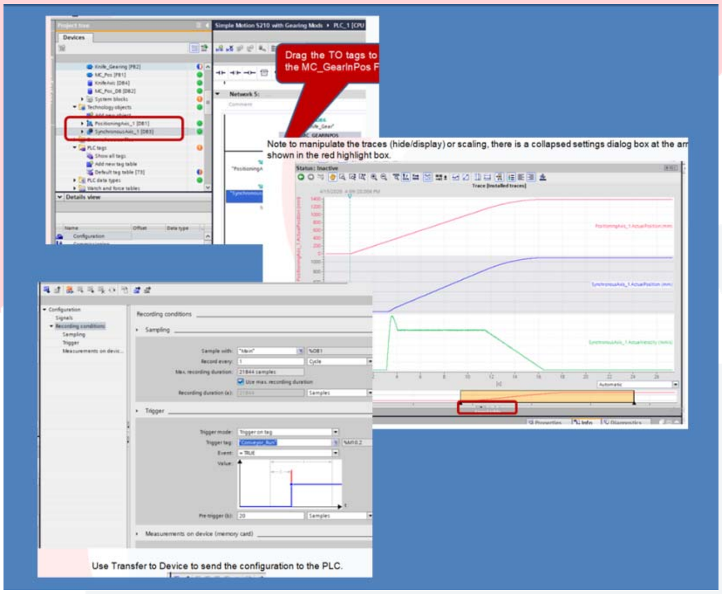
S210 Simple Motion with Gearing S7-1500T using TIA Portal V16 and Startdrive

Revision Date: 4-23-2020

This application note builds on the C&E Advanced Technologies application note “S210 Simple Motion” and will go into detail on adding a second S210 drive and motor, a Synchronous Technology Object and introduce many “value added” Portal usage tips along the way.

The ultimate goal of the exercise is to add a servo axis that can achieve a position synchronized electronic gearing relationship with the Master axis. Along the way, tips on finding the appropriate variables will be mentioned. Additionally the use of various motion control blocks and strategies will be discussed. Illustrated use of the oscilloscope Trace function is provided and a few notes on debugging typical start-up issues.



Drag the TO tags to the MC_GearInPos F

Note to manipulate the traces (hide/display) or scaling, there is a collapsed settings dialog box at the arr shown in the red highlight box.

Use Transfer to Device to send the configuration to the PLC.

All sample code illustrated is expected to be used

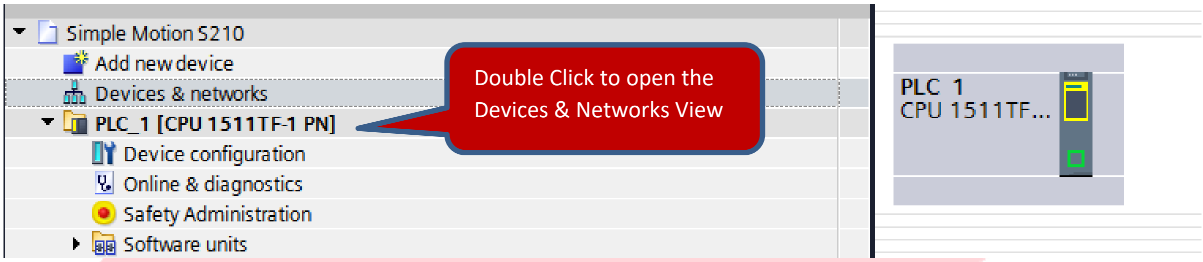
only on benchtop demo equipment without possibility

of harm to equipment or personnel.

Launch TIA Portal V16 and open the Project View.

In order to add a second servo drive, and referring back to the seed project document “S210 Simple Motion”, select the appropriate S210 drive from the Hardware Catalog at the right, illustrated below.

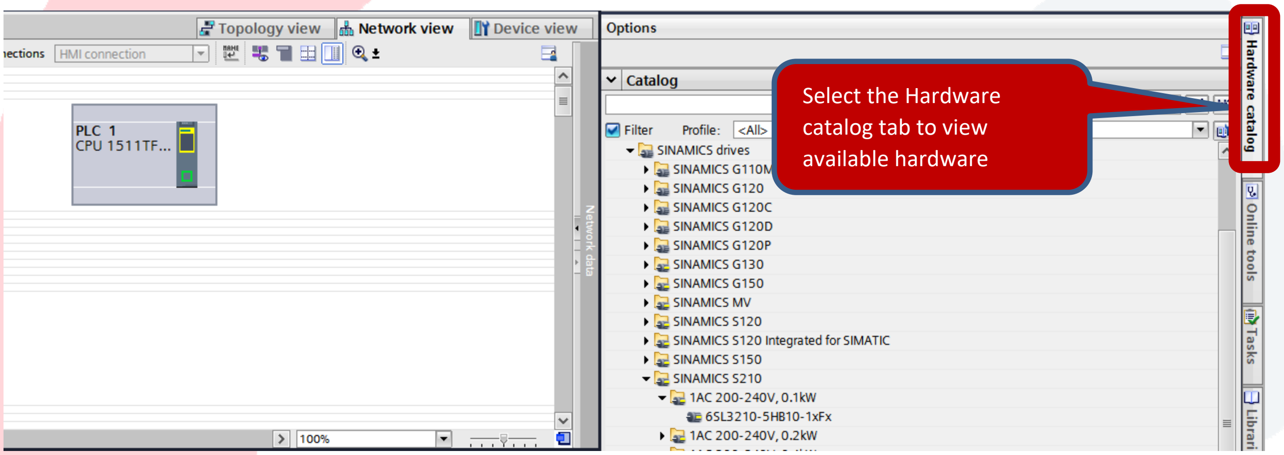
Open the Devices & Networks view to add the S210 Drive



- Simple Motion S210
  - Add new device
  - Devices & networks
  - PLC_1 [CPU 1511TF-1 PN]
    - Device configuration
    - Online & diagnostics
    - Safety Administration
    - Software units

Double Click to open the Devices & Networks View

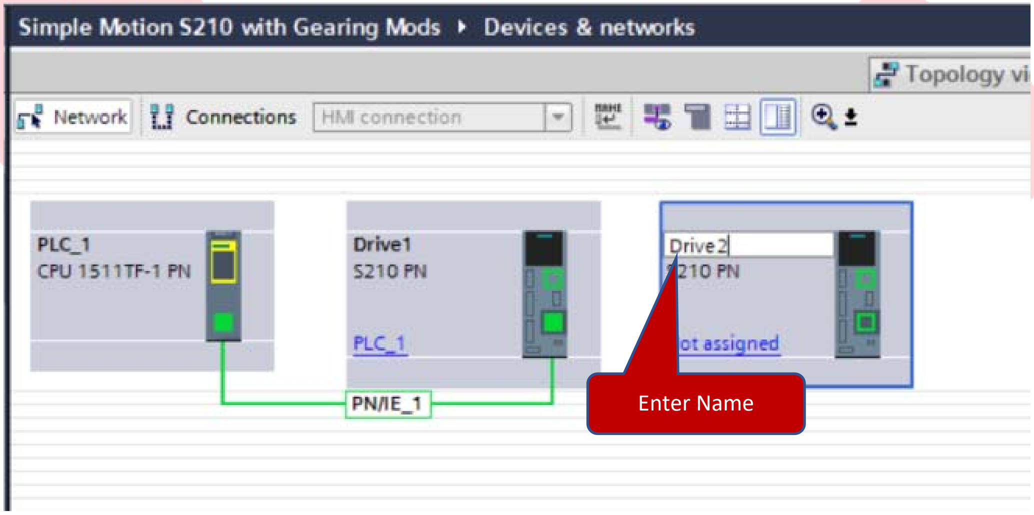
In the Devices and Networks Window, Drag and Drop in your S210 Drive model from the hardware catalog.



- SINAMICS drives
  - SINAMICS G110M
  - SINAMICS G120
  - SINAMICS G120C
  - SINAMICS G120D
  - SINAMICS G120P
  - SINAMICS G130
  - SINAMICS G150
  - SINAMICS MV
  - SINAMICS S120
  - SINAMICS S120 Integrated for SIMATIC
  - SINAMICS S150
  - SINAMICS S210
    - 1AC 200-240V, 0.1kW
      - 6SL3210-5HB10-1xFx
    - 1AC 200-240V, 0.2kW

Select the Hardware catalog tab to view available hardware

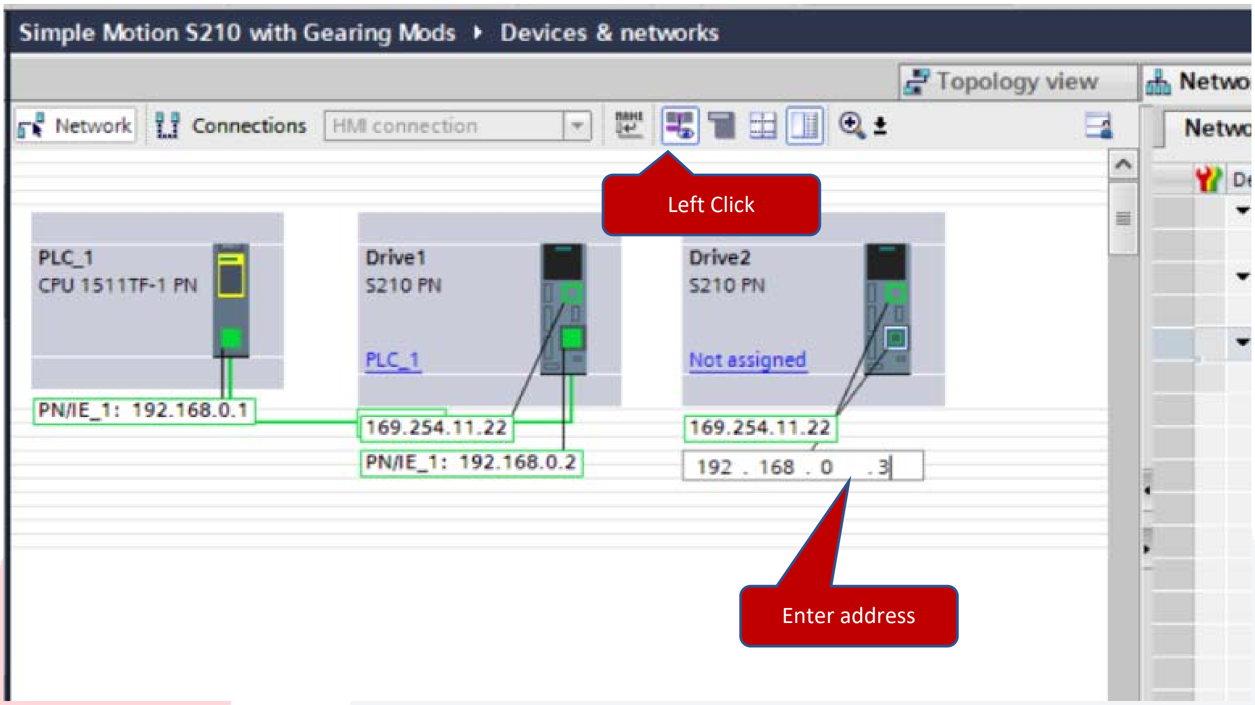
This will be the second S210 drive in the hardware configuration so name that drive “Drive2” by clicking on the default drive name given in the Network View where the drive is dropped into the project.



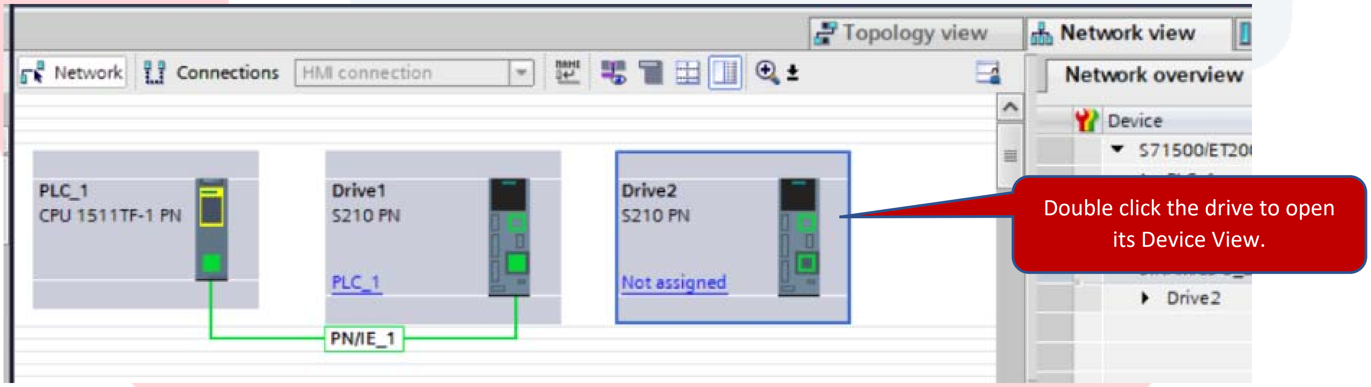
Simple Motion S210 with Gearing Mods ▸ Devices & networks

Enter Name

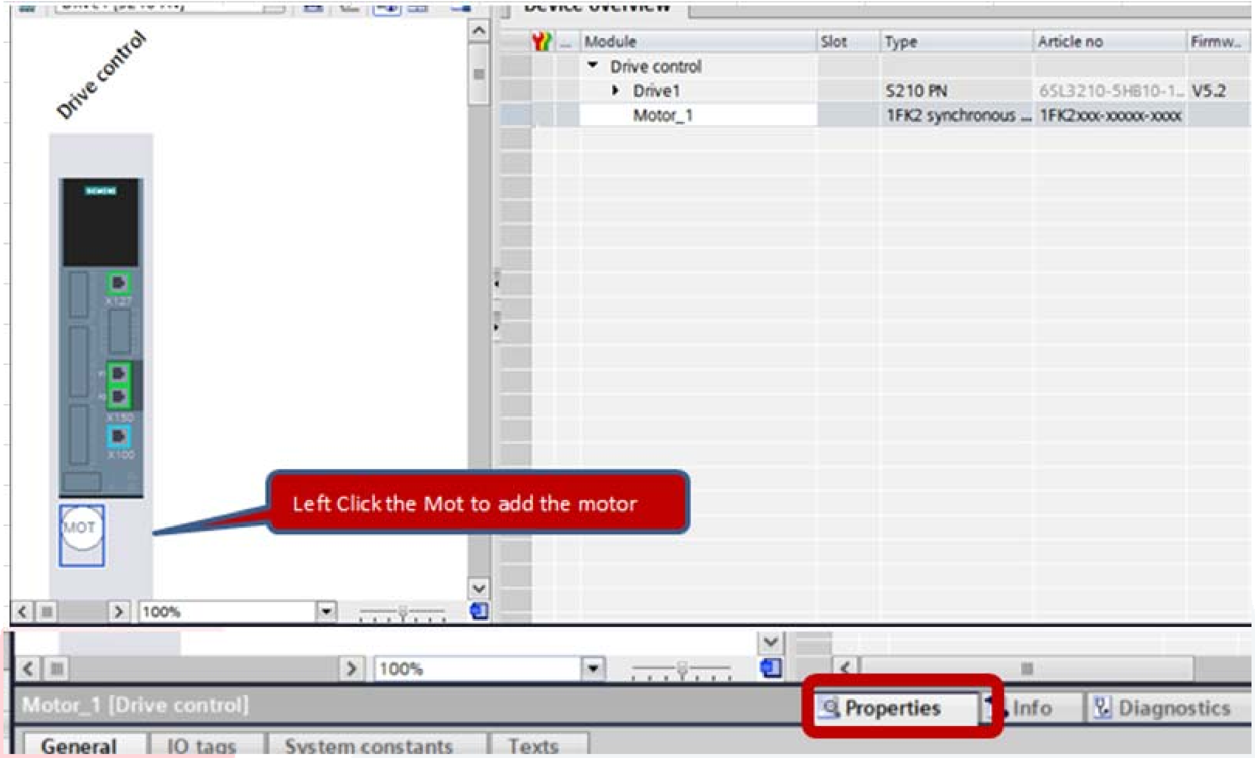
This drive should have an X150 address 192.168.0.3 so click on the icon to display the IP addresses and then by clicking directly on the displayed address, change the address as desired. Do not modify the 169.254.11.22 address; this is the Siemens service port default address for Port X127 on Sinamics drives.



Double Click on Drive2 to open the Device View for Drive2 to assign the servo motor next.



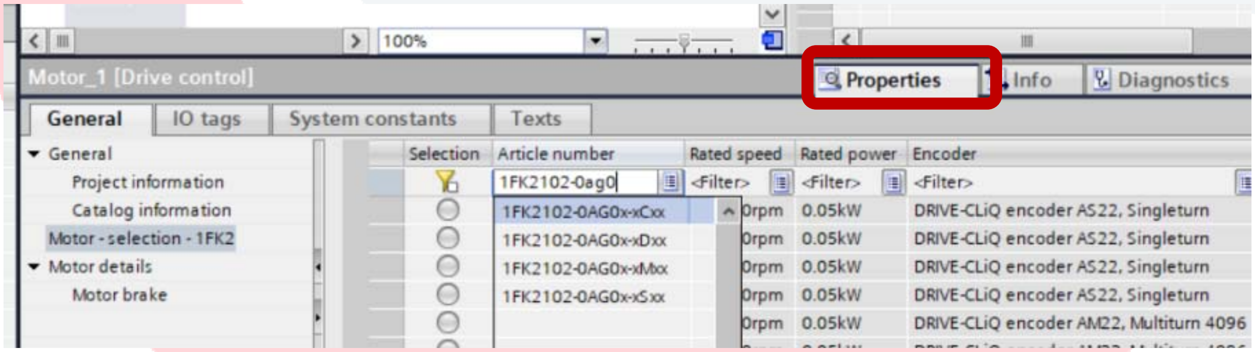
Click specifically on the motor outline to get the appropriate motor selections in the Properties view.



Module	Slot	Type	Article no	Firmw...
Drive control				
Drive1		S210 PN	6SL3210-5HB10-1...	V5.2
Motor_1		1FK2 synchronous ...	1FK2xxx-xxxxx-xxxx	

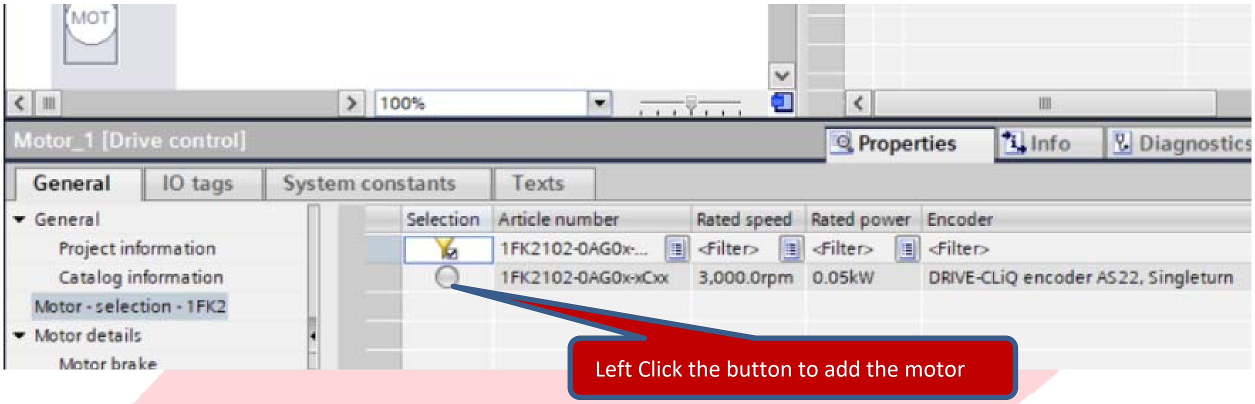
Left Click the Mot to add the motor

If you want to filter out the large list of motors, start typing in your motor part number in the "filter" until the matching list shows your motor and appropriate feedback type. Any value can be used where an x is shown. Even after picking that motor, the next click is what puts the motor into the project.



Selection	Article number	Rated speed	Rated power	Encoder
	1FK2102-0ag0	<Filter>	<Filter>	<Filter>
○	1FK2102-0AG0x-xCxx	0rpm	0.05kW	DRIVE-CLiQ encoder AS22, Singleturn
○	1FK2102-0AG0x-xDxx	0rpm	0.05kW	DRIVE-CLiQ encoder AS22, Singleturn
○	1FK2102-0AG0x-xMxx	0rpm	0.05kW	DRIVE-CLiQ encoder AS22, Singleturn
○	1FK2102-0AG0x-xSxx	0rpm	0.05kW	DRIVE-CLiQ encoder AS22, Singleturn
○		0rpm	0.05kW	DRIVE-CLiQ encoder AM22, Multiturn 4096

Click on the “selection” button to assign this motor to the drive hardware.

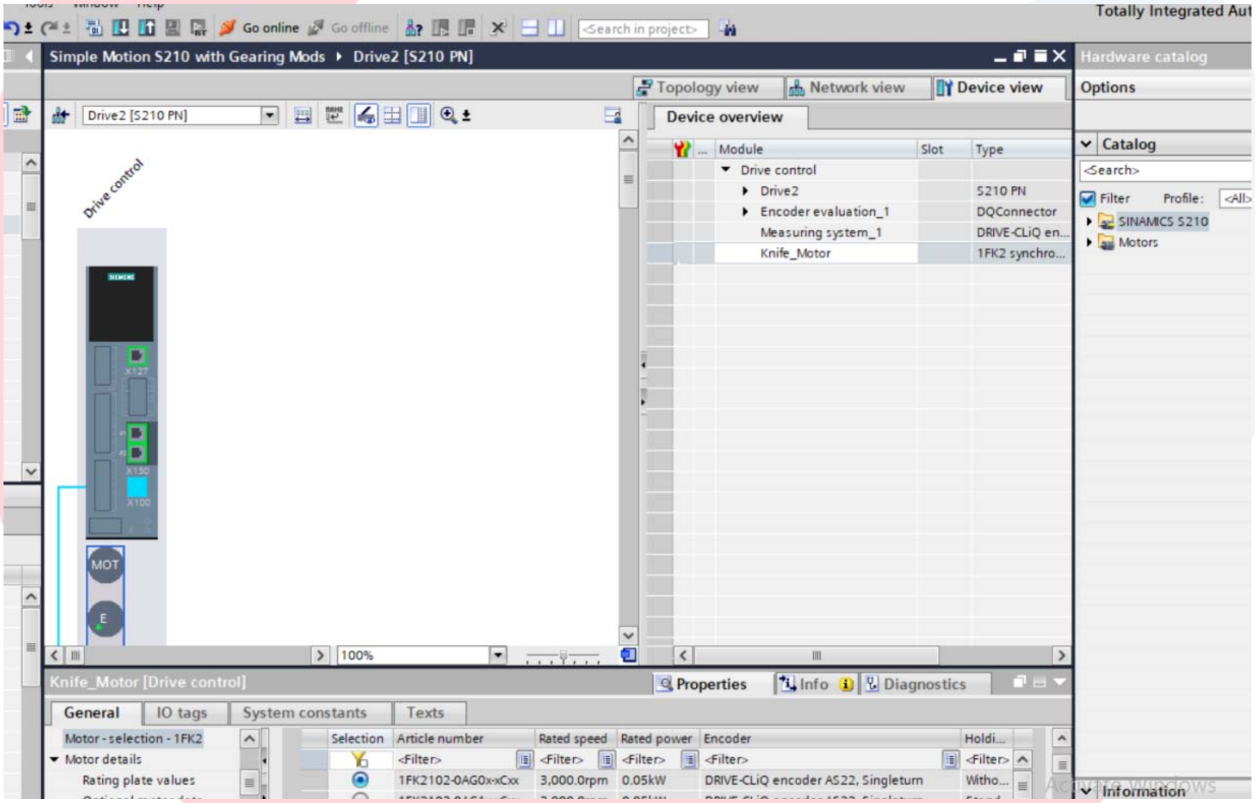


Motor_1 [Drive control]

Selection	Article number	Rated speed	Rated power	Encoder
	1FK2102-0AG0x-...	<Filter>	<Filter>	<Filter>
○	1FK2102-0AG0x-xCxx	3,000.0rpm	0.05kW	DRIVE-CLiQ encoder AS22, Singleturn

Left Click the button to add the motor

You should now have a motor and feedback device defined for Drive2 as shown below. Additionally the motor’s name on Drive2 is changed to “Knife_Motor” below.



Simple Motion S210 with Gearing Mods ▸ Drive2 [S210 PN]

Device overview

Module	Slot	Type
Drive control		
Drive2		S210 PN
Encoder evaluation_1		DQConnector
Measuring system_1		DRIVE-CLiQ en...
Knife_Motor		1FK2 synchro...

