks for communication in FB1 (FB14 and FB19) and on the other hand the user program FB2 (multi-point positioning) and FB4 (2-speed velocity) programmed as an example. The data required to control the drive from the user program are written directly to DB2 (Axis_Control_DB), which is connected to the axis communication FB (FB19) and controls the axis via this.

3.2 Data storage

The DB2 is structured as an array of 8 axes. Each axis is then divided into the following 6 sections:

1. Control_In
These are the control registers of the user program.
2. Control_Out
These are the status data of the user program.
3. Init_In
The initial values of the axis are controlled via this.
4. Init_Out

The initial values of the axis are mapped on this.

5. Config_Reference

The configuration values of the axis are saved here and called by the program.

6. Axis_Reference

The current status of the axis is saved here and called by the program.

3.3 FB1 – CallAxis

In this block, the blocks VMC_InitSigma_PN and VMC_ControlAxisSigma_PN required for communication are called and controlled. The variables are transferred via DB2, which is described in more detail in Chapter 3.2.

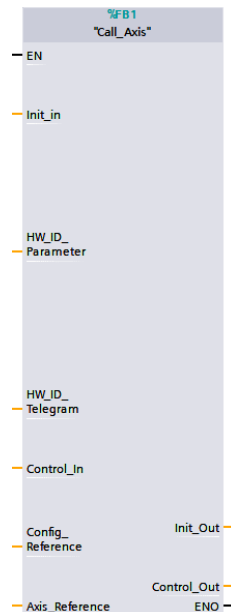


Figure 12: Inputs and outputs from FB1

The inputs "HW_ID_Parameter" and "HW_ID_Telegram" are used by the hardware addresses of the PROFINET drive. All other inputs and outputs are connected to DB2, which is controlled by the sample program.

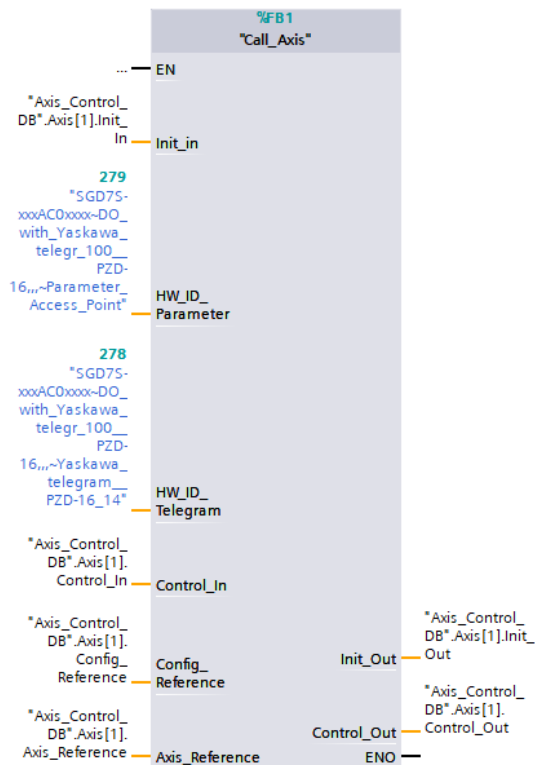


Figure 13: FB1 with all connected inputs and outputs

3.4 FB2 - multi-point positioning

This function block allows the drive to move to 3 predefined positions and repeats the process until the "Stop" input is activated. Homing with this block is also possible. The block was programmed entirely in the SCL language.

