LinMot® Rockwell Automation® Motion

Application Note / How-to

© 2022 NTI AG
This work is protected by copyright.
Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, microfilm, storing in an information retrieval system, not even for didactical use, or translating, in whole or in part, without the prior written consent of NTI AG.

LinMot® is a registered trademark of NTI AG.

Note
The information in this documentation reflects the stage of development at the time of press and is therefore without obligation. NTI AG reserves itself the right to make changes at any time and without notice to reflect further technical advance or product improvement.

NTI AG
LinMot®
Bodenaeckerstrasse 2
CH-8957 Spreitenbach

Tel.: +41 56 419 91 91
Fax: +41 56 419 91 92
Email: office@LinMot.com
Homepage: www.LinMot.com

Table of Contents

Attachments / File List

The package of which this documentation is part of should contain the following files and folders:

General (documentation and demo projects):
\0185-0105-E_1V4_AN_LinMot_Rockwell_CIPSyncMotion.pdf
\CIPSync_Demo_L320ERM_2Axes_V34_20220420.ACD
\CIPSync_Demo_L320ERM_2Axes_V30_20220420.ACD

Instructions:
\AOI\ > Instructions as single exports

Program exports:
\Exports\
\p10ms > Program exports of the p10ms task
\z_MotionGroupUpdate > Program exports of the z_MotionGroupUpdate task
\Optional > Example program exports for optional functionality

Drive EDS files:
\EDS\
- C1250-CM-XC-0S.eds
- C1250-CM-XC-1S.eds
- C1250-MICM-XC-0S.eds
- C1250-MICM-XC-1S.eds
- C1251-MICM-XC-2S.eds

Non motion demo projects:
\Misc\
- CM_EDS_NoMotion_Demo_L320ERM_V34_20220420.ACD
  > No motion demo for CM drives
- IP_AOP_NoMotion_Demo_L320ERM_V34_20220420.ACD
  > No motion demo for IP drives (C1250-IP)
- IP_EDS_NoMotion_Demo_L320ERM_V34_20220420.ACD
  > No motion demo for IP drives (E1250-IP, C1450-IP, E1450-IP)

Version History

Version	Date	Author	Description
1V0	March 20th, 2019	fj	Initial Version
1V1	August 27th, 2019	fj	- Optimized LM_ADV_PVATime: Integrated ADD64 instruction - Added PVAStatus flag in UDT tstLM_Axis which indicates whether virtual and physical axis are coupled or not - Added a new instruction where the EDS read&write and the PVATime instructions are nested (LM_ADV_CmdPosition) - Added force/torque control instructions - Added several instructions known from IP drives. - Moved CAInit instruction to the periodic task and cleaned code - Simplified demo project - Updated documentation - Added program exports to the package > check \Exports\ - Corrected tag name in UDT: tstLM_Axis.ActualPosition
1V2	September 23rd, 2019	fj	- LM_ADV_PVATime: PVAStatus was not always correctly reset - LM_LMDriveEDS_Read_78_64: Added CfgRWActive flag watchdog - LM_MAM: Added move types 2, 3 and 4 - Added new instructions: LM_CFG_CTAccess & LM_CFG_CurveAccess
1V3	November 5th, 2020	fj	- LM_MAM: added profile 3 (Sine) for move types 0 (absolute) and 1 (incremental) - LM_MAS: fixed wrong decelRate scaling - Added instruction LM_CFG_GetStringByUPID - Added instructions to support C1250-IP drives integrated using AOP (NO SYNCHRONIZED MOTION and with limitations!). LM_LMDriveAOP_Read_IP & LM_LMDriveAOP_Write_IP. No documentation of this instructions but an example project is available.
	September 6th, 2021	fj	- Added note in chapter 5.6.3 CIP Sync PVA Streaming (LM_ADV_PVATime)
1V4	July 07th, 2022	fj	- Added LM_MAH_MG instruction (2.0) support for master gantry homing on E1250-IP & E1450-IP drives - Added LM_LMDriveEDS_Read_IP instruction (2.0) support for E1250-IP & E1450-IP drives (EDS file import) - Added LM_LMDriveEDS_Write_IP instruction (2.0) support for E1250-IP & E1450-IP drives (EDS file import) - Added LM_LMDriveEDS_Write_IP_As27 instruction (2.0) support for E1250-IP & E1450-IP drives as Slave axis (EDS file import, ControlWord not transmitted) - Added LM_FC_FCST (2.0, force control standstill tara) - Added LM_FC_GTPT (2.0, go to pos tara) - Added LM_FC_FCCITF (2.0 **preliminary**, force/torque check in target force/torque) - Added instructions to read/write the drive configuration - LM_CFG_StopStartDefault: Fixed Mode 0 (Reboot) (version 2.0 > 2.1) - Added support for C1250-MI and C1251-MI (EDS files) - Updated example projects - Updated documentation - Updated/removed links in chapter Rockwell Automation

Use of This Document

Description: LinMot® Rockwell Automation® Motion CIP Sync Demo

Drive:
C1250-CM-XC-0S-x00
C1250-CM-XC-1S-x00
C1250-MI-XC-1S-x00
C1251-MI-XC-2S-x00
All LinMot -IP drives (No CIP Sync Motion supported! Check chapters 7.2.2 and 7.2.3)

Classification:
[x] Application Note
[ ] Installation Guide
[ ] User Manual
[ ] Documentation
[ ] LinMot internally

Status: Release

General

This application note shows how a LinMot CM interface (EtherNet/IP interface, e.g., C1250-CM-XC-xS drive) can be integrated and setup in a Rockwell Automation environment as a motion axis for coordinated motions.

Nevertheless, there are also instructions available (see chapters 5.9 and 5.10) that allow direct control of the physical axis.

Example projects can be downloaded from:
http://download.linmot.com/plc_lib/examples/Rockwell_CM/ (CIPSync_Demo...)

Components used:
- LinMot C1250-CM-XC-0S-000 (article number 0150-2900) with firmware 6.8 Build 20190315
- Allen-Bradley 5069-L320ERM CompactLogix™ 5380 Controller, Revision 30.011
- Studio 5000 Logix Designer Version 30.01.00 – Professional Edition

!!! Mandatory Prerequisites !!!
- Rockwell Automation controller revision must be **V30 or later**
- LinMot drive firmware must be **6.8 Build 20190315 or later**
- LinMot drive firmware must be **6.11 Build 20220628 or later** to use all functions

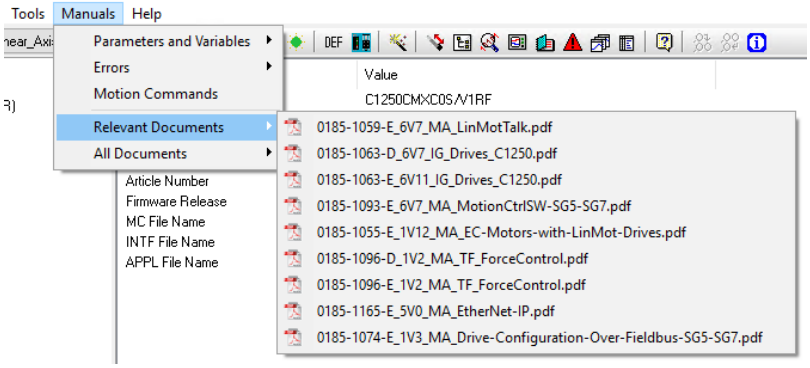
Hint:
The LinMot Drive is connected to a virtual axis. Virtual axes do not consume a motion axis.
In principle, a Rockwell Automation controller without motion (-M) can be used.

Recommended Documentation

LinMot

Reading the following user manuals is essential to understand the communication between the Rockwell Automation controller and the LinMot drive. The manuals are included in the LinMot-Talk software (*Menu Manuals → All Documents* or *Relevant Documents* if logged in to a drive) or can be downloaded from the LinMot eCatalogue (search by document reference): http://shop.linmot.com

User Manual	**Document Reference**
LinMot-Talk	0185-1059
Motion Control Software SG5 – SG7	0185-1093
Ethernet/IP CIP Sync	0185-1165
Drive Configuration Over Fieldbus SG5 – SG7	0185-1074
Application Notes	**Document Reference**
HT Config over Realtime	0185-0142-E / 0185-0142-D
HT Oscilloscope	0185-0132-E / 0185-0132-D
HT Position Loop Tuning	0185-1156-E / 0185-1156-D
HT Closed Loop Force/Torque Tuning	0185-0144-E / 0185-0144-D



Rockwell Automation

The following is an excerpt of the Rockwell Automation literature library concerning EtherNet/IP, CIP Sync, Motion, switches, and network architectures.

Design and Implementation Guides:
https://literature.rockwellautomation.com/idc/groups/literature/documents/qr/iasimp-qr019_-en-p.pdf
https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/enet-wp035_-en-p.pdf

Integrated Architecture and CIP Sync Configuration:
https://literature.rockwellautomation.com/idc/groups/literature/documents/at/ia-at003_-en-p.pdf

Stratix® Switches:
https://literature.rockwellautomation.com/idc/groups/literature/documents/um/1783-um007_-en-p.pdf

OEM Network Guidelines:
https://literature.rockwellautomation.com/idc/groups/literature/documents/rm/enet-rm001_-en-p.pdf
https://literature.rockwellautomation.com/idc/groups/literature/documents/rm/enet-rm002_-en-p.pdf

Troubleshoot EtherNet/IP Networks:
https://literature.rockwellautomation.com/idc/groups/literature/documents/at/enet-at003_-en-p.pdf

Integrated Motion on the EtherNet/IP Network: Configuration and Startup
https://literature.rockwellautomation.com/idc/groups/literature/documents/um/motion-um003_-en-p.pdf

1 System Overview

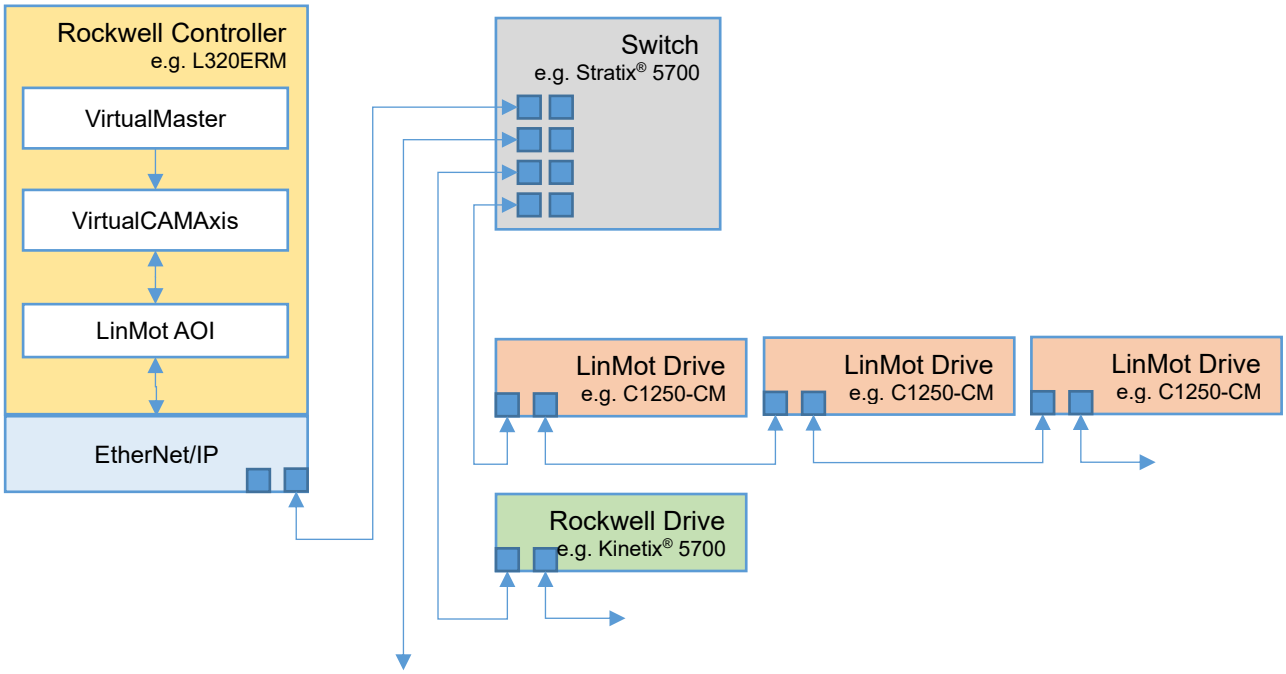
1.1 Network Topology

The same network limitations for Rockwell Automation motion drives must be considered for LinMot drives. (Check the chapter recommended documentation > Rockwell Automation).

It is recommended to use a dedicated switch port for the LinMot drives.

A potential network topology is shown below.

In this application note two LinMot C1250-CM-XC-0S-000 drives are connected to a Rockwell Automation CompactLogix™ L320ERM controller.



1.2 Comparison

The following table shows a comparison between the actual integration developed by LinMot & Rockwell Automation and between CIP Motion

	LinMot CIP Sync Motion	CIP Motion
Logix Version	V30 or later	V19 or later
Motion Update Rate	CUP min. 1ms	CUP min. 0.5ms
Configuration	Logix & LinMot-Talk	Logix
High network load?	no influence	no influence
Late packets?	no impact	no impact
Network topology	Integration in normal motion topology possible	Integration in normal motion topology possible
Actual values (D2C)	Actual values not synchronized with motion planner	Actual values synchronized with motion planner
Motion command (C2D)	Motion command synchronized with other motion	Motion command synchronized with other motion
DeviceLevelRing (DLR)	supported	supported
IEEE1588 – PTP V2	supported	supported

1.3 Limitations

- **This is an application-based solution.**

  LinMot CIP Sync Motion does not provide full CIP Motion integration!
- LinMot-Talk software required to configure drive/motor/tuning
- No CUR less than 1-2ms (up to processor generation L8/L7)
- No time changes during motion allowed (same with Kinetix® -> see KB or manuals[1])
- No ethernet topology changes (grandmaster change) during motion allowed
- Managed switch (Stratix® 5700/5400 -> CGP/CGN) recommended

[1] https://rockwellautomation.custhelp.com/app/answers/detail/a_id/730060/page/1

2 Configuration and Connection of the LinMot Drive(s)

2.1 Configuration and Connection of the LinMot Drive

The LinMot drive is configured using the LinMot-Talk software:
http://www.linmot.com/download/linmot-talk-drive-configuration/

To login into the drive over Ethernet see chapter *9 Appendix I: LinMot-Talk Configuration over EtherNet*/IP.

2.2 Motor Configuration

It is assumed that the motor connected to the drive is already configured.

Use the *Motor Wizard* inside LinMot-Talk to setup the motor (Menu -> Drive -> Motor Wizard):

Note:
If there is any doubt, reset the drive to default values and then configure the motor with the motor wizard. Be sure to save your actual configuration before this step!

Set drive to default values (E1100, E1200, E1400, B8050, MB8050, C1100-GP and C1250):
Remove 24V supply from drive.
Set both address selectors (S1 and S2) either rotary hex switches to F or all dip switches to 1
Restore 24V power to drive. The ERROR and WARN LEDs should blink alternately.
Set both address selectors (S1 and S2) to 0.
Wait until EN and WARN led blink together.
Remove and restore 24V power to drive.

Set drive to default values (C1x00, E1200 und E1400 series drives only):
In the LinMot-Talk Software the parameters can defaulted by selecting the “DEF” button:

2.3 EDS Files

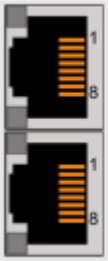
Install the EDS file(s) that is part of the LinMot-Talk software/firmware you are using with the Rockwell Automation software “EDS Hardware Installation Tool”:

The most recent device files are always part of the newest LinMot-Talk software. They are located by default:

- EtherNet/IP: \\LinMot-Talk X.X Build XXXXXXXX\Firmware\Interfaces\EtherNetIP_NX\EDS

2.4 EtherNet/IP Connection

The drive is connected to the EtherNet/IP network using the X17 & X18 connectors.

X17 – X18	RealTime Ethernet 10/100 Mbit/s
	X17 RT ETH In X18 RT ETH Out
RJ-45	

2.5 IP Address

The default IP address is 192.168.001.xxx, where the last byte xxx is defined via the two address selectors *S1 & S2*. *S1* sets the high and *S2* the low digit. E.g., S1 = 5, S2 = A -> 5A (hex) = 90 (dec) -> IP = 192.168.1.**90**

Set IP address (last byte, xxx from above) by S1 & S2

S1 - S2	Address Selectors		
E1100 E1200 V1	E1200 V2 E1400 C1x00		
		S1 (5..8)	Bus ID High (0 ... F). Bit 5 is LSB, bit 8 MSB.
		S2 (1..4)	Bus ID Low (0 ... F). Bit 1 is LSB, bit 4 MSB.

Attention:
The switch value **S1 = S2 = 0 (factory default setting)** is a special configuration which acquires the IP address via DHCP (e.g., for use with the BOOTP-DHCP Tool from Rockwell Automation)!

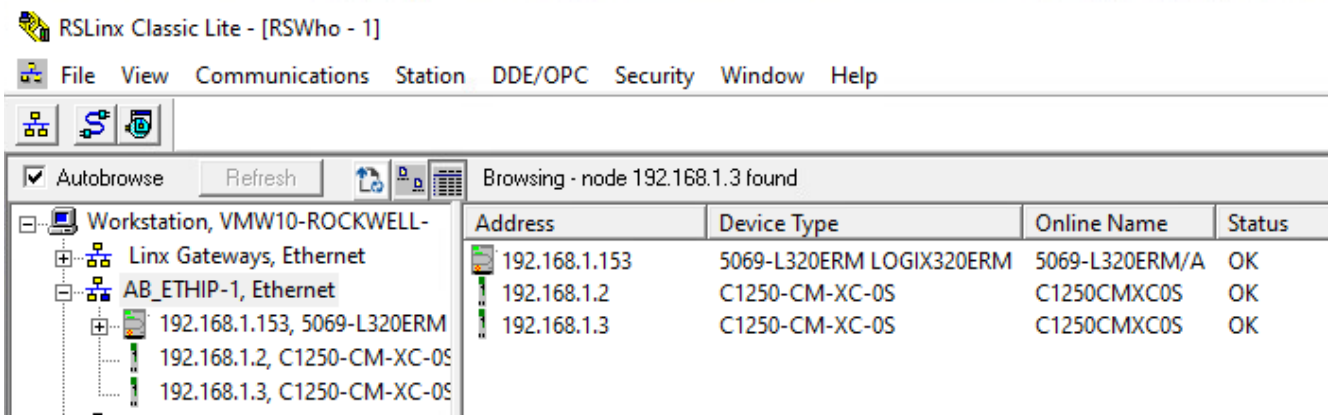
Note:
To set the IP address by parameter (assign a static IP address) login to the drive using LinMot-Talk.
Parameters > EtherNet/IP Intf > EtherNet Configuration

3 Controller Setup EtherNet/IP

3.1 RSLinx Classic

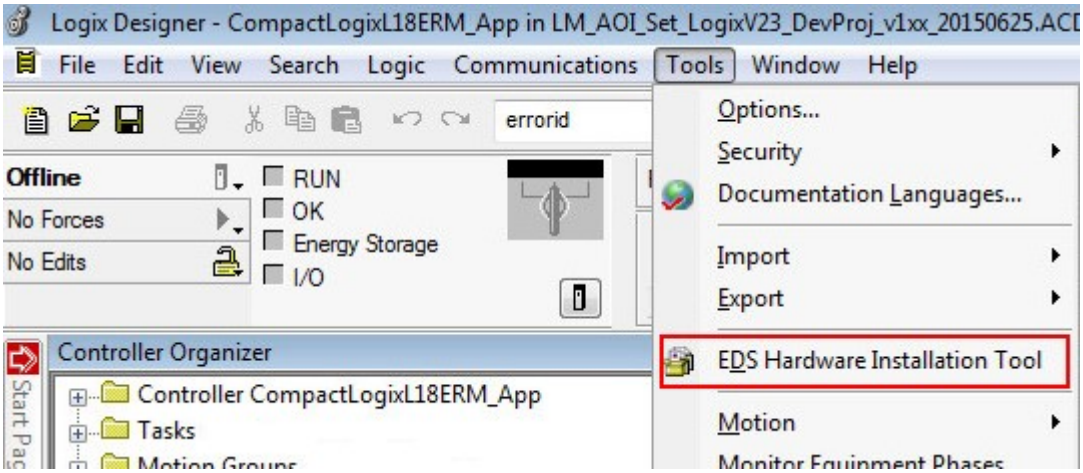
After setting the IP address the new LinMot device should appear in RSLinx Classic as “Unrecognized Device”.

If the EDS file is already installed, the LinMot drives will appear in RSLinx with its Device Type (e.g., IP 192.168.1.2).



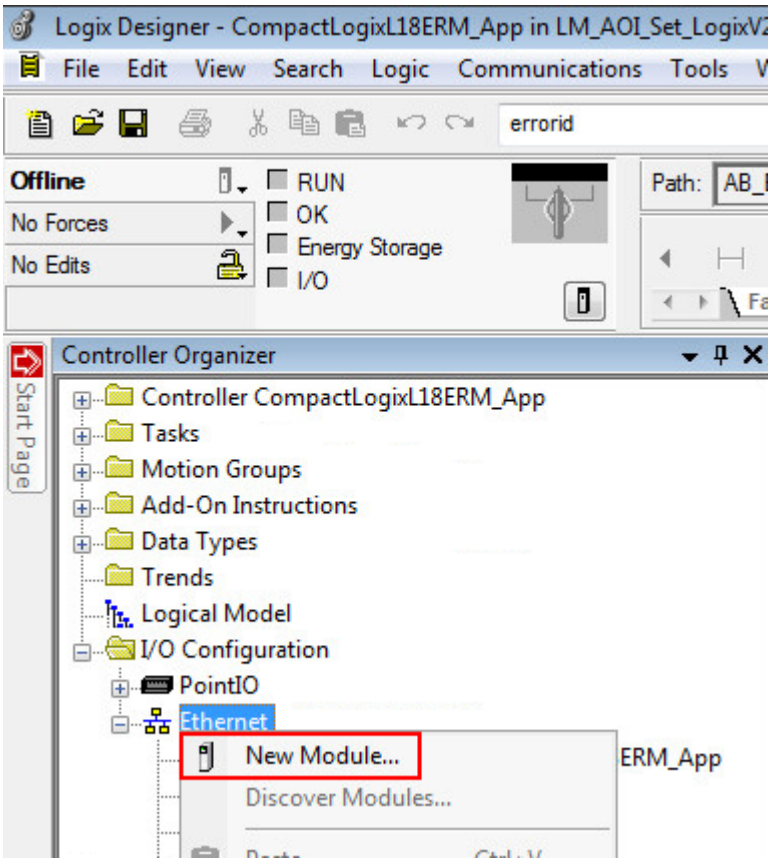
3.2 Add a LinMot Drive to the I/O configuration using the EDS file

1. The EDS files can be installed using the EDS Hardware Installation Tool:

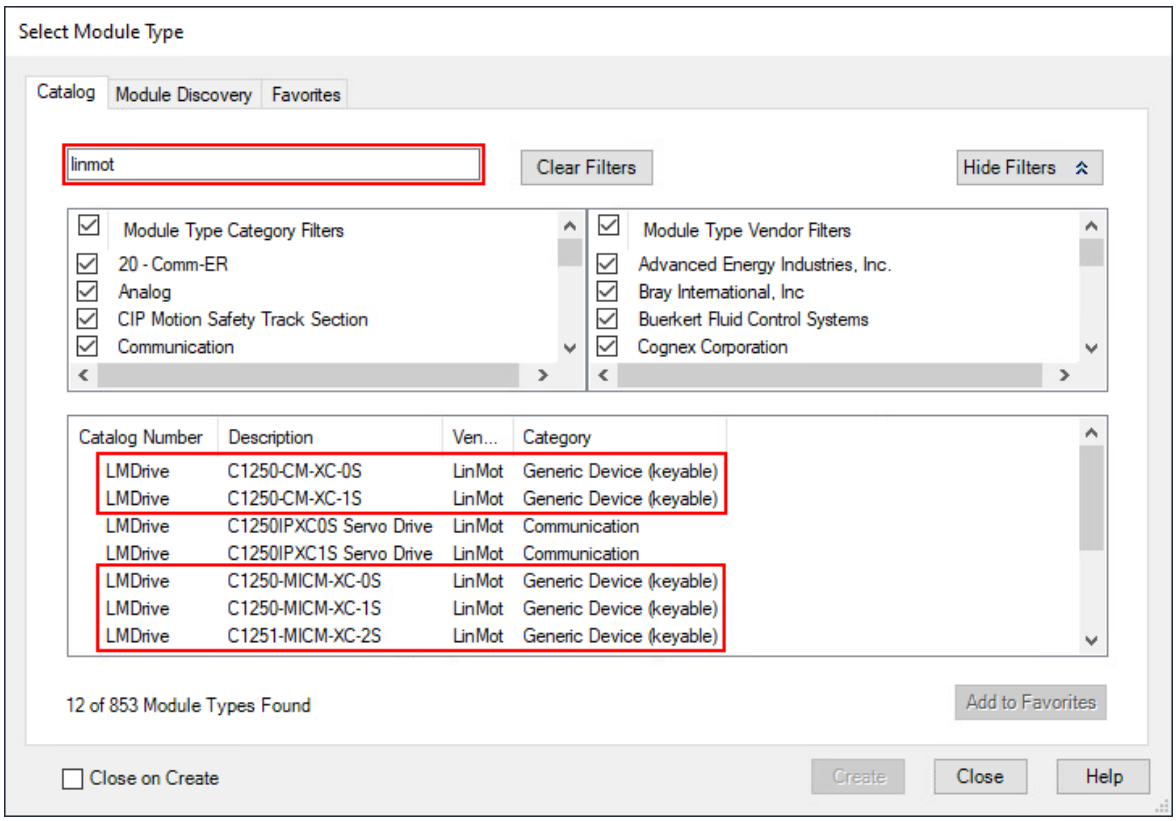


Follow the wizard and install the EDS files from the location mentioned in chapter 2.3.