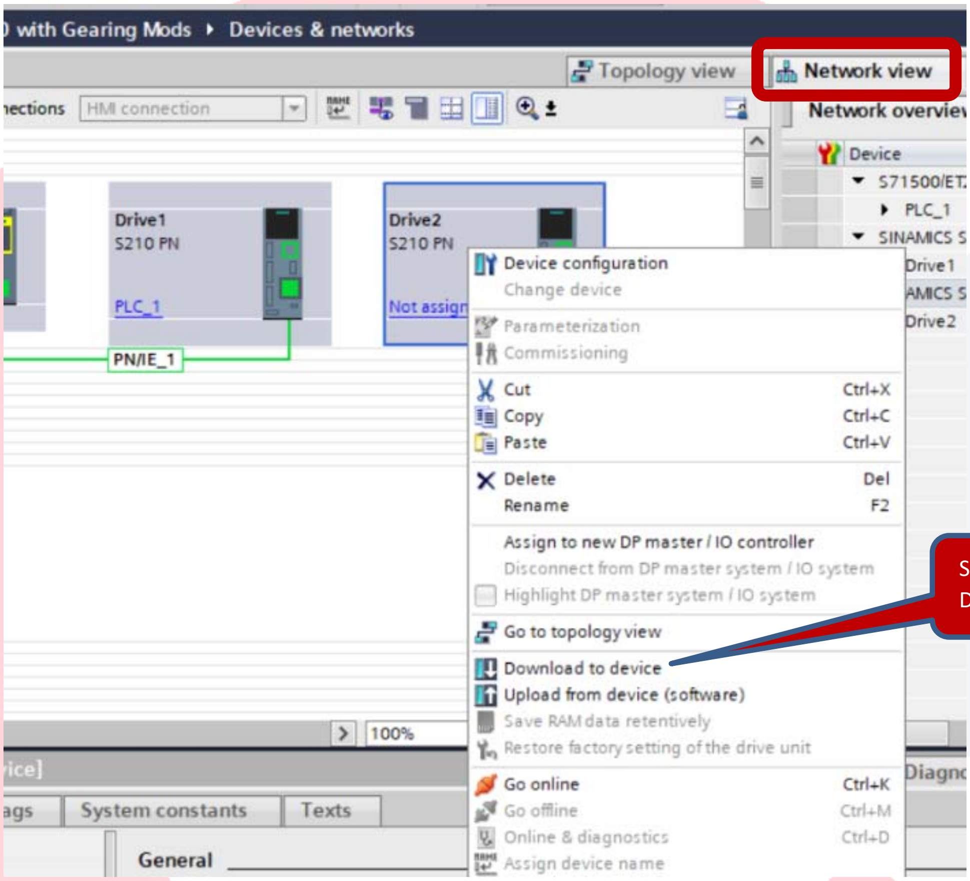
Knife_Motor [Drive control]

Selection	Article number	Rated speed	Rated power	Encoder	Holdi...
	<Filter>	<Filter>	<Filter>	<Filter>	<Filter>
●	1FK2102-0AG0x-xCxx	3,000.0rpm	0.05kW	DRIVE-CLiQ encoder AS22, Singleturn	Witho...

Next download to Drive2 specifically to install the basic parameters along with the IP Address and the Profinet Name of the device.

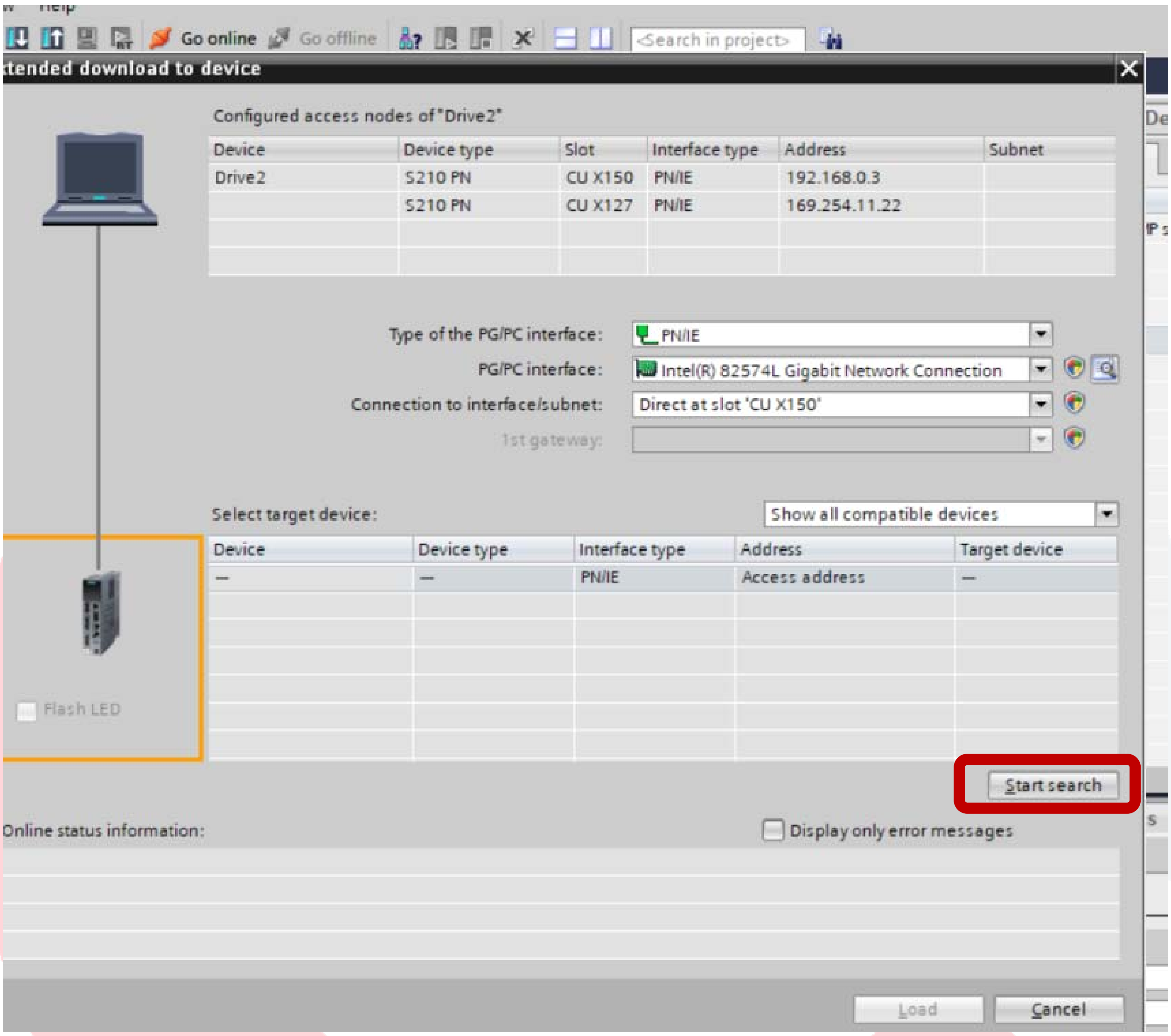
Make sure your Ethernet cable is plugged into one of the X150 ports on the Drive or at least on the common Profinet network that includes this drive.

In the “Network View” tab, as shown below, right click once on the Drive2 box to see the options menu that includes the Download to that specific device. Select the “Download to Device”.



Select Download to Device

Confirm the details of the address (192.168.0.3) and the Name (Drive2) and then begin the “Start Search” for the appropriate drive hardware on the network.



Configured access nodes of "Drive2"

Device	Device type	Slot	Interface type	Address	Subnet
Drive2	S210 PN	CU X150	PN/IE	192.168.0.3	
	S210 PN	CU X127	PN/IE	169.254.11.22	

Type of the PG/PC interface: PN/IE

PG/PC interface: Intel(R) 82574L Gigabit Network Connection

Connection to interface/subnet: Direct at slot 'CU X150'

1st gateway:

Select target device: Show all compatible devices

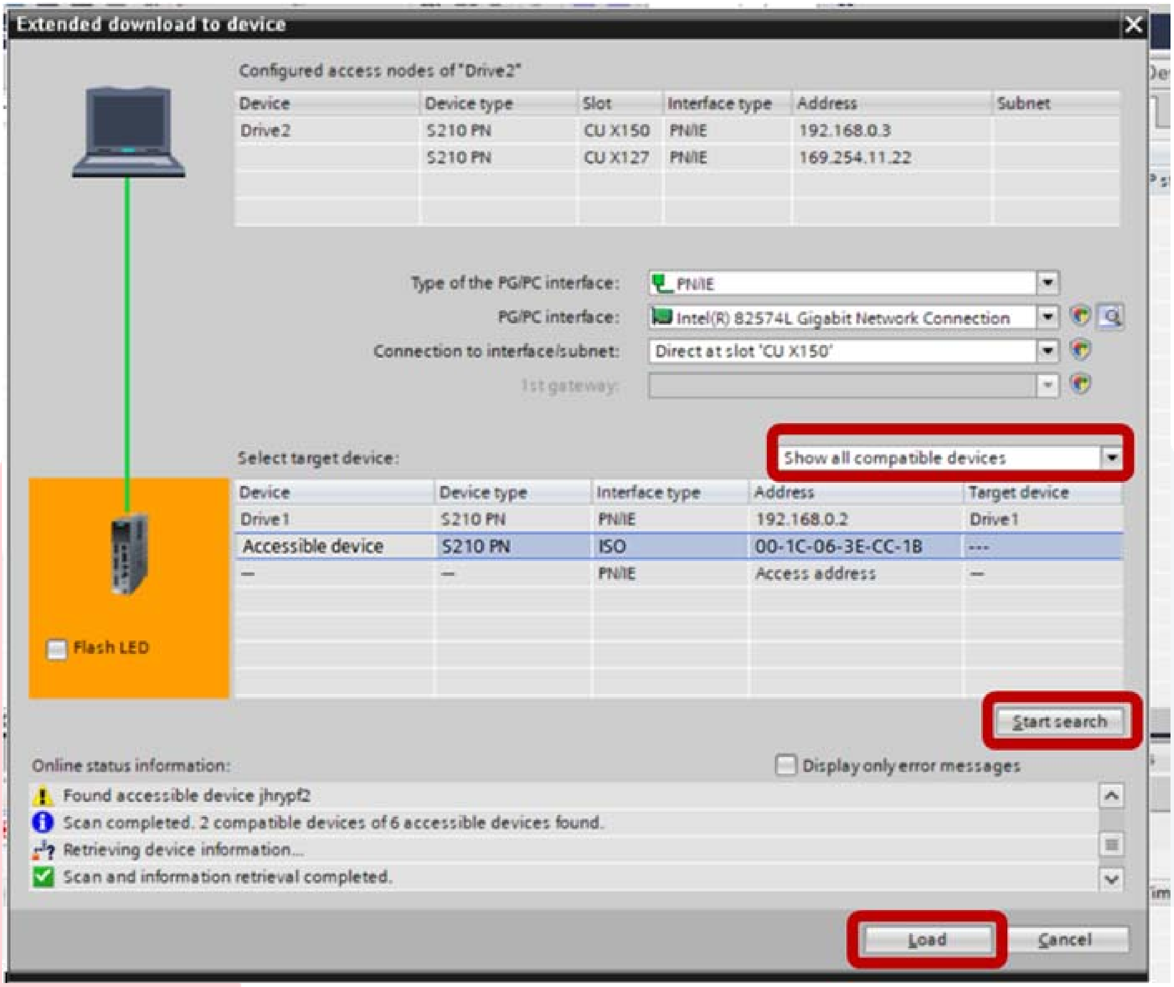
Device	Device type	Interface type	Address	Target device
—	—	PN/IE	Access address	—

□ Flash LED

Online status information:

□ Display only error messages

With “Show all compatible devices” selected for the Search, any S210 on the network should show up. Left click on the drive that corresponds and then select “Load”. With a factory fresh drive the address is the MAC address of the device.



Extended download to device

Configured access nodes of "Drive2"

Device	Device type	Slot	Interface type	Address	Subnet
Drive2	S210 PN	CU X150	PN/IE	192.168.0.3	
	S210 PN	CU X127	PN/IE	169.254.11.22	

Type of the PG/PC interface: PN/IE

PG/PC interface: Intel(R) 82574L Gigabit Network Connection

Connection to interface/subnet: Direct at slot 'CU X150'

1st gateway:

Select target device: Show all compatible devices

Device	Device type	Interface type	Address	Target device
Drive1	S210 PN	PN/IE	192.168.0.2	Drive1
Accessible device	S210 PN	ISO	00-1C-06-3E-CC-1B	---
—	—	PN/IE	Access address	—

□ Flash LED

Start search

Online status information:

□ Display only error messages

Found accessible device jhrypf2

Scan completed. 2 compatible devices of 6 accessible devices found.

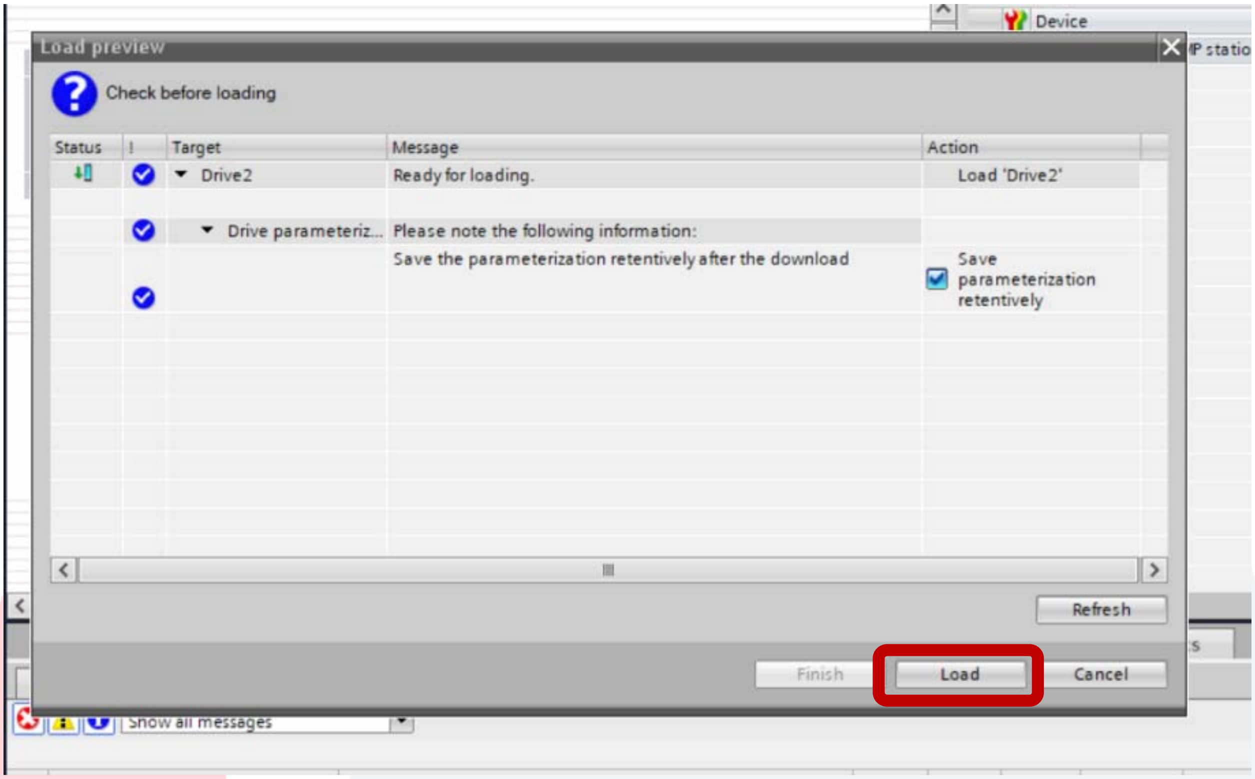
Retrieving device information...

Scan and information retrieval completed.

Load

Cancel

Confirm the “Save parameterization retentively” is selected and select the final “Load” button at the bottom of the window.

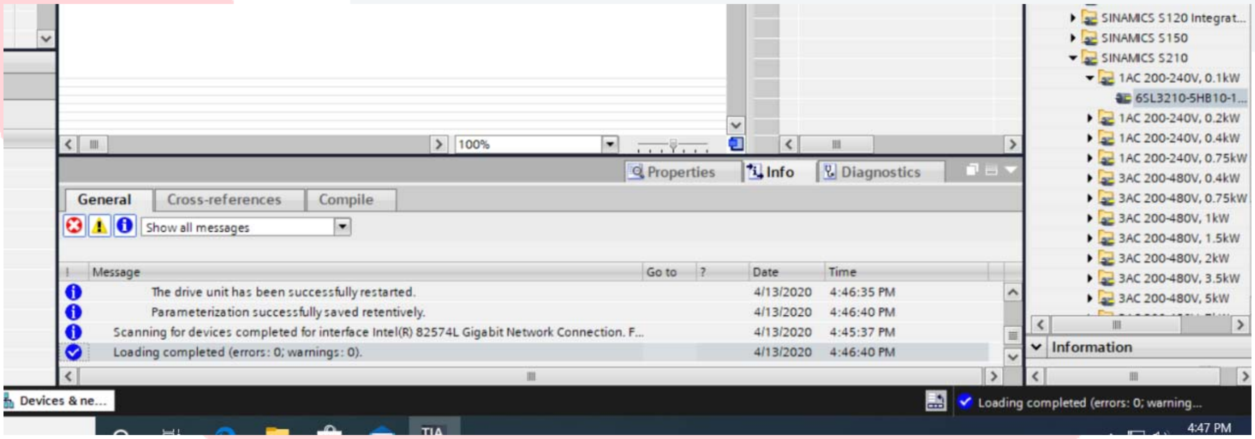


Load preview

Check before loading

Status	!	Target	Message	Action
	✓	▾ Drive2	Ready for loading.	Load 'Drive2'
	✓	▾ Drive parameteriz...	Please note the following information:	
			Save the parameterization retentively after the download	☑ Save parameterization retentively
	✓			

Portal provides some useful feedback as to the success of the hardware download in the “Info” dialog box and at the bottom right of the window:

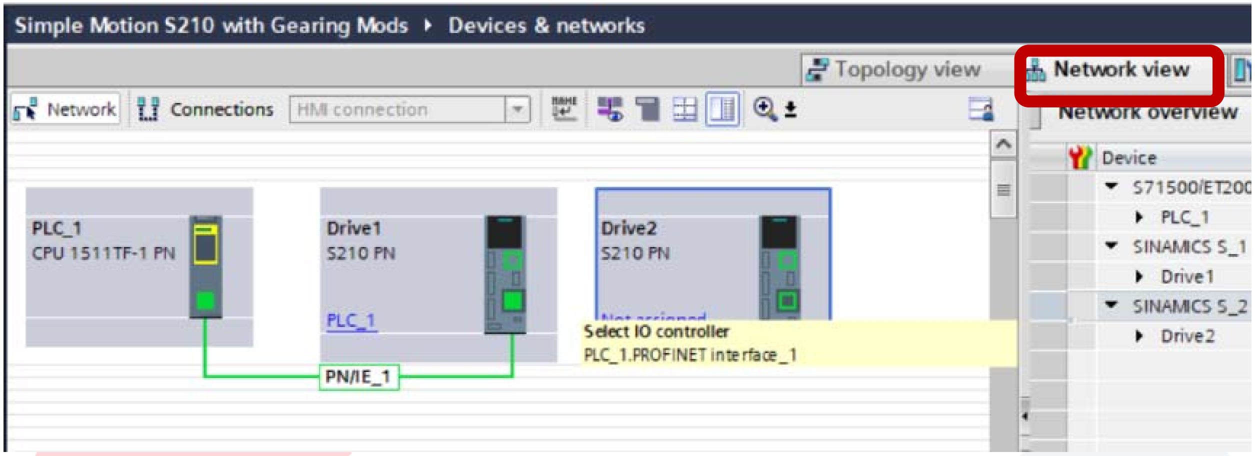


!	Message	Go to	?	Date	Time
ⓘ	The drive unit has been successfully restarted.			4/13/2020	4:46:35 PM
ⓘ	Parameterization successfully saved retentively.			4/13/2020	4:46:40 PM
ⓘ	Scanning for devices completed for interface Intel(R) 82574L Gigabit Network Connection. F...			4/13/2020	4:45:37 PM
✓	Loading completed (errors: 0; warnings: 0).			4/13/2020	4:46:40 PM

✓ Loading completed (errors: 0; warning...

Next, configure the Profinet Connection between Drive2 and PLC 1.

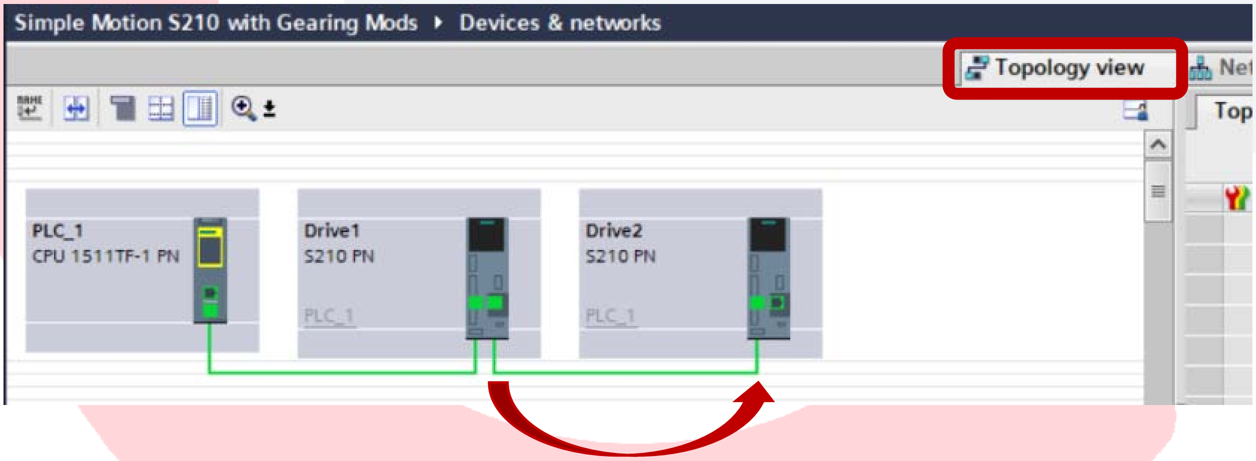
Open the "Network View" tab. Left Click on Not Assigned on the Drive and select your PLC as the IO controller. Left Click PLC_1.PROFINET interface_1 to connect the two devices.



Select IO controller
PLC_1.PROFINET interface_1

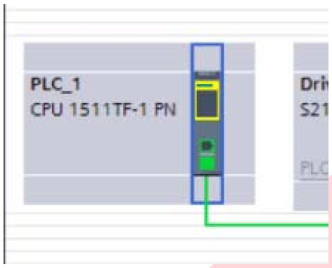
The Topology needs to match the exact physical port to port connections between the PLC and the Drive with Technology Objects and motion hardware when using Profinet IRT (Isochronous Real-Time) communications.

In this example, Port2 of Drive1 is connected to Port1 of Drive2. Select the "Topology View" tab. Left click on Drive1, Port2 and drag a connection from Drive1 to Port 1 of Drive2 as shown below.

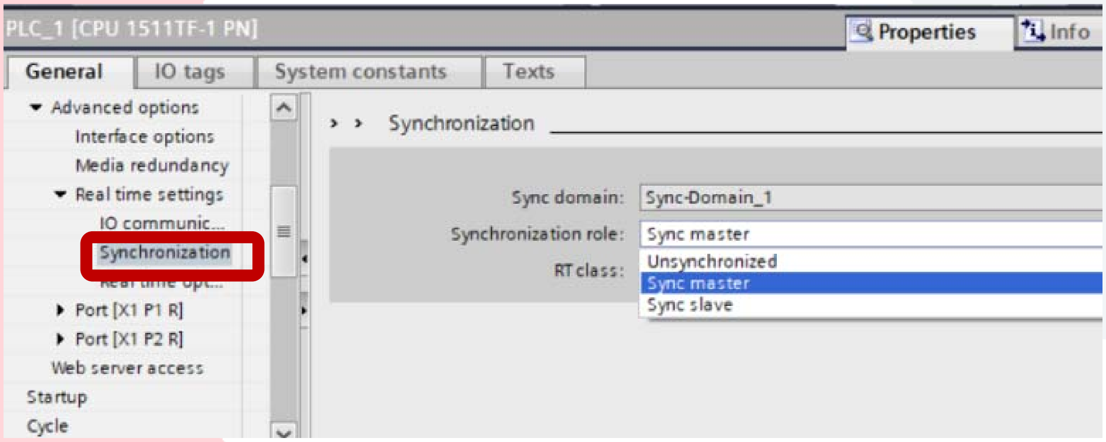


The next steps establish the IRT relationship and timing. This was covered in detail in the "S210 Simple Motion" app note, so will only be covered in bullet points here.

Click specifically on the PLC in the PLC_1 box "to put focus on the PLC" in the "Topology View" tab.



- At the PLC Properties dialog box
    - Expand "Profinet interface X1"
    - Expand "Advanced options"
    - Expand "Real time settings"
    - Select "Synchronization"
    - Confirm the PLC is set as the "Sync master"

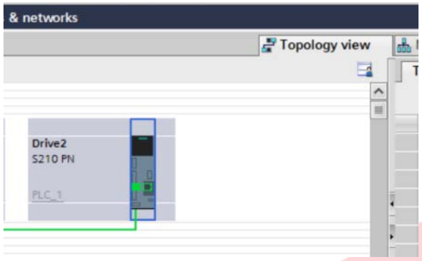


Sync domain: Sync-Domain_1

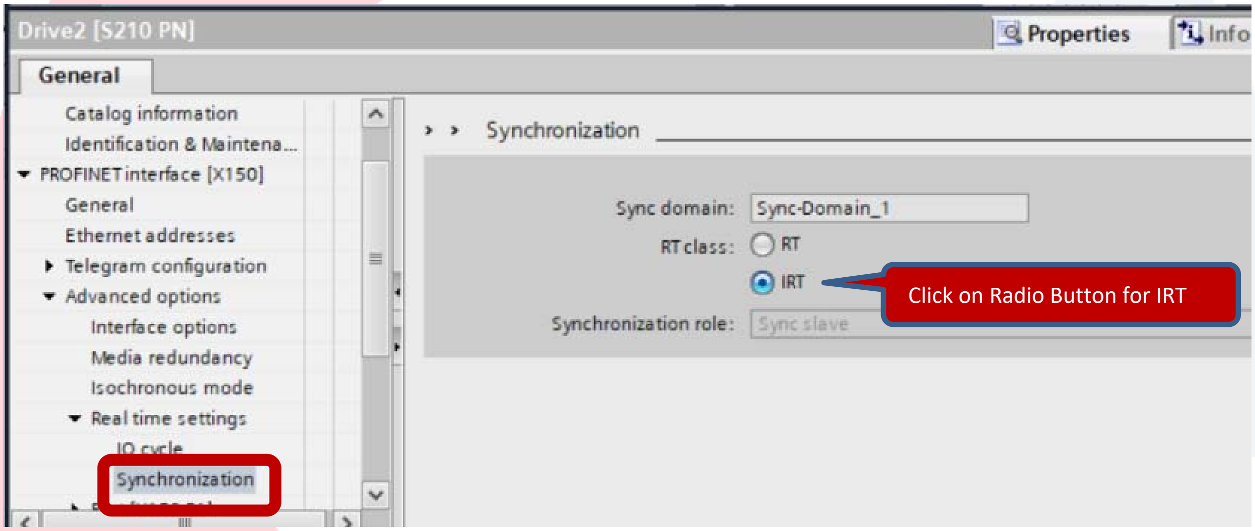
Synchronization role: Sync master

RT class:

Click specifically on the Drive2 in the Drive2 box in the “Topology View”.