3.4.1 Layout

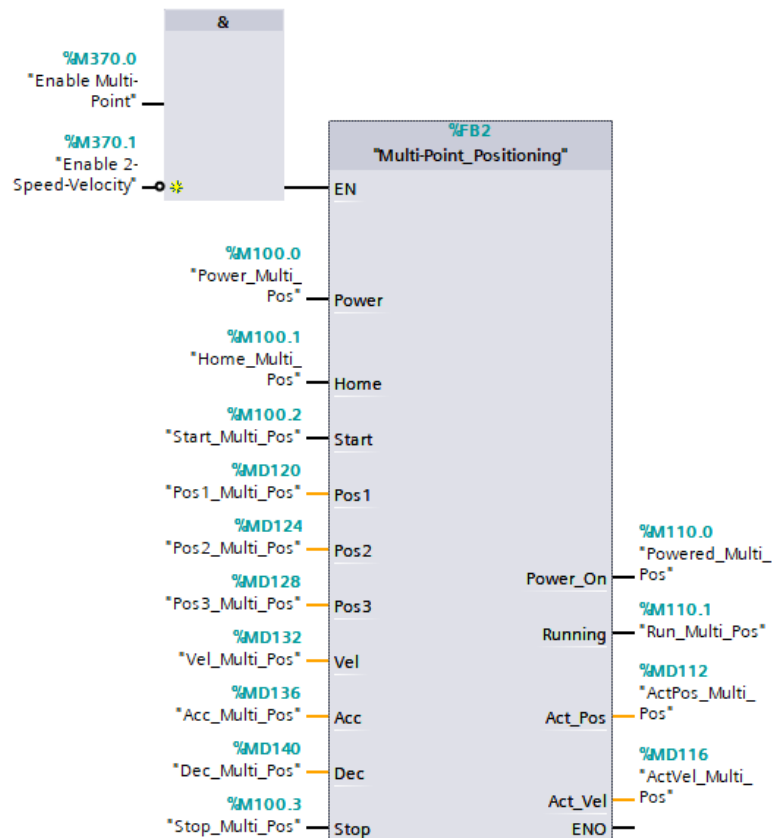


Figure 14: Layout of FB2 - multi-point positioning

3.4.2 Inputs

Name	Data type	Default	Comment
Input			
Power	BOOL	FALSE	1 = switches the drive ON
Home	Bool	FALSE	Starts the homing procedure
Start	Bool	0.0	Starts the positioning run
Pos1	Real	0.0	Target position 1
Pos2	Real	0.0	Target position 2
Pos3	Real	0.0	Target position 3
Vel	Real	0.0	Velocity of motion while driving
Acc	Real	0.0	User value acceleration If 0 then automatically set to 100.
Dec	Real	0.0	User value negative acceleration (braking). If 0 then automatically set to 100.
Stop	Bool	FALSE	Stops the drive

3.4.3 OUTPUT

Name	Data type	Default	Comment
Output			
Power_On	Bool	FALSE	1 = servo is switched ON
Running	Bool	FALSE	1 = positioning run has started
Act_Pos	DWord	16 # 0	Shows the current position
Act_Vel	DWord	16 # 0	Displays the current velocity

3.4.4 Usage

The following step chain describes how the FB works and in which order which bits have to be switched.

Annotation: For your own safety, please ensure that the drive is free and can turn.

1. Activate bit MB370.0 ("Enable_Multi-Point") in OB1
2. Deactivation bit MB370.1 ("Enable_2_Speed-Velocity") in OB1
 - a. Enables the function block
3. Enter user values:
 - 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. Activate the "Power" input with bit M100.0 ("Power_Multi_Pos")
5. Activation of the homing bit: "Home" (if required)
 - a. Starts the homing procedure of the controller
 - b. Homing is complete when the "ActPos_Multi_Pos" output shows 0
6. Disabling the "Home" input
7. Activate "Start" with bit MB100.2 ("Start_Multi_Pos")
 - a. The drive starts and moves to the first position
 - b. Once there, it moves straight to the second position
 - c. Once there, it moves straight to the third position
8. The "ActPos_Multi_Pos" output shows the current position
9. To stop the movement, activate the "Stop" input with the Stop_Multi_Pos bit (MB 100.3)

3.5 FBX – 2-Speed-Velocity-Block

In this function block, 2 velocities are specified, which are activated when the respective bit is set. The input with the lower velocity has priority here. Further information on the inputs and outputs can be found below.