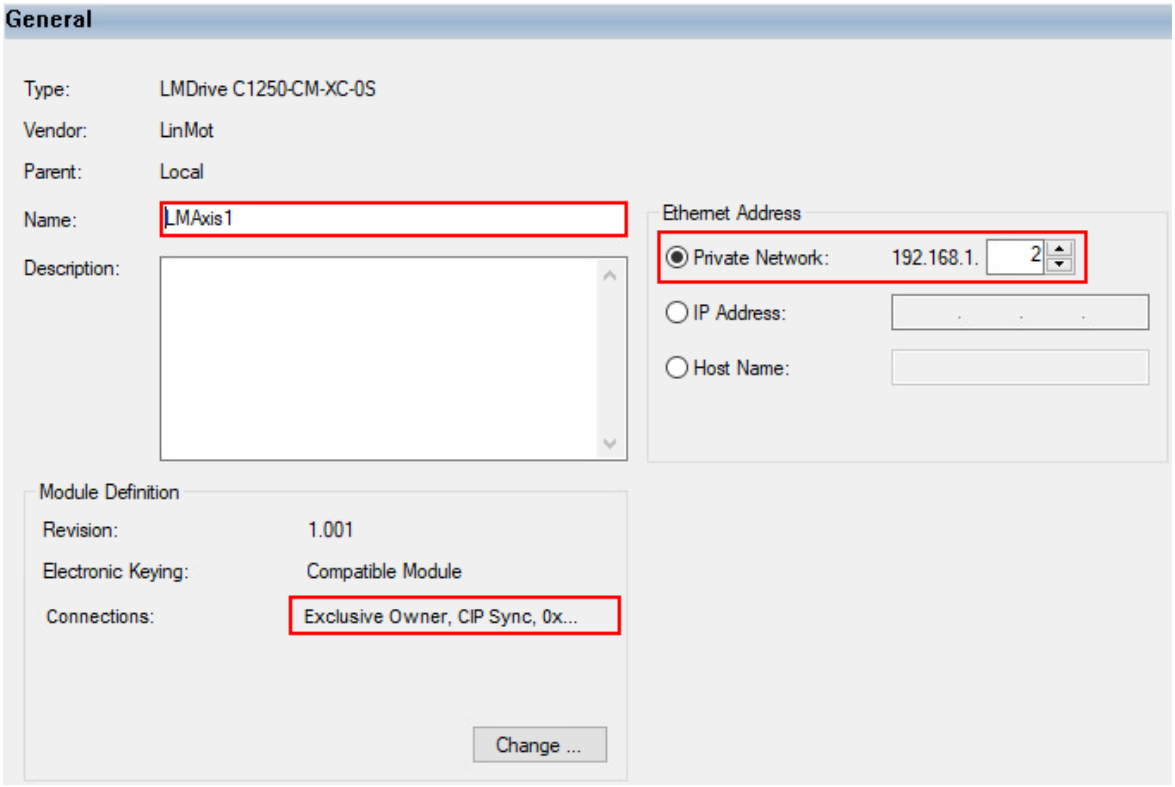
2. Add a new module by right clicking on Ethernet in the I/O configuration in the controller organizer:



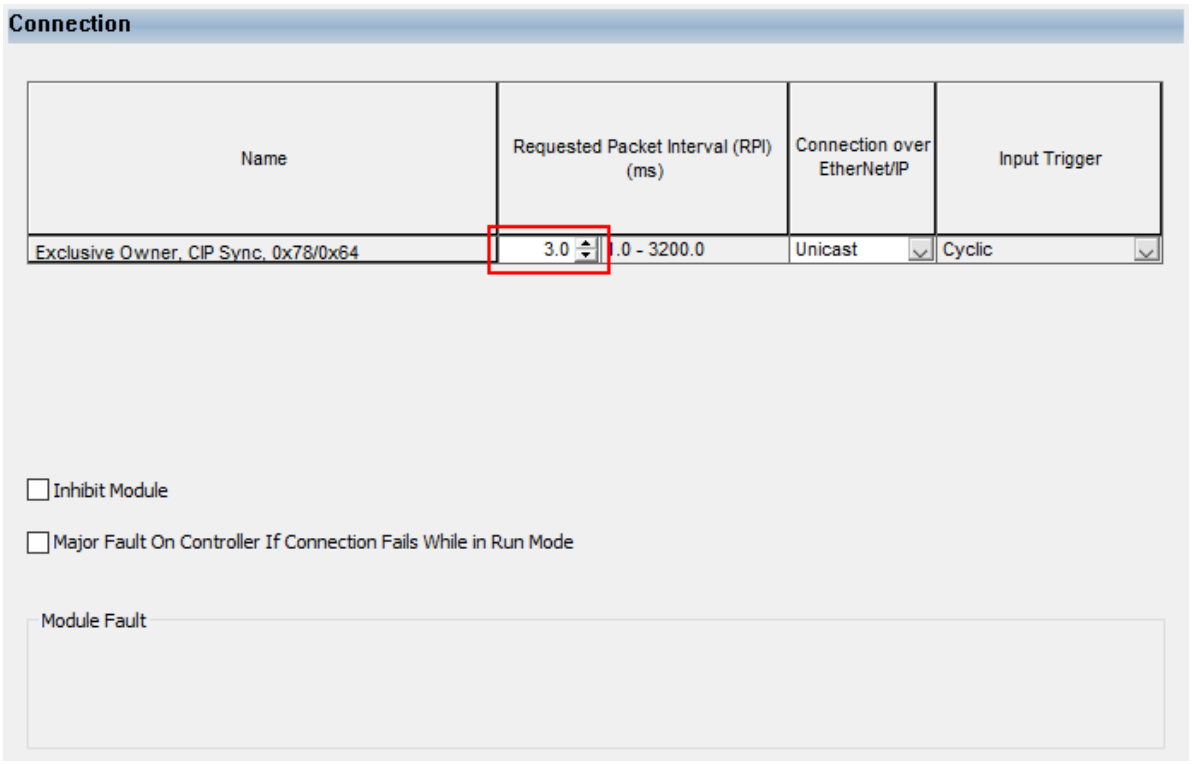
3. Choose the drive you want to add, and select “Create”:



4. Open the properties of the module and set Name & Ethernet Address in the *General* tab:

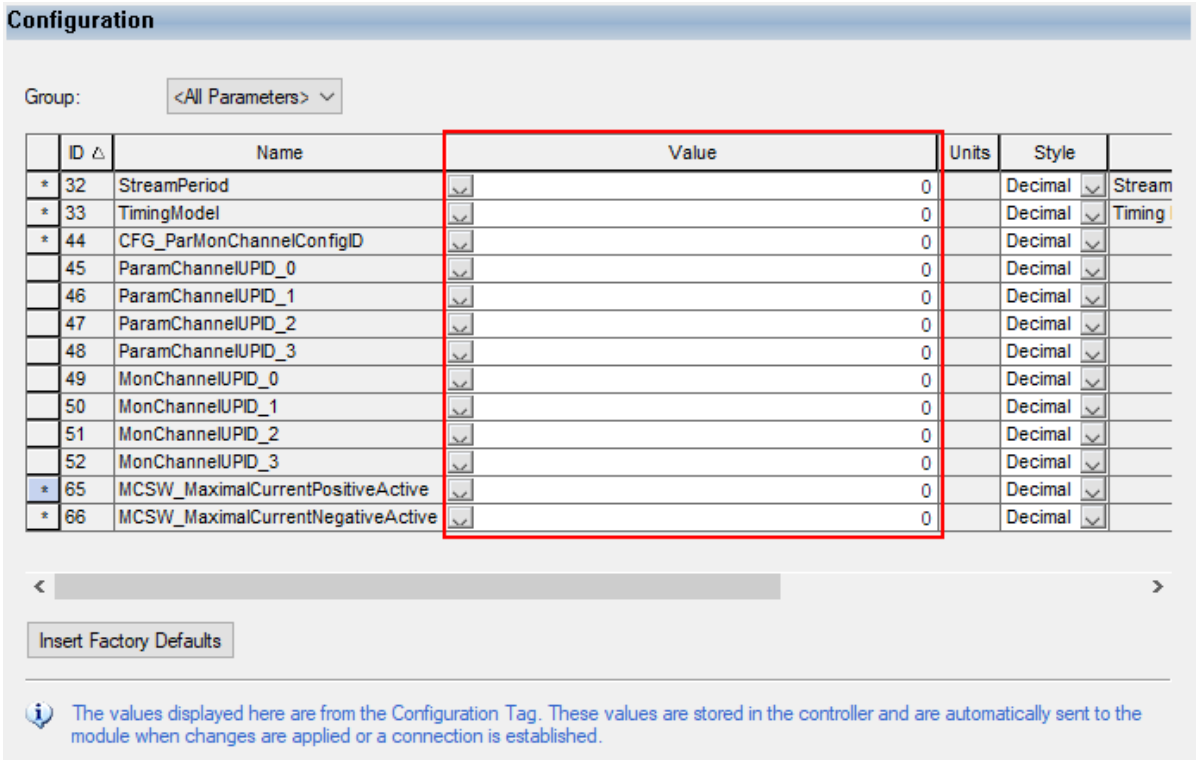


5. In the *Connection* tab set the RPI:



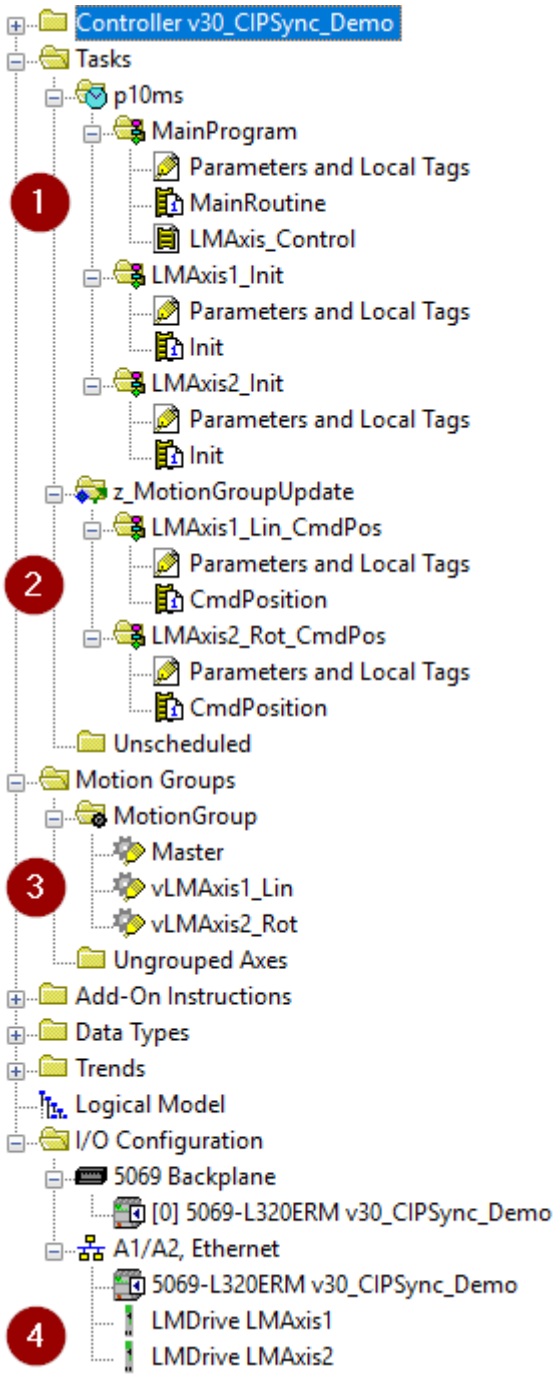
Attention:
Set the RPI a little longer than the update period of the motion group base.
The recommended RPI is the **Motion Group Update Period + 1ms**. E.g., 2ms + 1ms = **3ms.**

6. In the *Configuration* tab leave all values at 0 (zero). They are set by the LM_CFG_CAInit instruction.
It is very important to setup the corresponding MESSAGE correctly > see chapter 4.1.2



4 Controller Overview

The following image shows the Controller Organizer of the LinMot CIP Sync Demo project.



Attention:
- Rockwell Automation controller revision must be **V30 or later**
- LinMot drive firmware must be **6.8 Build 20190315 or later**

Note:
All programs shown in the following chapters are available as single program exports in the folders *\Exports\p10ms* and *\Exports\z_MotionGroupUpdate*

4.1 Task: p10ms

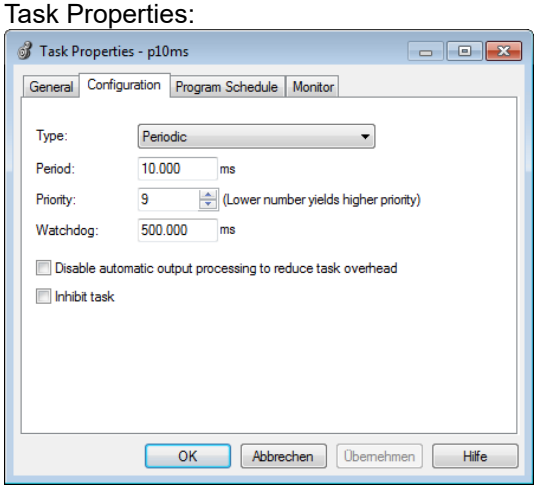
The **p10ms** task is the normal controller task where the general logic is implemented.

4.1.1 Program: MainProgram

The LinMot virtual axes (vLMAxis1_Lin and vLMAxis2_Rot) can be controlled in the **MainRoutine** with Rockwell motion commands and are coupled to the LinMot drives through the LM_ADV_CmdPosition AOI.

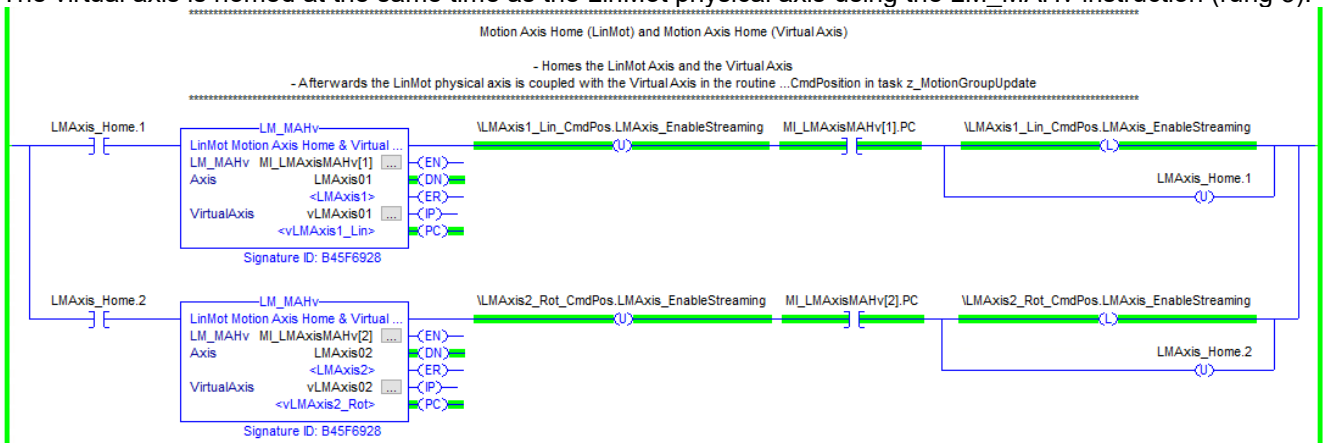
The virtual master axis (Master) is controlled in the **MainRoutine** through jogging.
A CAM curve (MAPC) can be used to synchronize the LinMot virtual axes to the virtual master axis.

Task Properties:



In the sub-routine **LMAxis_Control** the LinMot instructions for the LinMot drives like LM_MSO, LM_MSF, LM_MAFR, LM_MRPv and LM_MAHv are called.

The virtual axis is homed at the same time as the LinMot physical axis using the LM_MAHv instruction (rung 3).



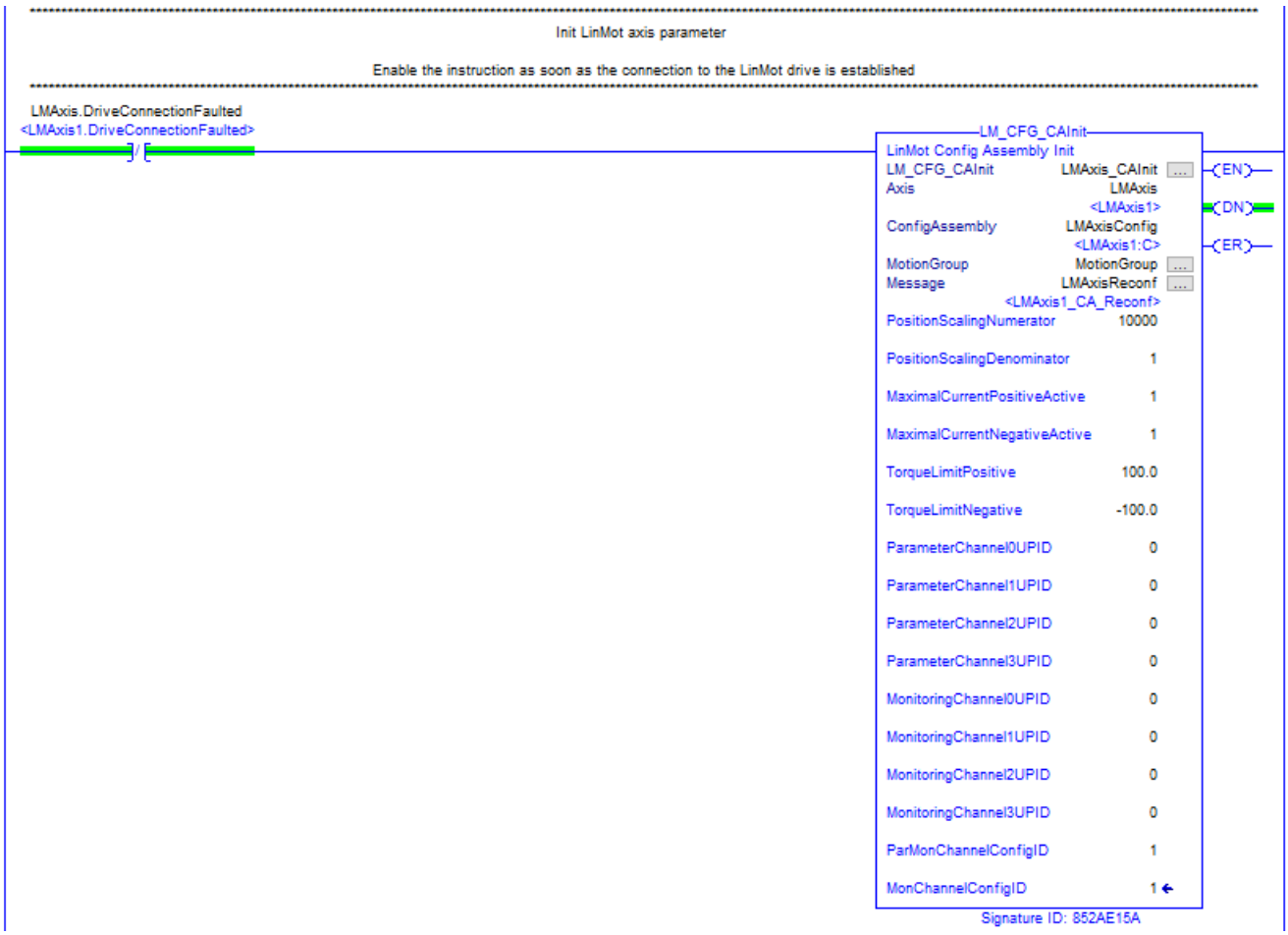
After homing is completed, streaming is enabled and the LinMot physical axis is coupled with the virtual axis.

4.1.2 Program: LMAxis1_Init & LMAxis2_Init

Attention: The correct setup of the configuration assembly is essential for a working application.

The **Init** sub-routine is very important as it sets the required parameters for time synchronization and unit scaling.

That happens once at system start/restart when the connection to the LinMot drive is established.
(Hint: The instruction executes on an enable false to true transition. That means it could be re-executed whenever required)



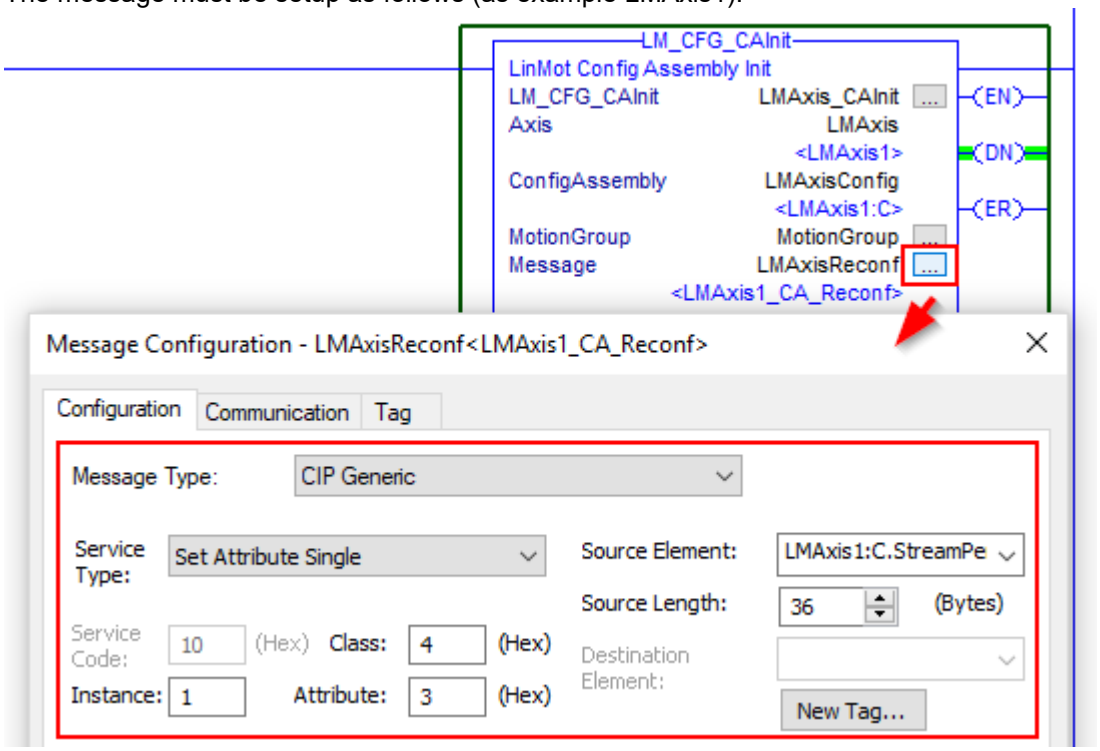
Mandatory parameters beside the in/out tags:

Tag Name	Description
PositionScalingNumerator	Numerator for unit scale. **Also check chapters 4.3.1 & 4.3.2**
PositionScalingDenominator	Denominator for unit scale. **Also check chapters 4.3.1 & 4.3.2**

Optional but recommended parameters:

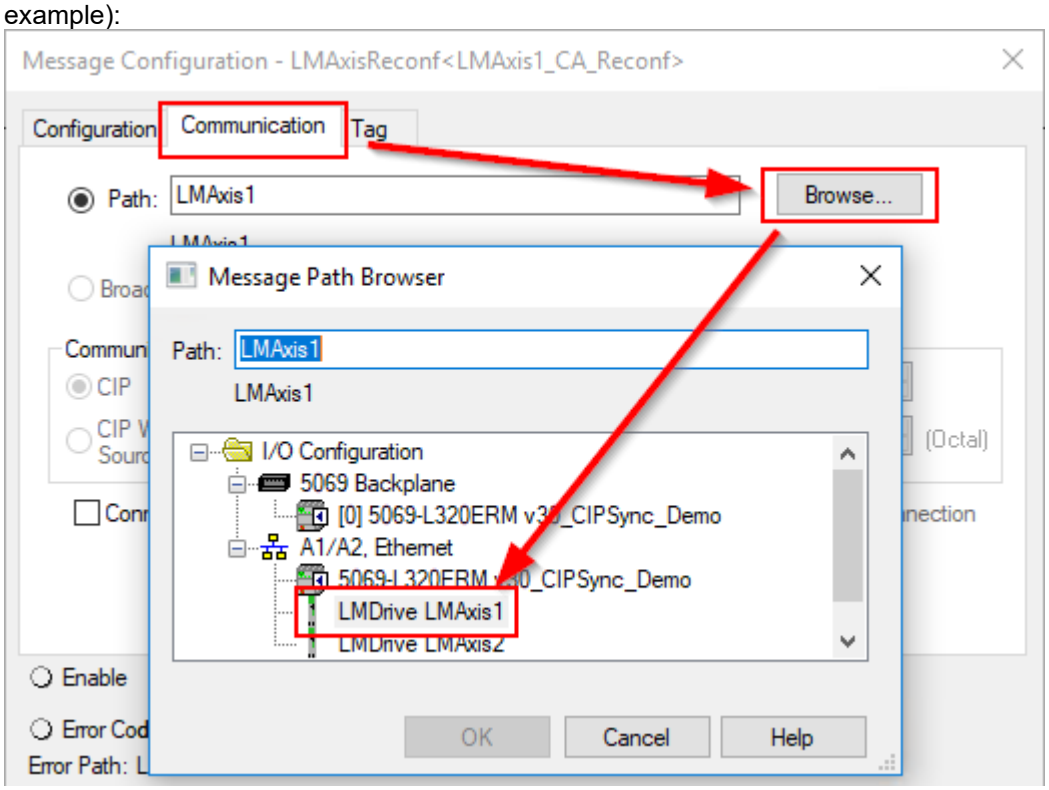
Tag Name	Description
MaximalCurrentPositiveActive	Config Assembly: Activate maximal current positive in output assembly
MaximalCurrentNegativeActive	Config Assembly: Activate maximal current negative in output assembly
TorqueLimitPositive	Init value of torque limit positive. Should be 100.0 as initialization value
TorqueLimitNegative	Init value of torque limit negative. Should be -100.0 as initialization value

These configuration parameters are set active by sending a CIP Generic message (Set Attribute Single).
The message must be setup as follows (as example LMAxis1):



For the ***Source Element*** configuration, the first entry of the configuration assembly must be selected which is *LMAxis1:C.StreamPeriod* in this example

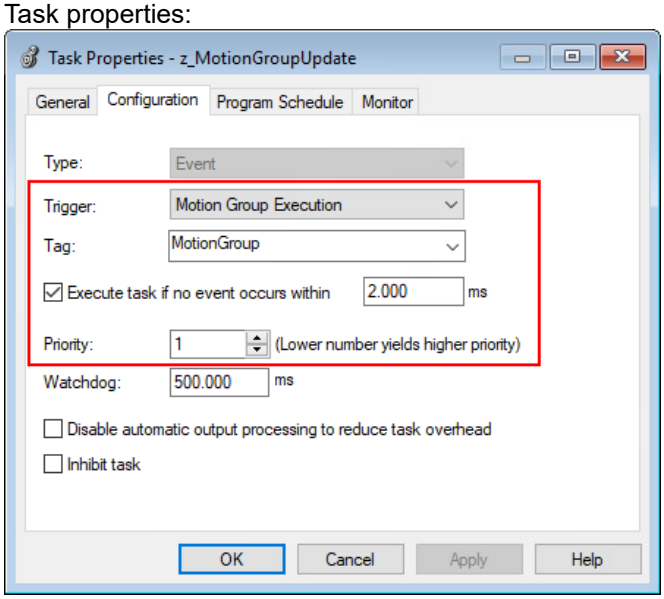
In the tab *Communication* the Path must be set to the corresponding I/O device (LMDrive LMAxis1 in this example):



4.2 Task: z_MotionGroupUpdate

This task is triggered by the motion group execution and therefore runs in sync with the interpolation of the axes command values. This is essential to get the application properly running!