- At the Drive2 Properties dialog box
  - o Expand “Profinet interface X150”
  - o Expand “Advanced options”
  - o Expand “Real time settings”
  - o Select “Synchronization”
  - o Select the “IRT” designation.



Synchronization

Sync domain: Sync-Domain_1

RT class: ○ RT ● IRT

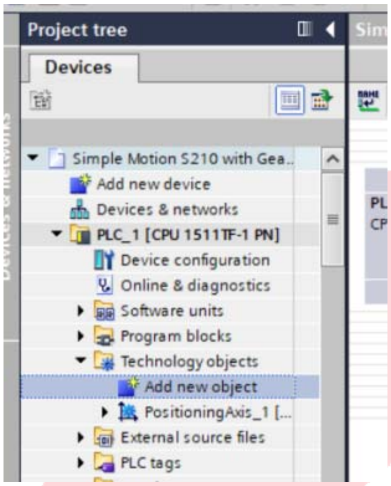
Synchronization role: Sync slave

Click on Radio Button for IRT

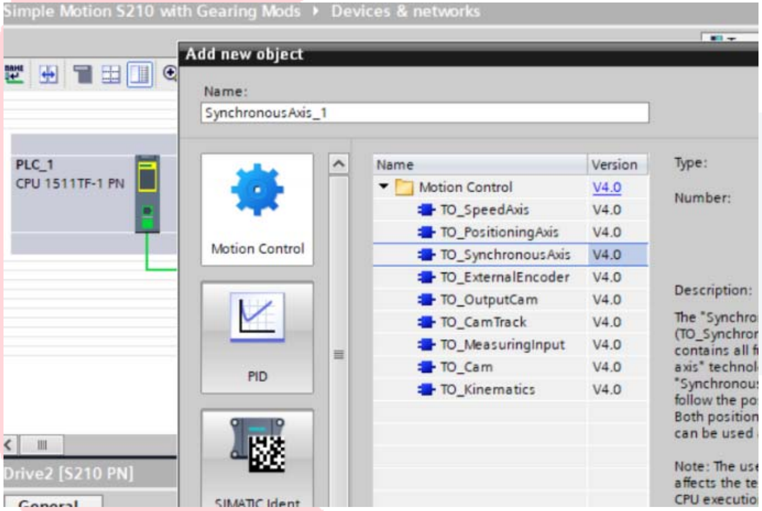
This is a good time to perform a project file Save.

Now that the hardware is defined for the second drive, an appropriate Technology Object needs added to the project and a link defined to assign this drive hardware to that Technology Object (TO). Since this TO will need to perform motion directly linked to another axis’ position, the TO type “TO_SynchronousAxis” will be used. Configuration of the TO is very similar to the TO_PositioningAxis used previously, with the additional need to define what other axes can be used as a master reference for the motion that will be position based and following the master position of another axis. Understanding this master axis will have a “Commanded Position” and an “Actual Position” based directly on the current encoder position during the move, a selection can be made to use the position reference that makes the most sense for the application.

Under the PLC in the project tree at the left, expand “Technology Objects” and double click on “Add new object”.



Select the “TO_SynchronousAxis” type and select OK.



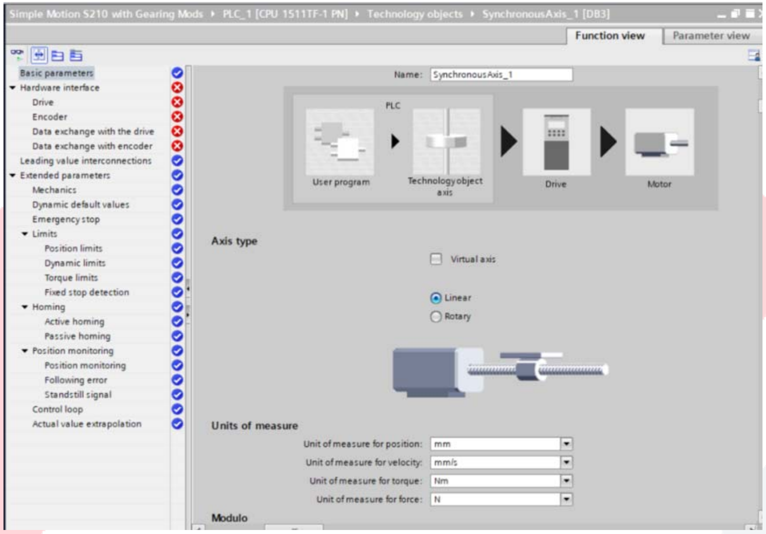
Name:
SynchronousAxis_1

Name	Version
Motion Control	V4.0
TO_SpeedAxis	V4.0
TO_PositioningAxis	V4.0
TO_SynchronousAxis	V4.0
TO_ExternalEncoder	V4.0
TO_OutputCam	V4.0
TO_CamTrack	V4.0
TO_MeasuringInput	V4.0
TO_Cam	V4.0
TO_Kinematics	V4.0

Description:
The "Synchro
(TO_Synchron
contains all f
axis" technol
"Synchronou
follow the po
Both position
can be used

Note: The use
affects the te
CPU executio

The “Function View” window will open next illustrating the needed assignments. Note this is where the TO is defined as a Virtual, Linear or a Rotary axis and is where the engineering units are selected. It is important to consider the relationship of a follower axis to a master axis during the configuration. This example uses default values for a linear axis to match the already configured PositionAxis.



Name: SynchronousAxis_1

Axis type

☐ Virtual axis

● Linear
○ Rotary

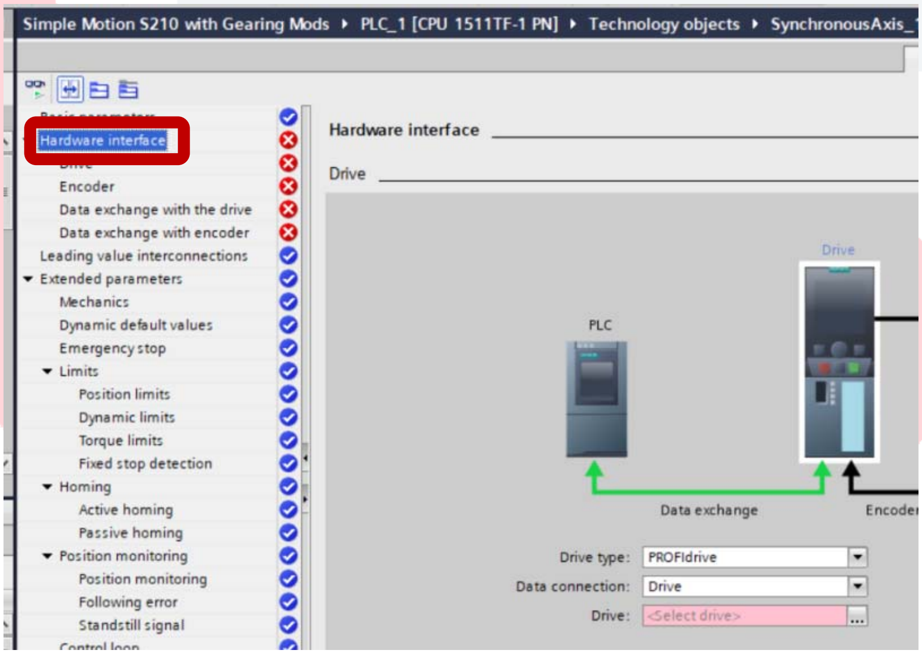
Units of measure

Unit of measure for position: mm
Unit of measure for velocity: mm/s
Unit of measure for torque: Nm
Unit of measure for force: N

Modulo

If a mechanical system has a naturally reoccurring product length, like a rotary table or flighted indexing conveyor, a repeat machine distance can be defined using the Modulo dialog. As an example, assuming a conveyor with product nest precisely occurring every 600mm, Modulo can be set for 600 and instead of incrementing the Position to 601, it will roll over, back to 0 and count only to 600. Quite useful for 360 degree applications too. See the “Modulo Addendum” at the end of this app note that builds on content still to come.

At the Project Tree, select “Hardware Interface” and select the physical drive hardware to assign to this TO as shown.

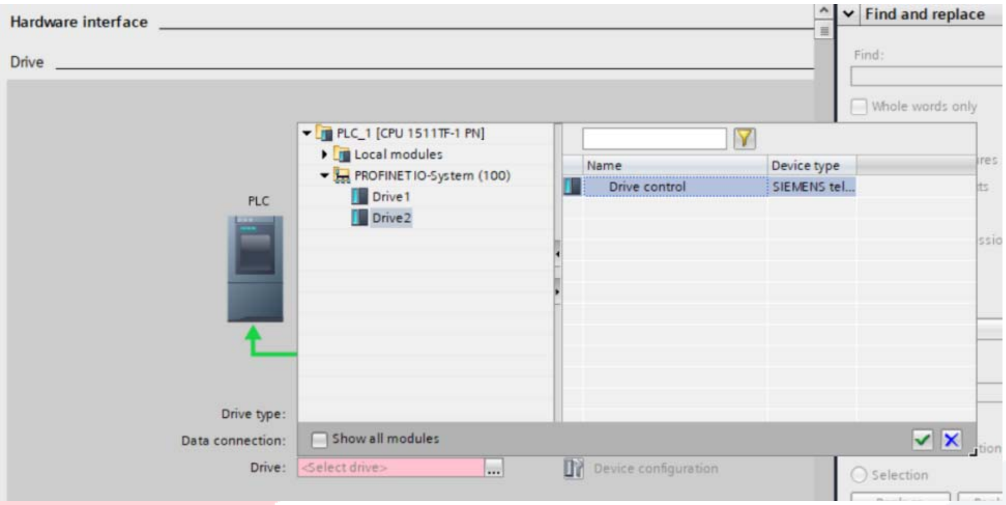


Hardware interface

Drive

Drive type: PROFIdrive
Data connection: Drive
Drive: <Select drive>

Expand ProfinetIO-System (100) and select Drive2. Use green check mark to accept.



Hardware interface

Drive

Drive type:

Data connection:

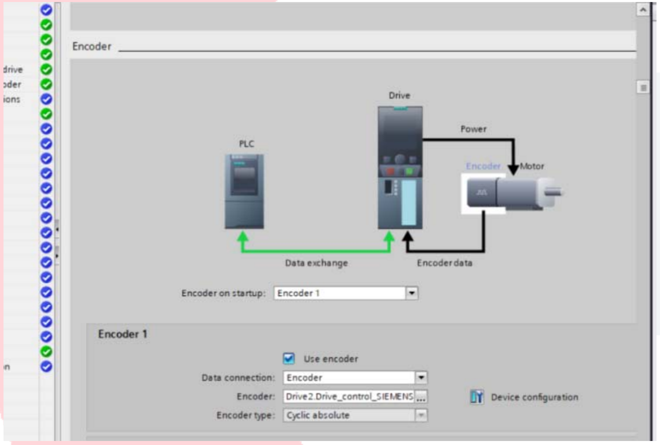
Drive: <Select drive>

- PLC_1 [CPU 1511TF-1 PN]
  - Local modules
  - PROFINET IO-System (100)
    - Drive1
    - Drive2

Name	Device type
Drive control	SIEMENS tel...

□ Show all modules

You can scroll down through the configuration and observe the encoder feedback for the TO is also assigned to Drive2 and further down observe that Portal has assigned Telegram 105 as the means for passing data cyclically between the PLC TO and the hardware.



Encoder

Encoder on startup: Encoder 1

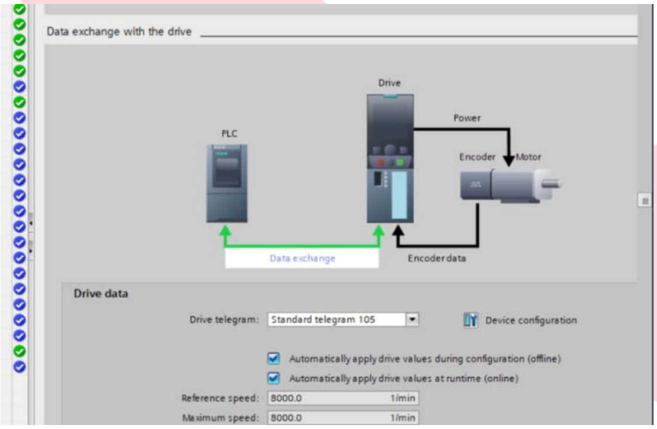
Encoder 1

☑ Use encoder

Data connection: Encoder

Encoder: Drive2.Drive_control_SIEMENS

Encoder type: Cyclic absolute



Data exchange with the drive

Drive data

Drive telegram: Standard telegram 105

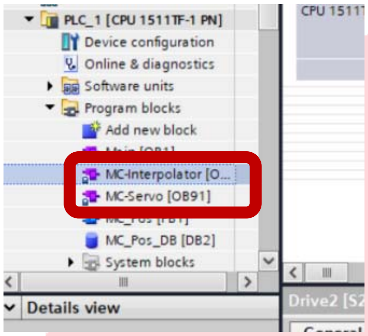
☑ Automatically apply drive values during configuration (offline)

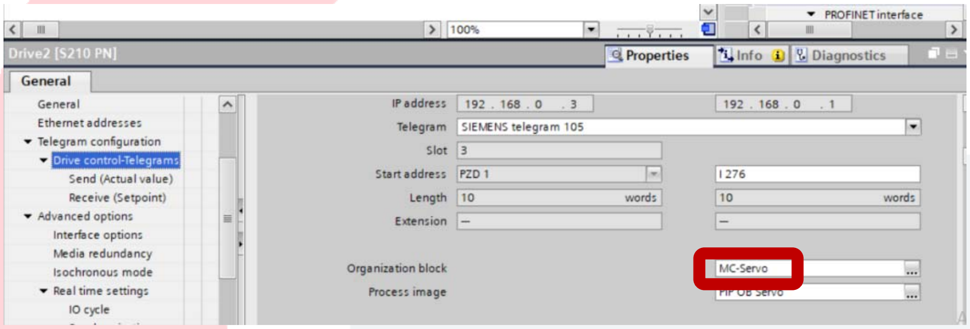
☑ Automatically apply drive values at runtime (online)

Reference speed: 8000.0 1/min

Maximum speed: 8000.0 1/min

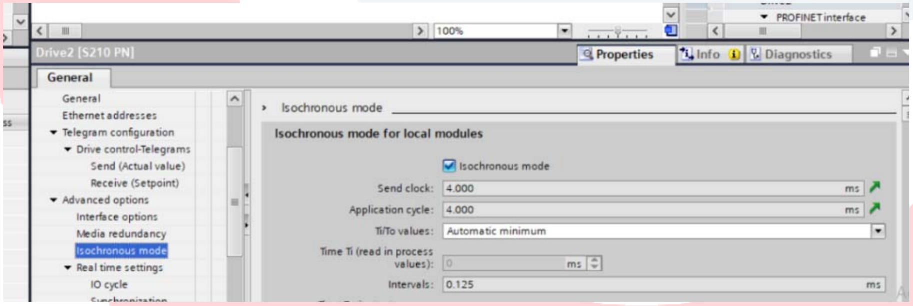
Worth mentioning, and useful for higher level troubleshooting, it is important to know Portal is performing high level configurations in the background, as long as the steps provided are followed sequentially. For example, when a servo technology object is defined, two PLC Organizational Blocks were inserted into the program. MC-Interpolator and MC-Servo. Only after the hardware is linked to the PLC in Topology View and hardware is assigned to the TO, Portal has inserted the appropriate instance of MC-Servo into the Siemens telegram data transfer configuration.





IP address: 192 . 168 . 0 . 3 | 192 . 168 . 0 . 1
Telegram: SIEMENS telegram 105
Slot: 3
Start address: PZD 1 | I 276
Length: 10 words | 10 words
Extension: — | —
Organization block: MC-Servo
Process image: PIP OB Servo

Additionally, Isochronous Mode has been selected and a matching sync time as defined earlier.

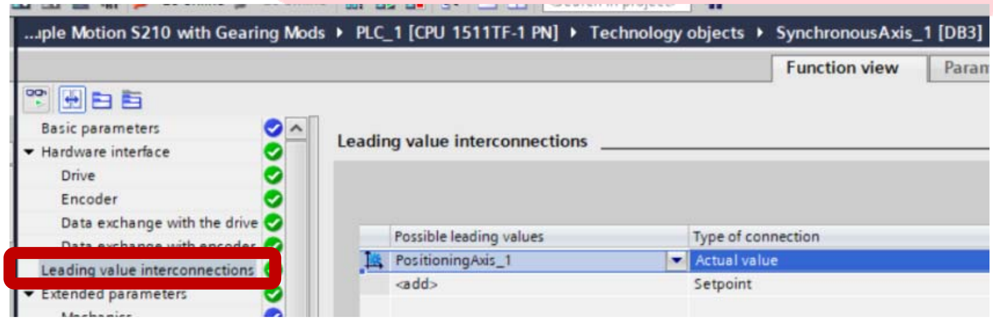


Isochronous mode

Isochronous mode for local modules

☑ Isochronous mode
Send clock: 4.000 ms
Application cycle: 4.000 ms
Ti/To values: Automatic minimum
Time Ti (read in process values): 0 ms
Intervals: 0.125 ms

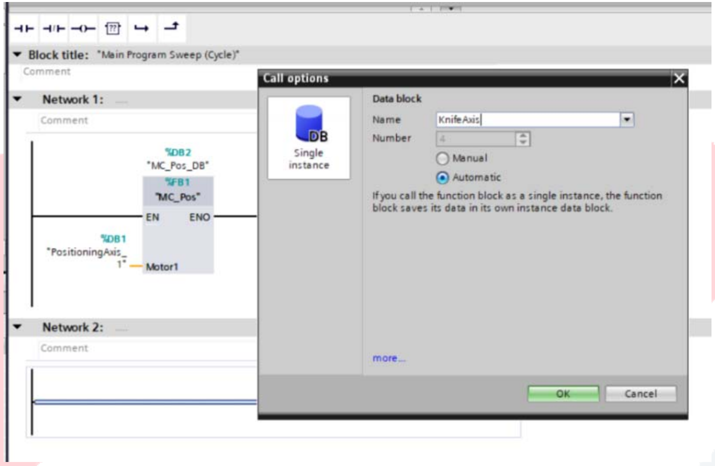
To assign the possible Master Axes for this Synchronous type TO, at “Hardware Interface” go to the “Leading Value Interconnection” and choose to allow the use of “PositioningAxis1” as a master position reference for this TO and select the “Actual Value” as the position value to use. There can be multiple axes listed.


...ple Motion S210 with Gearing Mods ▸ PLC_1 [CPU 1511TF-1 PN] ▸ Technology objects ▸ SynchronousAxis_1 [DB3]
Function view
Param
Basic parameters
Hardware interface
Drive
Encoder
Data exchange with the drive
Data exchange with encoder
Leading value interconnections
Extended parameters
Mechanics

Leading value interconnections

Possible leading values	Type of connection
PositioningAxis_1	Actual value
<add>	Setpoint

The next step takes full advantage of the programming strategy of using Multi Instance Function Blocks. Based on starting with the program developed previously from the "S210 Simple Motion" app note, there is already a Multi Instance FB for MC_Power, MC_Reset, Jog, Move Incremental, Move Absolute, Home, and Halt. There is no reason to build that PLC code again. Simply drop a second instance of that FB into OB1 on Network 2 for example. When the dialog box opens, assign a new name called KnifeAxis for this exercise.



Data block

Name: KnifeAxis

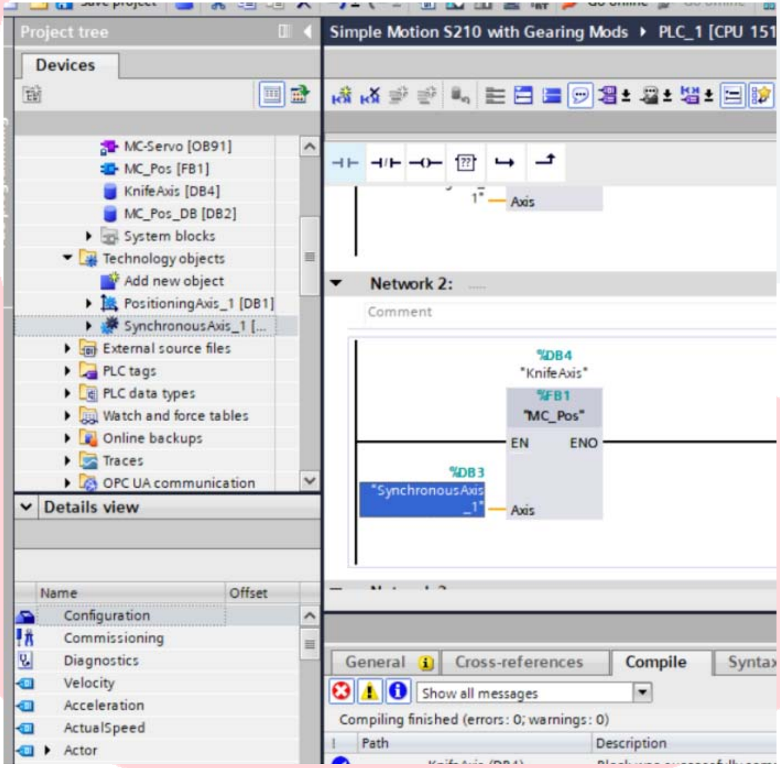
Number: 4

○ Manual

● Automatic

If you call the function block as a single instance, the function block saves its data in its own instance data block.

Drag the "SynchronousAxis_1" TO from Technology Objects in the Project Tree onto the "Axis" input of the MC_Pos function block. Now there is a duplicate set of motion control blocks available for the new axis.

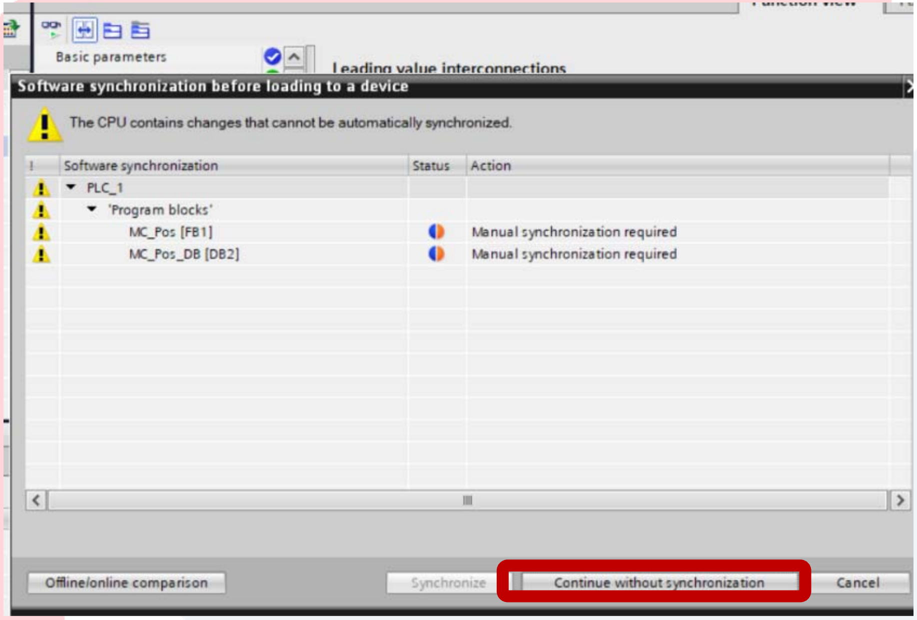


Compiling finished (errors: 0; warnings: 0)

The next step will compile all the existing code and download to the PLC and allow test operation of the second servo axis.

Click on PLC_1 in the project tree at the left. Either use the Download icon in the top menu bar or right click on PLC_1 in the tree and select "Download software and hardware" and then perform the steps to search for the PLC and Load to the PLC. When there is an existing program running in the PLC, Portal attempts to synchronize the new changes with the existing scanning PLC code. With major changes such as adding a new axis, that is not possible and the user must allow a complete, unsynchronized (PLC Stop) download.

Select "Continue without synchronization" at this point. Agree to Stop the PLC and then agree to Start the PLC upon download completion.