3.5.1 Layout

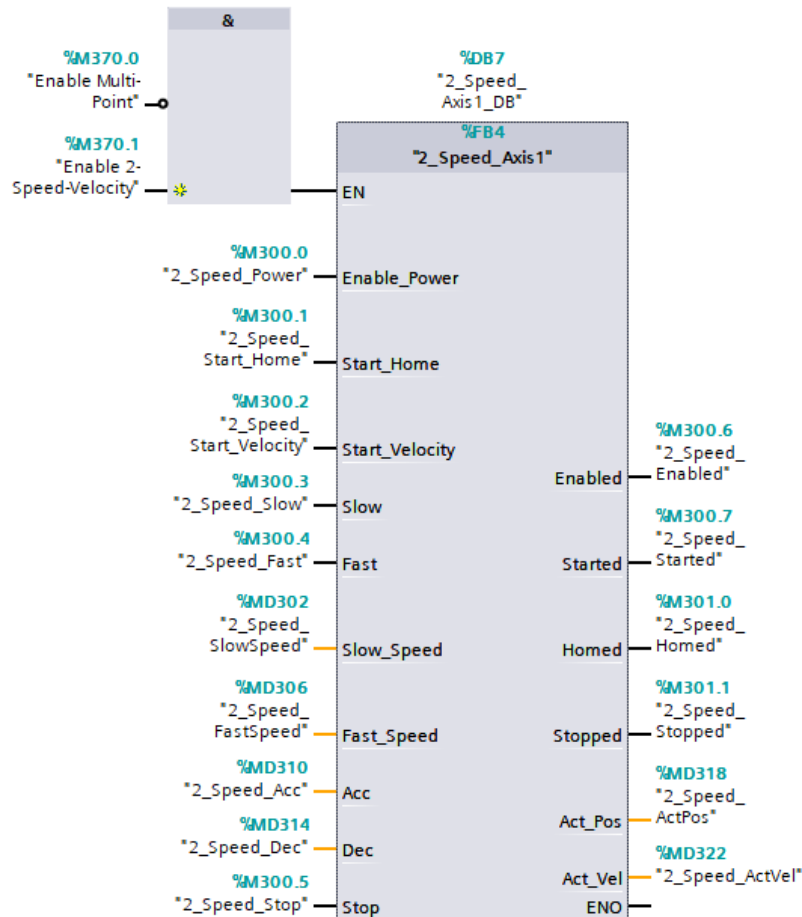


Figure 15: Layout FB2 – 2-Geschwindigkeiten-Block

3.5.2 Inputs

Name	Data type	Default	Comment
Input			
Enable_Power	Bool	FALSE	Switches the drive ON
Start_Home	Bool	FALSE	Enable the homing function
Start_Velocity	Bool	FALSE	Starts the servo drive
Slow	Bool	FALSE	Activates the slow velocity
Fast	Bool	FALSE	Activates the fast velocity
Slow_Speed	Real	0.0	Slow velocity user value
Fast_Speed	Real	0.0	Fast velocity user value
Acc	Real	0.0	User value acceleration.
Dec	Real	0.0	User value negative acceleration (braking).
Stop	Bool	FALSE	Starts the positioning run

3.5.3 Outputs

Name	Data type	Default	Comment
Output			
Enabled	Bool	FALSE	1 = drive is switched ON
Started	Bool	FALSE	1 = drive has started
Homed	Bool	FALSE	1 = drive has homed Current position = 0.0
Stopped	Bool	FALSE	1 = drive stands still
Act_Pos	Real	0.0	Shows the current position
Act_Vel	Real	0.0	Shows the current velocity

3.5.4 Usage

The following step chain describes how the FB works and in which order which bits have to be switched.

Annotation: For your own safety, please ensure that the drive is free and can turn

3.5.4.1 Fast and slow velocity

1. Activation of bit MB370.1 ("Enable 2-Speed-Velocity") in OB1
2. Deactivation of bit MB370.0 ("Enable_Multi-Point") in OB1
 - a. Enables the function block
3. Activation of input "Enable_Power"
4. Enter user values:
 - a. "Slow_Speed"
 - b. "Fast_Speed"
 - c. "Acc"
 - d. "Dec"
5. Activate the "Slow" or "Fast" input
 - a. "Slow" activates the slow velocity
 - b. "Fast" activates the fast velocity
 - c. If both inputs are activated, the slow velocity has priority!
6. Activation of the "Start_Velocity" input
 - a. Starts the drive with set velocity (see point 5)
7. While driving, the velocity to be driven can now be activated/deactivated with the inputs "Slow" and "Fast"

3.5.4.2 Homing

1. Activation of bit MB370.1 ("Enable 2-Speed-Velocity") in OB1
2. Deactivation of bit MB370.0 ("Enable_Multi-Point") in OB1
 - a. Enables the function block
3. Activation of input "Enable_Power"
4. Activation of the "Start_Home" input
 - a. Homing is complete when the "Act_Pos" output shows 0
 - b. The "Homed" output has the status TRUE

3.6 Variables

All variables used in this example program are listed and commented on below.