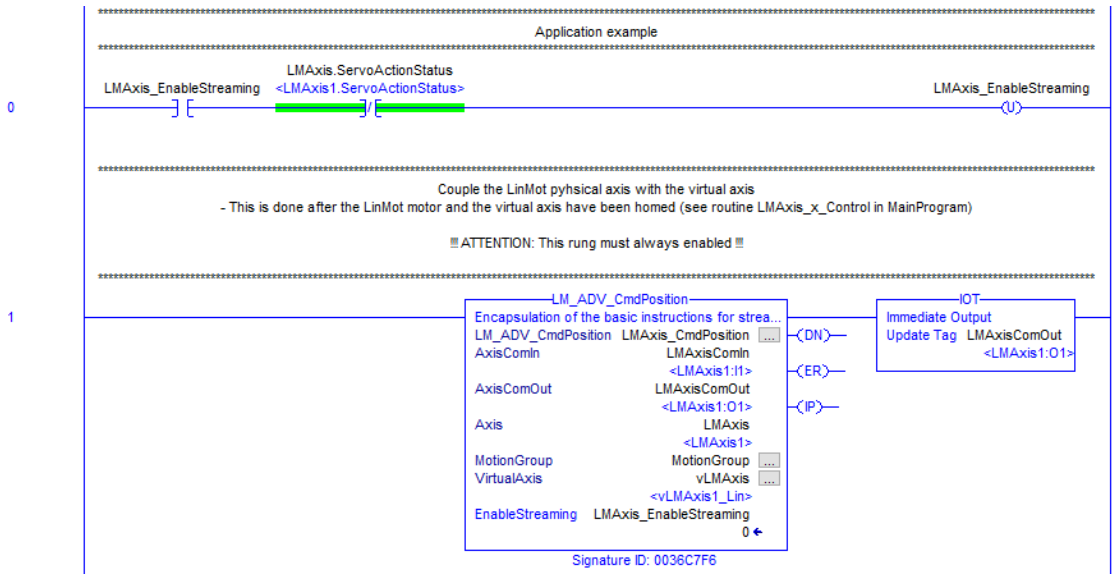
Task properties:



Optional: If the option “Execute task if no event occurs within” is enabled the time must be set the same as the Motion Group *Base Update Period*.

4.2.1 Program: LMAxis1_Lin_CmdPos & LMAxis2_Rot_CmdPos

In the routine **CmdPosition** the LM_ADV_CmdPosition & IOT instructions are called.
The LM_ADV_CmdPosition instruction reads and writes the LinMot drive I/O data and sends the position, velocity and acceleration set points of the virtual motion axis to the drive.



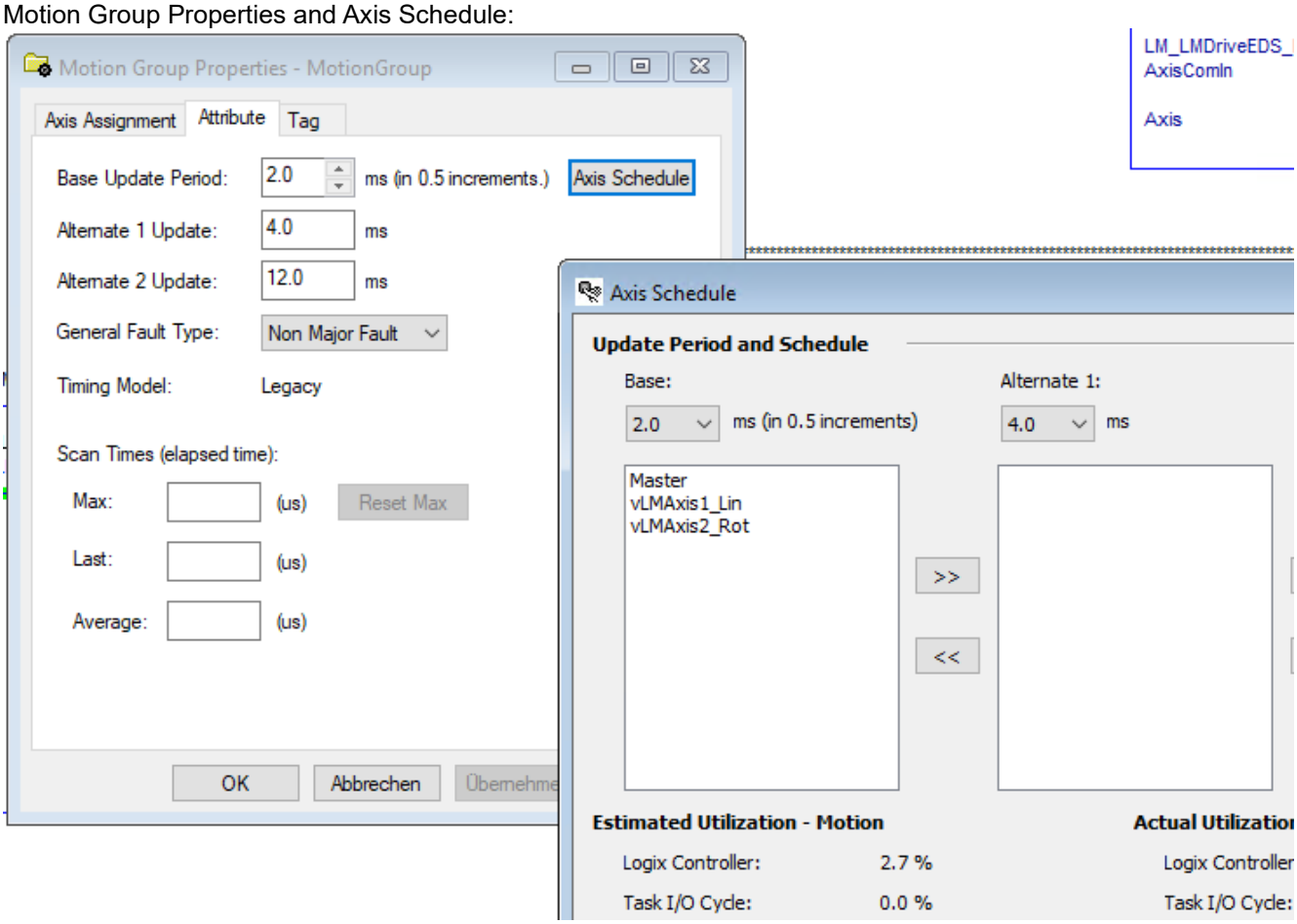
Attention:
The LM_ADV_CmdPosition instruction must be **always enabled!**
The coupling between real and virtual axis can be enabled and disabled using the *EnableStreaming* tag.
The IOT instruction is important to immediately write the drive data to the outputs!

4.3 Motion Group: MotionGroup

In this application note three axes are defined in the motion group:

1. Master: virtual encoder from 0 to 360 degrees
2. vLMAxis1_Lin: a virtual motion axis coupled with a LinMot Drive linear axis
3. vLMAxis1_Rot: a virtual motion axis coupled with a LinMot Drive rotary axis

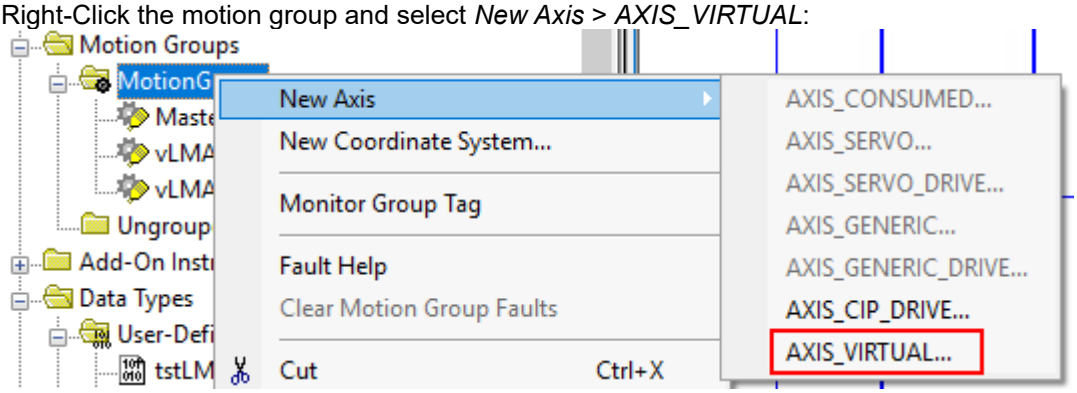
Motion Group Properties and Axis Schedule:



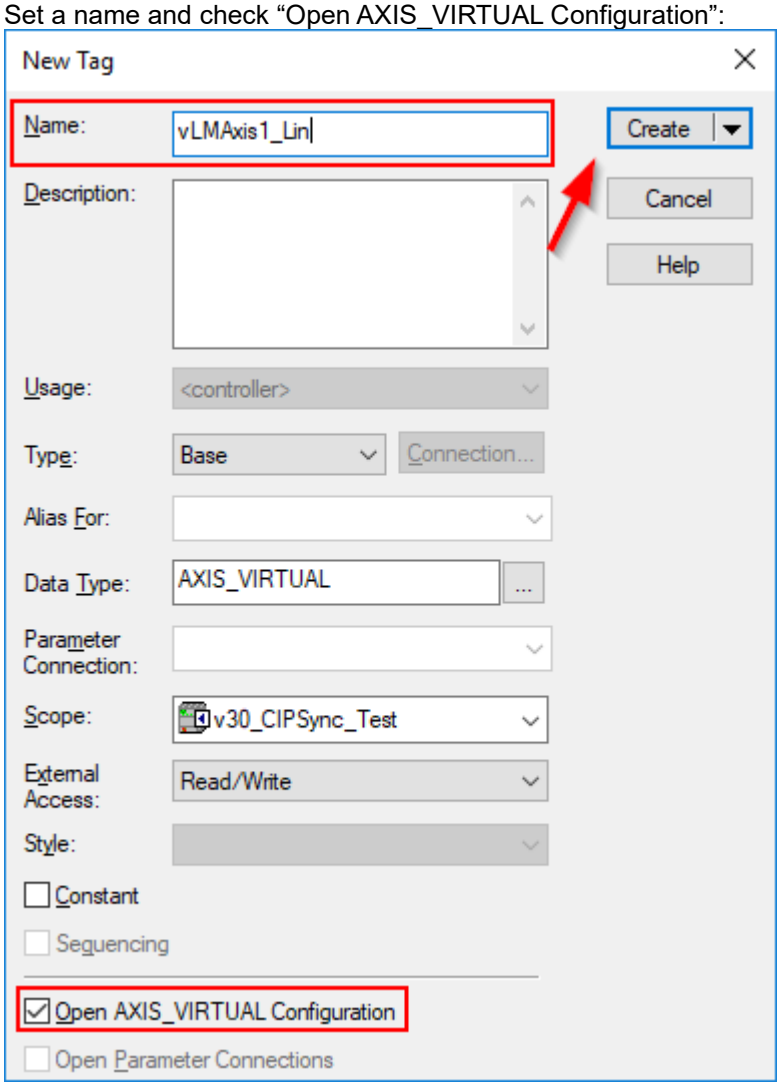
Attention:
It is recommended to have the LinMot axis/axes scheduled in the Base task (fastest).
The lower the update period the better the results regarding reactivity and positioning quality during motion.
For **highly dynamic** applications an update rate of **less than 5ms** is recommended.

4.3.1 Add virtual axis (linear)

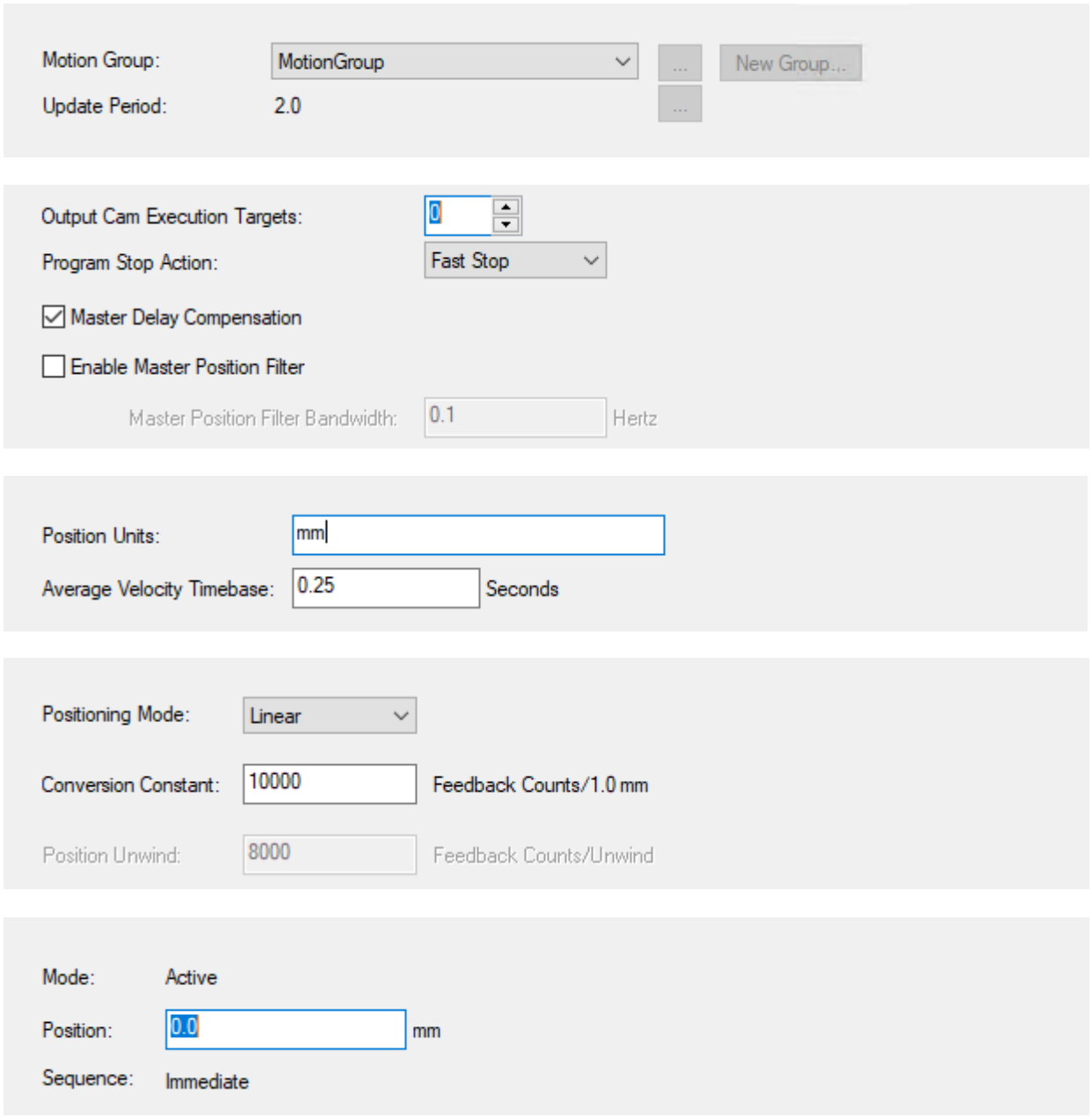
Right-Click the motion group and select *New Axis* > *AXIS_VIRTUAL*:



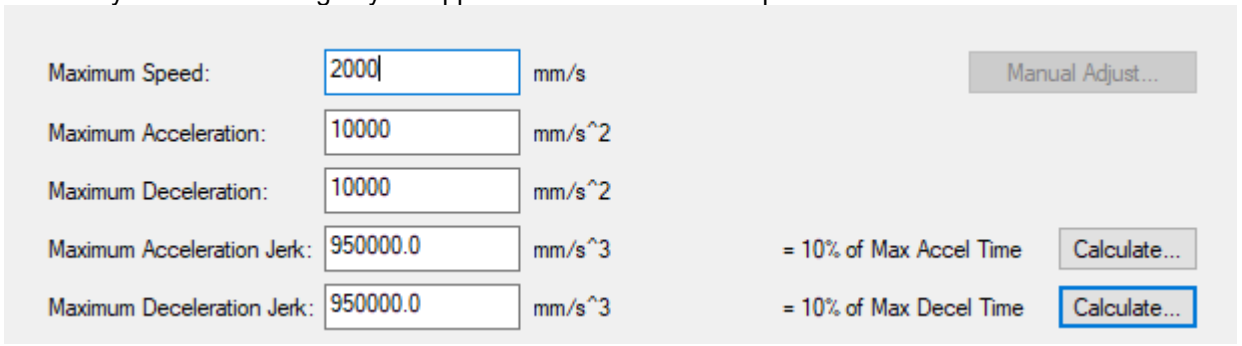
Set a name and check “Open AXIS_VIRTUAL Configuration”:



Set the following parameters in the wizard:



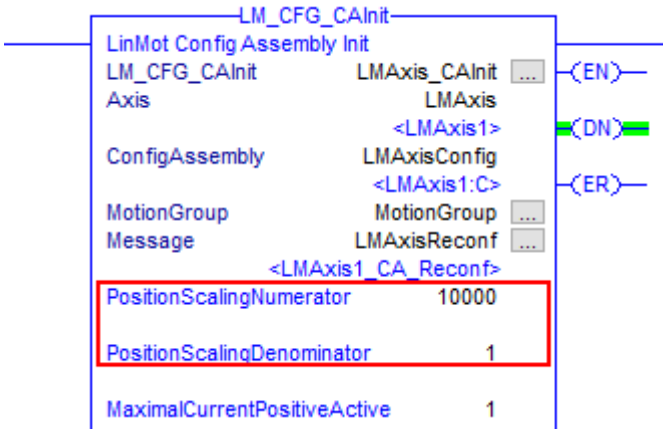
Set the dynamics according to your application and the motor capabilities:



For the final step, select the tag name and finish the wizard.

In the **Init** routine for this axis (see chapter 4.2.1) set the parameters of the LM_CFG_CAInit instruction to the following values:

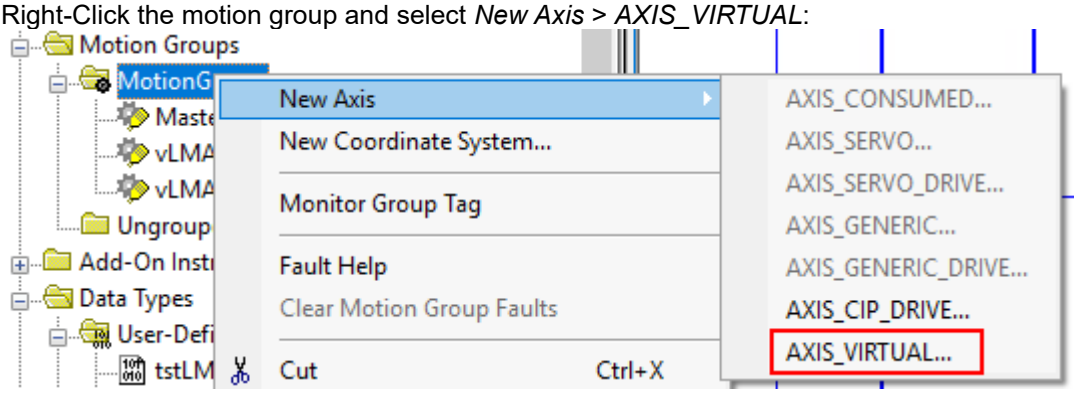
PositionScalingNumerator to 10000
PositionScalingDenominator to 1



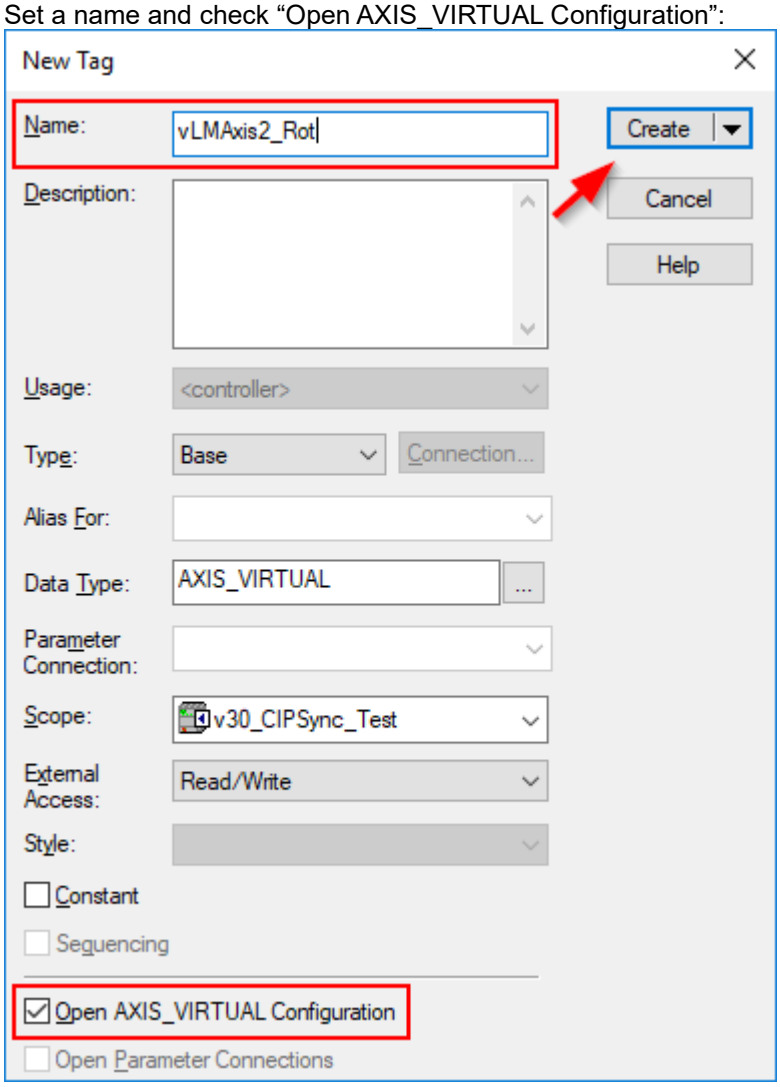
Note:
The LinMot linear motor has 0.1 um position resolution. Therefor for 1mm (1 motor rev) there are 10000 ticks.

4.3.2 Add virtual axis (rotary)

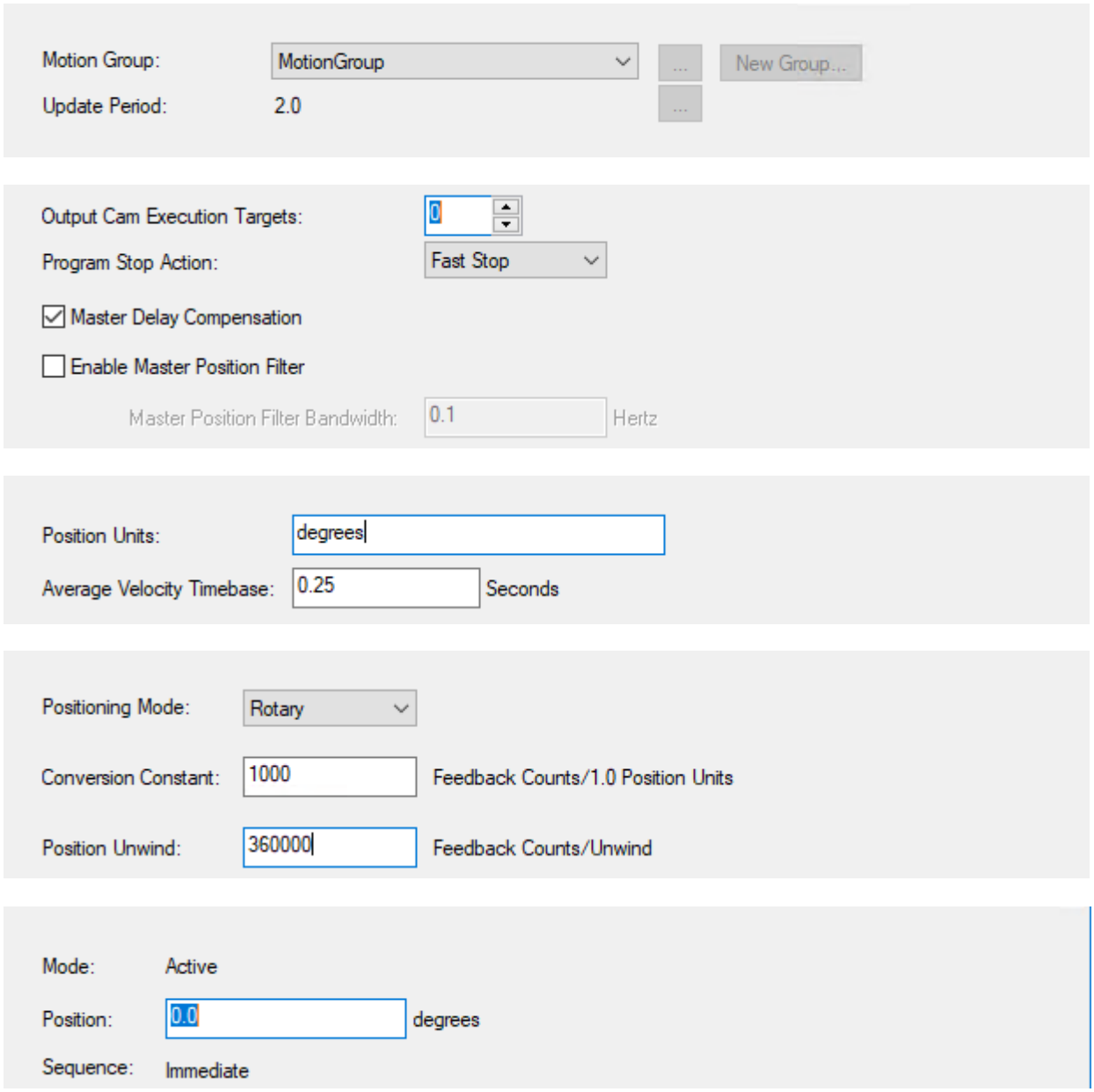
Right-Click the motion group and select *New Axis* > *AXIS_VIRTUAL*:



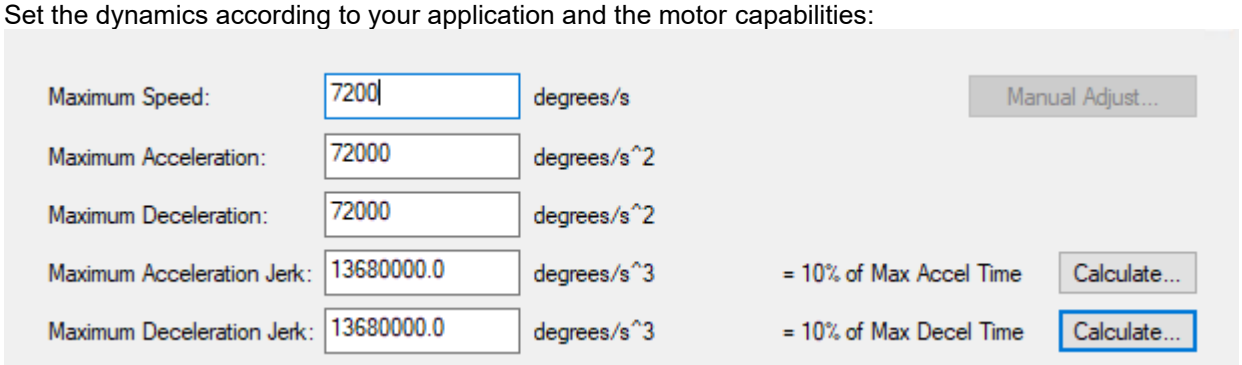
Set a name and check “Open AXIS_VIRTUAL Configuration”:



Set the following parameters in the wizard:



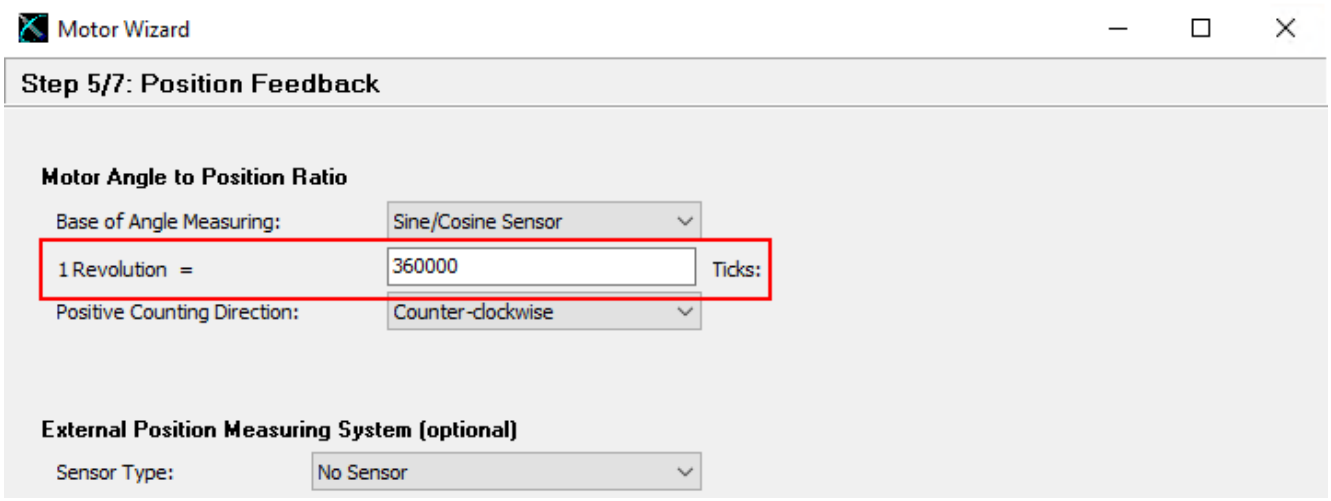
Set the dynamics according to your application and the motor capabilities:



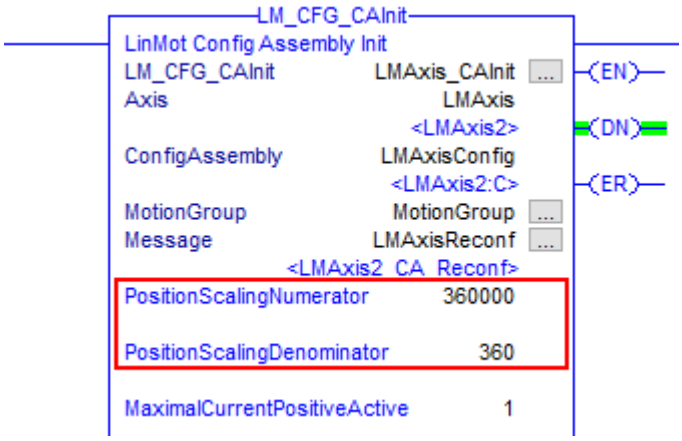
For the final step, select the tag name and finish the wizard.

In the **Init** routine for this axis (see chapter 4.2.1) set the *PositionScalingNumerator* and the *PositionScalingDenominator* according to the settings used in the LinMot-Talk motor wizard. It is important to note the number of ticks for one revolution of the motor.

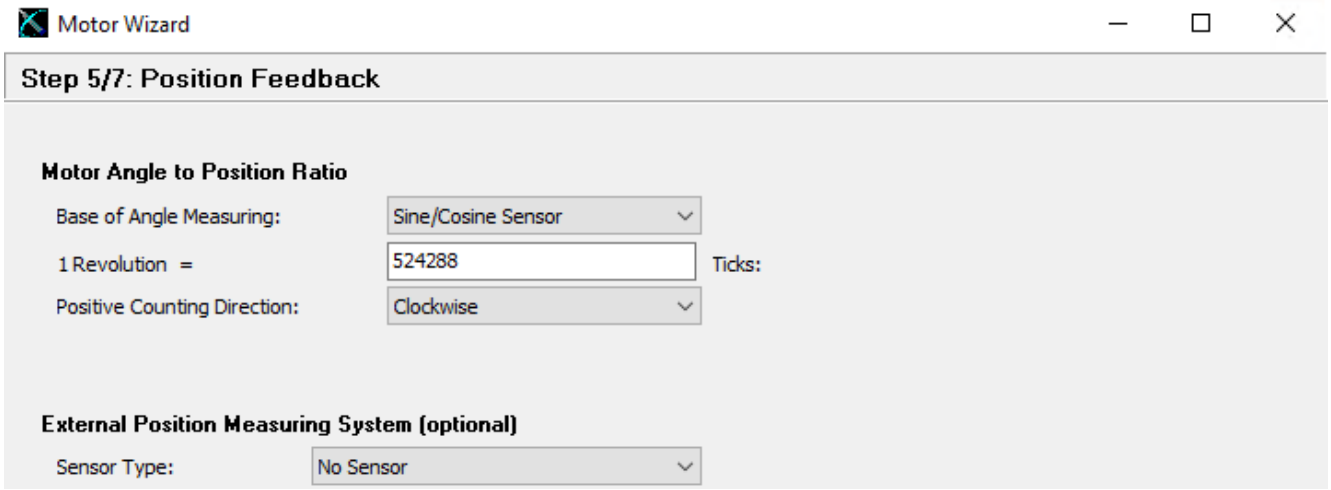
Example 1: For the rotary part of a LinMot PR01/PR02… with rotary units selected in the Motor Wizard 360'000 ticks per revolution are set:



Therefore, the *PositionScalingNumerator must be set to 360'000* and the *PositionScalingDenominator to 360*

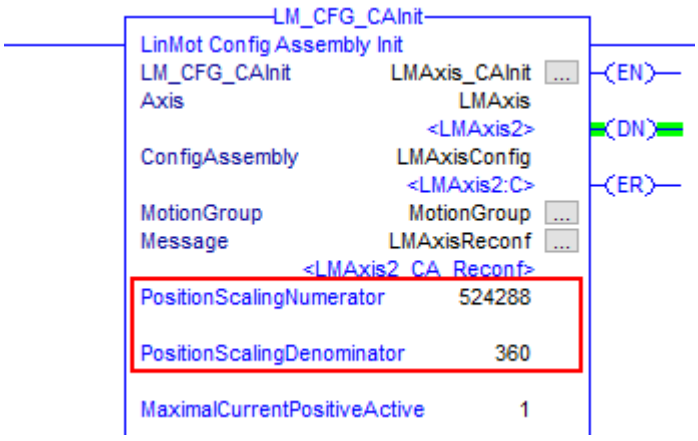


Example 2: For a LinMot EC02-40x… motor with rotary units selected in the Motor Wizard *524'288 ticks* per revolution are set:

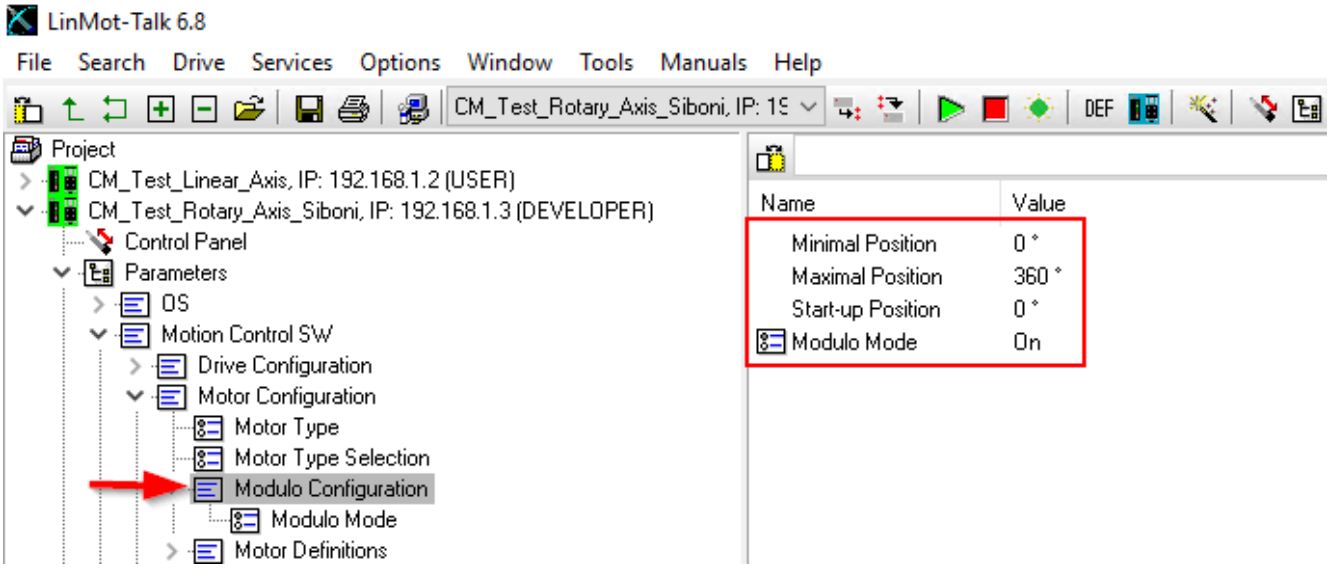


Or another motor set to the same amount of ticks:

Therefore, the *PositionScalingNumerator must be set to 524'288* and the *PositionScalingDenominator to 360*



For rotary axes the modulo functionality must be activated in the drive parameters.

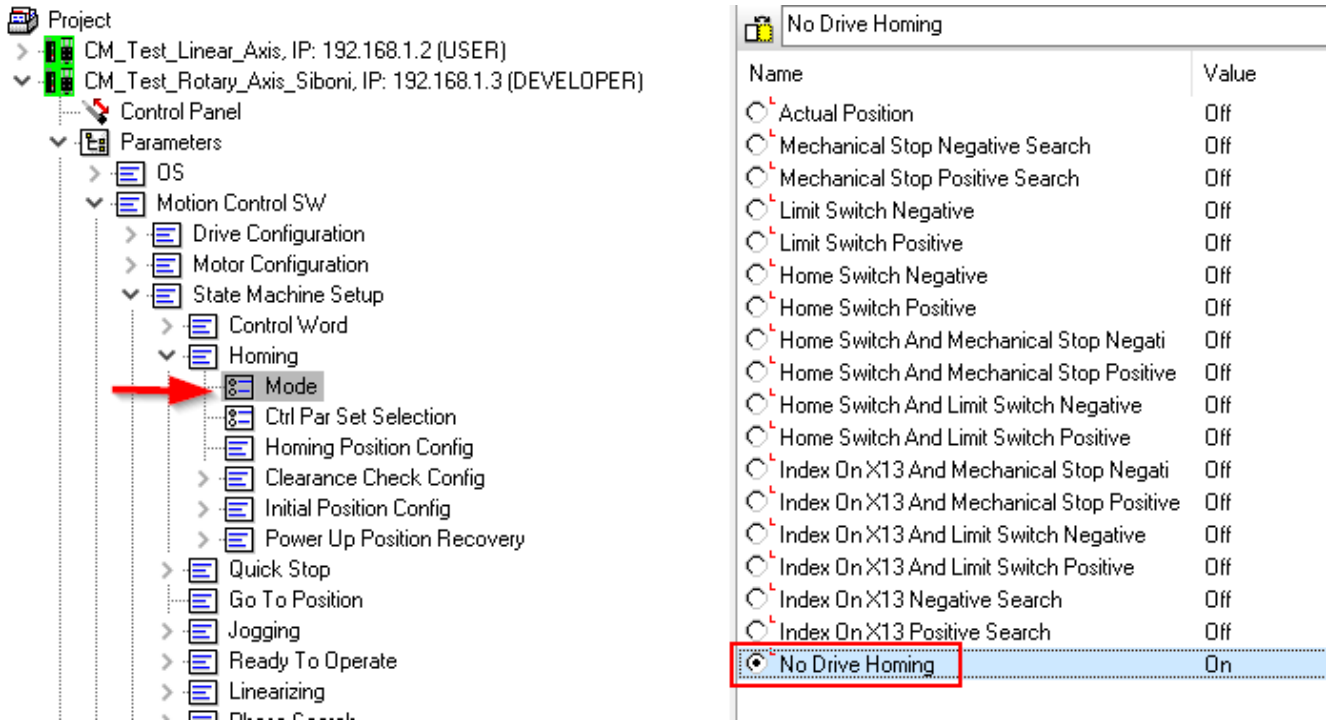


In the **Modulo Configuration** set *Modulo Mode to On*, *Minimal Position to 0°* and *Maximal Position to 360°*.

This allows the LinMot drive to correctly interpret the command values sent from the Rockwell Automation controller.

4.3.3 Absolute Position Sensor

When motors with absolute position sensors (SSI; BiSS, EnDat, etc.) are used the homing mode on the drive must be set to “No Drive Homing”.



Otherwise, the position may be shifted when executing the LM_MAHv to synchronize the virtual axis with the LinMot drive.

4.4 I/O Device: LMDrive

Add and setup the LinMot drive as shown in chapter 3.2.

5 Instructions

5.1 Overview

All instructions can be found as single L5X exports in the folder *AOI* of this package.

The following minimal instruction set is required for this application (coordinated motion using a LinMot CM drive coupled with a virtual motion axis):

- LM_LMDriveEDS_Read_78_64
- LM_LMDriveEDS_Write_78_64
- LM_MSO
- LM_MSF
- LM_MAFR
- LM_MAHv
- LM_ADV_PVATime
  - *LM_ParameterRead*
- LM_ADV_CmdPostion
  - *LM_LMDriveEDS_Read_78_64 & LM_LMDriveEDS_Write_78_64 & LM_ADV_PVATime*
- LM_CFG_CAInit
- LM_ParameterRead

Optional instructions for parameter access or advanced functionality:

- LM_ParameterWrite
- LM_MRPv
- LM_CFG_GetErrorText / LM_CFG_GetStringByUPID
- LM_CFG_GetModUPIDList / LM_CFG_WriteUPIDList
- LM_CFG_StopStartDefault
- LM_CFG_CTAccess
- LM_CFG_CurveAccess

Optional instructions for closed loop force/torque control (on the physical axis only, no coordinated moves):

- LM_FC_FCCTF
- LM_FC_FCST
- LM_FC_GTPFCHL & LM_FC_GTPFCLL
- LM_FC_GTPRFC

Optional instructions to use motion functionality without being coupled with a virtual motion axis > on the physical axis only:

- LM_MAM
- LM_MAS
- LM_MAH & LM_MAH_MG
- LM_MAIAP_I0
- LM_MAJ
- LM_MASC
- LM_MATC

Notes:
If an instruction is integrated in another one this is indicated by *italic* font. E.g., see LM_ADV_PVATime where *LM_ParameterRead* is integrated.

When working with parameter/configuration instructions it is sometimes required to know if any other instructions is already accessing parameters.
For this reason, the axis structure *tstLM_Axis* contains a flag ***CfgRWActive*** that indicates a running parameter access.

5.2 Instruction Signatures

To allow the user to check if the original signed instructions are used following version, ID, and timestamp.

Instruction name	Version	Signature ID	Timestamp
LM_ADV_CmdPosition	Version 2.0	0036C7F6	2019-08-27T14:01:15.584Z
LM_ADV_PVATime	Version 2.2	C9E5CEAE	2019-09-23T09:47:18.533Z
LM_LMDriveEDS_Read_78_64	Version 2.2	6C238E01	2019-09-18T11:45:28.527Z
LM_LMDriveEDS_Write_78_64	Version 2.0	4F5B1F59	2019-03-14T15:33:22.103Z
LM_MAFR	Version 2.0	CB16BA53	2019-03-14T15:34:46.343Z
LM_MAHv	Version 2.0	B45F6928	2019-03-14T15:36:01.268Z
LM_MAS	Version 2.2	DCCFA6FA	2020-11-05T07:22:01.000Z
LM_MSF	Version 2.0	E80449EA	2019-03-14T15:37:56.726Z
LM_MSO	Version 2.0	8545B4B3	2019-03-14T15:38:16.999Z
LM_ParameterRead	Version 2.0	EEC49A0A	2019-03-18T06:37:20.693Z
LM_ParameterWrite	Version 2.0	63E9E98A	2019-03-14T15:39:06.493Z
LM_DriveStatus	Version 2.1	C1E7C8AA	2019-08-27T14:08:28.164Z
LM_CFG_CAInit	Version 2.0	852AE15A	2019-03-14T15:29:52.232Z
LM_CFG_GetErrorText	Version 2.0	FDC70AD9	2019-03-18T06:28:56.627Z
LM_CFG_GetStringByUPID	Version 2.0	30F54DC9	2020-11-05T07:18:13.946Z
LM_CFG_GetModUPIDList	Version 2.0	09F030C7	2019-03-18T06:29:46.753Z
LM_CFG_StopStartDefault	Version 2.1	84717C75	2022-04-20T06:37:55.789Z
LM_CFG_WriteUPIDList	Version 2.0	F1BFDAEB	2019-03-18T06:31:28.117Z
LM_CFG_CTAccess	Version 2.0	0C466A8F	2019-09-18T12:30:35.562Z
LM_CFG_CurveAccess	Version 2.0	DD68406D	2019-09-18T12:29:24.729Z
LM_FC_FCCTF	Version 2.0	8418574B	2019-08-20T12:50:31.856Z
LM_FC_FCST	Version 2.0	31E73201	2022-07-06T05:15:28.872Z
LM_FC_GTPT	Version 2.0	B03E7239	2022-07-06T06:34:16.529Z
LM_FC_GTPFCHL	Version 2.0	4FA7BF47	2019-08-20T12:51:03.174Z
LM_FC_GTPFCLL	Version 2.0	6C4A43A7	2019-08-20T12:51:38.964Z
LM_FC_GTPRFC	Version 2.0	791B6BEE	2019-08-20T12:52:11.430Z
LM_ADV_CEE	Version 2.0	8A194FC6	2019-08-20T12:48:35.091Z
LM_ADV_CurrentCmdMode	Version 2.0	564D203B	2020-11-05T12:17:37.780Z
LM_MAM	Version 2.3	1BA7C3C4	2020-11-05T07:22:38.855Z
LM_MRPv	Version 2.0	77432C0C	2019-03-18T06:36:11.798Z
LM_MRP	Version 2.0	AA5DB58C	2019-08-28T06:44:10.766Z
LM_MAH	Version 2.0	3954BA0F	2019-08-20T12:58:01.654Z
LM_MAH_MG	Version 2.0	5B58F86A	2022-04-20T06:46:39.313Z
LM_MAIAP_I0	Version 2.0	01B89E43	2019-08-20T12:53:00.167Z
LM_MAJ	Version 2.0	52F3E012	2019-08-28T06:11:16.360Z
LM_MASC	Version 2.0	8EE9098B	2019-08-20T12:56:40.210Z
LM_MATC	Version 2.0	022BD204	2019-08-20T12:57:18.531Z
LM_LMDriveAOP_Read_IP	Version 2.0	22550FD2	2020-11-05T07:20:39.537Z
LM_LMDriveAOP_Write_IP	Version 2.0	8F8A3A5D	2020-11-05T07:21:29.348Z
LM_LMDriveEDS_Read_IP	Version 2.0	F77A8CF7	2022-04-20T06:45:15.610Z
LM_LMDriveEDS_Write_IP	Version 2.0	7812D7E1	2022-04-20T06:45:39.555Z
LM_LMDriveEDS_Write_IP_As27	Version 2.0	EDF7F44D	2022-04-20T06:46:07.232Z

5.3 Data Types

5.3.1 tstLM_Axis

The following table lists the members of the LinMot axis reference data type (UDT)

Name	D Type	Radix	Ext Access	Max	Min	Description	Eng. Unit
ControlWord	INT	Decimal	Read/Write			Internal: Drive Control Word	
MCHeader	INT	Decimal	Read/Write			Internal: Motion Command Header	
MCParaWord0	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 0	
MCParaWord1	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 1	
MCParaWord2	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 2	
MCParaWord3	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 3	
MCParaWord4	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 4	
MCParaWord5	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 5	
MCParaWord6	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 6	
MCParaWord7	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 7	
MCParaWord8	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 8	
MCParaWord9	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 9	
MCParaWord10	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 10	
MCParaWord11	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 11	
MCParaWord12	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 12	
MCParaWord13	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 13	
MCParaWord14	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 14	
MCParaWord15	INT	Decimal	Read/Write			Internal: Motion Command Parameter Word 15	
StatusWord	INT	Decimal	Read/Write			Internal: Drive Status Word	
StateVar	INT	Decimal	Read/Write			Internal: Drive State Var	
WarnWord	INT	Decimal	Read/Write			Internal: Drive Warn Word	
CfgControl	INT	Hex	Read/Write			Internal: Drive Config Module Control Word	
CfgIndexOut	INT	Hex	Read/Write			Internal: Drive Config Module Index Out	
CfgValueOut	DINT	Hex	Read/Write			Internal: Drive Config Module Value Out	
CfgStatus	INT	Hex	Read/Write			Internal: Drive Config Module Status Word	
CfgIndexIn	INT	Hex	Read/Write			Internal: Drive Config Module Index In	
CfgValueIn	DINT	Hex	Read/Write			Internal: Drive Config Module Value In	
DriveConnectionFaulted	BOOL	Decimal	Read/Write			Connection to drive faulted	
AxisFault	DINT	Hex	Read/Write			Bits used: 0 = Physical Axis Fault	
PhysicalAxisFault	BOOL	Decimal	Read/Write			The Physical Axis Fault bit is set. Check ErrorID	
AxisStatus	DINT	Hex	Read/Write			Bits used: 0 = Servo Action Status, 1 = Drive Enable Status	
ServoActionStatus	BOOL	Decimal	Read/Write			The ServoActionStatus bit attribute is set when the associated axis is under servo control. If the bit is not set, then servo action is disabled.	
DriveEnableStatus	BOOL	Decimal	Read/Write			The DriveEnableStatus bit attribute is set when the Drive Enable output of the associated physical axis is currently enabled. If the bit is not set, then physical servo axis Drive Enable output is currently disabled.	
MotionStatus	DINT	Hex	Read/Write			Motion Status	
JogStatus	BOOL	Decimal	Read/Write			Jog Status	
MoveStatus	BOOL	Decimal	Read/Write			Move Status	
HomingStatus	BOOL	Decimal	Read/Write			Homing Status	

StoppingStatus	BOOL	Decimal	Read/Write			Stopping Status	
AxisHomedStatus	BOOL	Decimal	Read/Write			Axis Homed Status	
PVAStatus	BOOL	Decimal	Read/Write			PVA Streaming Status	
ActualPosition	REAL	Float	Read/Write			Actual Position in Position Units	
CommandPosition	REAL	Float	Read/Write			Command Position in Position Units	
PositionError	REAL	Float	Read/Write			The error between commanded and actual position that is the output of the position loop-summing junction.	
ActualVelocity	REAL	Float	Read/Write			Actual Velocity in Position Units / Sec	
CurrentFeedback	REAL	Float	Read/Write			Actual current applied to the axis in Ampere	
ErrorID	INT	Hex	Read/Write			Drive Error ID	
TorqueLimitPositive	REAL	Float	Read/Write	100	0	Motor positive Torque/Force Limit [0...100% of max.]. If 0 the motor cannot move	%
TorqueLimitNegative	REAL	Float	Read/Write	0	-100	Motor negative Torque/Force Limit [0...-100% of max.]. If 0 the motor cannot move	%
PositionScalingNumerator	REAL	Float	Read/Write			Increments per motor rev	
PositionScalingDenominator	REAL	Float	Read/Write			Units per motor rev	
TimeStampDrvToCtrl	DINT[2]	Decimal	Read/Write			Internal: Time Stamp Drive to Controller	
TimeStampCtrlToDrv	DINT[2]	Decimal	Read/Write			Internal: Time Stamp Controller to Drive	
TimeOffsetDrvToCtrl	DINT[2]	Decimal	Read/Write			Internal: Time Offset Drive to Controller	
TimeOffsetCtrlToDrv	DINT[2]	Decimal	Read/Write			Internal: Time Offset Controller to Drive	
MonChannelConfigID	DINT	Decimal	Read/Write			Internal: Actual active monitoring configuration ID	
ParChannelConfigID	DINT	Decimal	Read/Write			Internal: Actual active parameter configuration ID	
CfgRWActive	BOOL	Decimal	Read/Write			Internal: A configuration instruction is active	

5.4 Read and Write Instructions

These instructions read and write the LinMot drive data transmitted over Ethernet/IP.

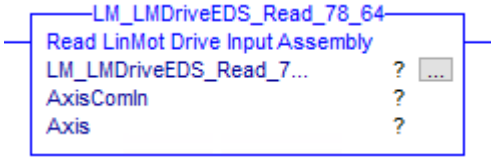
Attention: If the *LM_ADV_CmdPosition* instruction (chapter 5.6.2) is used (as shown in the example demo project), **DO NOT** use the following AOIs in the program logic:
LM_LMDriveEDS_Read_78_64
LM_LMDriveEDS_Write_78_64

The *LM_LMDriveEDS_Read_78_64* and *LMDriveEDS_Write_78_64* AOIs are included inside the logic of the *LM_ADV_CmdPosition* AOI, so those instructions will appear in the AOI list in the controller organizer, however they must not be utilized outside of *the LM_ADV_CmdPosition* AOI.