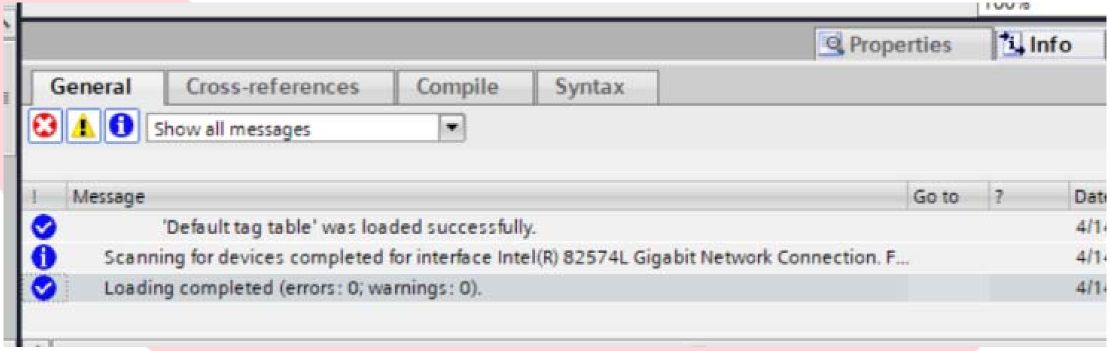
Software synchronization before loading to a device

The CPU contains changes that cannot be automatically synchronized.

!	Software synchronization	Status	Action
⚠	PLC_1		
⚠	'Program blocks'		
⚠	MC_Pos [FB1]	◐	Manual synchronization required
⚠	MC_Pos_DB [DB2]	◐	Manual synchronization required

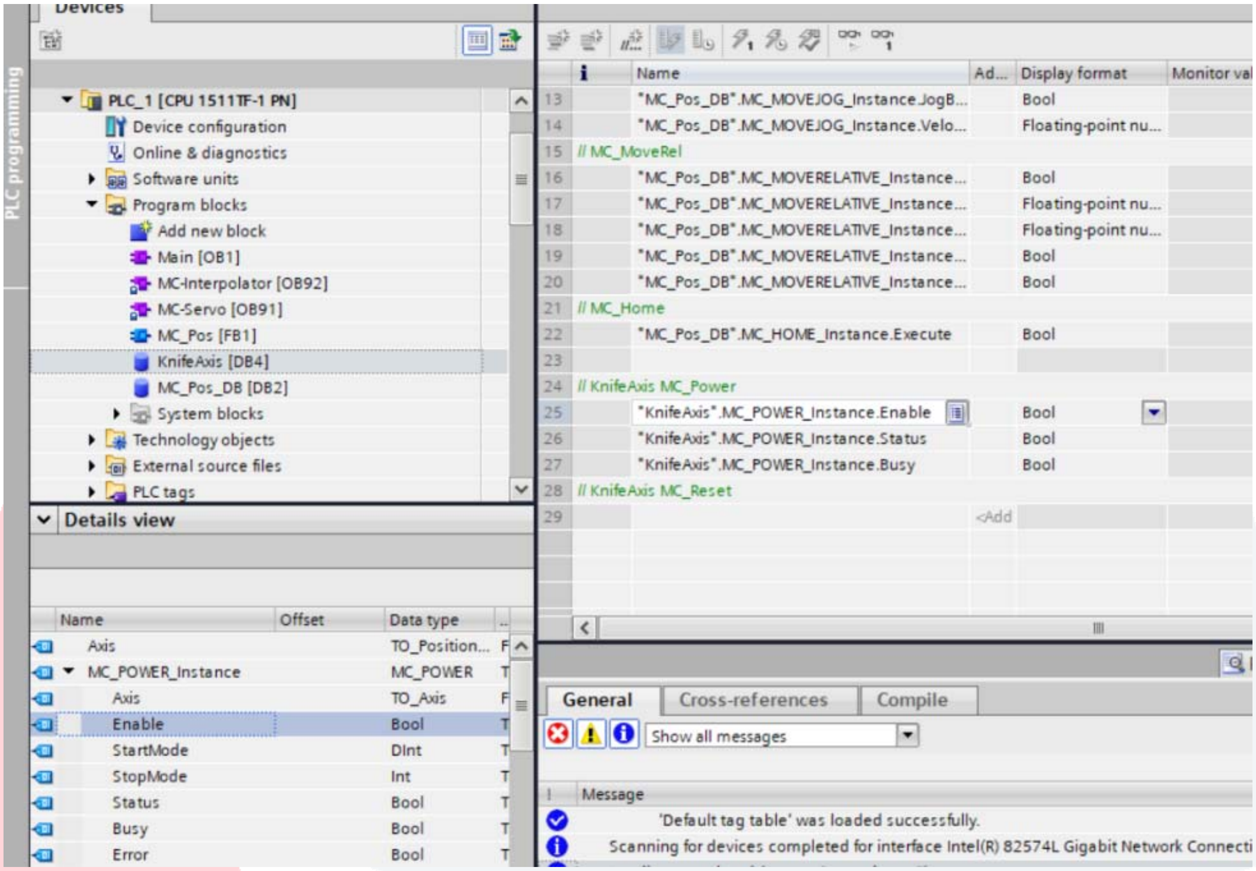
Basic parameters
Leading value interconnections
Offline/online comparison
Synchronize
Continue without synchronization
Cancel

A successful download is confirmed with zero errors in the Info dialog box.



!	Message	Go to	?	Date
✓	'Default tag table' was loaded successfully.			4/1
i	Scanning for devices completed for interface Intel(R) 82574L Gigabit Network Connection. F...			4/1
✓	Loading completed (errors: 0; warnings: 0).			4/1

To add the appropriate control variables to the existing Watch Table, open the watch table then click once on the KnifeAxis DB in the Project Tree. This puts the DB variables at the lower left details view box. Add the comments and useful variables to the Watch Table to test the KnifeAxis. Use “Ctrl” to select multiple variables and drag to the table all at one time.

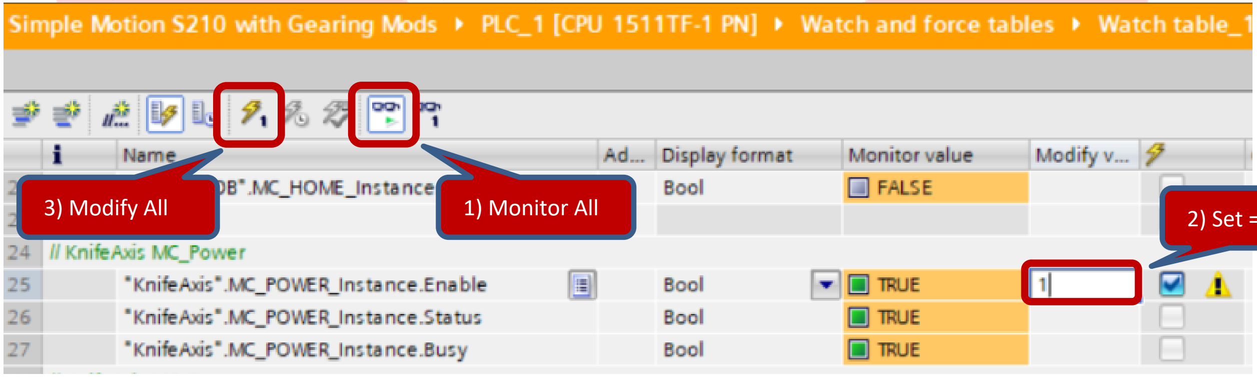


Name	Offset	Data type	
Axis		TO_Position...	F
MC_POWER_Instance		MC_POWER	T
Axis		TO_Axis	F
Enable		Bool	T
StartMode		DInt	T
StopMode		Int	T
Status		Bool	T
Busy		Bool	T
Error		Bool	T

	i	Name	Ad...	Display format	Monitor val
13		"MC_Pos_DB".MC_MOVEJOG_Instance.JogB...		Bool	
14		"MC_Pos_DB".MC_MOVEJOG_Instance.Velo...		Floating-point nu...	
15		// MC_MoveRel			
16		"MC_Pos_DB".MC_MOVERELATIVE_Instance...		Bool	
17		"MC_Pos_DB".MC_MOVERELATIVE_Instance...		Floating-point nu...	
18		"MC_Pos_DB".MC_MOVERELATIVE_Instance...		Floating-point nu...	
19		"MC_Pos_DB".MC_MOVERELATIVE_Instance...		Bool	
20		"MC_Pos_DB".MC_MOVERELATIVE_Instance...		Bool	
21		// MC_Home			
22		"MC_Pos_DB".MC_HOME_Instance.Execute		Bool	
23					
24		// KnifeAxis MC_Power			
25		"KnifeAxis".MC_POWER_Instance.Enable		Bool	
26		"KnifeAxis".MC_POWER_Instance.Status		Bool	
27		"KnifeAxis".MC_POWER_Instance.Busy		Bool	
28		// KnifeAxis MC_Reset			
29			<Add		

!	Message
✓	'Default tag table' was loaded successfully.
ℹ	Scanning for devices completed for interface Intel(R) 82574L Gigabit Network Connecti

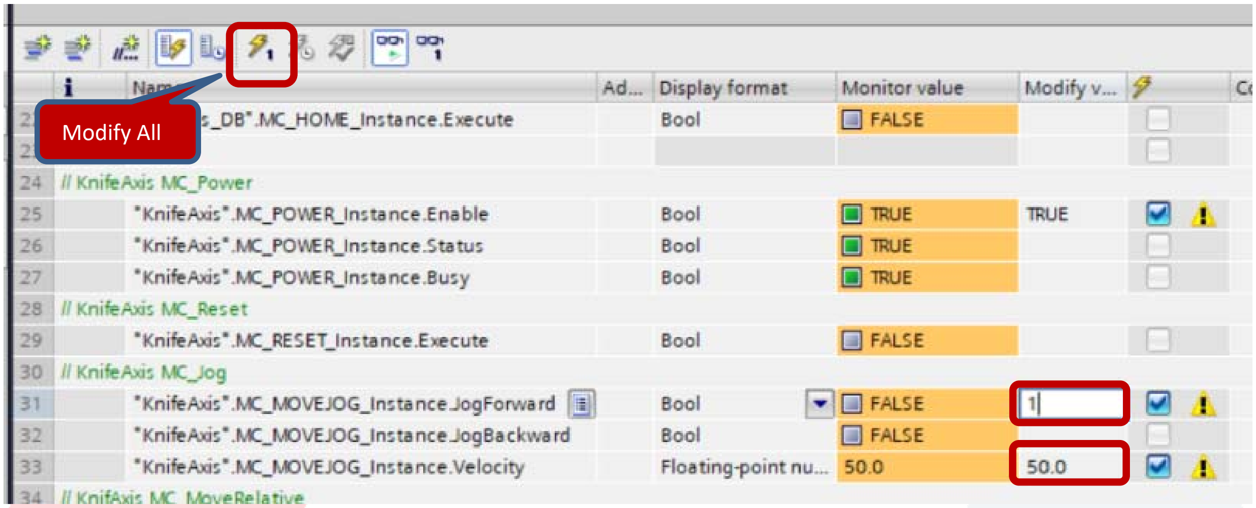
To perform a test jog, click on the “Monitor All” icon. Place a “1” (True) at MC_Power Enable. Send that to the PLC using the “Modify All” icon. MC_Power Enable is necessary first since many motion operations, like Jog, trigger on the rising edge of an Execute input. Check mark next to Modify Value column is important. Only items with the check mark will be modified by the next Modify All. The “exclamation point warning” indicates the value is only sent with the next Modify All trigger.



Simple Motion S210 with Gearing Mods ▸ PLC_1 [CPU 1511TF-1 PN] ▸ Watch and force tables ▸ Watch table_1
3) Modify All
1) Monitor All
2) Set =1

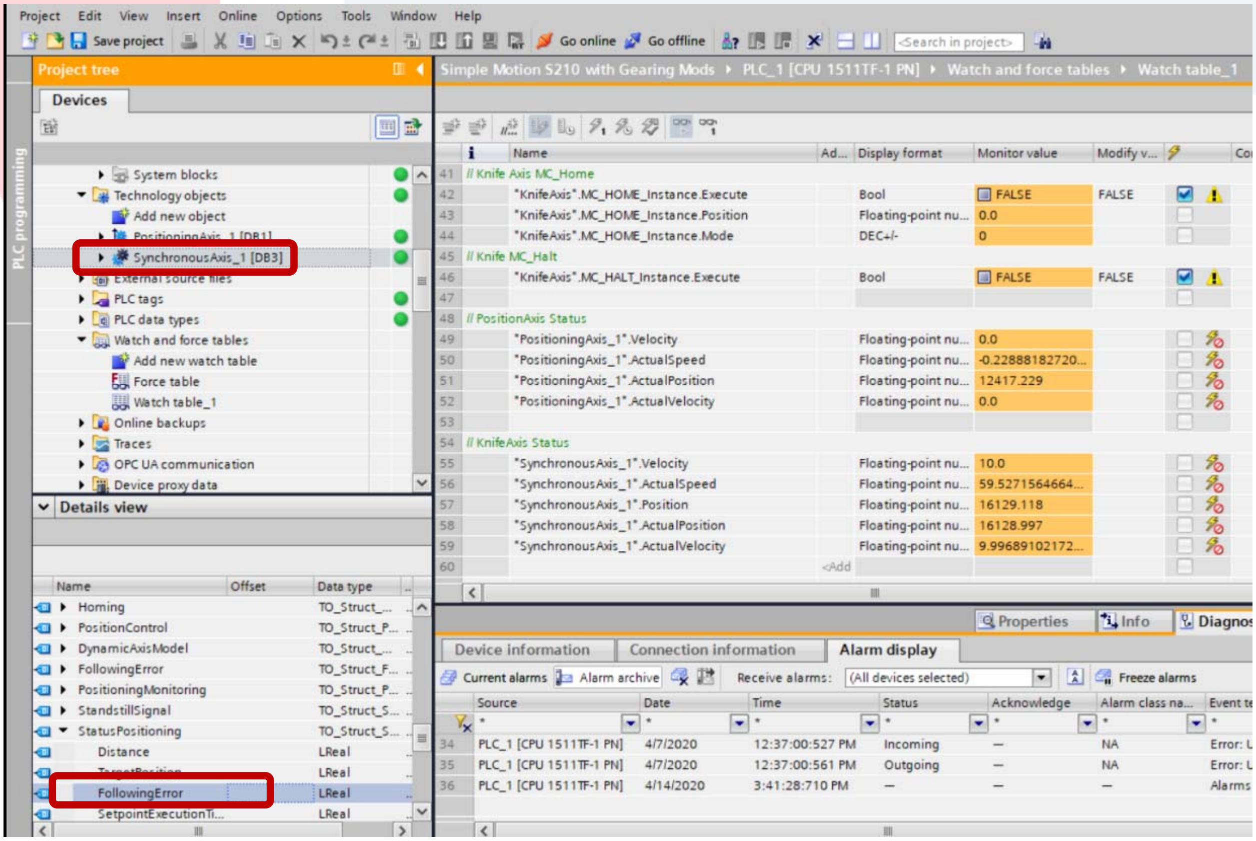
	i	Name	Ad...	Display format	Monitor value	Modify v...	
2		...DB".MC_HOME_Instance...		Bool	FALSE		□
2							
24		// KnifeAxis MC_Power					
25		"KnifeAxis".MC_POWER_Instance.Enable		Bool	TRUE	1	☑ ⚠
26		"KnifeAxis".MC_POWER_Instance.Status		Bool	TRUE		□
27		"KnifeAxis".MC_POWER_Instance.Busy		Bool	TRUE		□

Next put a “1” at “MC_MoveJog…Forward” and put 50 (mm/sec) as a “KnifeAxis MC_MoveJog …Velocity”. Follow with clicking on “Modify All” icon. The KnifeAxis should be rotating. To stop the Jog, put a “0” (False) in for MC_MoveJog…Forward” and trigger “Modify All”. Next set MC_Power to “0” and “Modify All” again to disable the axis.



	i	Name	Ad...	Display format	Monitor value	Modify v...	⚡	Co
22		…s_DB".MC_HOME_Instance.Execute		Bool	▢ FALSE		☐	
23							☐	
24	// KnifeAxis MC_Power							
25		"KnifeAxis".MC_POWER_Instance.Enable		Bool	🟩 TRUE	TRUE	☑ ⚠	
26		"KnifeAxis".MC_POWER_Instance.Status		Bool	🟩 TRUE		☐	
27		"KnifeAxis".MC_POWER_Instance.Busy		Bool	🟩 TRUE		☐	
28	// KnifeAxis MC_Reset							
29		"KnifeAxis".MC_RESET_Instance.Execute		Bool	▢ FALSE		☐	
30	// KnifeAxis MC_Jog							
31		"KnifeAxis".MC_MOVEJOG_Instance.JogForward		Bool	▢ FALSE	1	☑ ⚠	
32		"KnifeAxis".MC_MOVEJOG_Instance.JogBackward		Bool	▢ FALSE		☐	
33		"KnifeAxis".MC_MOVEJOG_Instance.Velocity		Floating-point nu...	50.0	50.0	☑ ⚠	
34	// KnifAxis MC_MoveRelative							

Other data points important to a motion configuration are the axis related status registers such as Actual Position, Actual Velocity, Position Error, etc. These are found in the DB associated with the Technology Object. To add these to a Watch Table, click on the TO name and from the Details View, these can drag and drop into the Watch Table as shown. Note that Speed is motor speed in rpm where a Velocity value is in engineering units, taking into account the mechanics defined. The differences between a commanded value and an actual value can be observed here as well.



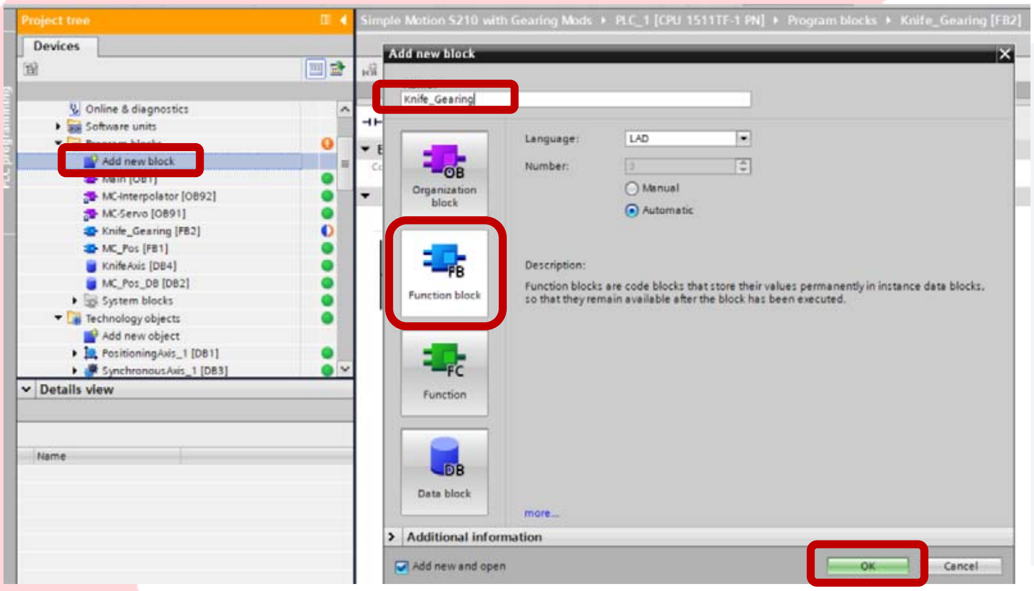
Name	Offset	Data type
Homing		TO_Struct_...
PositionControl		TO_Struct_P...
DynamicAxisModel		TO_Struct_...
FollowingError		TO_Struct_F...
PositioningMonitoring		TO_Struct_P...
StandstillSignal		TO_Struct_S...
StatusPositioning		TO_Struct_S...
Distance		LReal
TargetPosition		LReal
FollowingError		LReal
SetpointExecutionTi...		LReal

	i	Name	Ad...	Display format	Monitor value	Modify v...	⚡	Co
41	// Knife Axis MC_Home							
42		"KnifeAxis".MC_HOME_Instance.Execute		Bool	▢ FALSE	FALSE	☑ ⚠	
43		"KnifeAxis".MC_HOME_Instance.Position		Floating-point nu...	0.0		☐	
44		"KnifeAxis".MC_HOME_Instance.Mode		DEC+/-	0		☐	
45	// Knife MC_Halt							
46		"KnifeAxis".MC_HALT_Instance.Execute		Bool	▢ FALSE	FALSE	☑ ⚠	
47							☐	
48	// PositionAxis Status							
49		"PositioningAxis_1".Velocity		Floating-point nu...	0.0		☐	
50		"PositioningAxis_1".ActualSpeed		Floating-point nu...	-0.22888182720...		☐	
51		"PositioningAxis_1".ActualPosition		Floating-point nu...	12417.229		☐	
52		"PositioningAxis_1".ActualVelocity		Floating-point nu...	0.0		☐	
53							☐	
54	// KnifeAxis Status							
55		"SynchronousAxis_1".Velocity		Floating-point nu...	10.0		☐	
56		"SynchronousAxis_1".ActualSpeed		Floating-point nu...	59.5271564664...		☐	
57		"SynchronousAxis_1".Position		Floating-point nu...	16129.118		☐	
58		"SynchronousAxis_1".ActualPosition		Floating-point nu...	16128.997		☐	
59		"SynchronousAxis_1".ActualVelocity		Floating-point nu...	9.99689102172...		☐	
60			<Add				☐	

	Source	Date	Time	Status	Acknowledge	Alarm class na...	Event te
	*	*	*	*	*	*	*
34	PLC_1 [CPU 1511TF-1 PN]	4/7/2020	12:37:00:527 PM	Incoming	—	NA	Error: L
35	PLC_1 [CPU 1511TF-1 PN]	4/7/2020	12:37:00:561 PM	Outgoing	—	NA	Error: L
36	PLC_1 [CPU 1511TF-1 PN]	4/14/2020	3:41:28:710 PM	—	—	—	Alarms

Adding the capability of the Knife axis to match speed with the PositionAxis and do so with a goal to dictate the positional relationship between the two moving axes is the next exercise goal, and is easy to achieve. To just match speed is a simple electronic gearing function block called MC_GearIn. In this exercise, the next level of complexity will be illustrated, and expose the difference between gearing one axis to another randomly versus gearing in with a predefined positional relationship between the two axes. This capability is one of the deciding factors when evaluating the use of a standard S7-1500 PLC or a S7-1500T PLC, where only the 1500T (Technology PLC) can accommodate the positional relationship.

To create a new Function Block "container" for the gearing control and example cut-off knife PLC code, select "Add new block" at the Program Blocks in the Project Tree. Use Knife_Gearing as the name of this Function Block and click on the Function Block choice and then the OK button.


Simple Motion S210 with Gearing Mods ▸ PLC_1 [CPU 1511TF-1 PN] ▸ Program blocks ▸ Knife_Gearing [FB2]

Add new block

Knife_Gearing

Language: LAD

Number: 3

○ Manual

● Automatic

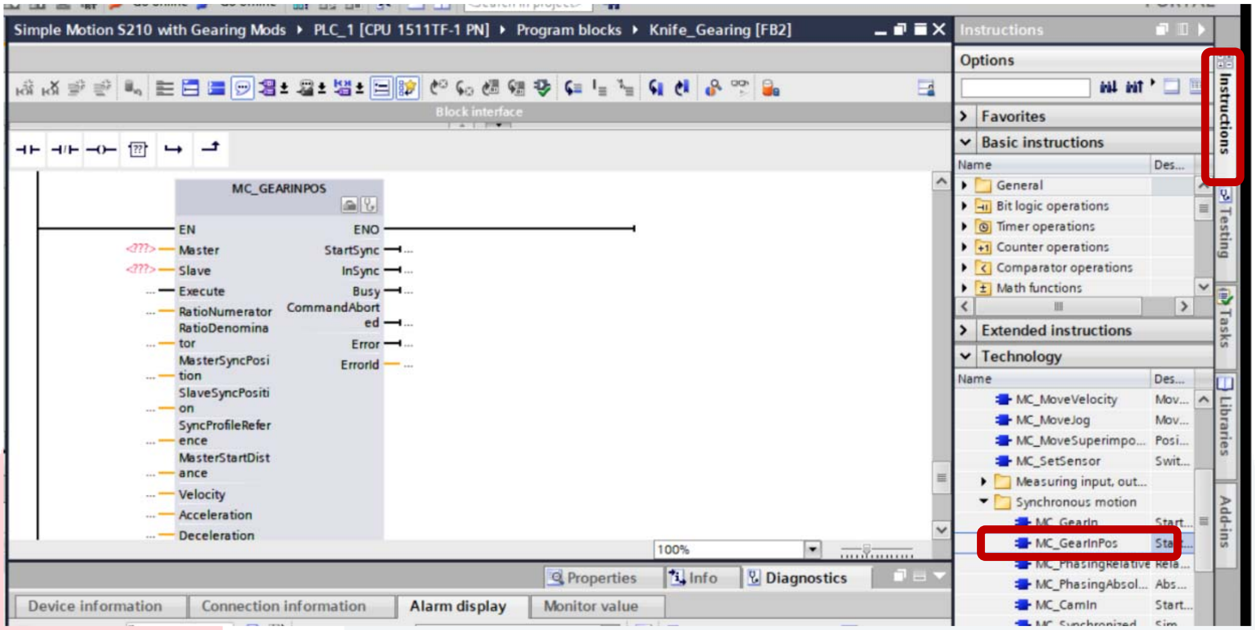
Description:

Function blocks are code blocks that store their values permanently in instance data blocks, so that they remain available after the block has been executed.