Name	Data type	Address	Comment
General			
System_Byte	Byte	%MB0	
FirstScan	Bool	%M0.0	Changes the status during the first scan after switching on the CPU
DiagStatusUpdate	Bool	%M0.1	Changes the status during a diagnostic status update
AlwaysTRUE	Bool	%M0.2	Is always "TRUE"
AlwaysFALSE	Bool	%M0.3	Is always "FALSE"
Clock_Byte	Byte	%MB1	
Clock_10Hz	Bool	%M1.0	Changes the status at 10Hz intervals
Clock_5Hz	Bool	%M1.1	Changes the status at 5Hz intervals
Clock_2.5Hz	Bool	%M1.2	Changes the status at 2.5Hz intervals
Clock_2Hz	Bool	%M1.3	Changes the status at 2Hz intervals
Clock_1.25Hz	Bool	%M1.4	Changes the status at 1.25Hz intervals
Clock_1Hz	Bool	%M1.5	Changes the status at 1Hz intervals
Clock_0.625Hz	Bool	%M1.6	Changes the status at 0.625Hz intervals
Clock_0.5Hz	Bool	%M1.7	Changes the status at 0.5Hz intervals
Enable			
Enable Multi-Point	Bool	%M370.0	Activates the multi-point positioning and blocks the 2 speed velocity function
Enable 2-Speed-Velocity	Bool	%M370.1	Activates the 2 speed velocity function and blocks multi-point positioning
Multi-point positioning			
Power_Multi_Pos	%M100.0	Bool	Switches the drive ON
Home_Multi_Pos	%M100.1	Bool	Starts the homing function
Start_Multi_Pos	%M100.2	Bool	Starts the positioning run
Stop_Multi_Pos	%M100.3	Bool	Stops the positioning run
Powered_Multi_Pos	%M110.0	Bool	Saves the current status of the "Power_On" output
Run_Multi_Pos	%M110.1	Bool	The drive is in the active positioning run
ActPos_Multi_Pos	%MD112	Real	Shows the current position
ActVel_Multi_Pos	%MD116	Real	Displays the current velocity
Pos1_Multi_Pos	%MD120	Real	Target position 1
Pos2_Multi_Pos	%MD124	Real	Target position 2
Pos3_Multi_Pos	%MD128	Real	Target position 3
Vel_Multi_Pos	%MD132	Real	Velocity while movement Can be adjusted and changed while movement
Acc_Multi_Pos	%MD136	Real	Acceleration while movement Can be adjusted and changed while movement
Dec_Multi_Pos	%MD140	Real	Negative acceleration while movement Can be adjusted and changed while movement
Name	Data type	Address	Comment

2-Speed-Velocity			
2_Speed_Vel_Power	%M300.0	Bool	Switches the drive ON
2_Speed_Vel_Start_Home	%M300.1	Bool	Starts the homing function of the drive
2_Speed_Vel_Start	%M300.2	Bool	Starts the servo drive
2_Speed_Vel_Slow	%M300.3	Bool	Activates the slow velocity
2_Speed_Vel_Fast	%M300.4	Bool	Activates the fast velocity
2_Speed_Vel_SlowSpeed	%MD302	Real	Low velocity Can be adjusted and changed while movement
2_Speed_Vel_FastSpeed	%MD306	Real	Fast velocity Can be adjusted and changed while movement
2_Speed_Vel_Acc	%MD310	Real	Acceleration while movement Can be adjusted and changed while movement
2_Speed_Vel_Dec	%MD314	Real	Negative acceleration while movement Can be adjusted and changed while movement
2_Speed_Vel_Stop	%M300.5	Bool	Stops the drive
2_Speed_Vel_Enabled	%M300.6	Bool	Drive is switched ON
2_Speed_Vel_Started	%M300.7	Bool	Drive has started
2_Speed_Vel_Homed	%M301.0	Bool	Drive is homed
2_Speed_Vel_Stopped	%M301.1	Bool	Drive is stopped
2_Speed_Vel_ActPos	%MD318	Real	Shows the current position
2_Speed_Vel_ActVel	%MD322	Real	Shows the current velocity