5.4.1 LM_LMDriveEDS_Read_78_64

Reads the communication data (input assembly) of a LinMot drive.

Supported LinMot drives:
- C1250-CM-xx

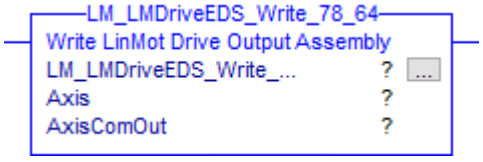


Operand	Type	Format	Description
LM_LMDriveEDS_Read_78_64	LM_LMDriveEDS_Read_78_64	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
AxisComIn	_024D:LMDrive_A3FF6A28:I:0	Tag	LinMot axis input assembly
Axis	tstLM_Axis	Tag	LinMot axis reference

5.4.2 LM_LMDriveEDS_Write_78_64

Writes the communication data (output assembly) of a LinMot drive.

Supported LinMot drives:
- C1250-CM-xx

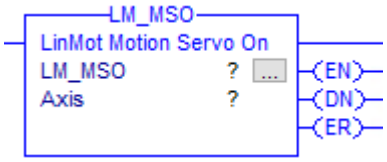


Operand	Type	Format	Description
LM_LMDriveEDS_Write_78_64	LM_LMDriveEDS_Write_78_64	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
AxisComOut	_024D:LMDrive_AD451A70:O:0	Tag	LinMot axis output assembly
Axis	tstLM_Axis	Tag	LinMot axis reference

5.5 Motion State Instructions

5.5.1 Motion Servo On (LM_MSO)

The LM_MSO instruction is the equivalent of the Rockwell Automation MSO instruction. Use this instruction to enable the drive amplifier and the axis' servo control loop.



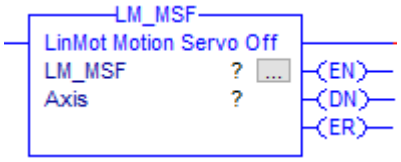
Operand	Type	Format	Description
LM_MSO	LM_MSO	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	LinMot axis reference.

LM_MSO Structure Descriptions

Enumerations	Description
EN (Enable)	It is set when the rung makes a false-to-true transition and remains set until the servo message transaction is completed and the rung goes false.
DN (Done)	It is set when the axis' servo action has been successfully enabled and the drive enable, and servo active status bits have been set.
ER (Error)	It is set to indicate that the instruction detected an error.

5.5.2 Motion Servo Off (LM_MSF)

The LM_MSF instruction is the equivalent for the Rockwell Automation MSF instruction. Use this instruction to disable the drive output for the specified axis and to deactivate the axis' servo loop.



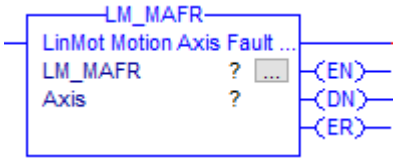
Operand	Type	Format	Description
LM_MSF	LM_MSF	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	LinMot axis reference.

LM_MSF Structure Descriptions

Enumerations	Description
EN (Enable)	It is set when the rung makes a false-to-true transition and remains set until the servo message transaction is completed and the rung goes false.
DN (Done)	It is set when the axis' servo action been successfully disabled, and the drive enable, and servo active status bits have both been cleared.
ER (Error)	It is set to indicate that the instruction detected an error.

5.5.3 Motion Axis Fault Reset (LM_MAFR)

The LM_MAFR instruction is the equivalent for the Rockwell Automation MAFR instruction. Use this instruction to clear all motion faults for an axis. This is the only method for clearing axis motion faults.



Operand	Type	Format	Description
LM_MAFR	LM_MAFR	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	LinMot axis reference.

LM_MAFR Structure Descriptions

Enumerations	Description
EN (Enable)	It is set when the rung makes a false-to-true transition and remains set until the servo message transaction is completed and the rung goes false.
DN (Done)	It is set when the axis' faults have been successfully cleared.
ER (Error)	It is set to indicate that the instruction detected an error.

5.6 Motion Move Instructions

5.6.1 Motion Axis Home (LM_MAHv)

The LM_MAHv instruction is the equivalent for the Rockwell Automation MAH instruction. Use this instruction to home an axis.
The homing sequence can be selected during axis configuration (LinMot-Talk Motor Wizard). It's an active homing. The connected virtual axis is homed as well to the actual command position of the LinMot motor.

Operand	Type	Format	Description
LM_MAHv	LM_MAHv	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	LinMot axis reference.
VirtualAxis	AXIS_VIRTUAL	Tag	Virtual axis reference

LM_MAHv Structure Descriptions

Enumerations	Description
EN (Enable)	It is set when the rung makes a false-to-true transition and remains set until the servo message transaction is completed and the rung goes false.
DN (Done)	It is set when axis home has been successfully completed or is aborted.
ER (Error)	It is set to indicate that the instruction detected an error.
IP (In Process)	It is set on positive rung transition and cleared after the Motion Home Axis is complete, or terminated by a stop command, shutdown, or a servo fault.
PC (Process Compete)	It is set when axis home is successfully completed (real and virtual axis).

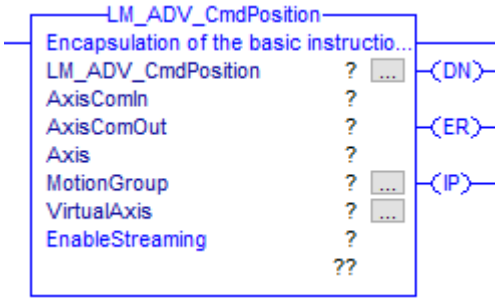
Note:
The homing sequence is configured using LinMot-Talk. The simplest method is to configure homing through the LinMot-Talk motor wizard.

5.6.2 CIP Sync Command Position (LM_ADV_CmdPosition)

This instruction is the main part of this solution as it couples the virtual motion axis and the physical axis.
It encapsulates the read and write instructions of chapter 5.4 and the PVA Streaming instruction of chapter 5.6.3 to simplify the code. The encapsulated instructions will be visible in the controller organizer.

This instruction is used in the demo project > check chapter 4.2.1

Attention:
When this instruction is used do not use the following instructions anywhere else in the program!
LM_LMDriveEDS_Read_78_64, LM_LMDriveEDS_Write_78_64 & LM_ADV_PVATime



Operand	Type	Format	Description
LM_ADV_CmdPosition	LM_ADV_CmdPosition	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
AxisComIn	_024D:LMDrive_A3FF6A28:I:0	Tag	LinMot axis input assembly
AxisComOut	_024D:LMDrive_AD451A70:O:0	Tag	LinMot axis output assembly
Axis	tstLM_Axis	Tag	LinMot axis reference.
MotionGroup	MOTION_GROUP	Tag	Motion Group reference.
VirtualAxis	VIRTUAL_AXIS	Tag	Virtual axis reference.
EnableStreaming	Bool	Tag	Couples virtual and physical axis

LM_ADV_CmdPosition Structure Descriptions

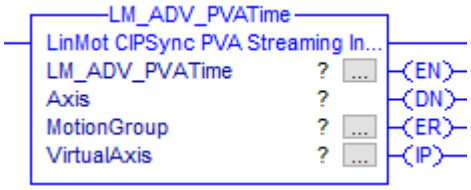
Enumerations	Description
DN (Done)	The streaming was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	The LinMot axis is coupled with the virtual axis and command values are transmitted.

Attention:
The LM_ADV_CmdPosition instruction must be **always enabled!**
The coupling between physical and virtual axis can be enabled and disabled using the *EnableStreaming* tag.
The IOT instruction is important to immediately write the drive data to the outputs!

Attention:
- Rockwell Automation controller revision must be **V30 or later**
- LinMot drive firmware must be **6.8 Build 20190315 or later**

5.6.3 CIP Sync PVA Streaming (LM_ADV_PVATime)

This instruction couples the LinMot drive with a virtual axis. As soon as IP is set the command values (position, velocity & acceleration) of the virtual axis are transmitted to the LinMot drive. This includes a time stamp to guarantee a synchronized move together with CIP Motion axes.

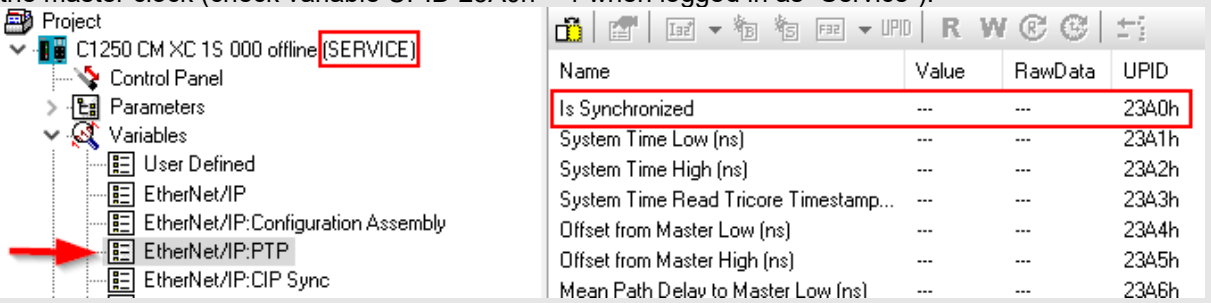


Operand	Type	Format	Description
LM_ADV_PVATime	LM_ADV_PVATime	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	LinMot axis reference.
MotionGroup	MOTION_GROUP	Tag	Motion Group reference.
VirtualAxis	VIRTUAL_AXIS	Tag	Virtual axis reference.

LM_ADV_PVATime Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The streaming was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	The LinMot axis is coupled with the virtual axis and command values are transmitted.

Note:
If **DN =True, ER = False and IP = False** then the drive is waiting to have its clock synchronized with the master clock (check variable UPID 23A0h = 1 when logged in as “Service”).



Check that the Controller / Ethernet card the LinMot is connected to has “Time Synchronization” enabled.

Attention:
If the LM_ADV_CmdPosition instruction (chapter 5.6.2) is used (as it is in the demo project) do not use this instruction. It is already part of LM_ADV_CmdPosition.

Attention:
- Rockwell Automation controller revision must be **V30 or later**
- LinMot drive firmware must be **6.8 Build 20190315 or later**

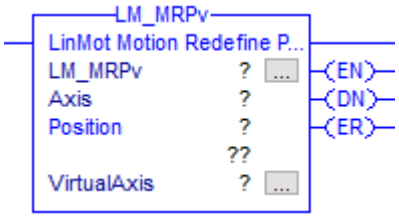
5.6.4 Motion Axis Redefine Position (LM_MRPv)

Motion Axis Redefine Position (LinMot)

Use this instruction to change the command and actual position of both the virtual and real (LinMot) axis.

Attention:
This instruction is thought for special use cases only!
The type is always absolute and performs on both command and actual position.

The instruction must only be used when the axis is not in coupled mode (tstLM_Axis.PVAStatus = FALSE)!



Operand	Type	Format	Description
LM_MRPv	LM_MRPv	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
Position	REAL	Immediate Tag	The value to use to change the axis position to
VirtualAxis	VIRTUAL_AXIS	Tag	Virtual axis reference.

LM_MRPv Structure Descriptions

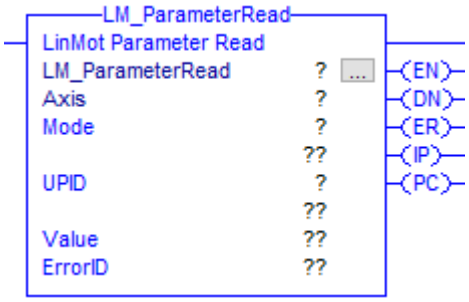
Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit remains set until the process is complete and the rung goes false.
DN (Done)	Is set when the axis position has been successfully redefined.
ER (Error)	An error occurred.

Note:
There is also a version for the physical axis only available > LM_MRP

5.7 LinMot Parameter Instructions

5.7.1 Read Parameter (LM_ParameterRead)

Use this instruction to read a parameter (RAM or ROM) of the LinMot drive.



Operand	Type	Format	Description
LM_ParameterRead	LM_ParameterRead	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	LinMot Axis reference.
Mode	INT	Immediate Tag	Set access mode: 0 = ROM 1 = RAM
UPID	INT	Immediate Tag	Unique Parameter ID. Parameter to be read.
Value	DINT	Immediate Tag	Parameter value read. Valid if PC bit set.
ErrorID	INT	Immediate Tag	Error ID if error is detected.

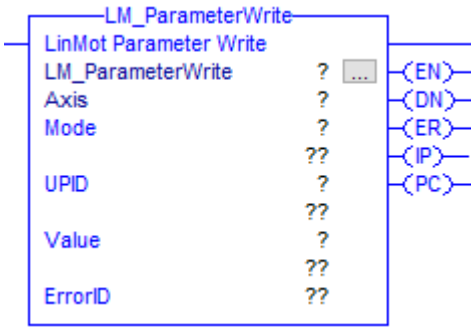
LM_ParameterRead Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred.
IP (In Process)	Drive is reading the parameter.
PC (Process Complete)	Parameter read complete.

Note:
Be aware of the parameter scale

5.7.2 Write Parameter (LM_ParameterWrite)

Use this instruction to write a parameter (RAM, ROM or RAM&ROM) of the LinMot drive.



Operand	Type	Format	Description
LM_ParameterWrite	LM_ParameterWrite	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	LinMot axis reference
Mode	INT	Immediate Tag	Set access mode: 0 = ROM 1 = RAM 2 = RAM & ROM
UPID	INT	Immediate Tag	Unique Parameter ID. Parameter to be written.
Value	DINT	Immediate Tag	Parameter value to be written.
ErrorID	INT	Immediate Tag	Error ID if error is detected.

LM_ParameterWrite Structure Descriptions

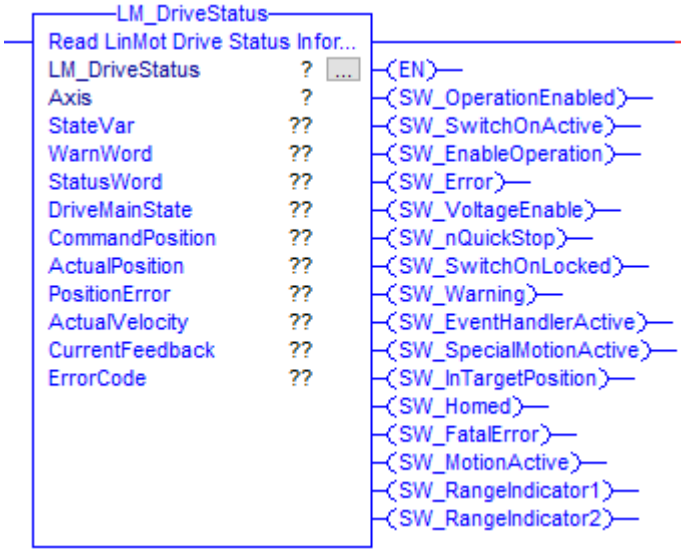
Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred.
IP (In Process)	Drive is writing the parameter.
PC (Process Complete)	Parameter writing complete.

Note:
Be aware of the parameter scale

Attention:
Intense use of writing into the ROM memory can reduce the lifetime of the drive memory!
More details can be found in the *Drive Configuration Over Fieldbus (0185-1074)* user manual
> see chapter Recommended Documentation

5.7.3 LinMot Drive Status (LM_DriveStatus)

Use this instruction to get LinMot specific drive status information.



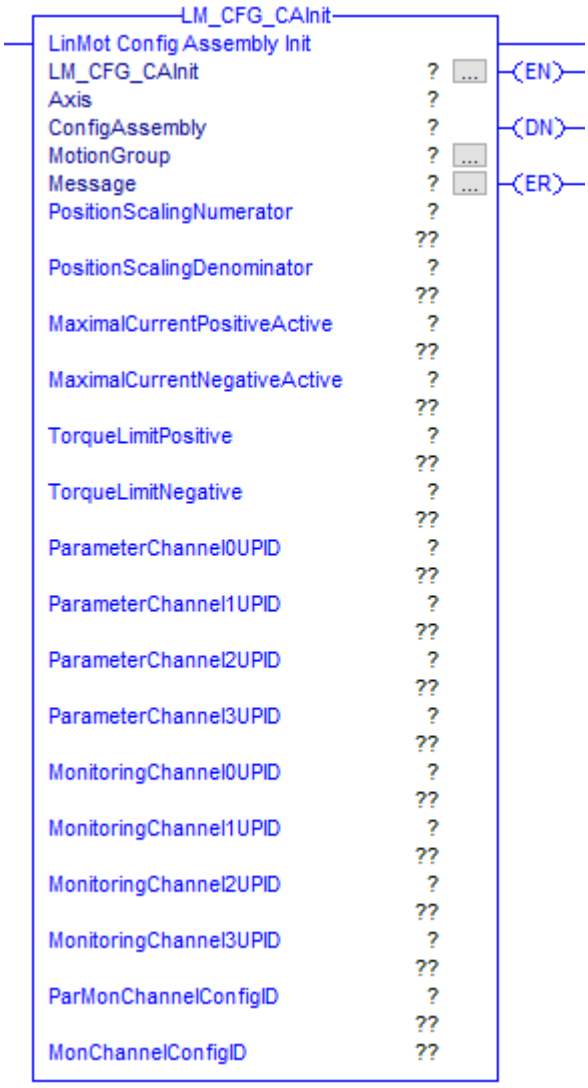
Operand	Type	Format	Description
LM_DriveStatus	LM_DriveStatus	Tag	Instance of the Addon Instruction. Access instruction status parameters
Axis	tstLM_Axis	Tag	Axis to perform the action on

Note:
For a detailed description of the status information refer to the Motion Control Software manual -> see chapter Recommended Documentation

5.8 LinMot Config Instructions

5.8.1 Config Assembly Init (LM_CFG_CAInit)

Use this instruction to setup and activate the configuration assembly of the LinMot drive.



Note:
Setup the connected MESSAGE as shown in chapter 4.1.2

Attention:
It is very important that the configuration assembly is setup correctly!

Operand	Type	Format	Description
LM_CFG_CAInit	LM_CFG_CAInit	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	LinMot Axis reference
ConfigAssembly	_024D:LMDrive_A8562DE1:C:0	Immediate Tag	LinMot Config Assembly
MotionGroup	MOTION_GROUP	Immediate Tag	Motion Group
Message	MESSAGE	Immediate Tag	Message
PositionScalingNumerator	DINT	Immediate Tag	Increments per Motor Revolution > **see chapters 4.3.1 & 4.3.2**
PositionScalingDenominator	DINT	Immediate Tag	Units per Motor Revolution > **see chapters 4.3.1 & 4.3.2**
MaximalCurrentPositiveActive	BOOL	Immediate Tag	Activate Force/Torque Limit Positive
MaximalCurrentNegativeActive	BOOL	Immediate Tag	Activate Force/Torque Limit Negative
TorqueLimitPositive	REAL	Immediate Tag	Force/Torque Limit Positive [0...100%]
TorqueLimitNegative	REAL	Immediate Tag	Force/Torque Limit Negative [0...-100%]
ParameterChannel0UPID	INT	Immediate Tag	UPID to be written in Parameter Channel 0
ParameterChannel1UPID	INT	Immediate Tag	UPID to be written in Parameter Channel 1
ParameterChannel2UPID	INT	Immediate Tag	UPID to be written in Parameter Channel 2
ParameterChannel3UPID	INT	Immediate Tag	UPID to be written in Parameter Channel 3
MonitoringChannel0UPID	INT	Immediate Tag	UPID to be read in Monitoring Channel 0
MonitoringChannel1UPID	INT	Immediate Tag	UPID to be read in Monitoring Channel 1
MonitoringChannel2UPID	INT	Immediate Tag	UPID to be read in Monitoring Channel 2
MonitoringChannel3UPID	INT	Immediate Tag	UPID to be read in Monitoring Channel 3
ParMonChannelConfigID	DINT	Immediate Tag	Configuration ID to use
MonChannelConfigID	DINT	Immediate Tag	Actual Active Configuration ID (Monitoring)

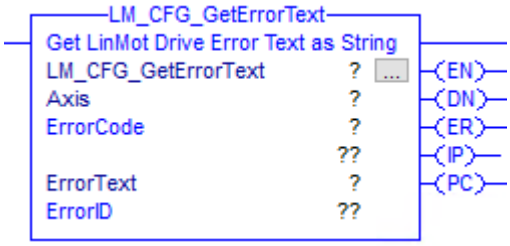
LM_CFG_CAInit Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred. Check ERR to get the error ID of the internal MESSAGE instruction.

5.8.2 Get Error Text (LM_CFG_GetErrorText)

This instruction returns a STRING containing the LinMot drive Error Text.
Input the ErrorCode to receive the corresponding ErrorText.

The actual error code of the drive can be found either in the axis data structure (tstLM_axis.ErrorID) or as ErrorCode output of the LM_DriveStatus instruction.



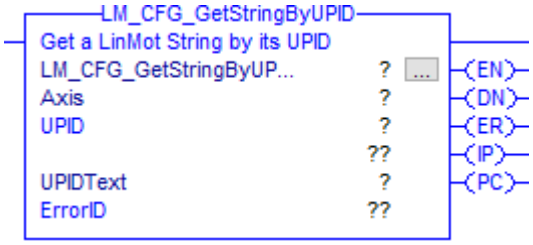
Operand	Type	Format	Description
LM_CFG_GetErrorText	LM_CFG_GetErrorText	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
ErrorCode	SINT	Immediate Tag	ErrorCode of the LinMot drive from which the error text should be read
ErrorText	STRING	Tag	Error text returned.
ErrorID	INT	Immediate Tag	Error ID if error is detected.

LM_CFG_GetErrorText Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred.
IP (In Process)	Drive is processing the command.
PC (Process Complete)	Process complete.

5.8.3 Get String by UPID (LM_CFG_GetStringByUPID)

This instruction returns a STRING containing the text of a string parameter/variable of the LinMot drive.
Input the UPID to receive the corresponding text.



Operand	Type	Format	Description
LM_CFG_GetStringByUPID	LM_CFG_GetStringByUPID	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
UPID	SINT	Immediate Tag	UPID of the LinMot drive from which the text should be read
UPIDText	STRING	Tag	Error text returned.
ErrorID	INT	Immediate Tag	Error ID if error is detected.

LM_CFG_GetStringByUPID Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred.
IP (In Process)	Drive is processing the command.
PC (Process Complete)	Process complete.

5.8.4 Get Modified UPID List (LM_CFG_GetModUPIDList)

This instruction reads a list of drive parameters and their values that have been modified (compared to factory defaults) and stores them in the array. Can be used to save the configuration of a drive in the controller.

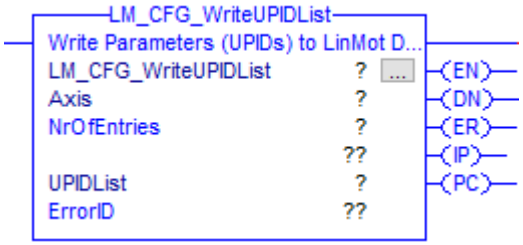
Operand	Type	Format	Description
LM_CFG_GetModUPIDList	LM_CFG_GetModUPIDList	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
NrOfEntries	INT	Immediate Tag	Number of parameters to be read at maximum. Recommended: 200
UPIDList	tstLM_CfgUPIDListEntry[200]	Tag	Array where the parameters are stored.
ErrorID	INT	Immediate Tag	Error ID if error is detected.

LM_CFG_GetModUPIDList Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred.
IP (In Process)	Drive is processing the command.
PC (Process Complete)	Process complete.

5.8.5 Write UPID List (LM_CFG_WriteUPIDList)

This instruction writes a list of drive parameters (UPID & Value) stored in an array to the drive.



Operand	Type	Format	Description
LM_CFG_WriteUPIDList	LM_CFG_WriteUPIDList	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
NrOfEntries	INT	Immediate Tag	Number of parameters to be written at maximum. Recommended: 200
UPIDList	tstLM_CfgUPIDListEntry[200]	Tag	Array where the parameters are stored.
ErrorID	INT	Immediate Tag	Error ID if error is detected.

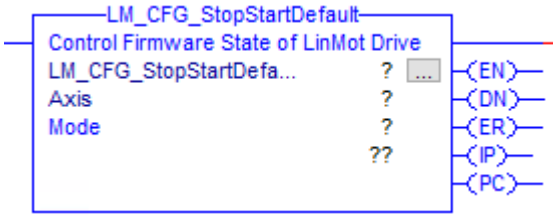
LM_CFG_WriteUPIDList Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred.
IP (In Process)	Drive is processing the command.
PC (Process Complete)	Process complete.

Attention:
The instruction stops writing as soon as it finds an entry with UPID = 0 in the array. Therefore, the last parameter entry in the array must be followed by an entry with UPID = 0.
Or as soon as the number at “NrOfEntries” has been reached.

5.8.6 Stop Start Default (LM_CFG_StopStartDefault)

This instruction provides the functionality to restart the drive, stop and start single firmware layers, or set the parameters of each firmware layer to factory defaults.



Operand	Type	Format	Description
LM_CFG_StopStartDefault	LM_CFG_StopStartDefault	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
Mode	INT	Immediate Tag	Set access mode: 0 = Restart Drive 1 = Set parameter ROM values to default (OS SW) 2 = Set parameter ROM values to default (MC SW) 3 = Set parameter ROM values to default (Interface SW) 4 = Set parameter ROM values to default (Application SW) 5 = Stop MC and Application Software (for Flash Access) 6 = Start MC and Application Software
ErrorID	INT	Immediate Tag	Error ID if error is detected.

LM_CFG_StopStartDefault Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred.
IP (In Process)	Drive is processing the requested mode.
PC (Process Complete)	Process complete.

Attention:
When mode 0 (Restart Drive) is executed then the EtherNet/IP connection of the drive is interrupted (same as power cycling the drive).

5.8.7 Command Table Access (LM_CFG_CTAccess)

This instruction provides access to the command table of a LinMot drive. Read and write entries, delete entries, delete the complete command table, save the command table from RAM to flash memory.

Operand	Type	Format	Description
LM_CFG_CTAccess	LM_CFG_CTAccess	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
Mode	INT	Immediate Tag	Set access mode: 0 = Save to Flash: Saves the Command Table to the flash memory (MC_SW must be stopped!) 1 = Delete all Entries (RAM) 2 = Delete Entry (RAM) 3 = Write Entry (RAM) 4 = Get Entry (is stored in CTEntry) 5 = Read presence list from drive
CTID	INT	Immediate Tag	Command Table ID (line number). Range: 1..255
CTEntry	tstLM_CfgCTEntry	Tag	Command Table Entry. Data type that contains all data for one command table line.
PresenceList	DINT[8]	Tag	Bit addressed array of DINT that shows if a command table ID is defined. 0 = Line defined 1 = Line empty
ErrorID	INT	Immediate Tag	Error ID if error is detected.