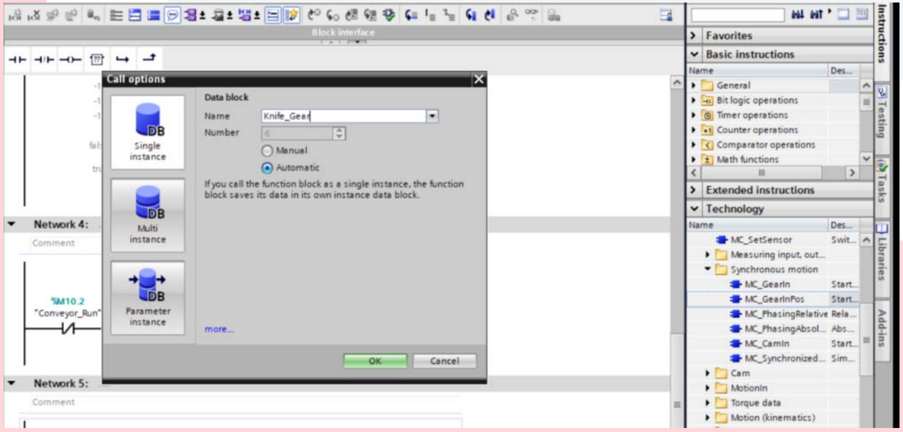
☑ Add new and open

The function block that allows for electronic gearing with positional synchronization is the MC_GearInPos found in the Instructions/Technology libraries, under “Synchronous Motion”.

Drop in MC_GearInPos from the Instructions onto a ladder network in the new FB. As this occurs, a dialog box will open offering to create the associated Data Block for the MC_GearInPos. Name that offered DB “Knife_Gear”. This can be a Single Instance DB.


Simple Motion S210 with Gearing Mods ▸ PLC_1 [CPU 1511TF-1 PN] ▸ Program blocks ▸ Knife_Gearing [FB2]

Naming the DB for MC_GearInPos.


Call options

Data block

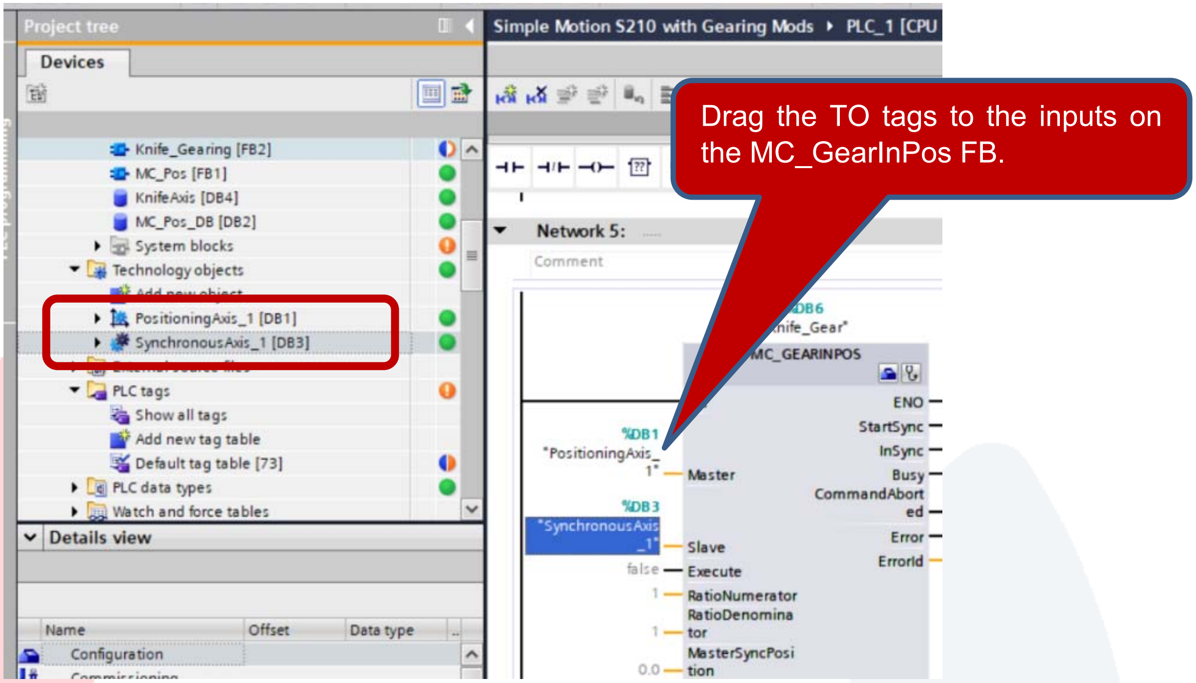
Name: Knife_Gear

Number: 6

○ Manual
● Automatic

If you call the function block as a single instance, the function block saves its data in its own instance data block.

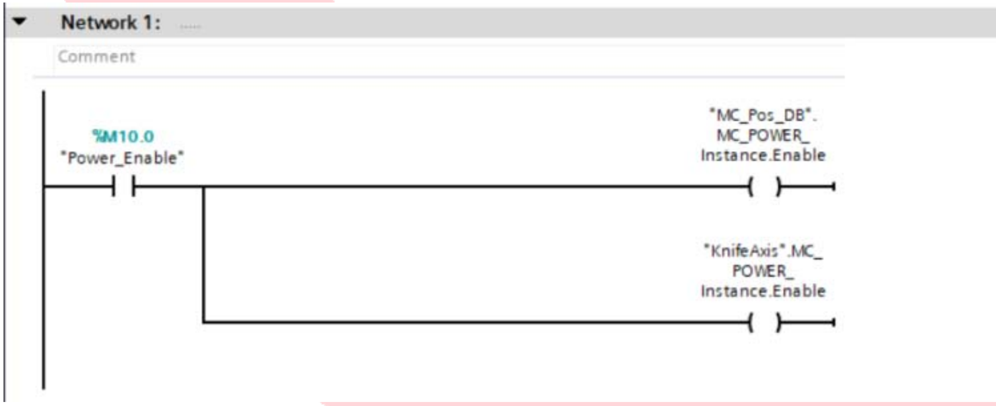
From the Technology Objects in the Project Tree, drag and drop “PositioningAxis_1” as the Master axis and drag “SynchronousAxis_1” as the Slave axis.



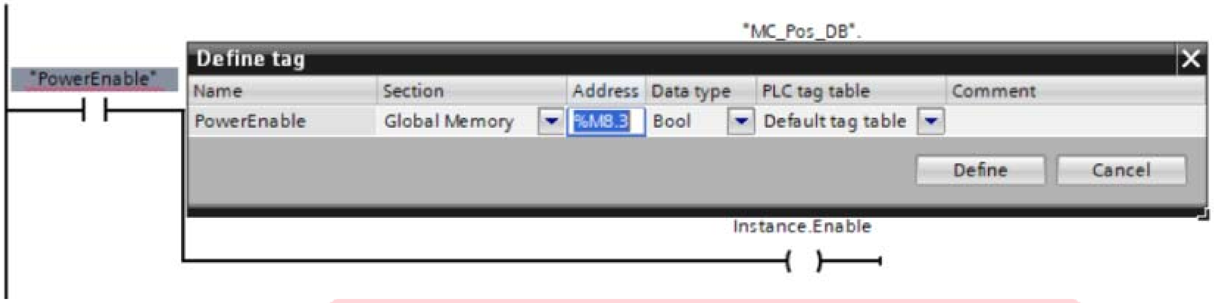
Drag the TO tags to the inputs on the MC_GearInPos FB.

Position synchronized operations will require both axes to be “Homed” initially. To simplify operation, add a network to “Enable” both axes and a network to “Home” both axes in this function block as shown below.

This approach will access the MC_Power “Enable” bit from the existing axes DB to enable both axes. The new internal memory Boolean tags were defined as Global Memory locations and the controlled tags are the same as what was being used in the Watch Table utilized previously. These can be drag and dropped from having “focus” on the respective DB in the project tree. See below for dragging DB tags to the coils.


Network 1:
Comment
%M10.0
"Power_Enable"
"MC_Pos_DB".MC_POWER_Instance.Enable
"KnifeAxis".MC_POWER_Instance.Enable

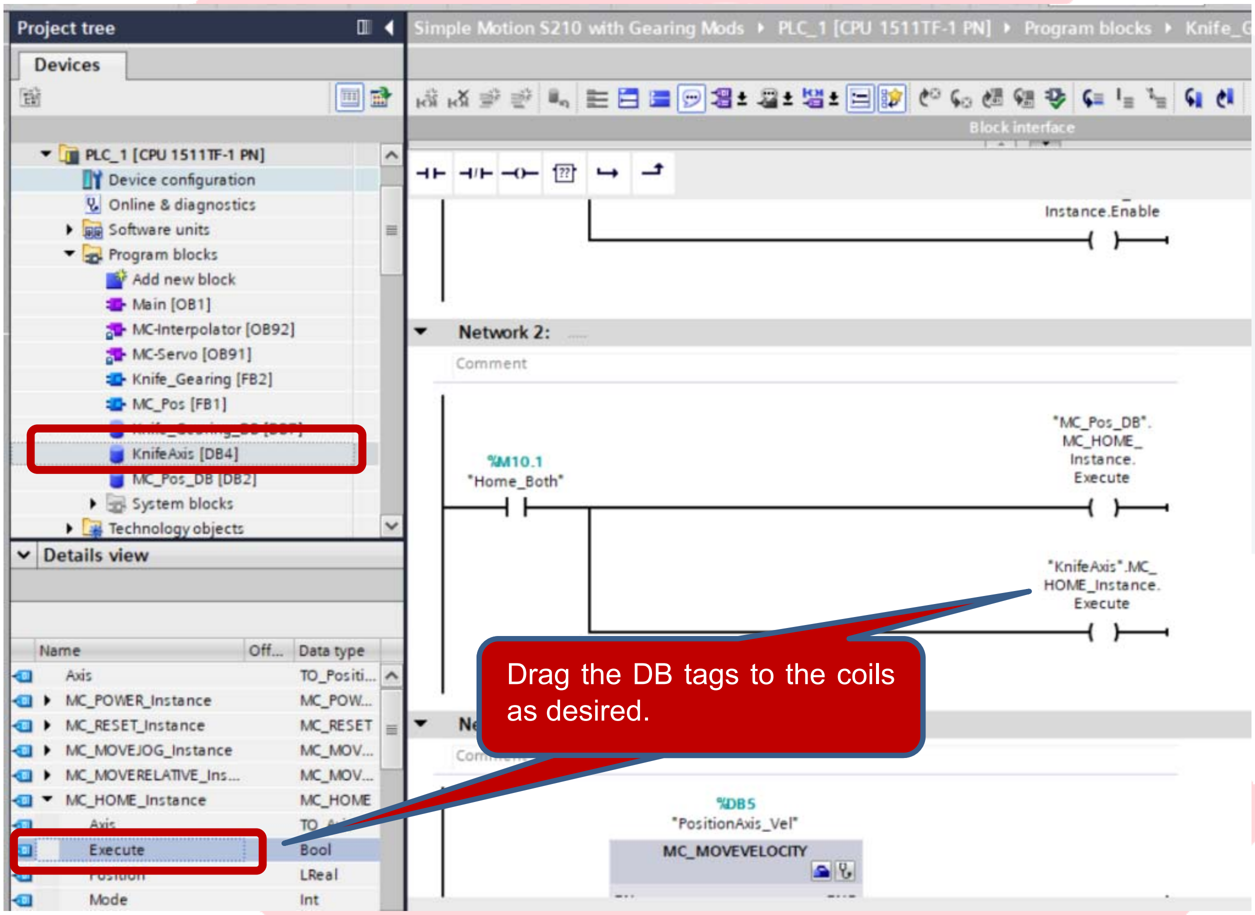
Right click to allow “Define Tag” as shown for new tags.



Define tag

Name	Section	Address	Data type	PLC tag table	Comment
PowerEnable	Global Memory	%M8.3	Bool	Default tag table	

Below, MC_Home to “Home” both axes, and showing what to have focus on to be able to drag and drop relevant tags.



Simple Motion S210 with Gearing Mods ▸ PLC_1 [CPU 1511TF-1 PN] ▸ Program blocks ▸ Knife_G

Project tree

Devices

- PLC_1 [CPU 1511TF-1 PN]
  - Device configuration
  - Online & diagnostics
  - Software units
  - Program blocks
    - Add new block
    - Main [OB1]
    - MC-Interpolator [OB92]
    - MC-Servo [OB91]
    - Knife_Gearing [FB2]
    - MC_Pos [FB1]
    - Knife_Gearing_DB [DB3]
    - KnifeAxis [DB4]
    - MC_Pos_DB [DB2]
  - System blocks
  - Technology objects

Details view

Name	Off...	Data type
Axis		TO_Positi...
MC_POWER_Instance		MC_POW...
MC_RESET_Instance		MC_RESET
MC_MOVEJOG_Instance		MC_MOV...
MC_MOVERELATIVE_Ins...		MC_MOV...
MC_HOME_Instance		MC_HOME
Axis		TO_A...
Execute		Bool
Position		LReal
Mode		Int

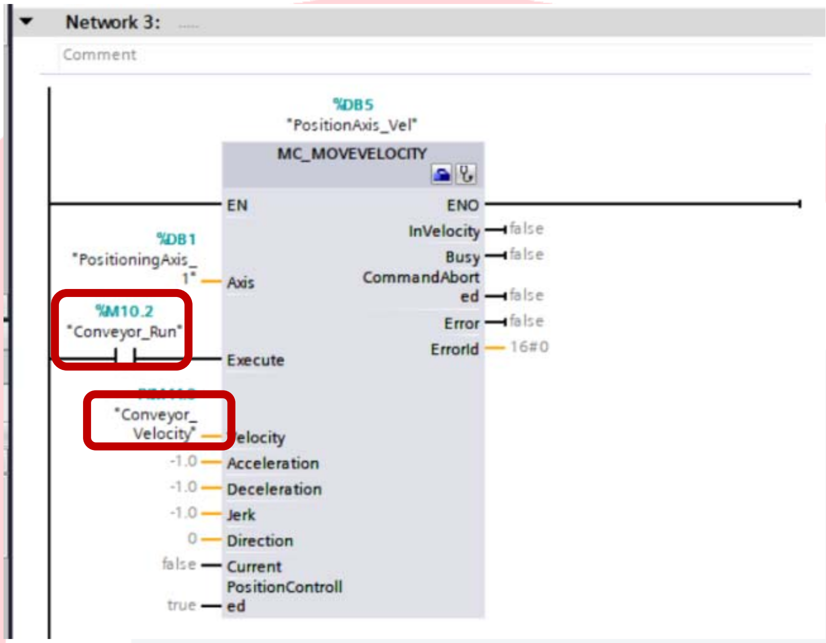
Network 2:

Comment

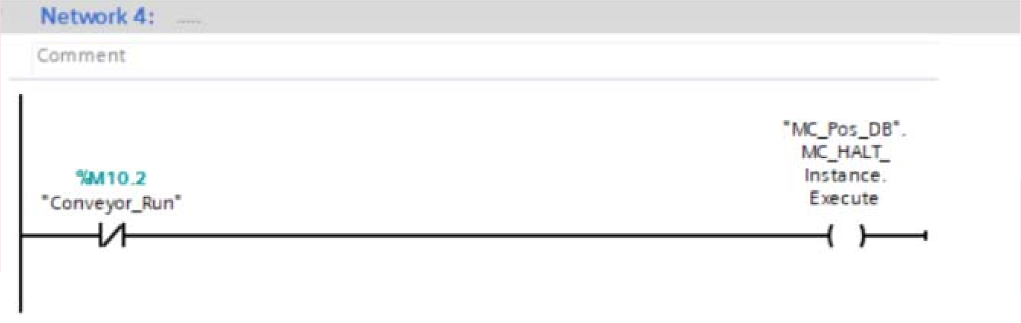
%M10.1 "Home_Both" → "MC_Pos_DB".MC_HOME_Instance.Execute, "KnifeAxis".MC_HOME_Instance.Execute

Drag the DB tags to the coils as desired.

Additionally, an MC_MoveVelocity block was added to start motion on the Conveyor axis. A Normally Open contact was placed on the Execute and named “Conveyor_Run”. The tag “Conveyor_Velocity” was added for the commanded velocity and defined as a Global variable. With values “-1” at the Acceleration and Deceleration inputs, the values set up as the default values back at the TO configuration and under Extended Parameters, “Dynamic default values” will be used.

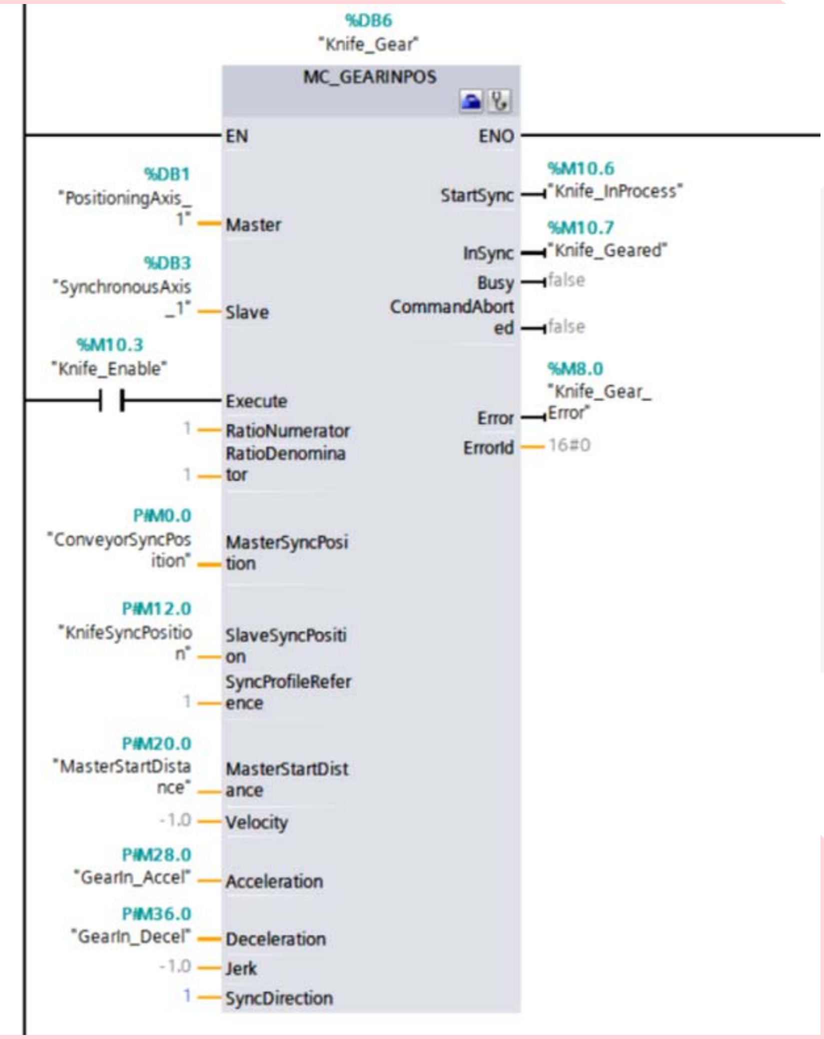


Unique to a PLC Open type Velocity moves, the motion will begin on the rising edge of the Execute input, but the axis will not stop the velocity move when the input goes away. Instead, an MC_Halt can be used to define a deceleration and stop motion. Shown below a normally closed contact of the same input used to start the velocity move is used to trigger the stop using the MC_Halt brought over from axis DB tags.



To fully understand and manipulate the synchronized gearing of the follower axis, additional I/O tags are placed on the MC_GearInPos FB as shown below.
These include:

“Knife_Enable” as a normally open input contact.
“ConveyorSyncPosition” as the target synchronization position of the Master Axis.
“KnifeSyncPosition” as the target synchronization position of the Follower Axis.
“MasterStartDistance” as the allowed distance for synchronization to take place.
Notice SyncProfileReference is hard coded to a “1”. Accel and Decel are not needed.
Additionally, see Siemens definitions by clicking on the FB and pressing F1 for the help dialog.



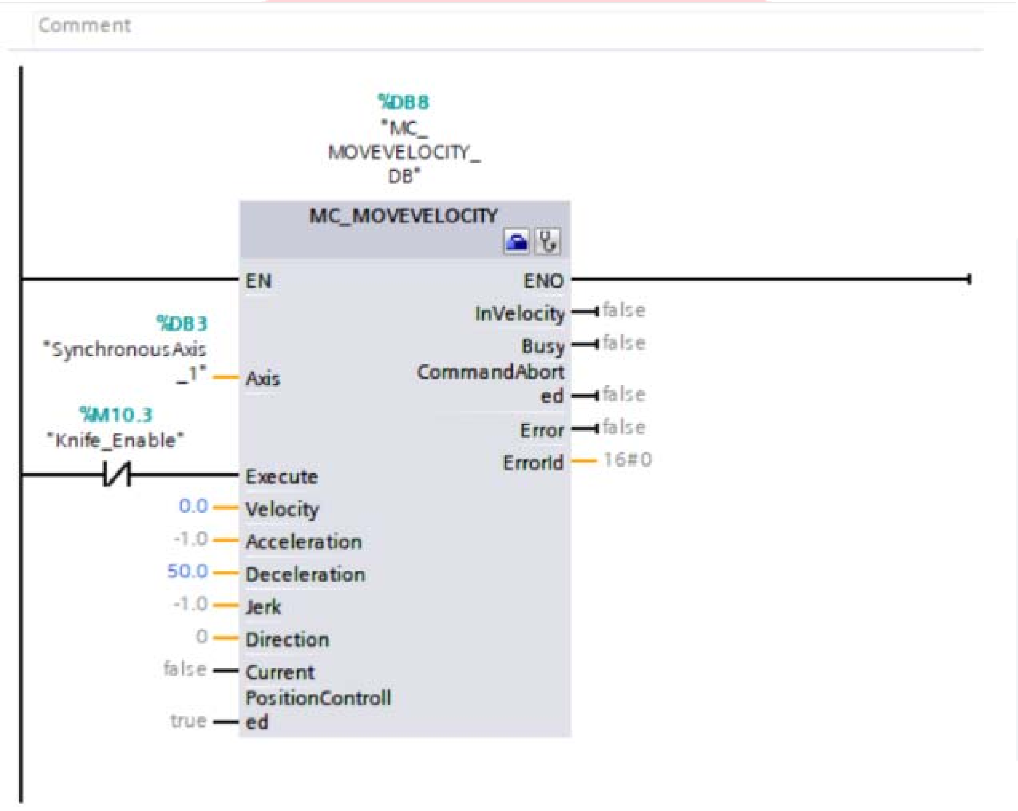
%DB6
"Knife_Gear"
MC_GEARINPOS

EN — ENO

%DB1 "PositioningAxis_1" — Master
%DB3 "SynchronousAxis_1" — Slave
%M10.3 "Knife_Enable" — Execute
1 — RatioNumerator
1 — RatioDenominator
P#M0.0 "ConveyorSyncPosition" — MasterSyncPosition
P#M12.0 "KnifeSyncPosition" — SlaveSyncPosition
1 — SyncProfileReference
P#M20.0 "MasterStartDistance" — MasterStartDistance
-1.0 — Velocity
P#M28.0 "GearIn_Accel" — Acceleration
P#M36.0 "GearIn_Decel" — Deceleration
-1.0 — Jerk
1 — SyncDirection

StartSync — %M10.6 "Knife_InProcess"
InSync — %M10.7 "Knife_Geared"
Busy — false
CommandAborted — false
Error — %M8.0 "Knife_Gear_Error"
ErrorId — 16#0

Since the “Execute” is like the MC_MoveVelocity input discussed previously, triggering on the rising edge of the input, this input can go away and the GearInPos is still active. As with many motion commands, an MC_Halt can be utilized to stop the axis. MC_Halt is often used as a generic operation to stop any motion in progress and may have a particularly aggressive deceleration, for example. An optional method to disengage gearing is to trigger an MC_MoveVelocity with a target velocity of 0 mm/sec for the Knife axis. The new motion task will override the existing one and the deceleration value assigned to the MC_MoveVelocity will be utilized.



Comment

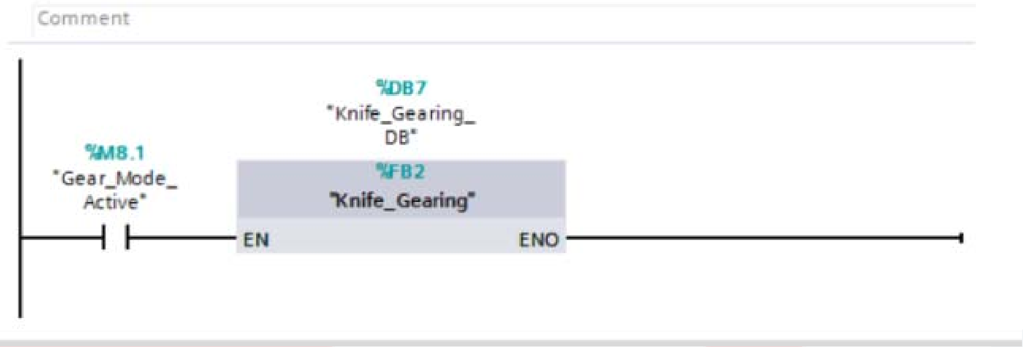
%DB8
"MC_MOVEVELOCITY_DB"

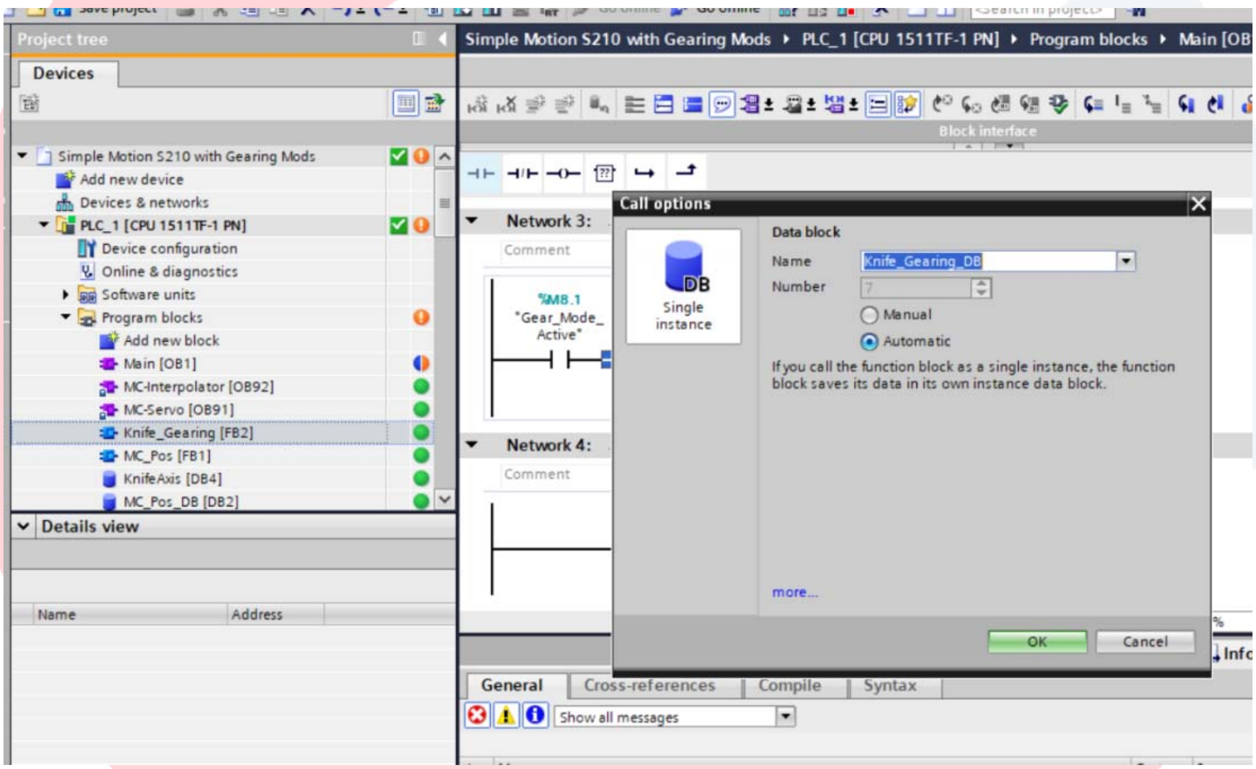
MC_MOVEVELOCITY

EN — ENO
%DB3 "SynchronousAxis_1" — Axis
%M10.3 "Knife_Enable" — Execute
0.0 — Velocity
-1.0 — Acceleration
50.0 — Deceleration
-1.0 — Jerk
0 — Direction
false — Current
true — PositionControlled

InVelocity — false
Busy — false
CommandAborted — false
Error — false
ErrorId — 16#0

Once the FB that includes the ancillary gearing control is complete, compile the PLC code to check for errors and then, back at OB1, you can add a contact called Gear_Mode_Active and then drag and drop the Knife_Gearing FB onto the same ladder rung in OB1 after the Gear_Mode_Active contact so this FB is only called when desired. Click OK and the Knife_Gearing DB will be created as shown below.





Call options

DB
Single instance

Data block

Name: Knife_Gearing_DB

Number: 7

○ Manual

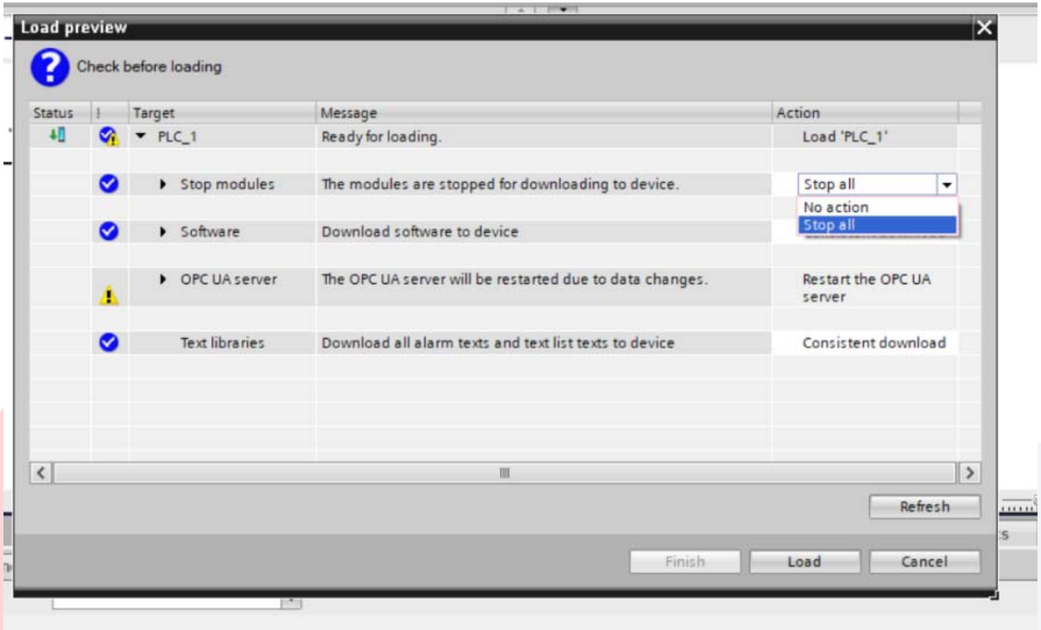
● Automatic

If you call the function block as a single instance, the function block saves its data in its own instance data block.

more...

OK | Cancel

Download all changes to the PLC. With major changes a PLC Stop will be required. Confirm the PLC is started following the Load.



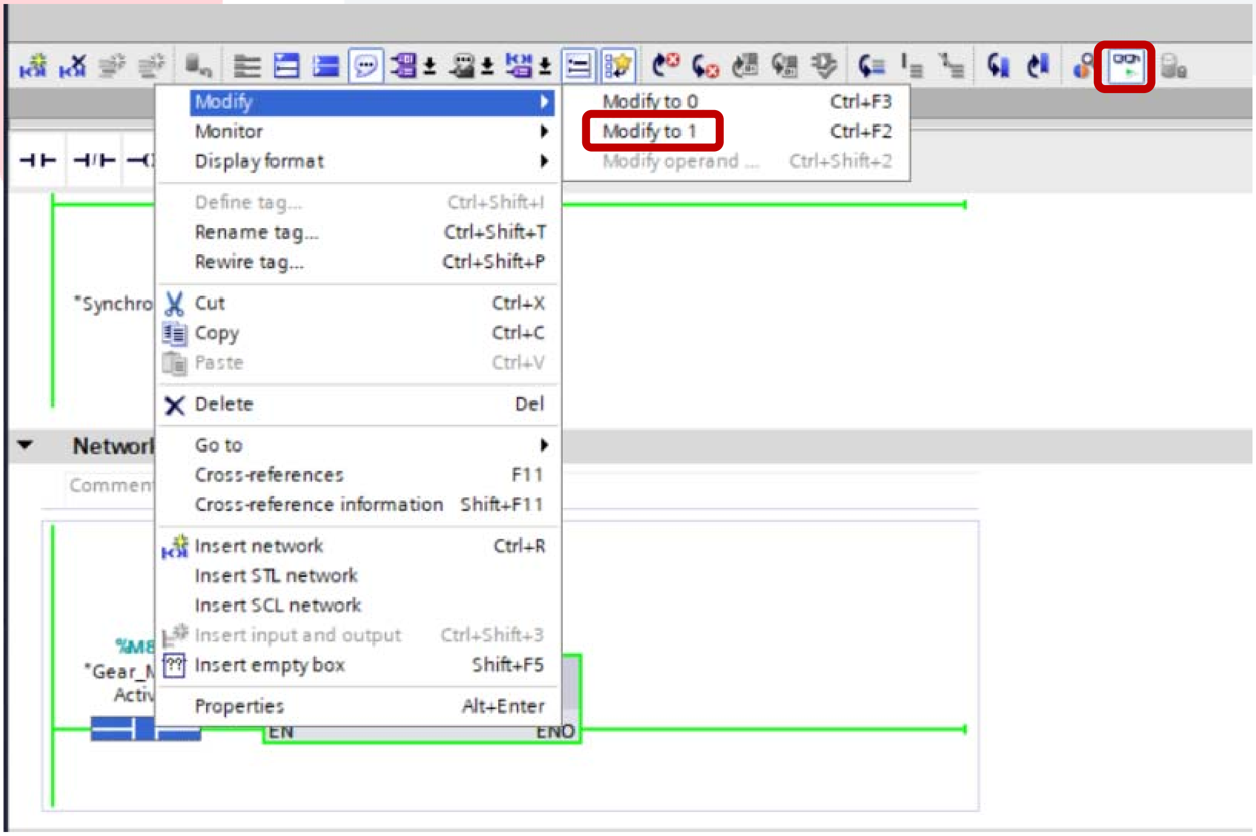
Load preview

Check before loading

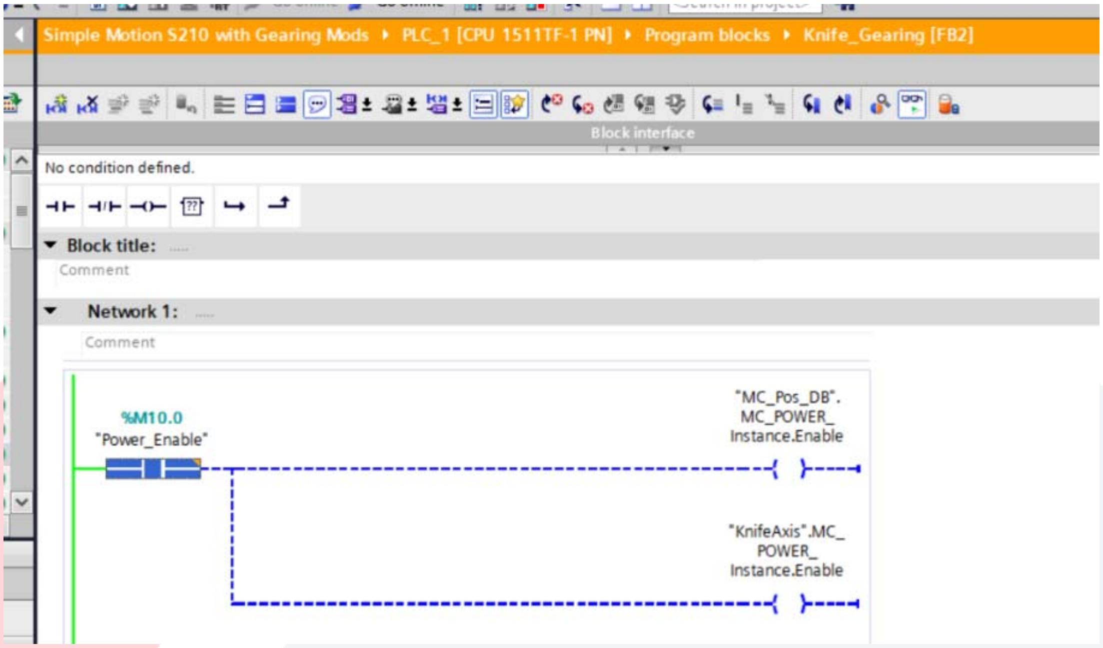
Status	!	Target	Message	Action
		PLC_1	Ready for loading.	Load 'PLC_1'
		Stop modules	The modules are stopped for downloading to device.	Stop all
		Software	Download software to device	
		OPC UA server	The OPC UA server will be restarted due to data changes.	Restart the OPC UA server
		Text libraries	Download all alarm texts and text list texts to device	Consistent download

Good time to Save the project…

A point being made here, is that if a newly created FB is not called from an active OB (i.e. OB1) then that code will not be active on the PLC, so the first step to testing is to “Modify to 1” the “GearModeActive” input contact in OB1. Select the “Monitoring On/Off” icon on the Menu bar then put focus on the normally open contact and right click to select Modify and “Modify to 1” as shown.



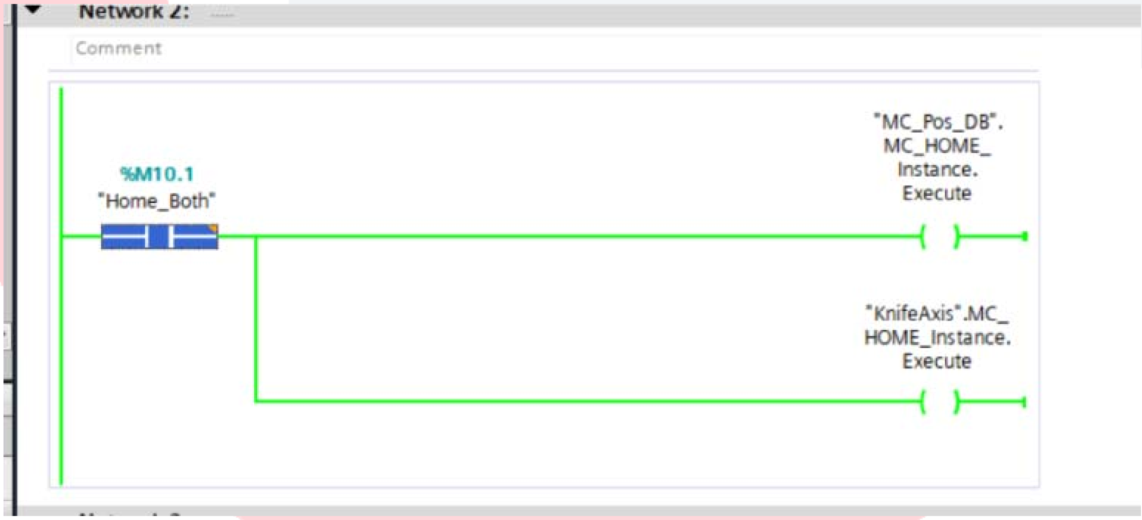
Now that the FB is being evaluated in the PLC (note green ladder rail), select the Monitor On/Off and Modify Power_Enable as shown. Short cut key is F2 to “Modify to 1” and F3 to “Modify to 0” once focus is on the desired contact.



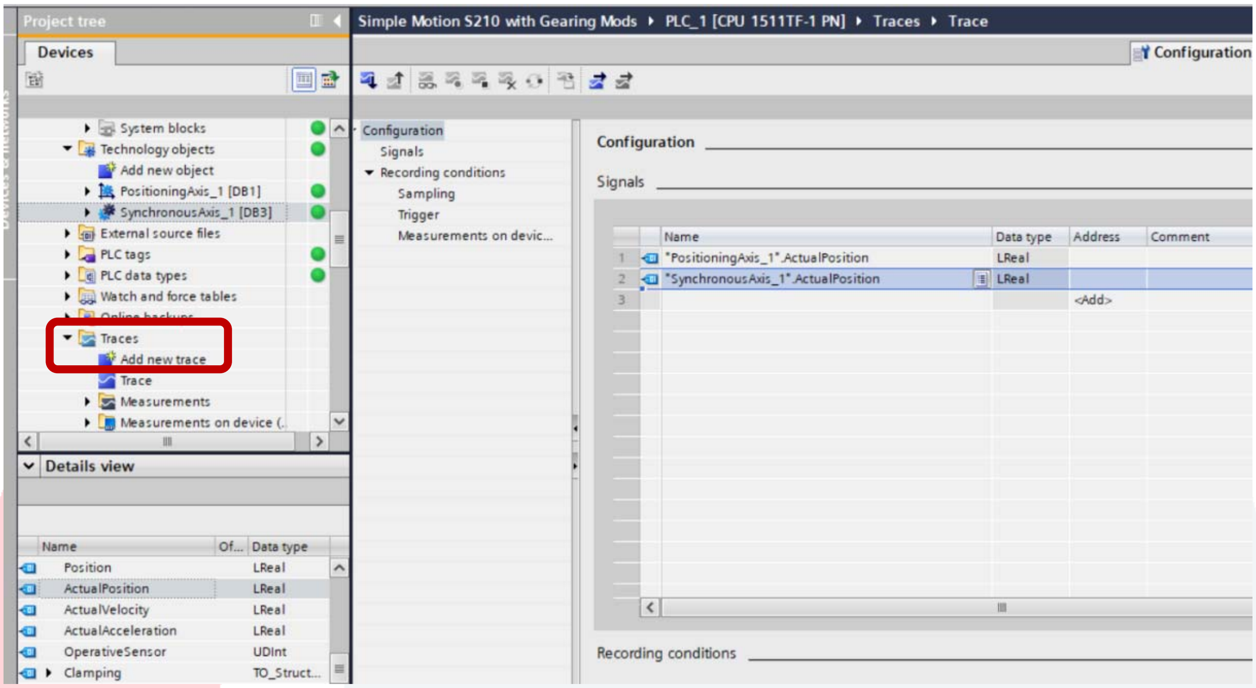
Simple Motion S210 with Gearing Mods ▸ PLC_1 [CPU 1511TF-1 PN] ▸ Program blocks ▸ Knife_Gearing [FB2]

Toggle “Home_Both” On and then Off to establish the current motor locations as Zero since all default settings are being used for the Home operation.

NOTE: Unexpected motion WILL HAPPEN if a geared axis is actively following a master axis and the master is Homed this way.



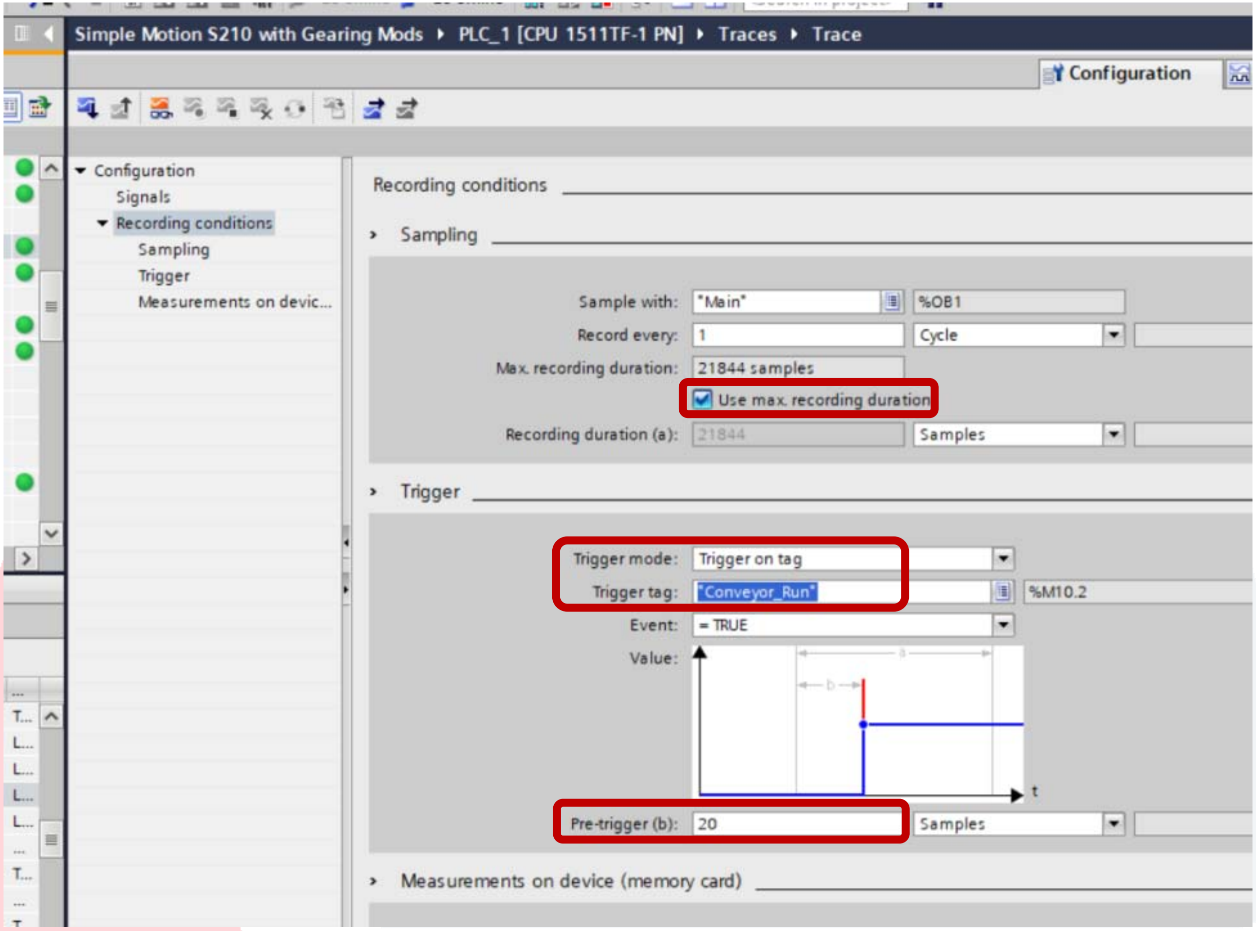
Next, configure a Trace to confirm precise operation of the servo axes. In the Project Tree under PLC_1, expand “Traces”, and select “Add new trace”. When the “Configuration” dialog opens, using the same drag and drop methods, add the “PositioningAxis_1ActualPosition” (conveyor axis) and “SynchronousAxis_1 ActualPosition” (Knife Axis) to the traces to be captured.



	Name	Data type	Address	Comment
1	"PositioningAxis_1".ActualPosition	LReal		
2	"SynchronousAxis_1".ActualPosition	LReal		
3			<Add>	

Name	Of...	Data type
Position		LReal
ActualPosition		LReal
ActualVelocity		LReal
ActualAcceleration		LReal
OperativeSensor		UDInt
Clamping		TO_Struct...

At “Recording Conditions” select “Use max recording duration” and set Trigger Mode to “Trigger on Tag” and select the “Conveyor_Run” tag used to start the conveyor motion. Set a pretrigger capture to 20 samples, based on OB1 cycle duration.



Simple Motion S210 with Gearing Mods ▸ PLC_1 [CPU 1511TF-1 PN] ▸ Traces ▸ Trace

Configuration

- Configuration
  - Signals
  - Recording conditions
    - Sampling
    - Trigger
    - Measurements on devic...

Recording conditions

› Sampling

Sample with: "Main" %OB1

Record every: 1 Cycle

Max. recording duration: 21844 samples

☑ Use max. recording duration

Recording duration (a): 21844 Samples

› Trigger

Trigger mode: Trigger on tag

Trigger tag: "Conveyor_Run" %M10.2

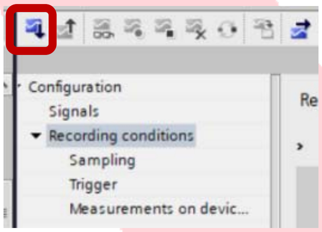
Event: = TRUE

Value:

Pre-trigger (b): 20 Samples

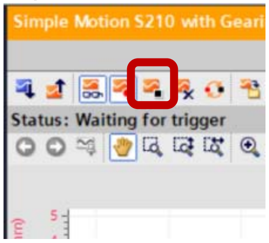
› Measurements on device (memory card)

Use Transfer to Device to send the configuration to the PLC.



- Configuration
  - Signals
  - Recording conditions
    - Sampling
    - Trigger
    - Measurements on devic...

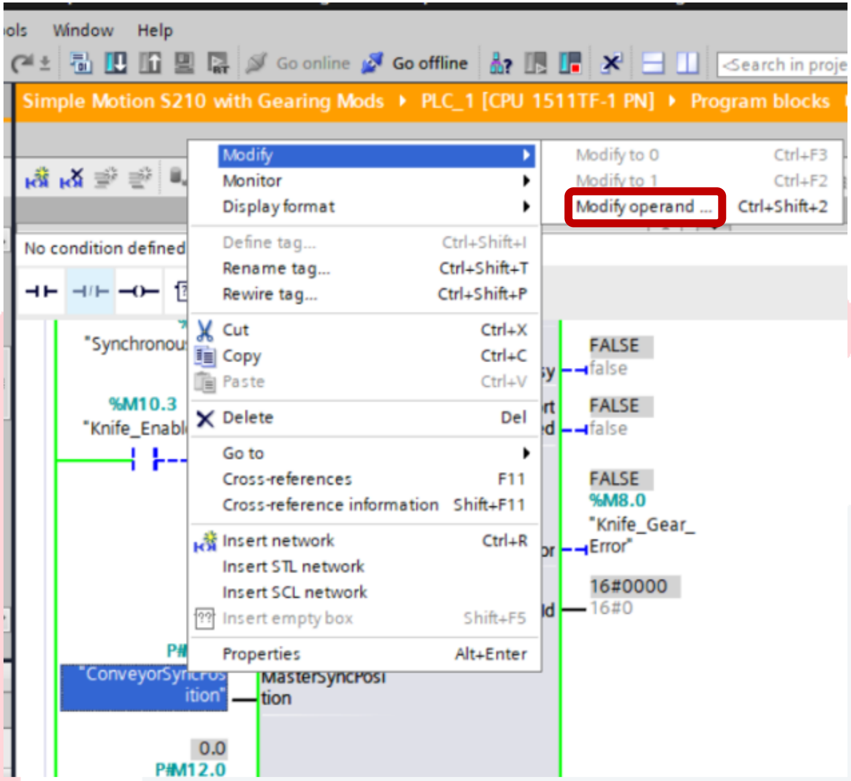
Select “Activate Recording” and observe “Waiting for Trigger”. Note, this operation exist on the PLC, not on the connected laptop, so a strategic trigger could be selected to capture events that happen randomly while the laptop was not even connected, and then pulled up later and analyzed.