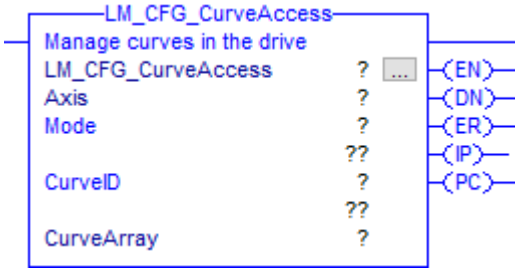
LM_CFG_CTAccess Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred.
IP (In Process)	Drive is processing the requested mode.
PC (Process Complete)	Process complete.

Attention:
Before executing mode 0 the MC_SW of the drive must be stopped. Use LM_CFG_StartStopDefault Mode 5 to do so.

5.8.8 Curve Access (LM_CFG_CurveAccess)

This instruction provides access to motion profiles (curves) on the LinMot drive. It is possible to read, write, modify and delete curves as well as saving all curves from the RAM to the flash memory.



Operand	Type	Format	Description
LM_CFG_CurveAccess	LM_CFG_CurveAccess	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
Mode	INT	Immediate Tag	Set access mode: 0 = Save all curves from RAM to Flash. (MC_SW must be stopped!) 1 = Delete all Curves (RAM) 2 = Add Curve (RAM) 3 = Modify Curve (RAM) 4 = Get curve (-> stored in CurveArray)
CurveID	INT	Immediate Tag	Curve ID (number). Range: 1..255
CurveArray	DINT[1019]	Tag	Curve data. Contains all important curve data.
ErrorID	INT	Immediate Tag	Error ID if error is detected.

LM_CFG_CurveAccess Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command.
ER (Error)	An error occurred.
IP (In Process)	Drive is processing the requested mode.
PC (Process Complete)	Process complete.

Attention:
Before executing mode 0 the MC_SW of the drive must be stopped. Use LM_CFG_StartStopDefault Mode 5 to do so.

Attention:
The curve stored on the LinMot drive must not exceed 1001 set-points!

5.9 Optional Instructions: Force / Torque Control

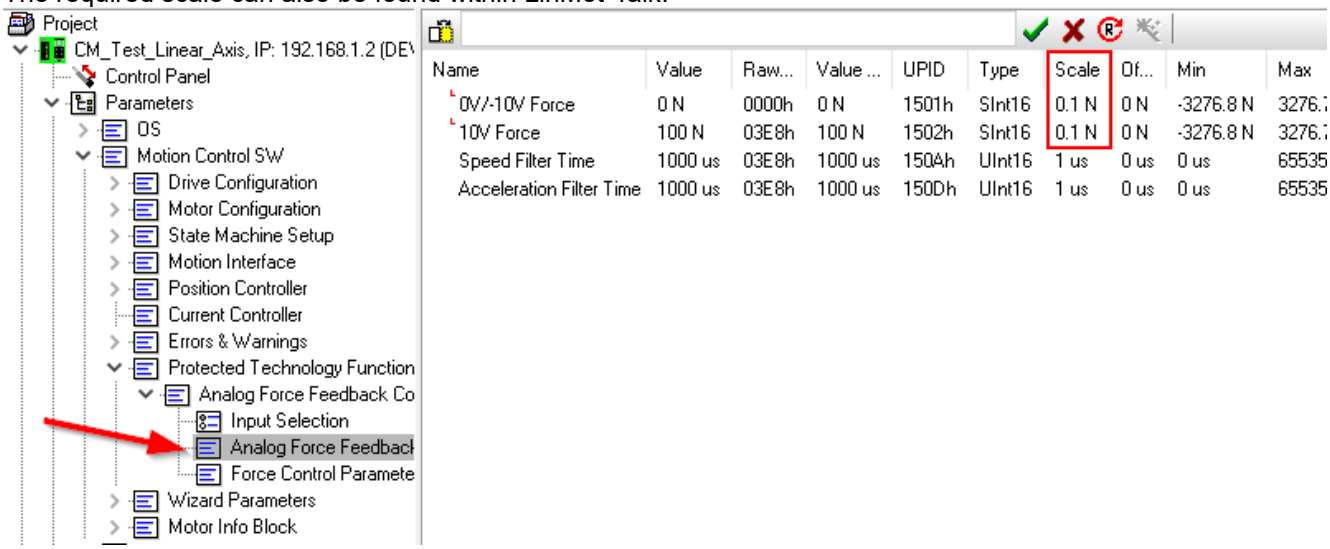
This chapter lists instructions for closed loop force/torque control.

These instructions can only be used when the virtual axis is not coupled with the physical axis!
Use *tstLM_Axis.PVAStatus* to check if the axis is in streaming mode.

The input tag ***TFScale*** of the instructions shown in this chapter must be set to the correct scaling factor depending on the motor that is used.

Motor Type	Encoder Settings	TFScale
PS01	For linear motor it is always 1 mm	0.1 (N)
PR01 & PR02	360'000 ticks per motor revolution	0.00057295779513082 (Nm)
EC02	524'288	0.000834430268037632 (Nm)

The required scale can also be found within LinMot-Talk:



Attention:
The closed loop force / torque control technology function must be ordered separately:
Technology Function: Closed Loop FC (Part Nr. 0150-2503)

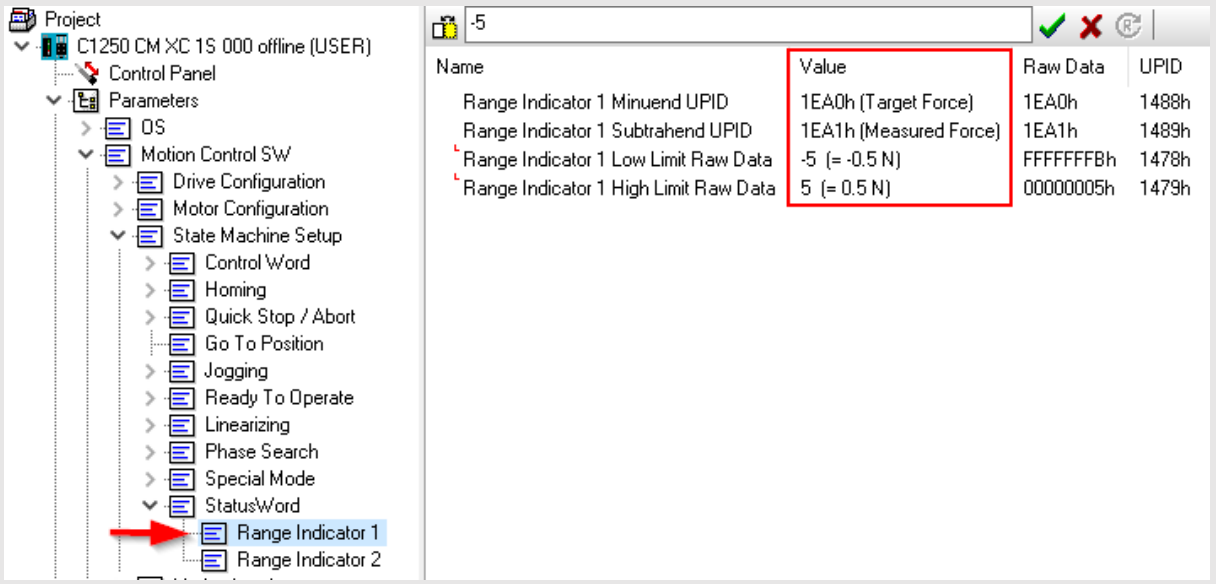
Note:
Additional information about the commands used in this chapter can be found in the Motion Control Software user manual > See Recommended Documentation

Attention: Check if the Target Force is reached

If the Target Force is reached must be approved in the application itself as the force/torque control instructions set the DN output TRUE when the drive has switched from position to closed loop force/torque control.

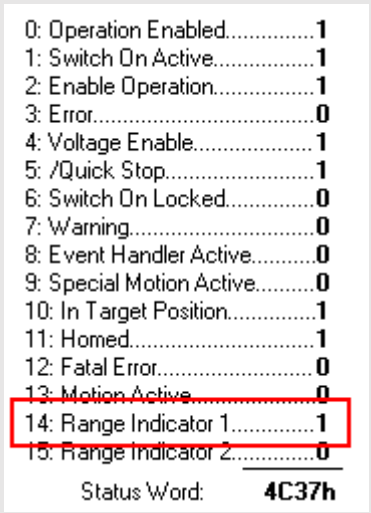
Either you can use a Monitoring Channel (CM / MI drives only) to have the measured force transmitted from the drive to the controller in the bus cycle time, and implement your compare logic,

or you can set the Range Indicator 1 accordingly and check Bit 14 of the StatusWord to become TRUE.



Check the Bit in the axis UDT to become TRUE:

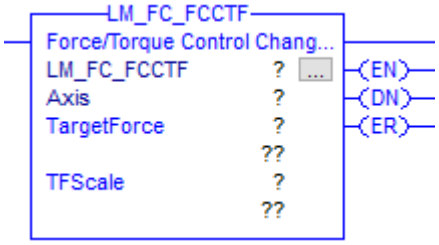
> ***tstLMAxis.StatusWord.14***



The Range Indicator 1 Low/High Limits can be modified using the LM_ParameterWrite instruction by accessing UPID 1478h or 1479h.

5.9.1 Force / Torque Control Change Target Force (LM_FC_FCCTF)

Use this instruction to change the target force / torque if the drive is in force / torque control mode.



Operand	Type	Format	Description
LM_FC_FCCTF	LM_FC_FCCTF	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
TargetForce	REAL	Immediate Tag	Target force [N] / torque [Nm].
TFScale	REAL	Immediate Tag	Force [N] / Torque [Nm] scale

LM_FC_FCCTF Structure Descriptions

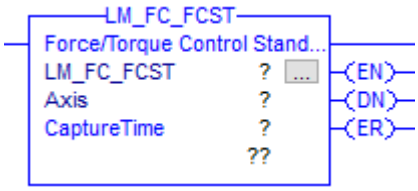
Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The command was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	Drive is setting new target force / torque.
PC (Process Complete)	Drive has set new target force / torque.

Note:
- This instruction implements the LinMot motion command: “Force Ctrl Change Target Force (382xh)”

5.9.2 Force / Torque Control Standstill Tara (LM_FC_FCST)

Use this instruction to tara the differential analogue input -10V..10V.

This command captures the Measured Force/Torque (UPID 0x1EA1) over the parametrized Capture Time and stores its average as 0V Raw Data Offset (UPID 0x1798).



Operand	Type	Format	Description
LM_FC_FCST	LM_FC_FCST	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
CaptureTime	INT	Immediate Tag	Capture time in milliseconds [ms]

LM_FC_FCST Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The command was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	Drive is capturing and calculating the average.
PC (Process Complete)	Drive has completed capturing and calculating the average

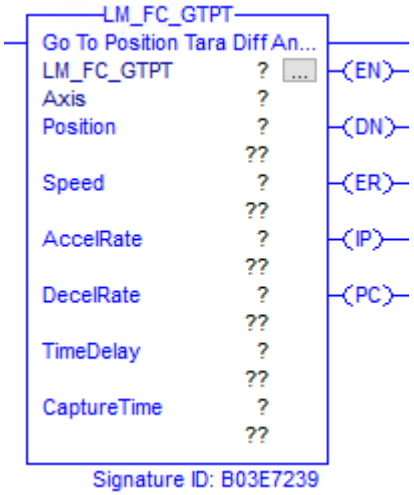
Note:
- This instruction implements the LinMot motion command: “Standstill Tara Diff Analog In -10V..10V (38Bxh)”

Attention:
This command is available on C1250 series drives and from firmware **6.10 and higher** only.

5.9.3 Go To Position Tara Diff Analog In -10..10V (LM_FC_GTPT)

Use this instruction to tara the differential analogue input -10V..10V.

This command sets a new target position and defines the maximal speed, acceleration, and deceleration rate for the movement. The command execution starts immediately when the command has been sent.
This command additionally captures the Measured Force (UPID 0x1EA1) while the Demand Velocity is in maximal velocity state and stores its average as 0V Raw Data Offset (UPID 0x1798).
The measurement of the Measured Force can be delayed by the parameter Time Delay and is captured during the time Capture Time. More information about the command is found in the manual.



Operand	Type	Format	Description
LM_FC_FCST	LM_FC_FCST	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
Position	REAL	Immediate Tag	Absolute position to move to
Speed	REAL	Immediate Tag	Speed to move the axis in [Units/s]
AccelRate	REAL	Immediate Tag	Acceleration rate of the axis in [Units/s^2]
DecelRate	REAL	Immediate Tag	Deceleration rate of the axis in [Units/s^2]
TimeDelay	REAL	Immediate Tag	Time delay in milliseconds [ms]
CaptureTime	REAL	Immediate Tag	Capture time in milliseconds [ms]

LM_FC_FCGTPT Structure Descriptions

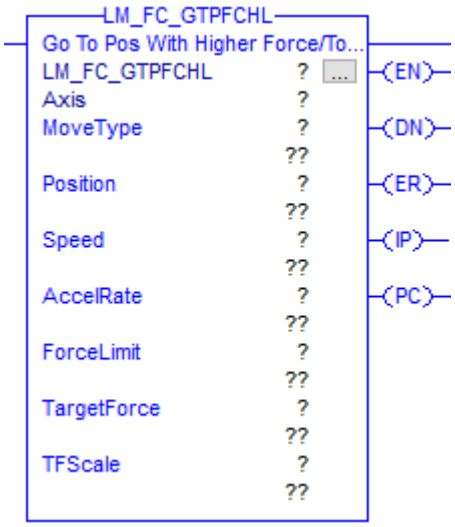
Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The command was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	Drive is moving, capturing, and calculating the average.
PC (Process Complete)	Drive has reached the position and completed capturing and calculating the average.

Note:
This instruction implements the LinMot motion command:
“Go To Position Tara Diff Analog In -10..10V (38Axh)”

Attention:
This command **only supports linear axis** at this point and is available on C1250 series drives from firmware **6.10** and higher.

5.9.4 Go To Position Force / Torque Control High Limit (LM_FC_GTPFCHL)

With this AOI the motor starts moving towards the target position (*Position*). If the measured force / torque **exceeds** the *ForceLimit* the drive switches to force / torque control mode with target force / torque = *TargetForce*.
Hint: If the target position is reached without switching to force / torque control mode an error is generated (ErrorID = 07h).



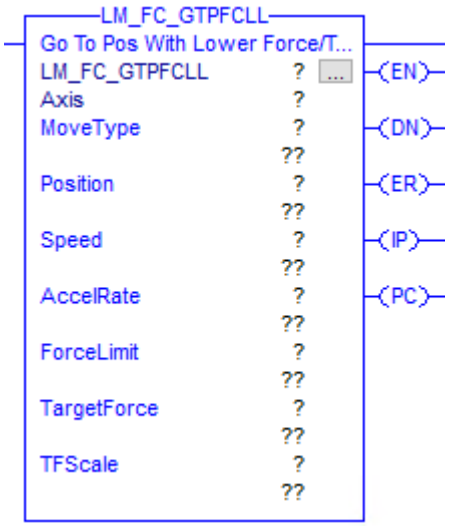
Operand	Type	Format	Description
LM_FC_GTPFCHL	LM_FC_GTPFCHL	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
MoveType	DINT	Tag Immediate	Absolute (0), Incremental (1)
Position	REAL	Immediate Tag	Target position / distance. Should not be reached but defines the maximum the motor moves if no force is detected. [Units]
Speed	REAL	Immediate Tag	Speed for the force search movement. [Units/s]
Acceleration	REAL	Immediate Tag	Acceleration for the force search movement. [Units/s^2]
ForceLimit	REAL	Immediate Tag	Force / torque limit that must be **exceeded** to switch to force / torque control mode in [N] / [Nm].
TargetForce	REAL	Immediate Tag	Target force [N] / [Nm].
TFScale	REAL	Immediate Tag	Force [N] / Torque [Nm] scale.

LM_FC_GTPFCHL Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The command was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	Motor is moving to detect the Force Limit.
PC (Process Complete)	Force / Torque Limit reached, and drive is operating in force / torque control mode.

5.9.5 Go To Position Force / Torque Control Low Limit (LM_FC_GTPFCLL)

With this AOI the motor starts moving towards the target position (*Position*). If the measured force / torque **undercuts** the *ForceLimit* the drive switches to force / torque control mode with target force / torque = *TargetForce*.
Hint: If the target position is reached without switching to force / torque control mode an error is generated (ErrorID = 07h).



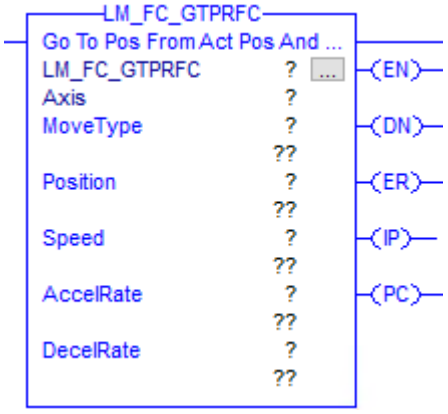
Operand	Type	Format	Description
LM_FC_GTPFCHL	LM_FC_GTPFCHL	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
MoveType	DINT	Tag Immediate	Absolute (0), Incremental (1) **Mode 1 requires firmware 6.9 and higher!**
Position	REAL	Immediate Tag	Target position / distance. Should not be reached but defines the maximum the motor moves if no force is detected. [Units]
Speed	REAL	Immediate Tag	Speed for the force search movement. [Units/s]
Acceleration	REAL	Immediate Tag	Acceleration for the force search movement. [Units/s^2]
ForceLimit	REAL	Immediate Tag	Force / torque limit that must be **undercut** to switch to force / torque control mode in [N] / [Nm].
TargetForce	REAL	Immediate Tag	Target force [N] / [Nm].
TFScale	REAL	Immediate Tag	Force [N] / Torque [Nm] scale.

LM_FC_GTPFCLL Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The command was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	Motor is moving to detect the Force Limit.
PC (Process Complete)	Force / Torque Limit reached, and drive is operating in force / torque control mode.

5.9.6 Go To Position Reset Force Control (LM_FC_GTPRFC)

With this AOI the drive can be set back to position control loop and moved from its actual position to the position defined with *Position, Speed, Acceleration* and *Deceleration*.



Operand	Type	Format	Description
LM_FC_GTPRFC	LM_FC_GTPRFC	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
MoveType	DINT	Tag Immediate	Absolute (0), Incremental (1)
Position	REAL	Immediate Tag	Target position / distance in [Units].
Speed	REAL	Immediate Tag	Speed in [Units/s].
Acceleration	REAL	Immediate Tag	Acceleration in [Units/s^2].
Deceleration	REAL	Immediate Tag	Deceleration in [Units/s^2].

LM_FC_GTPRFC Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The command was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	Motor is moving to position.
PC (Process Complete)	Axis position controlled and in target position.

Note:
- This instruction implements the LinMot motion commands:
MoveType 0 (Absolute): “VAI Go To Pos From Act Pos And Reset Force Control Set I (386xh)”
MoveType 1 (Incremental): “VAI Increment Act Pos And Reset Force Control Set I (387xh)”

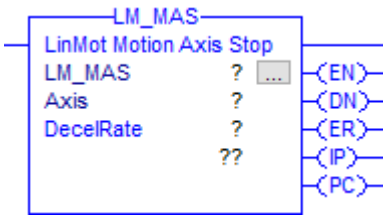
5.10 Optional Instructions: Motion Move Instructions known from LinMot -IP Drives

This chapter lists a couple of motion instructions that directly control the physical axis. They can be used if no coordinated motion is required or if you do not want to use the physical axis in a motion group.

These instructions can only be used when the virtual axis is not coupled with the physical axis!
Use *tstLM_Axis.PVAStatus* to check if the axis is in streaming mode.

5.10.1 Motion Axis Stop (LM_MAS)

The LM_MAS instruction is the equivalent for the Rockwell Automation MAS instruction. Use this instruction to stop a specific motion process on an axis or to stop the axis completely.



Operand	Type	Format	Description
LM_MAS	LM_MAS	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
DecelRate	REAL	Immediate Tag	Deceleration rate of the axis in [Units/s^2] **Important**: The axis could overshoot its target position if you reduce the deceleration while a move is in process.

LM_MAS Structure Descriptions

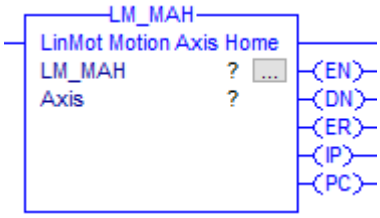
Enumerations	Description
EN (Enable)	The EN bit stays set until the process is complete and the rung goes false. A false-to-true transition caused the instruction to execute.
DN (Done)	The stop was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	The axis is stopping.
PC (Process Compete)	The axis stopped. The PC bit stays set until the rung makes a false-to-true transition.

Note:
- Stop Type is always **All (0)**
- DecelRate is in position units per second squared [Units/s^2]
- DecelRate is 10’000 Units/s^2 by default
- A trapezoidal profile is always used for LM_MAS

- This instruction implements the LinMot motion command “VAI Stop (017xh)”

5.10.2 Motion Axis Home (LM_MAH)

The LM_MAS instruction is the equivalent for the Rockwell Automation MAH instruction. Use this instruction to home an axis.
The homing sequence can be selected during axis configuration (LinMot-Talk Motor Wizard). It's an active homing.



Operand	Type	Format	Description
LM_MAH	LM_MAH	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.

LM_MAH Structure Descriptions

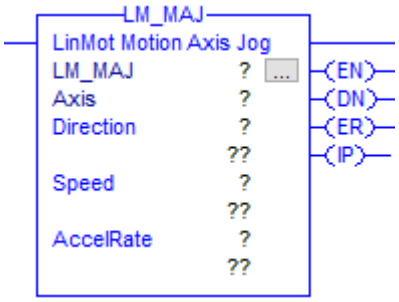
Enumerations	Description
EN (Enable)	It is set when the rung makes a false-to-true transition and remains set until the servo message transaction is completed and the rung goes false.
DN (Done)	It is set when axis home has been successfully completed or is aborted.
ER (Error)	It is set to indicate that the instruction detected an error.
IP (In Process)	It is set on positive rung transition and cleared after the Motion Home Axis is complete, or terminated by a stop command, shutdown, or a servo fault.
PC (Process Compete)	It is set when axis home is successfully completed.

Note:
The homing sequence can be set-up using LinMot-Talk. The easiest way is to use the Motor Wizard.

Note:
If a master gantry system is used (E1250-IP or E1450-IP drives) a special instruction is available for homing (LM_MAH_MG) which guarantees a proper homing of all involved axes.

5.10.3 Motion Axis Jog (LM_MAJ)

The LM_MAJ instruction is the equivalent for the Rockwell Automation MAJ instruction. Use this instruction to move an axis at a constant speed until you tell it to stop using the LM_MAS instruction above.



Operand	Type	Format	Description
LM_MAJ	LM_MAJ	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
Direction	DINT	Immediate Tag	Forward = 0 Reverse = 1
Speed	REAL	Immediate Tag	Speed to move the axis in [Units/s]
AccelRate	REAL	Immediate Tag	Acceleration rate of the axis in [Units/s^2]

LM_MAJ Structure Descriptions

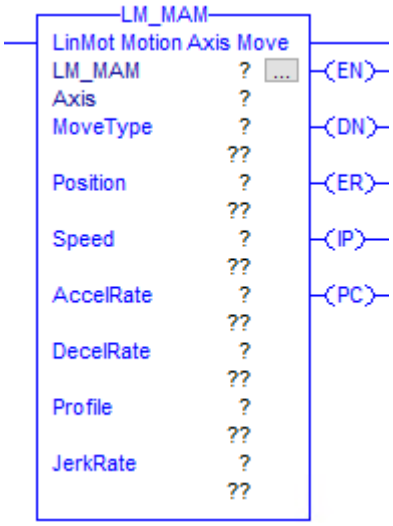
Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The jog was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	The axis is jogging.

Note:
- Profile is always trapezoidal
- Default Speed = 10 Units/s, Default AccelRate = 10 Units/s^2

- This instruction implements the LinMot motion commands:
VAI Dec=Acc VAI Infinite Motion Positive Direction (0CExh) and
VAI Dec=Acc VAI Infinite Motion Negative Direction (0CFxh)

5.10.4 Motion Axis Move (LM_MAM)

The LM_MAM instruction is the equivalent for the Rockwell Automation MAM instruction. Use this instruction to move an axis to a specified position.



Operand	Type	Format	Description	
LM_MAM	LM_MAM	Tag	Instance of the Add-on Instruction. Access instruction status parameters.	
Axis	tstLM_Axis	Tag	Axis to perform the action on.	
MoveType	DINT	Immediate Tag	Absolute	0
			Incremental	1
			Rotary Shortest Path	2 profile 0 only
			Rotary Positive	3 profile 0 only
			Rotary Negative	4 profile 0 only
Position	REAL	Immediate Tag	Absolute position or incremental distance for the move.	
			For this Move Type	**Enter this Position value in [Units]**
			Absolute	Position to move to
			Incremental	Distance to move
Speed	REAL	Immediate Tag	Speed to move the axis in [Units/s].	
AccelRate	REAL	Immediate Tag	Acceleration rate of the axis in [Units/s^2].	
DecelRate	REAL	Immediate Tag	Deceleration rate of the axis in [Units/s^2] Only used with Trapezoidal (0) and S-Curve (2) profile.	
Profile	DINT	Immediate Tag	Select the profile to run for the move. 0 = Trapezoidal 1 = Bestehorn 2 = S-Curve 3 = Sine	
JerkRate	REAL	Immediate Tag	Jerk rate of the axis in [Units/s^3] The instruction only uses the JerkRate operand if the Profile is Bestehorn (1) or S-Curve (2). Setting a LM_MAM to Bestehorn with 0 as JerkRate value will not produce a Trapezoidal profile.	

LM_MAM Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The move was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	The axis is moving. Any of these actions stop this move and clear the IP bit: • The axis gets to the end Position • Another MAM instruction supersedes this MAM instruction • MAS instruction • Shutdown command • Fault Action
PC (Process Compete)	• The PC bit stays set until the rung makes a false-to-true transition. • The PC bit stays cleared if some other action stops the move before the axis gets to the end Position.