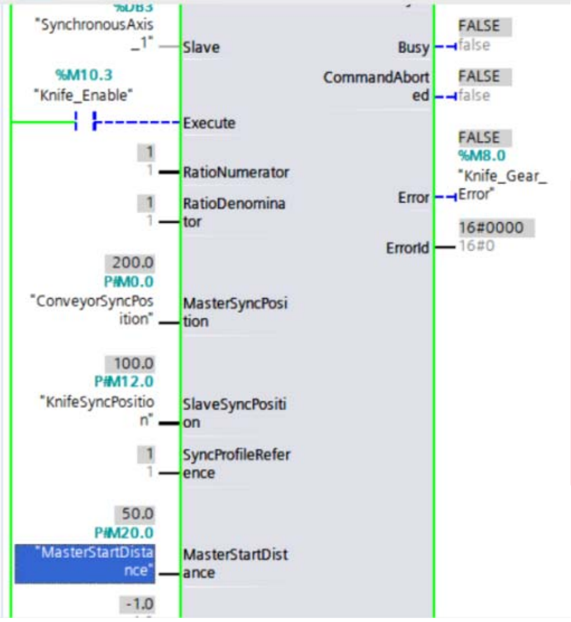
Simple Motion S210 with Geari

Status: Waiting for trigger

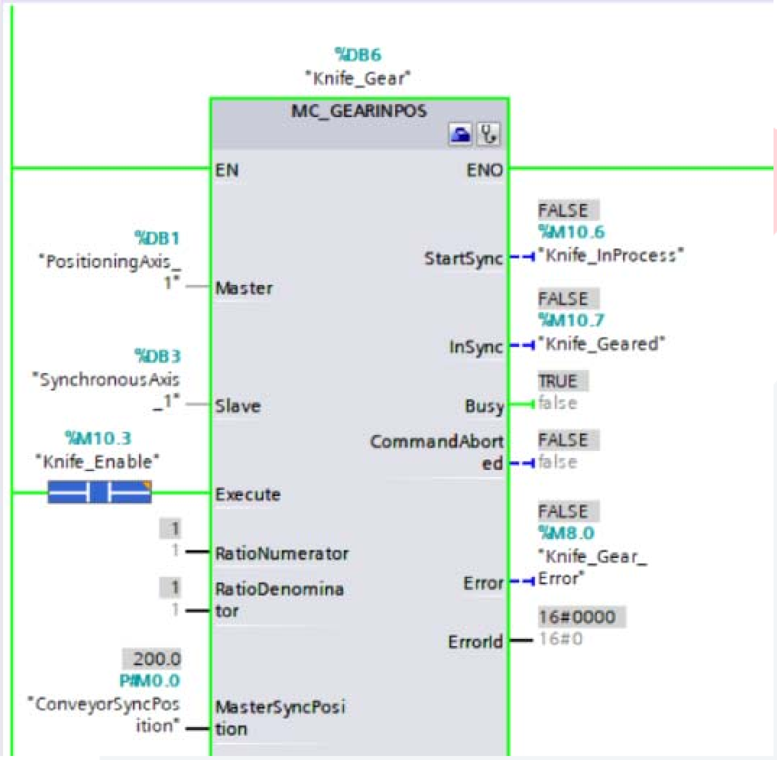
Use these values for operating the synchronous knife gearing. Values can be set utilizing “Modify Operand” as shown here, with focus on each input variable respectively.



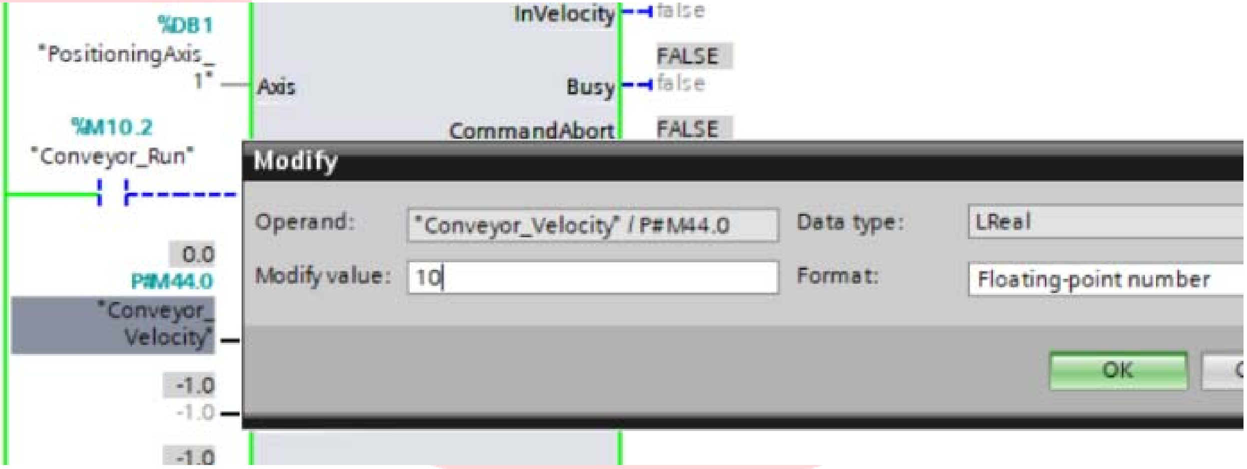
ConveyorSyncPosition = 200mm
KnifeSyncPosition = 100mm
MasterStartDistance = 50mm



Next, trigger the MC_GearInPos FB with “Knife_Enable” and notice the “InSync” output is still False however the “Busy” output is now True. Also notice there is no ErrorID reported and should show 16#0000. The FB is actively monitoring the Master Position and will ramp the Follower at the appropriate position so that when the Master hits 200mm location, the Follower will hit 100mm and with be locked on at a 1:1 ratio. The synchronization will begin 50mm before the target coincidence location of the Master, so at 150mm.



Next, set a relatively low velocity for the “Conveyor” and trigger the “Conveyor_Run” input knowing this will also trigger the Trace to begin recording the two servo axes Actual Positions.



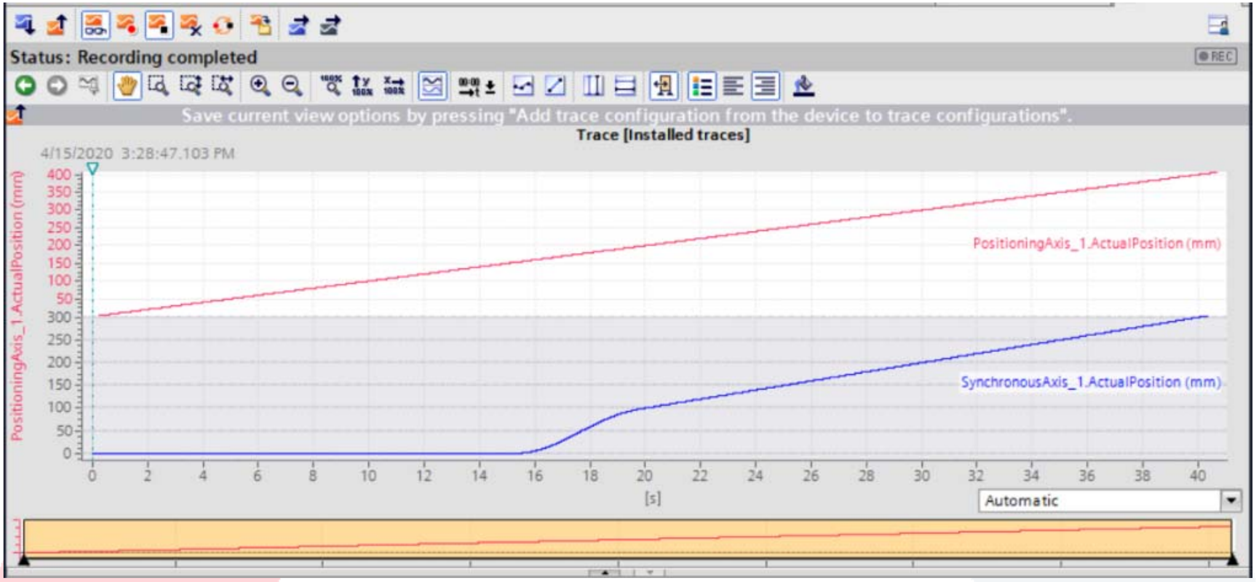
Operand: "Conveyor_Velocity" / P#M44.0

Data type: LReal

Modify value: 10

Format: Floating-point number

At first glance, the Following Axis started motion at about the 150mm position of the Master Axis (graph axis at left is position), accelerated to a velocity required to catch the synchronization points of its own 100mm position and the Master 200mm position, then matched Velocity (position change) at the 1 to 1 ratio.



Status: Recording completed

Save current view options by pressing "Add trace configuration from the device to trace configurations".

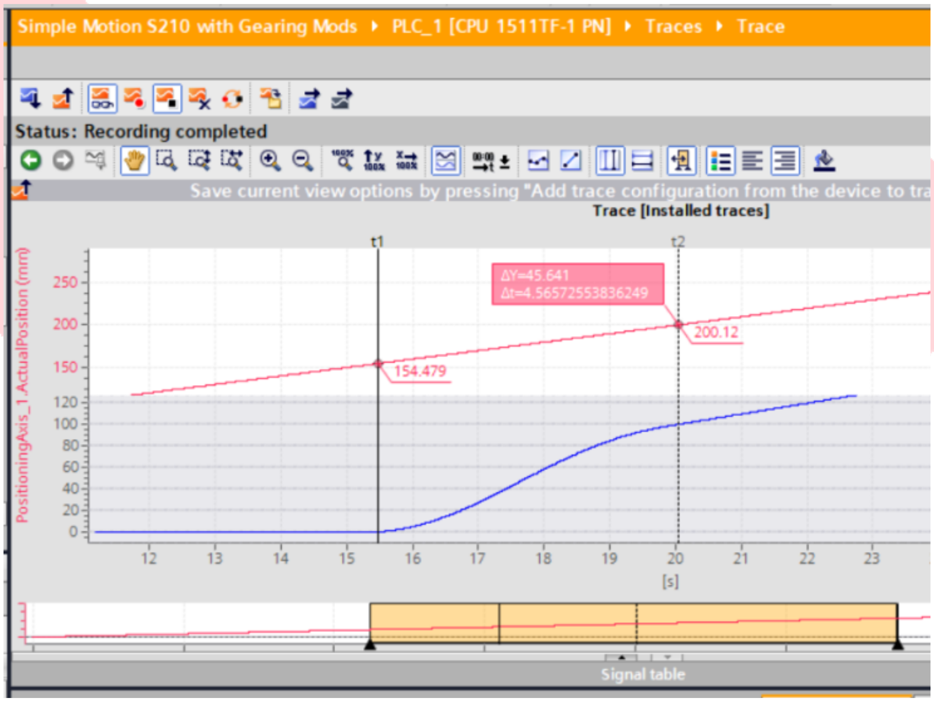
Trace [Installed traces]

4/15/2020 3:28:47.103 PM

REC
PositioningAxis_1.ActualPosition (mm)
400
350
300
250
200
150
100
50
300
250
200
150
100
50
0
PositioningAxis_1.ActualPosition (mm)
SynchronousAxis_1.ActualPosition (mm)
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
[s]
Automatic

To be more critical and specific, multiple tools are available for zoom and adding measurement cursor bars to help evaluate the servo accuracy. By zooming in and focusing on Conveyor trace (red) around the 100mm to 200mm positions it is confirmed the Knife Axis started moving at about 155mm (Master) and the Knife was at 100mm when the Conveyor was at 200mm (look to the left where position values are shown). Observing the slope of the Knife trace, it is known the velocity of the knife servo exceeded the Conveyor to catch up at the synchronization location, and then decelerated to match speed 1:1 as shown with parallel slopes from 100mm of the knife position onward.

See the “Modulo Addendum” attached for additional scope trace tips.



Simple Motion S210 with Gearing Mods ▸ PLC_1 [CPU 1511TF-1 PN] ▸ Traces ▸ Trace

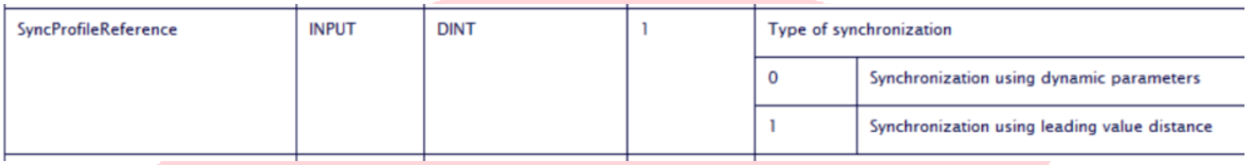
Status: Recording completed

Save current view options by pressing "Add trace configuration from the device to tra

Trace [Installed traces]

ΔY=45.641
Δt=4.56572553836249

To illustrate a Conveyor velocity 10x faster and the relationship of SyncProfileReference=1 versus 0 , it would be expected to allow the Knife axis more “distance” to come up to speed and catch the Conveyor, since it was already at about twice the 1:1 speed to catch the Conveyor at the slow conveyor speed. With SyncProfileReference=1, this synchronization is based on the user defined distance. If SyncProfileReference=0, then the dynamics defined in Maximum Speed, Acceleration and Deceleration are utilized. More details can be reviewed in the Help dialog.

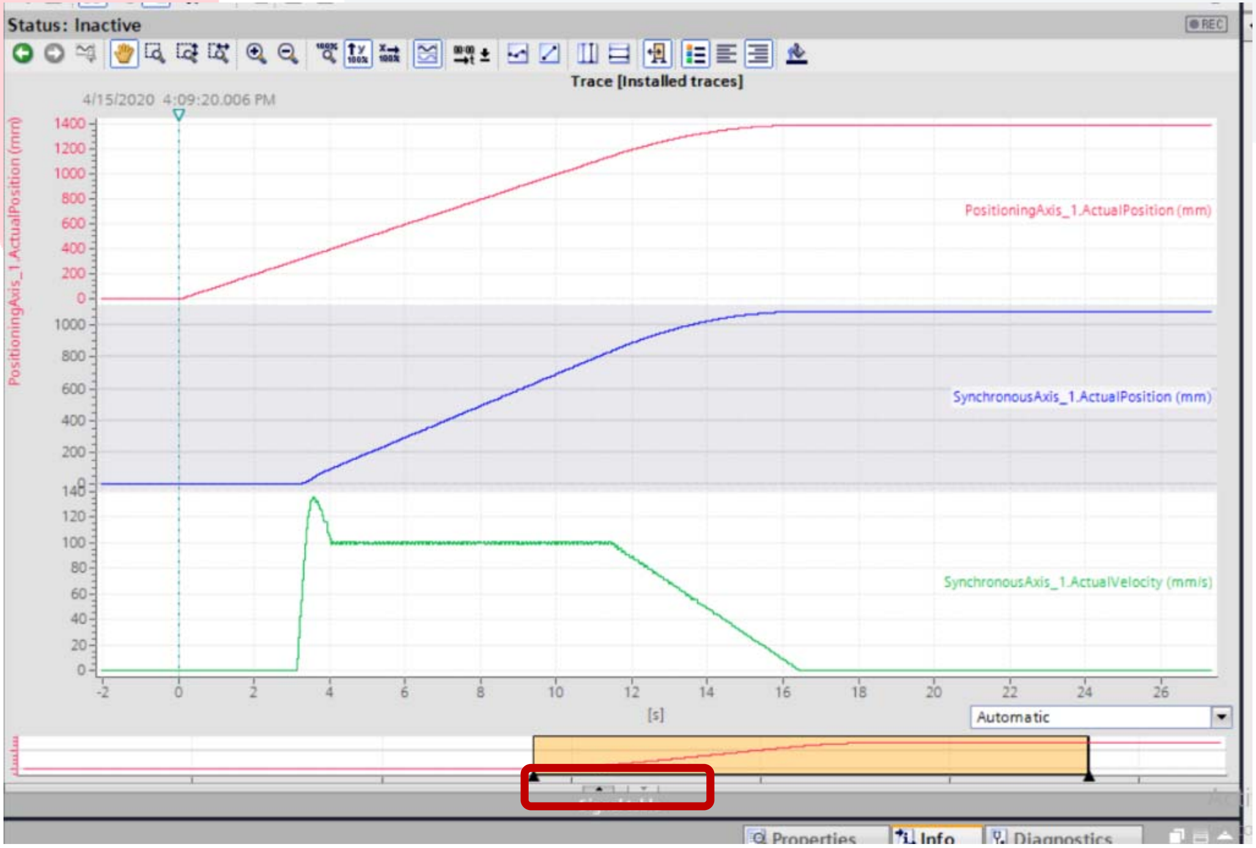


SyncProfileReference	INPUT	DINT	1	Type of synchronization	
				0	Synchronization using dynamic parameters
				1	Synchronization using leading value distance

Below, the faster values used are:

Conveyor_Velocity = 100 mm/sec
ConveyorSyncPosition = 400mm
KnifeSyncPosition = 100mm
MasterStartDistance = 100mm

The Velocity of the Knife axis has been added to the Trace.
Based on the values listed above, it is expected the Knife motion begins around the 300mm location of the Conveyor and will be at 1:1 synchronized Velocity at Conveyor location 400mm and Knife location 100mm.
This trace also captured the synchronized deceleration of stopping the Conveyor at 1400mm.
Note: To manipulate the traces hide/display or scaling, there is a collapsed settings dialog box at the arrows shown in the red highlight box.



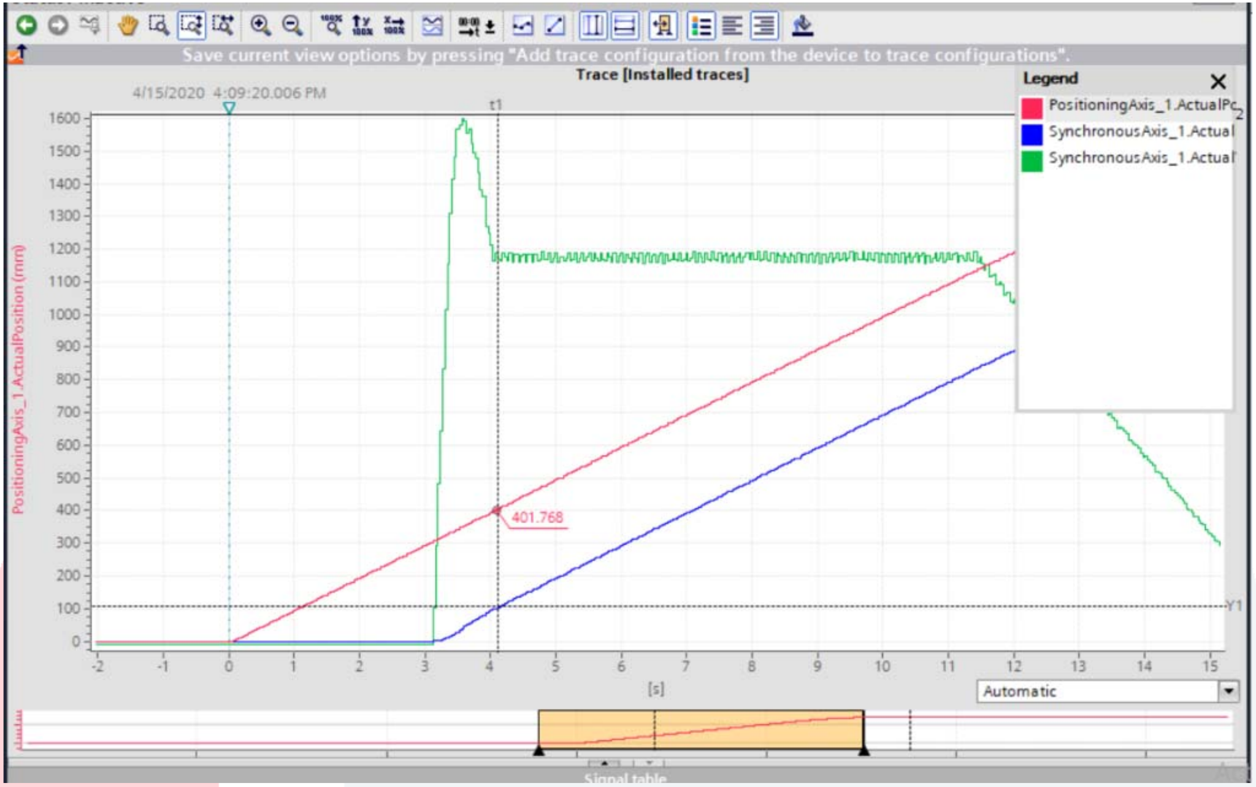
Status: Inactive

Trace [Installed traces]

4/15/2020 4:09:20.006 PM

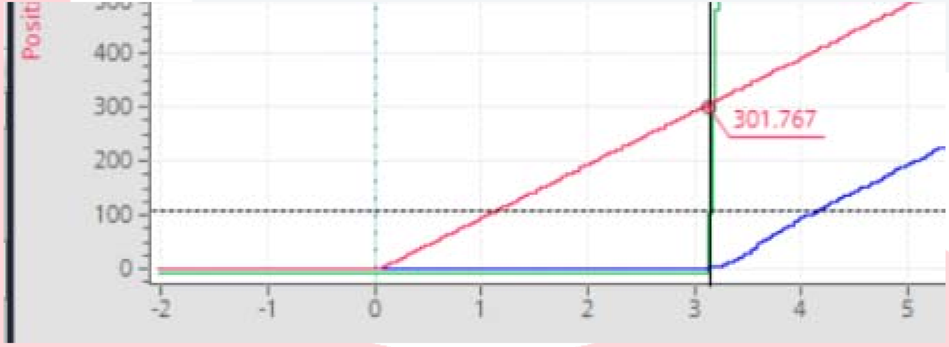
REC
PositioningAxis_1.ActualPosition (mm)
1400
1200
1000
800
600
400
200
0
1000
800
600
400
200
140
120
100
80
60
40
20
0
PositioningAxis_1.ActualPosition (mm)
SynchronousAxis_1.ActualPosition (mm)
SynchronousAxis_1.ActualVelocity (mm/s)
-2
0
2
4
6
8
10
12
14
16
18
20
22
24
26
[s]
Automatic
Properties
Info
Diagnostics

To be critical, and zoomed in, it is shown when the Conveyor (red) is at 400mm the Knife has ramped up, and decelerated back to be at 1:1 Velocity when the Knife (blue) reached 100mm. Also to be confirmed, Knife motion started 100mm before the synchronization location of the Conveyor sync target location of 400mm.



Save current view options by pressing "Add trace configuration from the device to trace configurations".

Knife motion began at Conveyor 300mm location:



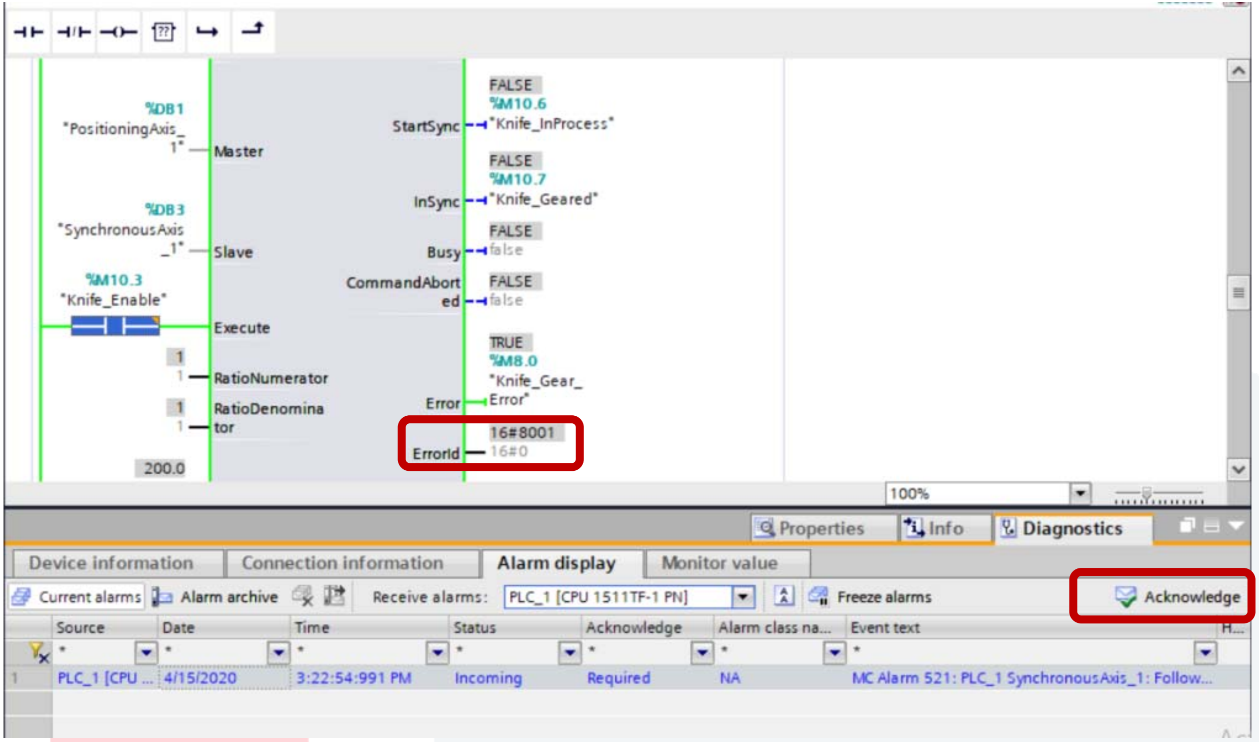
Although this Start location is defined as:

Start position = Synchronous position of leading axis - Synchronization length

…the dynamic limits of the Following axis may affect the actual start location relationship.

Save Project

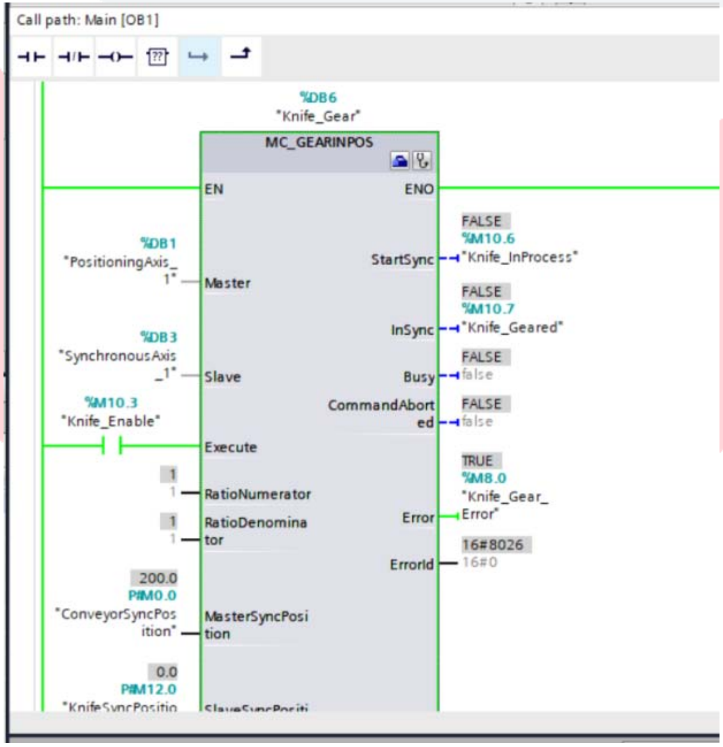
A common error, especially if an over aggressive acceleration has occurred is ErrorID 8001 (Technology Error) and further code evaluation shows MC Alarm 521, Following Error limit exceeded. This can be acknowledged by putting focus on the Alarm, in the “Alarm display” dialog window, and clicking on the “Acknowledge” icon.



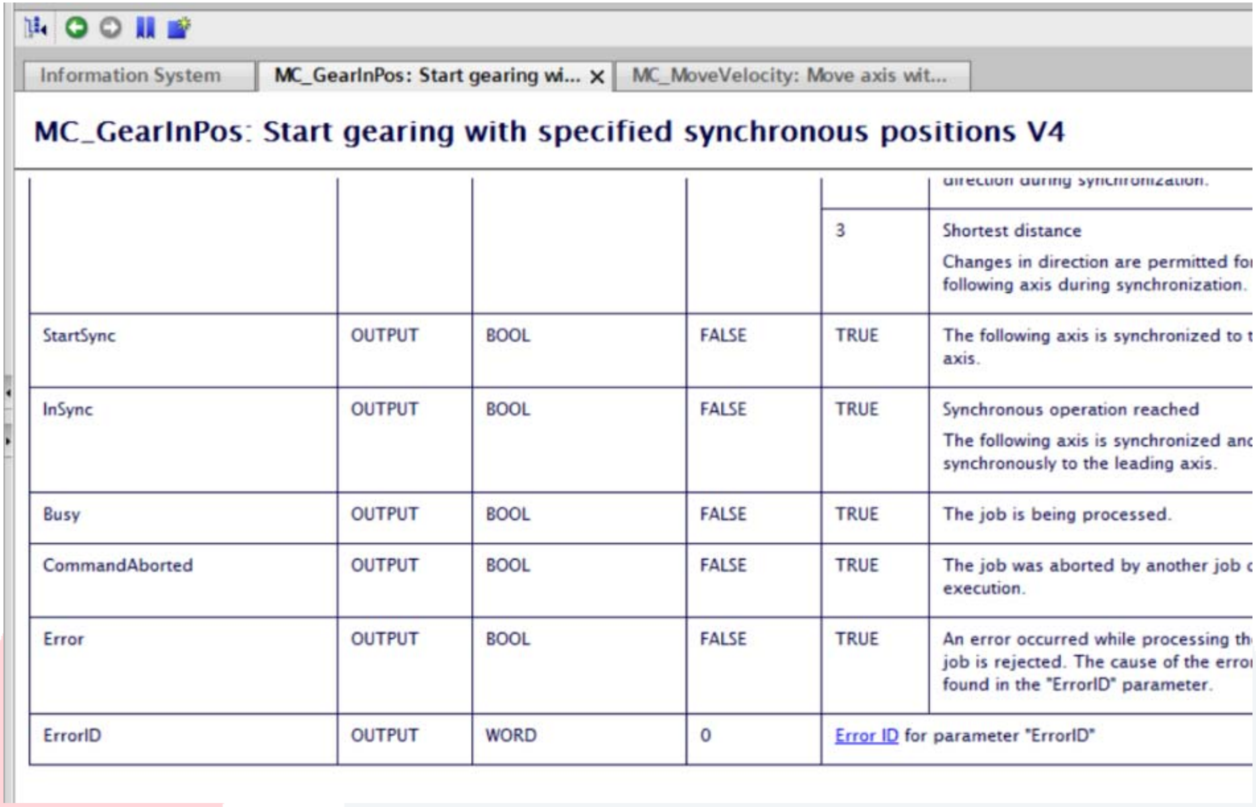
Receive alarms: PLC_1 [CPU 1511TF-1 PN]

	Source	Date	Time	Status	Acknowledge	Alarm class na...	Event text	H...
1	PLC_1 [CPU ...	4/15/2020	3:22:54:991 PM	Incoming	Required	NA	MC Alarm 521: PLC_1 SynchronousAxis_1: Follow...	

In order to track down a Technology Block error (which is not reported as a Hardware fault) one needs to observe the Error ID reported from the effected Technology Block, select F1 Help while having Focus on that Technology Block and reference the information in the related Help description or additionally, follow the link to the decoded error codes as shown below with Error ID of (Hex)8026 displayed:



The Help information provides the link to the Error ID table.

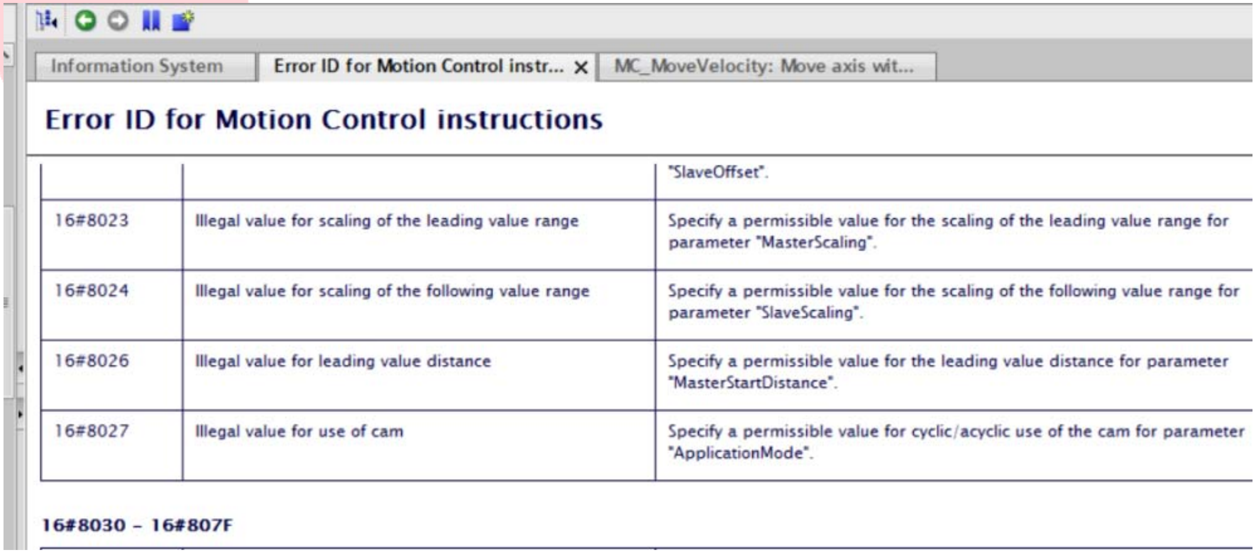


Information System
MC_GearInPos: Start gearing wi... ×
MC_MoveVelocity: Move axis wit...

MC_GearInPos: Start gearing with specified synchronous positions V4

					direction during synchronization.
				3	Shortest distance Changes in direction are permitted fo following axis during synchronization.
StartSync	OUTPUT	BOOL	FALSE	TRUE	The following axis is synchronized to t axis.
InSync	OUTPUT	BOOL	FALSE	TRUE	Synchronous operation reached The following axis is synchronized anc synchronously to the leading axis.
Busy	OUTPUT	BOOL	FALSE	TRUE	The job is being processed.
CommandAborted	OUTPUT	BOOL	FALSE	TRUE	The job was aborted by another job c execution.
Error	OUTPUT	BOOL	FALSE	TRUE	An error occurred while processing th job is rejected. The cause of the erroı found in the "ErrorID" parameter.
ErrorID	OUTPUT	WORD	0	Error ID for parameter "ErrorID"	

The Error ID table lists the following, and a correlates that a negative value was being used as a Sync Distance in this example.



Information System
Error ID for Motion Control instr... ×
MC_MoveVelocity: Move axis wit...

Error ID for Motion Control instructions

		"SlaveOffset".
16#8023	Illegal value for scaling of the leading value range	Specify a permissible value for the scaling of the leading value range for parameter "MasterScaling".
16#8024	Illegal value for scaling of the following value range	Specify a permissible value for the scaling of the following value range for parameter "SlaveScaling".
16#8026	Illegal value for leading value distance	Specify a permissible value for the leading value distance for parameter "MasterStartDistance".
16#8027	Illegal value for use of cam	Specify a permissible value for cyclic/acyclic use of the cam for parameter "ApplicationMode".

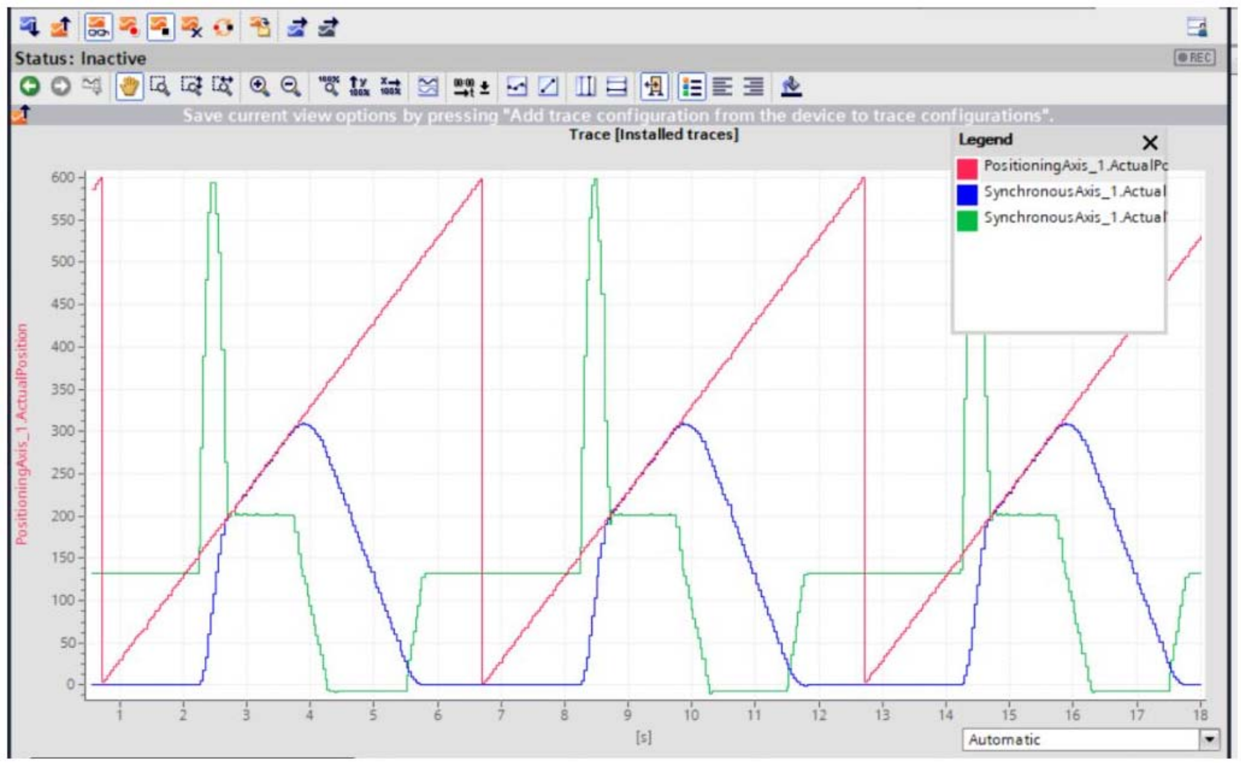
16#8030 – 16#807F

MODULO ADENDDUM
With Scope Trace Tips

Modulo is a configuration value found in the TO configuration that can be selected to automatically create a resetting position value for mechanical systems that actually repeat cyclically, and precisely. Modulo is very common with rotary table applications where the same product nest will be presented to a given work station every 360 degrees. Recently, modulo has been made available for axes defined as Linear. More typical of a lower precision linear application would be to reset a zero position (Home) based on the input of a sensor that confirm the leading edge of a product cell, but if a linear application is precise enough to know exactly the distance from nest to nest datum, then modulo could be used instead of a registration sensor.

To base a modulo example on the exercise on hand, the assumption is that product is made available every 600mm on a segmented conveyor. The knife must match conveyor speed and position at the 200mm location, and maintain that 1:1 speed lock until the 300mm location of the conveyor to allow a knife to cut across the product, and then return to the Knife starting point. Although the conveyor moves forward continuously, with Modulo set to 600mm, the reported “Actual Position” of the Conveyor (red trace) drops from 600 to 0 at the Modulo point. Based on the “Override” rules of the motion blocks, PLC logic has been added to the FB to trigger an MC_MoveAbsolute of the Knife at the desired “Reset Position”. Override rules dictate that a new PLC Open function will abort and override a currently operating function. This is strategic and useful. The assigned acceleration and deceleration make the transition predictable.

Below, Conveyor Velocity= 100, MC_GearInPos MasterSyncPosition=200, FollowerSyncPosition=200, MasterStartDistance=50. Since the target sync positions are both 200mm, the positions of the two axes lie right on top of each other in the trace below during the synchronous period at the 1:1 ratio defined. This could also be done as a CAM profile…

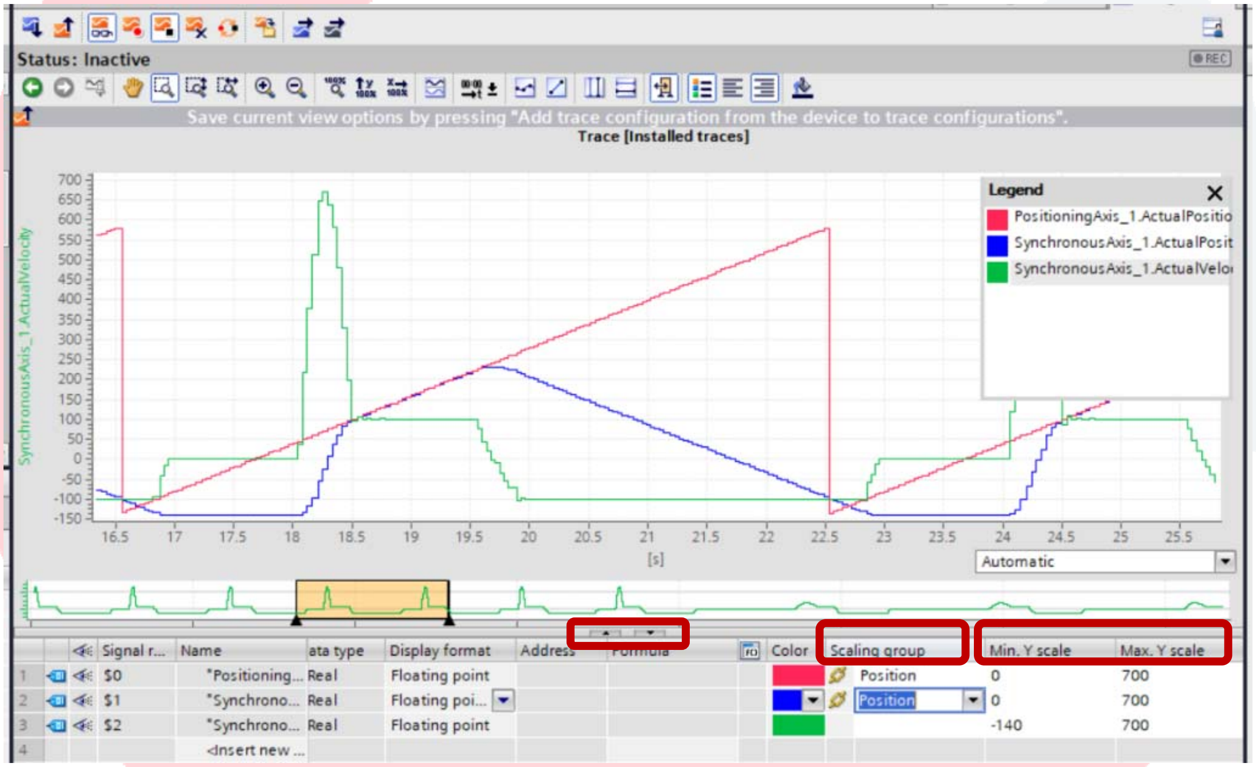

Status: Inactive

Save current view options by pressing "Add trace configuration from the device to trace configurations".

REC
Trace [Installed traces]
Legend
🟥 PositioningAxis_1.ActualPo
🟦 SynchronousAxis_1.Actual
🟩 SynchronousAxis_1.Actual
PositioningAxis_1.ActualPosition
600
550
500
450
400
350
300
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
[s]
Automatic

From the scope trace above (zoomed in below), it can be seen that the Knife axis needed to accelerate aggressively to catch up to the sync position yet was sitting stationary for part of the cycle. To reduce wear and tear on the mechanics, it would be beneficial to adjust the sync distance relative to the Master Axis speed to reduce the required acceleration and peak velocity needed for the specific settings.

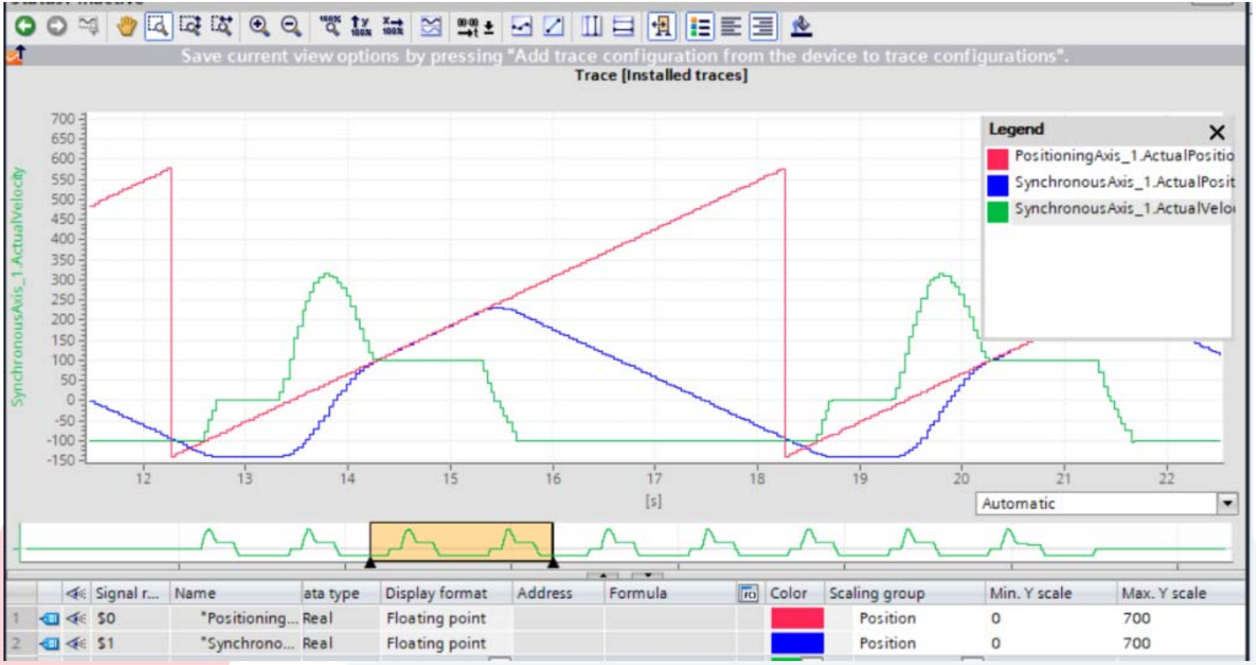
Notice below, to specify desired Y axis scale and to lock one trace to another traces scaling, expand the hidden “Signal Table” then set the Min and Max Y scale values for one channel, type in a label for “Scaling Group” (i.e. “Position”) then select that group for any other channel that should use the same scaling. Additionally, below, to change the Y axis reference values displayed at the left, use the icon for “Show Chart Legend” in the menu bar and when a tag name is clicked on, the Y axis scale to the left will show the respective scale. The icon to switch the traces from individual trace areas to one shared area is the “Arrange in tracks” icon in the menu bar.



Save current view options by pressing "Add trace configuration from the device to trace configurations".

	Signal r...	Name	ata type	Display format	Address	Formula	Color	Scaling group	Min. Y scale	Max. Y scale
1	$0	"Positioning...	Real	Floating point			🟥	Position	0	700
2	$1	"Synchrono...	Real	Floating poi...			🟦	Position	0	700
3	$2	"Synchrono...	Real	Floating point			🟩		-140	700
4		<Insert new ...								

Shown below, the only difference in the configuration was that the “MasterSyncDistance” was allowed to be 100mm. The maximum speed of the Knife is drastically reduced. If the value is too large, the Follower could miss the next synchronization cycle, therefore setting up a CAM profile would be a better solution if the Master conveyor speed was going to vary much.



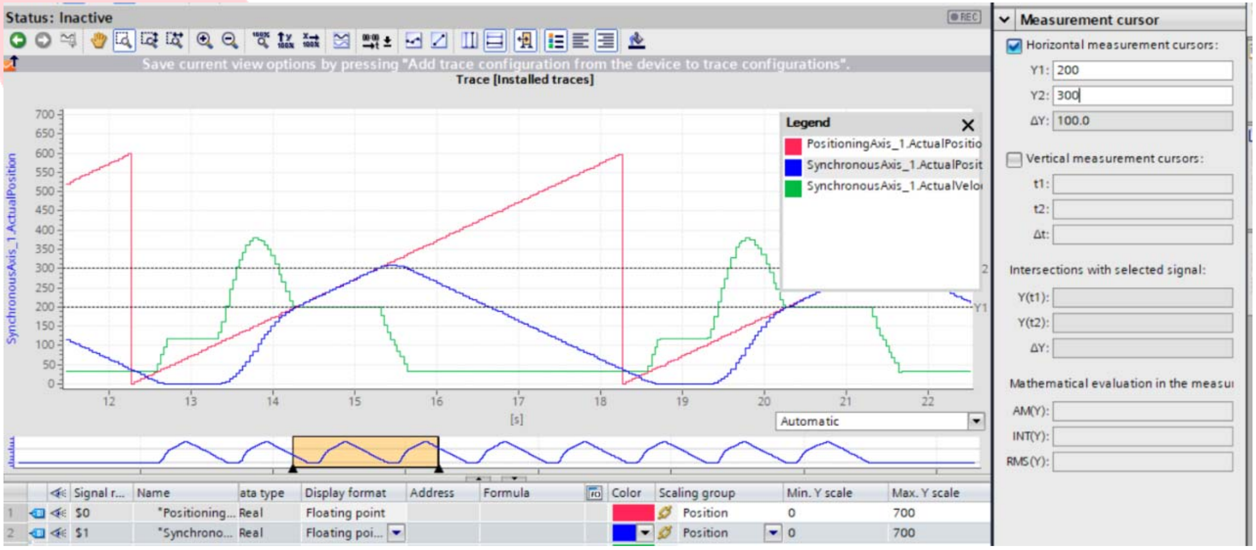
Save current view options by pressing "Add trace configuration from the device to trace configurations".

Trace [Installed traces]

SynchronousAxis_1.ActualVelocity
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
-50
-100
-150
12
13
14
15
16
17
18
19
20
21
22
[s]
Legend
PositioningAxis_1.ActualPositio
SynchronousAxis_1.ActualPosit
SynchronousAxis_1.ActualVelo
Automatic

	Signal r...	Name	ata type	Display format	Address	Formula	Color	Scaling group	Min. Y scale	Max. Y scale
1	$0	"Positioning...	Real	Floating point			🟥	Position	0	700
2	$1	"Synchrono...	Real	Floating point			🟦	Position	0	700

One more useful tool is the use of the measurement cursors in the menu bar, which can either be positioned by dragging the lines to desired locations, or if there is a known specific location they should be at, at the right, the desired locations can be manually typed in and the cursors will move to that location. Below, “Horizontal Measurement Cursor” positions are placed at 200mm and 300mm to confirm the 1 to 1 synchronous segment.



Status: Inactive

Save current view options by pressing "Add trace configuration from the device to trace configurations".

Trace [Installed traces]

REC
SynchronousAxis_1.ActualPosition
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
12
13
14
15
16
17
18
19
20
21
22
[s]
Y1
Legend
PositioningAxis_1.ActualPositio
SynchronousAxis_1.ActualPosit
SynchronousAxis_1.ActualVelo
Automatic

	Signal r...	Name	ata type	Display format	Address	Formula	Color	Scaling group	Min. Y scale	Max. Y scale
1	$0	"Positioning...	Real	Floating point			🟥	Position	0	700
2	$1	"Synchrono...	Real	Floating poi...			🟦	Position	0	700

Measurement cursor

☑ Horizontal measurement cursors:

Y1: 200

Y2: 300

ΔY: 100.0

□ Vertical measurement cursors:

t1:

t2:

Δt:

Intersections with selected signal:

Y(t1):

Y(t2):

ΔY:

Mathematical evaluation in the measu

AM(Y):

INT(Y):

RMS(Y):