Note: If an input tag is 0 (zero) the following default values are taken
Default Speed = 10 Units/s,
Default AccelRate & DecelRate = 10 Units/s^2,
Default JerkRate = 10 Units/s^3

Note:
The MAM instruction implements the following LinMot motion commands:
Profile 0 Trapezoidal:
- Absolute: VAI Go To Pos (010xh)
- Incremental: VAI Increment Dem Pos (011xh)
- Rotary Shortest Path : VAI Modulo Go To Pos Shortest Way (3C2xh)
- Rotary Positive: VAI Modulo Go To Pos Positive Direction (3C0xh)
- Rotary Negative: VAI Modulo Go To Pos Negative Direction (3C1xh)

Profile 1 Bestehorn:
- Absolute: Bestehorn VAJ Go To Pos (0F0xh)
- Incremental: Bestehorn VAJ Increment Demand Pos (0F1xh)

Profile 2 S-Curve:
- Absolute: VAJI Go To Pos (3A0xh)
- Incremental: VAJI Increment Dem Pos (3A1xh)

Profile 3 Sine:
- Absolute: Sin VA Go To Pos (0E0xh)
- Incremental: Sin VA Increment Demand Pos (0E1xh)

Attention: Risk of Velocity and/or End Position Overshoot
If you change move parameters dynamically by any method, for example by starting a new instruction before the last one has completed, be aware of the risk of velocity and/or end position overshoot.

A Trapezoidal profile can overshoot if maximum deceleration is decreased while the move is decelerating or is close to the deceleration point.

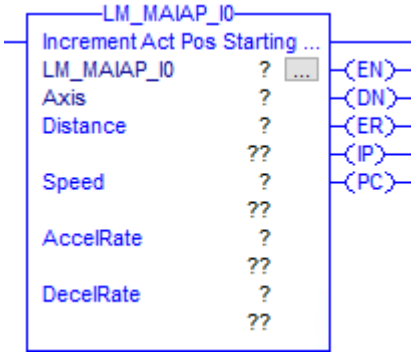
A Bestehorn profile can overshoot if a Bestehorn move is overwritten by another Bestehorn move.

Attention:
The Bestehorn profile has same acceleration and deceleration rate. So only AccelRate has influence.

5.10.5 Motion Axis Increment Actual Position (LM_MAIAP_I0)

Use the LM_MAIAP_I0 instruction to increment the actual position and reset the I part of the position controller of a LinMot motor.

The most common use case is to reset the position lag after pressing on an object.



Operand	Type	Format	Description
LM_MAIAP	LM_MAIAP	Tag	Instance of the Add-on Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
Distance	REAL	Immediate Tag	Distance to move
Speed	REAL	Immediate Tag	Speed to move the axis in [Units/s]
AccelRate	REAL	Immediate Tag	Acceleration rate of the axis in [Units/s^2]
DecelRate	REAL	Immediate Tag	Deceleration rate of the axis in [Units/s^2]

LM_MAIAP_I0 Structure Descriptions

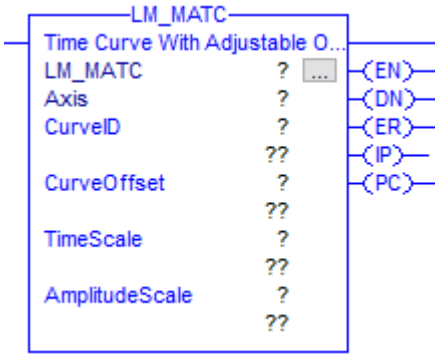
Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The move was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	The axis is moving.
PC (Process Complete)	The move was successfully completed.

Note:
- Profile is always trapezoidal
- Default Speed = Units/s, Default AccelRate & DecelRate = 10 Units/s^2

- The instruction LM_MAIAP_I0 implements the LinMot motion command:
VAI Increment Act Pos Starting with Dem Vel = 0 Reset I-Contr (0D9xh)

5.10.6 Motion Axis Time Curve (LM_MATC)

Use this instruction to start a curve (motion profile) stored on the LinMot drive.



Operand	Type	Format	Description
LM_MATC	LM_MATC	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
CurveID	DINT	Immediate Tag	Curve ID to start. Range: 1...255
CurveOffset	REAL	Immediate Tag	Curve Offset in [Units].
TimeScale	REAL	Immediate Tag	Time scaling in [%]. Range: 0.1 to 200.0%
AmplitudeScale	REAL	Immediate Tag	Amplitude scaling in [%]. Range: -2000.0 to +2000.0%

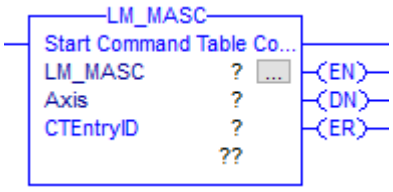
LM_MATC Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The move was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	The axis is running the curve.
PC (Process Complete)	The axis finished the curve.

Note:
- This instruction implements the LinMot motion command:
“Time Curve With Adjustable Offset, Time Scale & Amplitude Scale (044xh)”

5.10.7 Motion Axis Start Command Table (LM_MASC)

Use this instruction to start an entry of the Command Table stored on the LinMot drive.



Operand	Type	Format	Description
LM_MATC	LM_MATC	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.
CTEntryID	DINT	Immediate Tag	Command Table entry ID to start. Range: 1..255

LM_MASC Structure Descriptions

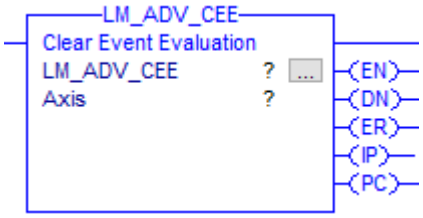
Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The drive has accepted command and is executing the Command Table entry.
ER (Error)	An error occurred.

Note:
If the Command Table Entry started is linked to other entries, they will be executed as well!

- This instruction implements the LinMot motion command: “Start Command Table Command (200xh)”

5.10.8 Clear Event Evaluation (LM_ADV_CEE)

Use this instruction to clear the event evaluation inside the LinMot drive. E.g., to stop a running command table.



Operand	Type	Format	Description
LM_ADV_CEE	LM_ADV_CEE	Tag	Instance of the Addon Instruction. Access instruction status parameters.
Axis	tstLM_Axis	Tag	Axis to perform the action on.

LM_ADV_CEE Structure Descriptions

Enumerations	Description
EN (Enable)	A false-to-true transition caused the instruction to execute. The EN bit stays set until the process is complete and the rung goes false.
DN (Done)	The command was successfully initiated.
ER (Error)	An error occurred.
IP (In Process)	Drive is clearing event evaluation.
PC (Process Complete)	Event evaluation cleared.

Note:
- This instruction implements the LinMot motion command: “Clear Event Evaluation (008xh)”

6 Error Descriptions

6.1 Motion Instructions Errors

ERR	Corrective Action or Cause	Notes
1	Home axis with LM_MAH	Axis Not Homed
5	Close the servo loop before you execute this instruction. E.g., Use LM_MSO	Servo Off State Error
7	Target position reached (FC instructions only)	LM_FC_GTPFC only > see chapter 5.9
13	An operand is outside its range	
16	Wait until the homing process is done.	Home In Process Error
20	The axis is in the faulted state.	Axis In Faulted State
>=4096	Internal instruction has an error > Err = 1000h + Error ID see 6.2	

6.2 Configuration Instructions Errors

If the ER output is set the following Error IDs are possible

Error ID	Error Text	Description
00h	Instruction aborted	The instruction was aborted by another instruction.
01h	TimeOut (No response from drive)	Drive is not responding within the requested time. Check fieldbus connection
02h	ConfigChannel already busy	Configuration channel is already busy. Check if another configuration instance is in process.
03h	Invalid Mode selected	Invalid mode selected. Please check Mode input
06h	Array size to small	Connected Array is too small
C0h	UPID error	Unknown UPID selected, check UPID input
C1h	Parameter Type Error	
C2h	Range Error	The value to be written is outside the parameters range
C3h	Address Usage Error	There is an attempt to write a read only parameter
C5h	Error: Command 21h	
D0h	Odd Address	
D1h	Size Error (Curve Service)	
D4h	Curve already defined / Curve not present (Curve Service)	
D7h	MC Software is still running, should be stopped	To store the Command Table / Curves into the Flash memory of the drive the MC_SW must be stopped (LM_CFG_StopStartDefault, Mode 5)
other		Contact LinMot technical support

7 Optional Example Programs

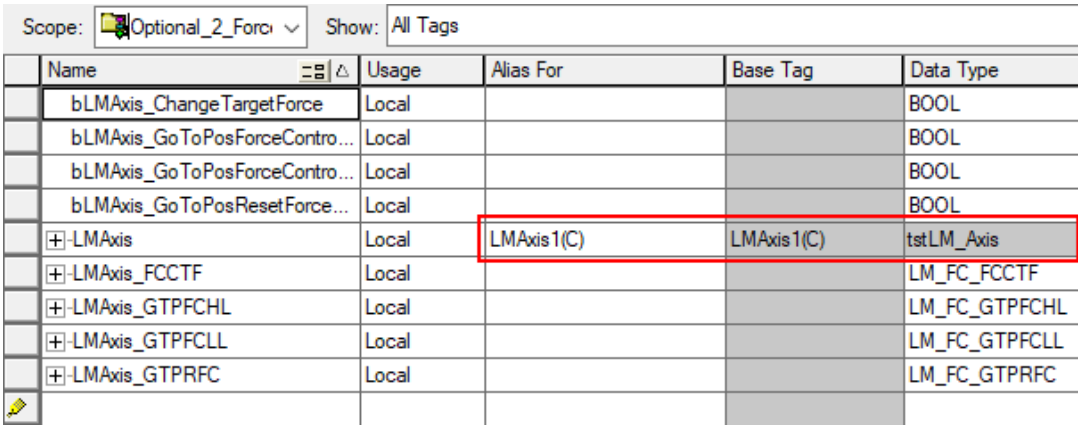
7.1 Functionality Beside Synchronized Motion

Beside the demo project (CIPSync_Demo_...) the package also contains some exported programs in the folder *\Exports\Optional*.

Optional_1_Parameters…L5X shows the use of the parameter and configuration instructions in a very basic manner.

Optional_2_ForceTorqueControl…L5X shows the use of the force / torque control instructions in a very basic manner.

Both program exports can be imported into a, preferably new, project to look at the implementation.
After importing do not forget to set a valid alias for the tstLM_Axis type, e.g.:



7.2 Example Projects for Non-Synchronized Motion

In the folder *\Misc* of this package two demo projects can be found that show how the physical axis can be controlled without a virtual motion axis.

7.2.1 CM Drive (C1250-CM-…, C1250-MI-…, C1251-MI-…)

A demo project is available called:

- *...\Misc\CM_EDS_NoMotion_Demo_L320ERM_Vxx_xxxxxxxx*

7.2.2 IP Drive (C1250-IP-…, E1250-IP-…) as AOP

With some limitations also C1250-IP and E1250-IP drives can use the version 2 instructions. To read and write the drive communication two instructions are available in the folder *\AOI*:

- LM_LMDriveAOP_Read_IP.L5X
- LM_LMDriveAOP_Write_IP.L5X

The demo project showing this integration can be found in the folder *\Misc*.

- *...\Misc\IP_AOP_NoMotion_Demo_L320ERM_Vxx_xxxxxxxx*

7.2.3 IP Drive (C1450-IP-…, E1450-IP-…) as EDS

With some limitations also all other LinMot -IP drives can use the version 2 instructions. To read and write the drive communication two instructions are available in the folder *\AOI*:

- LM_LMDriveEDS_Read_IP.L5X
- LM_LMDriveEDS_Write_IP.L5X

The demo project showing this integration can be found in the folder *\Misc*.

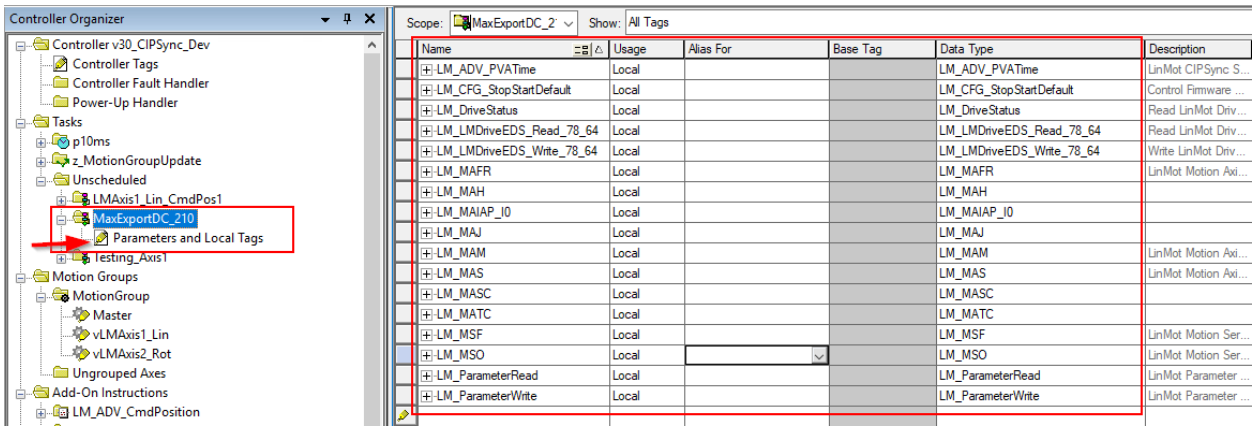
- *...\Misc\IP_EDS_NoMotion_Demo_L320ERM_Vxx_xxxxxxxx*

8 Principle of Upgrading a Project from IP to CM Drives

The following steps illustrate a possible way to upgrade an existing project from the IP to CM drives. There are a variety of implementation philosophies and there may be other steps required. Please contact our technical support if you have any problems.

Note:
It is not foreseen that both -IP and -CM/MI type drives are used in the same project!
V2 instructions are not compatible with IP drives.

1. In an empty or test project create a program where you add all new V2 instructions in the tags that you used in the old (IP) project.
   E.g.:

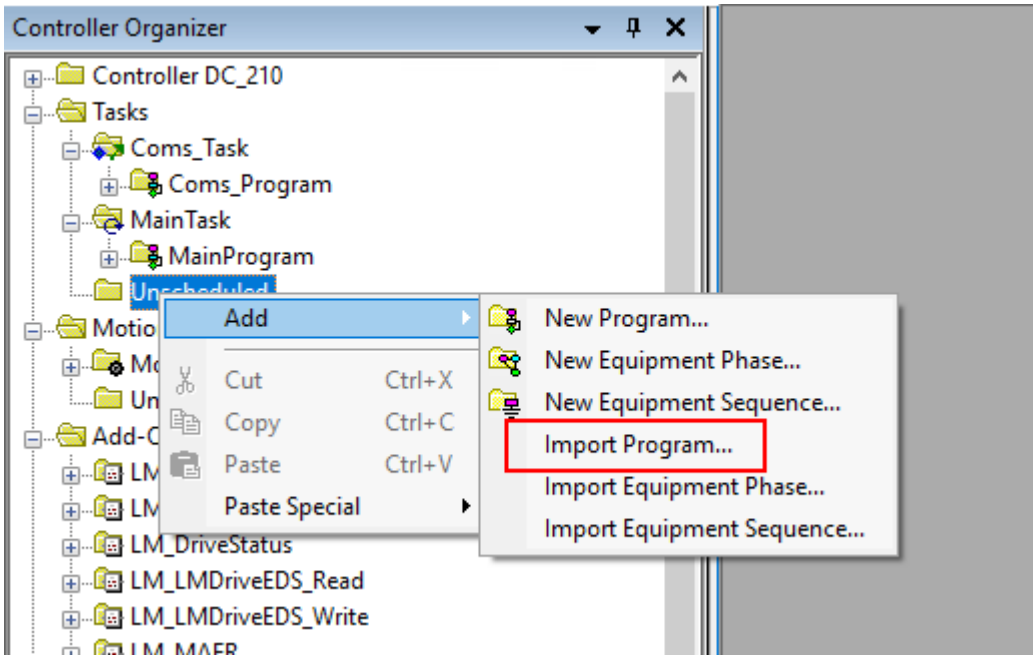


2. Export this program.

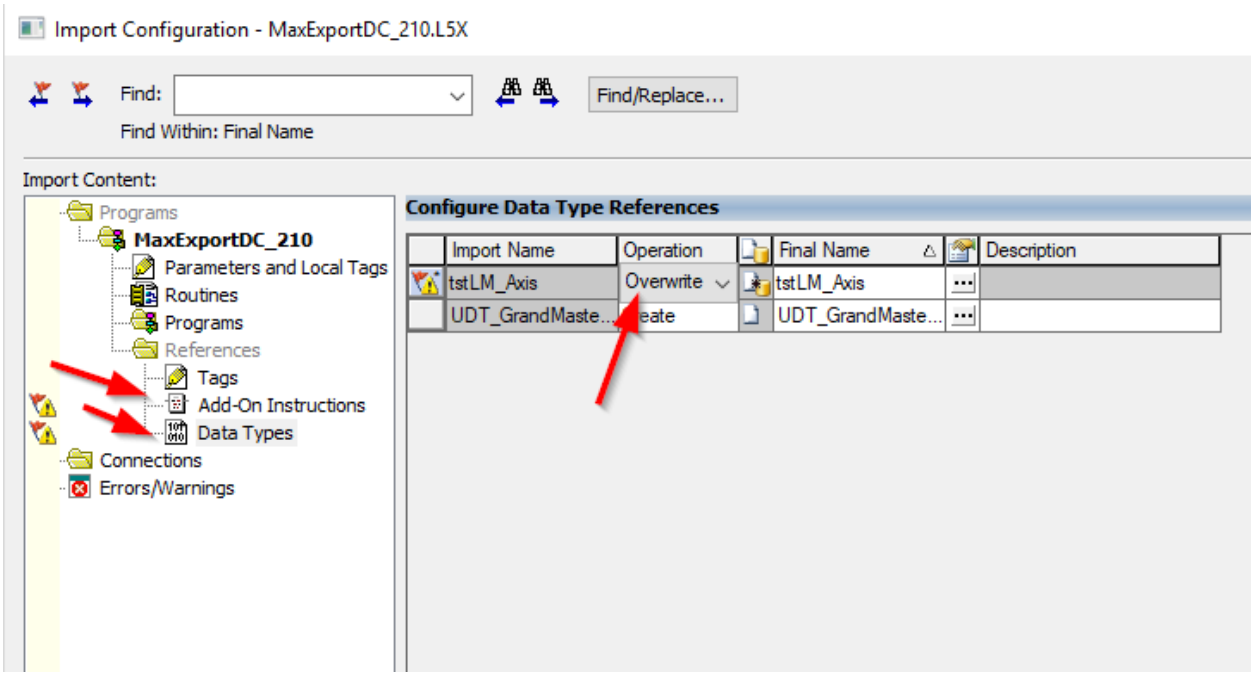
3. Install the EDS file of the CM/MI drive you want to use.

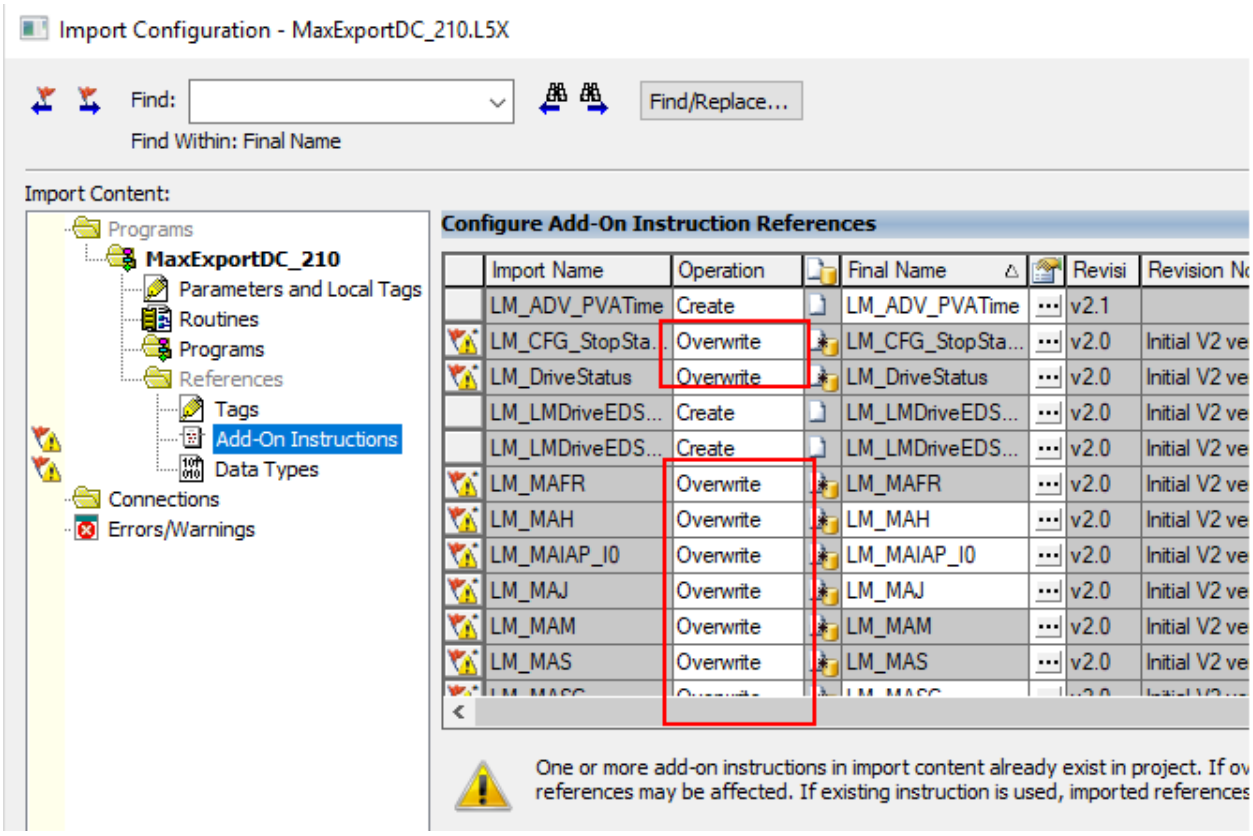
4. Add CM/MI drive(s) to your existing project replacing the existing IP drive(s).

5. Import the exported program from above into your existing project.

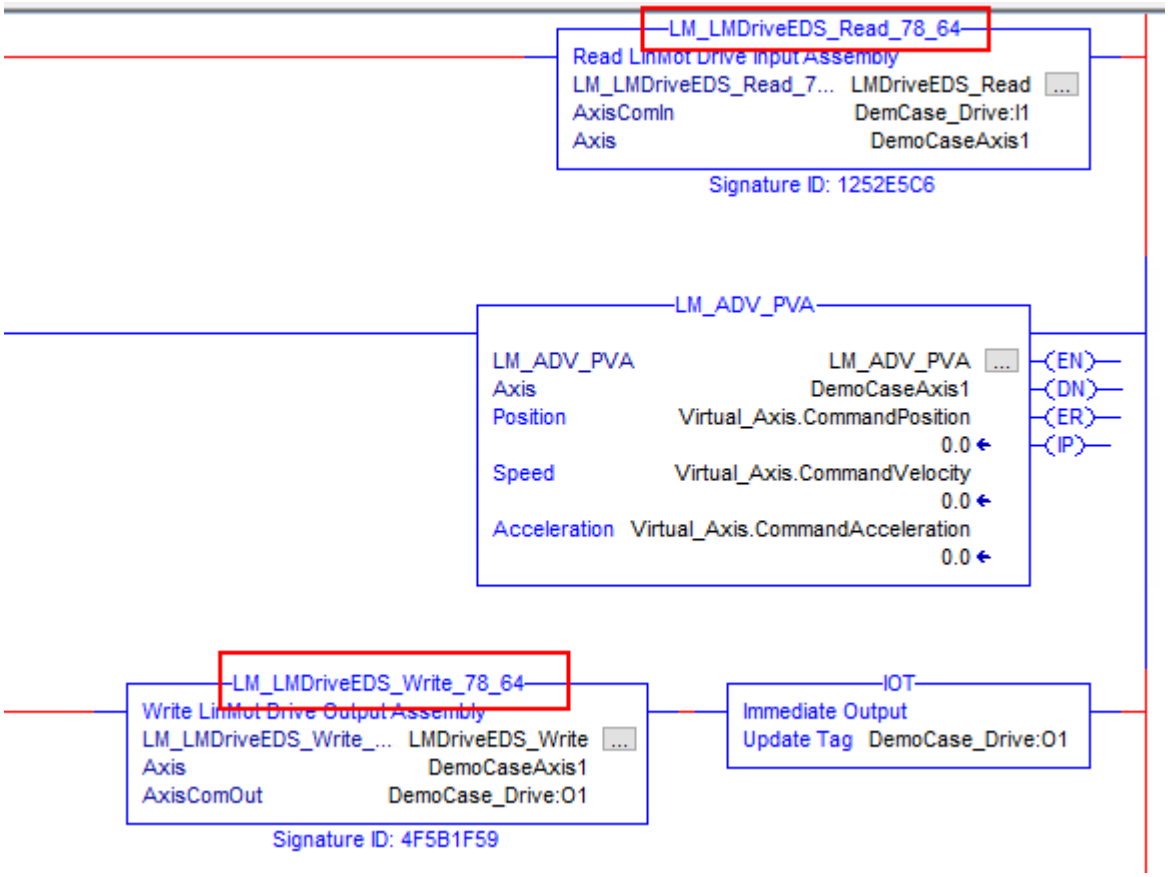


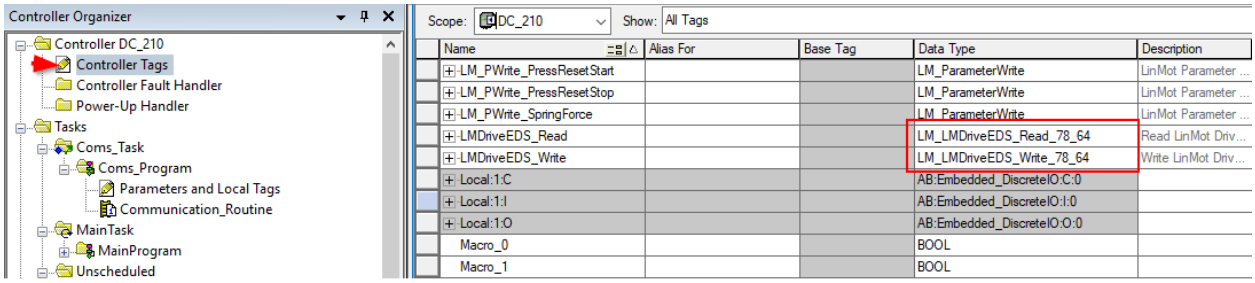
6. **Attention:** In the import configuration set all Add-On Instructions and Data Types to ***Overwrite***!





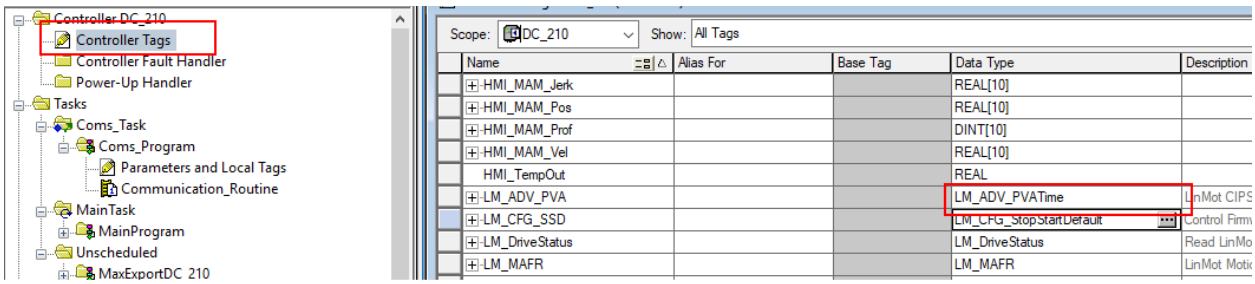
7. Replace the old read and write instructions in the project (and then delete them from the project).

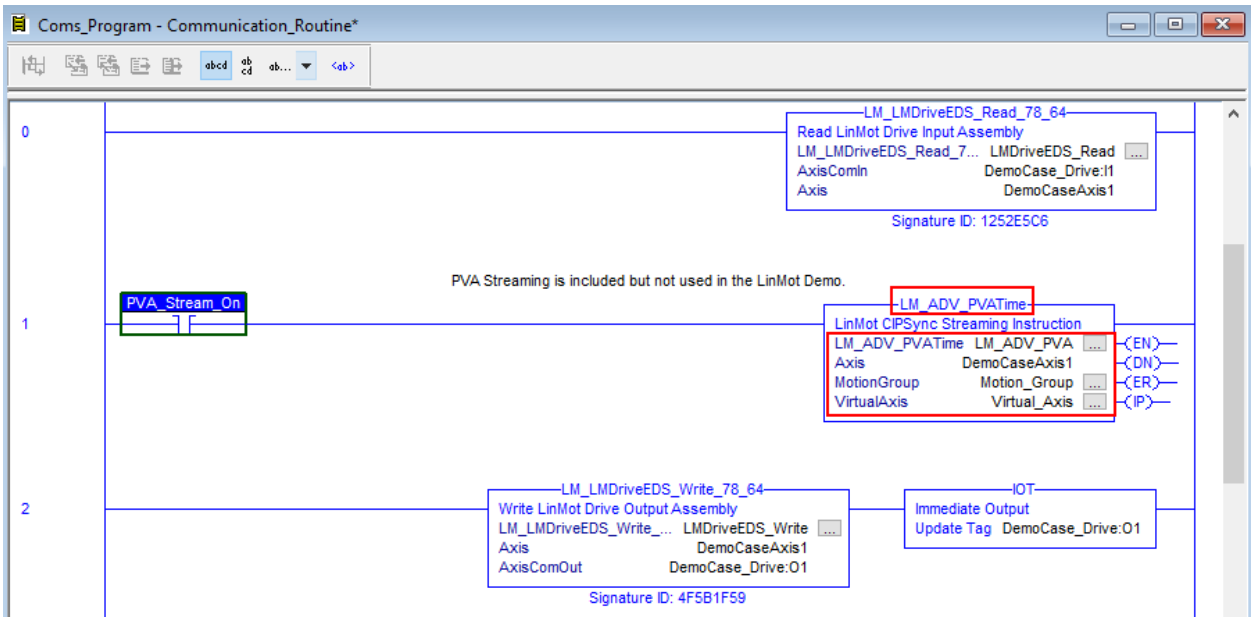




8. In this example it would make sense to rename the LM_ADV_PVA instruction to LM_ADV_PVATime first, so that it is overwritten during import.

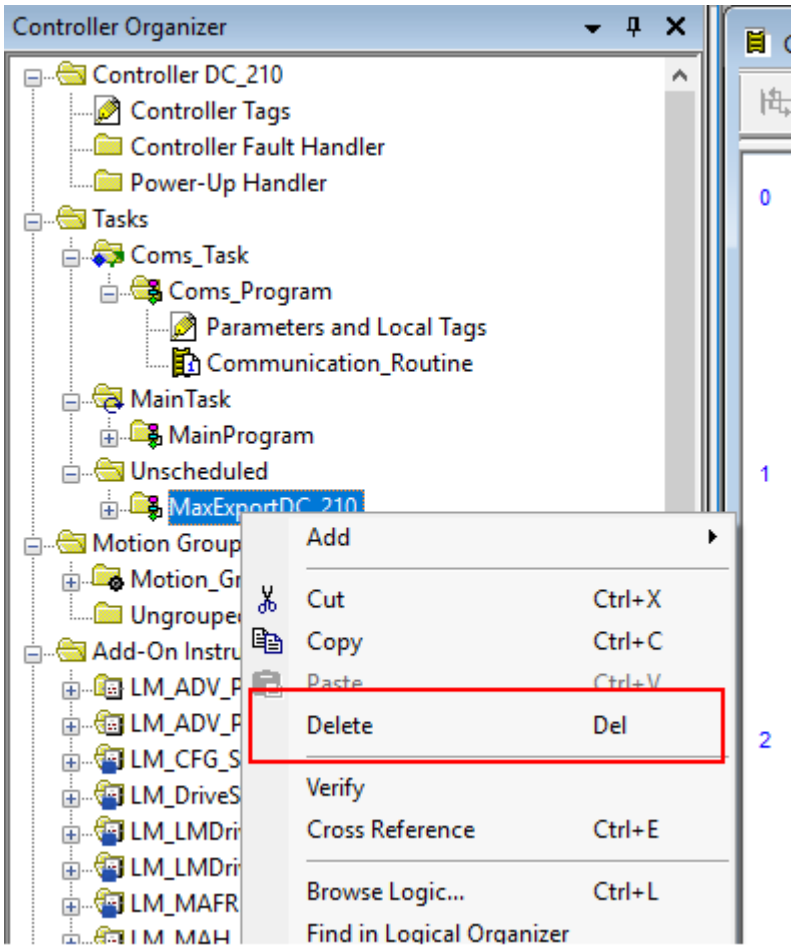
   If not then replace the instructions LM_ADV_PVA > LM_ADV_PVATime in both the routine and tags. Then correct the tags at the instruction.





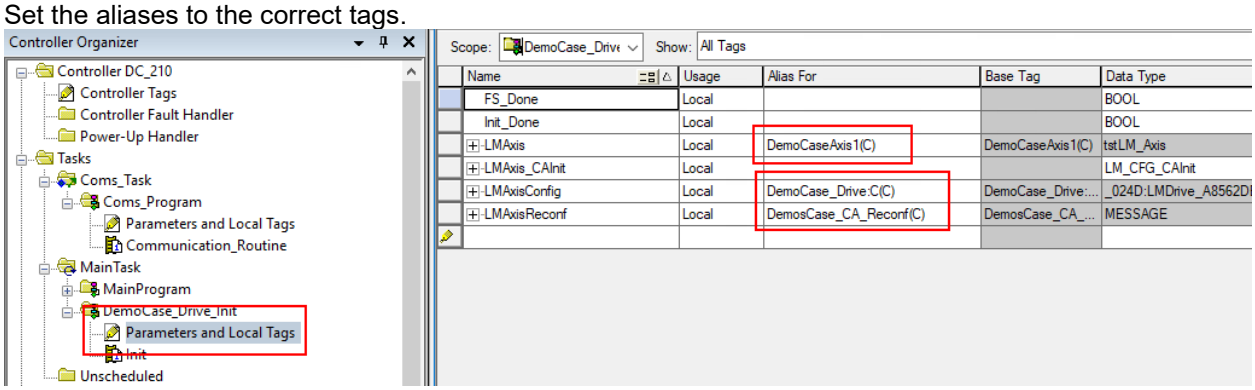
9. Build the controller and check for errors like ActualCurrent <> CurrentFeedback, invalid input and output tags, and so on.

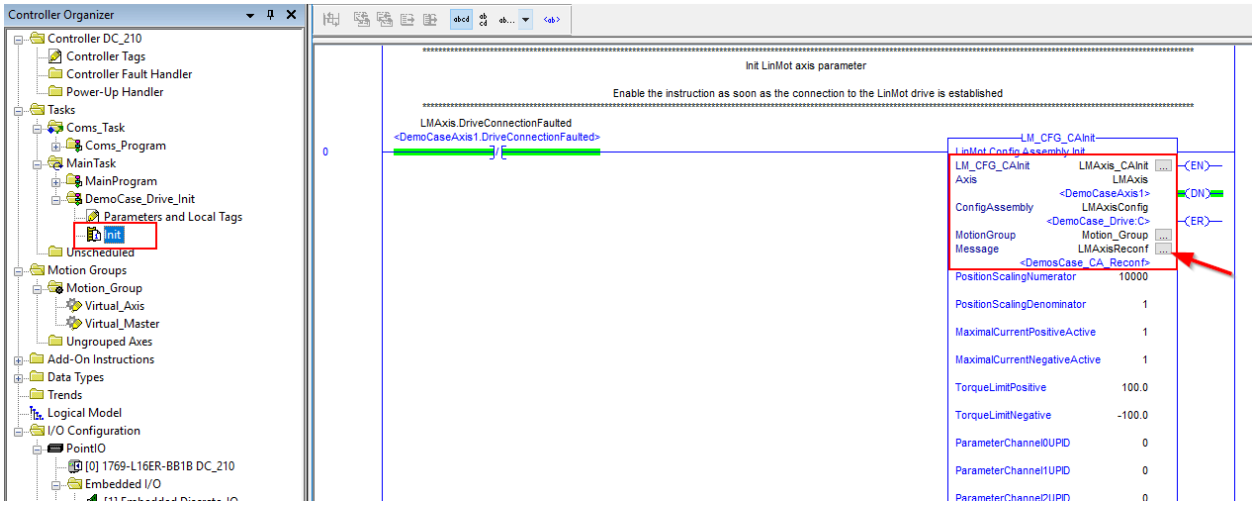
10. Delete the imported program.



11. To have the PVA CIP Sync running you must add the *LM_CFG_CAInit* instruction (copy paste or import), add an init program, add a message tag in the controller tags and set it up as shown in chapters 4.1.2 & 5.8.1.

    Set the aliases to the correct tags.

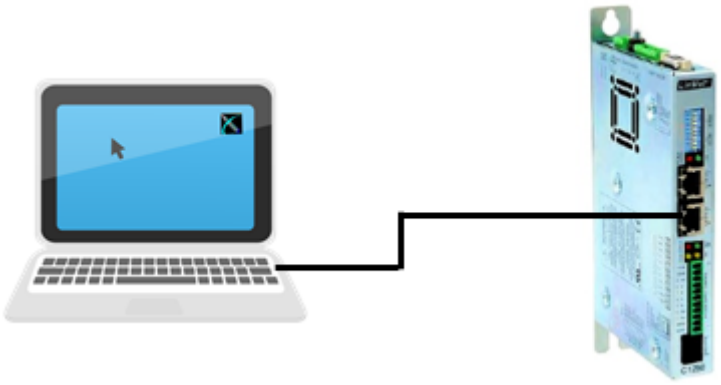




12. Now the project should be ready.

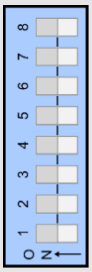
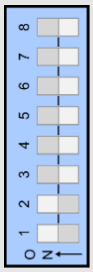
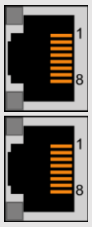
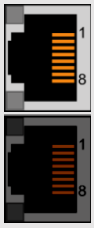
9 Appendix I: LinMot-Talk Configuration over EtherNet/IP

9.1 Login with static IP address



9.1.1 Settings on the LinMot Drive

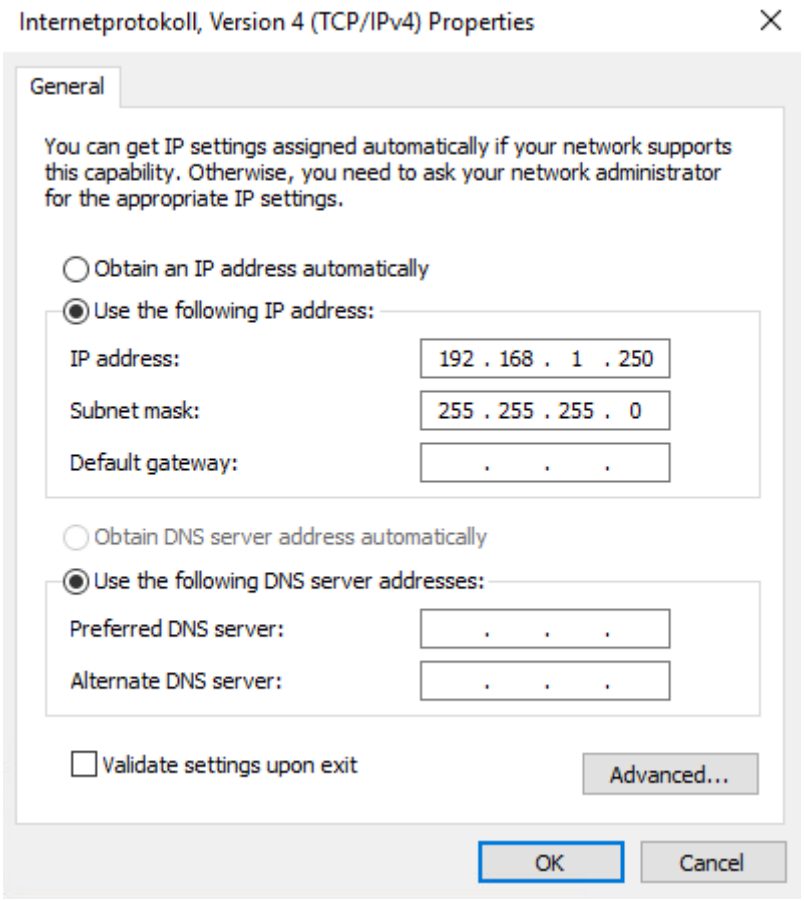
The default setting of the EtherNet/IP port address is manual IP Configuration using hex switches. With the standard settings, the IP address will be in the following range 192.168.001.xxx. The last number can be set with the hex switches S1 and S2.

Identification	Description	Example
S1 – S2 	S1: (5..8) Bit 5 is LSB Bit 8 is MSB S2: (1...4) Bit 1 is LSB Bit 4 is MSB	IP address with the settings below:  S1 = binary 0000, dec 0, hex 0 ---------- S2 = binary 0011, dec 3, hex 3 (Bits 1 and 2 are switched to on) Address = 192.168.001.003
X17 – X18 	X17 RT ETH In X18 RT ETH Out	Connect network cable on X17:  X17 RT ETH In

Note: Changes on S1 and S2 require a power up cycle to refresh the values in the drive.

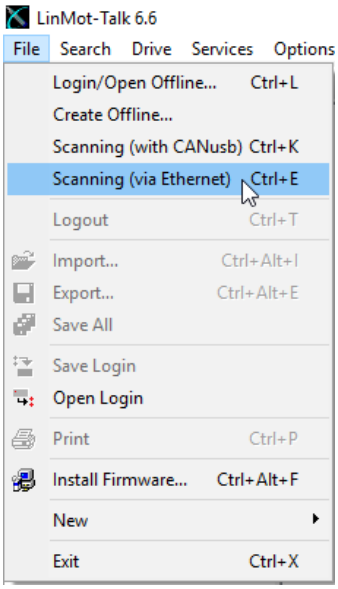
9.1.2 Settings on the PC

The computer needs to be in the same IP subnet as the servo drive. Set the network settings in the PC to an IP address with 192.168.1.xxx. The IP address needs to be different from the IP address of the drive.



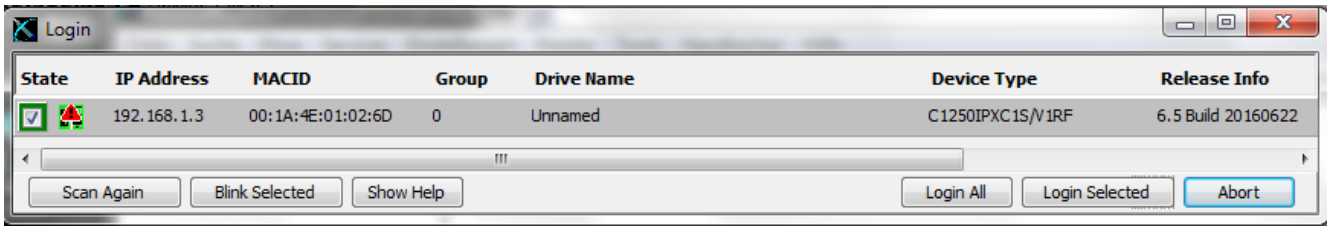
9.1.3 Login with LinMot-Talk

Open the LinMot-Talk Software and start the Scanning (via Ethernet) in the menu tab:
File –> Scanning (via Ethernet).



Select the network card of the laptop. Make sure that the IP address of the Laptop is in the following sub net (192.168.1.xxx).

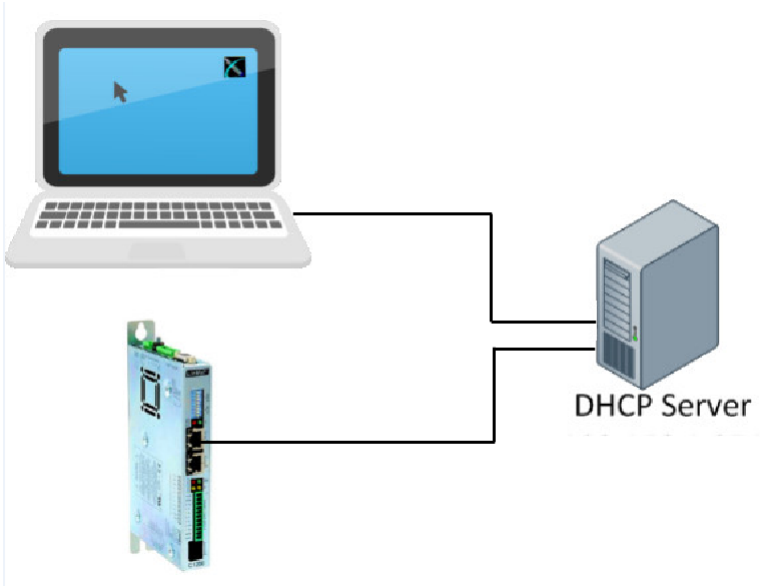
The following table will show all connected LinMot drives in the network.



Login to the drive to adjust drive settings.

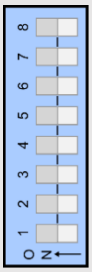
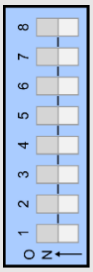
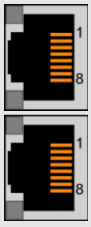
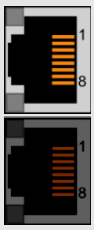
Note: When configuring the IP address of the drive, a new scan is necessary to reconnect the LinMot-Talk communication if the IP address is changed. The IP address of the drive can be configured with LinMot Talk:
Parameter/ Ethernet/IP Intf/ IP Configuration Mode

9.2 Login with dynamic IP address (DHCP)



9.2.1 Settings on the LinMot Drive

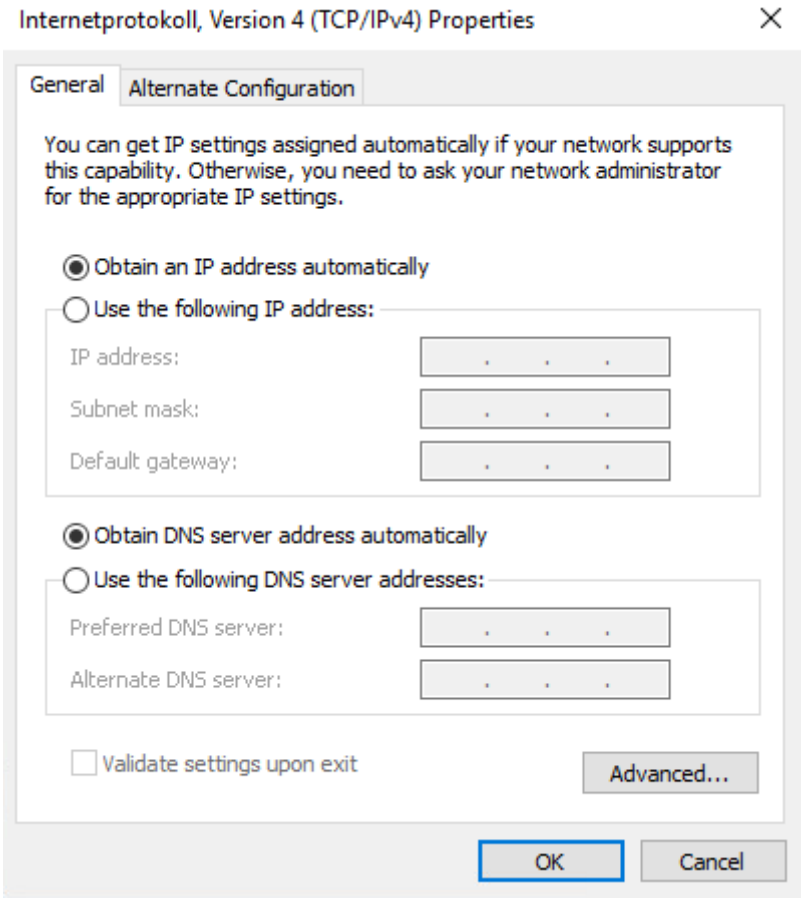
If the hex switches are set to S1=0 and S2=0, the servo drive is in the dynamic IP configuration mode. The servo drive doesn't support APIPA (Automatic Private IP Addressing). Connect the servo drive with a DHCP Server to receive a TCP/IP address.

Identification	Description	Example
S1 – S2 	S1: (5..8) Bit 5 is LSB Bit 8 is MSB S2: (1...4) Bit 1 is LSB Bit 4 is MSB	S1 to OFF / S2 to OFF = All Dip Switches OFF = DHCP. 
X17 – X18 	X17 RT ETH In X18 RT ETH Out	Connect the network cable to X17:  X17 RT ETH In

Note: Changes on S1 and S2 need a power up cycle to refresh the values in the drive.

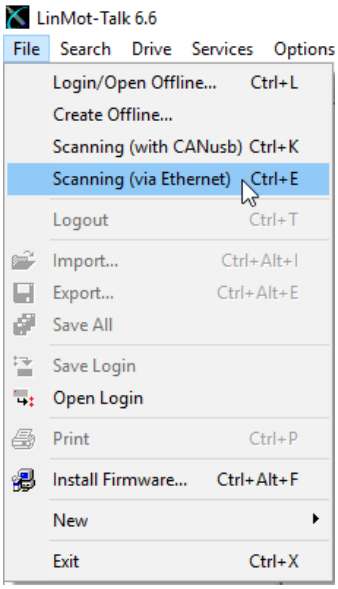
9.2.2 Setting on PC

Set the Network Configuration IPv4 to “Obtain an IP address automatically”.

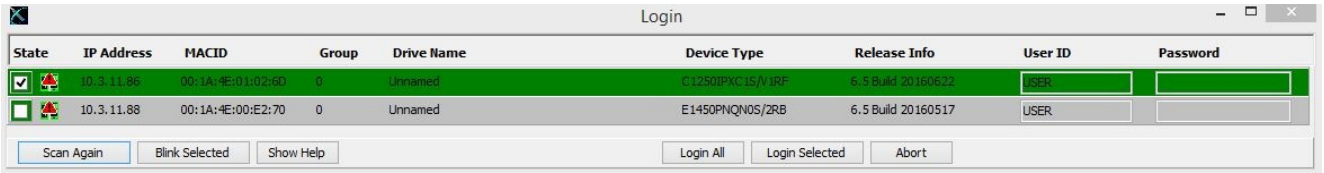


9.2.3 Login with LinMot-Talk

Open the LinMot-Talk software and select “Scanning (via Ethernet)” in the menu tab:



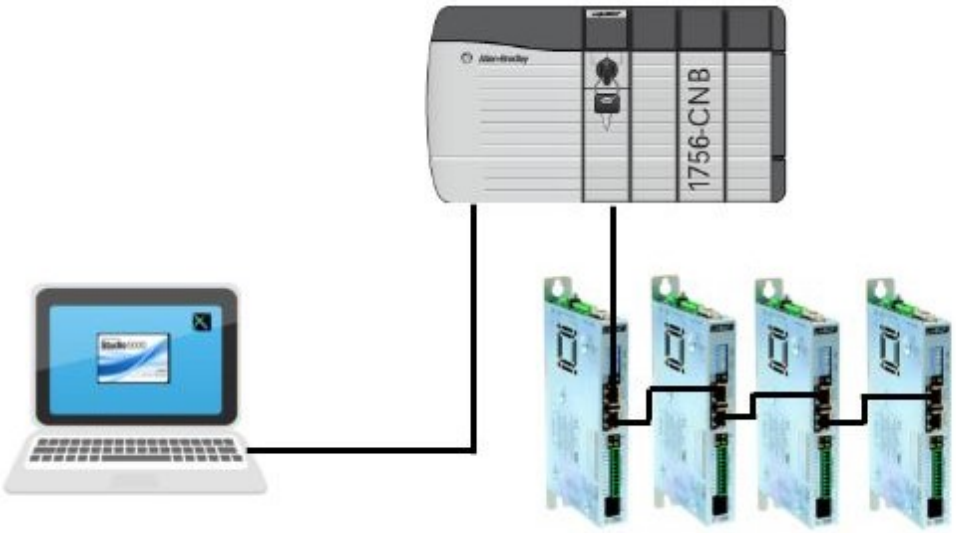
Attention: Use the **“Blink Selected”** function if you have more than one drive in the network.



Login to the drive to adjust the drive settings.

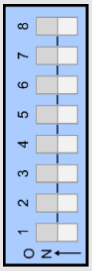
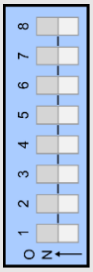
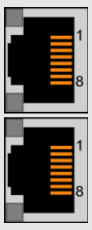
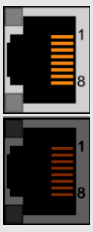
Note: When configuring the IP address of the drive, a new scan is necessary to reconnect the LinMot-Talk communication if the IP address is changed. The IP address of the drive can be configured with LinMot Talk:
Parameter/ Ethernet/IP Intf/ IP Configuration Mode

9.3 IP address configuration with BOOTP/ DHCP Server



9.3.1 Settings on the LinMot Drive

If the hex switches are set to S1=0, and S2 =0, the servo drive is in the dynamic IP configuration. The servo drive doesn't support APIPA (Automatic Private IP Addressing). Connect the servo drive with a DHCP Server to receive a TCP/IP address.

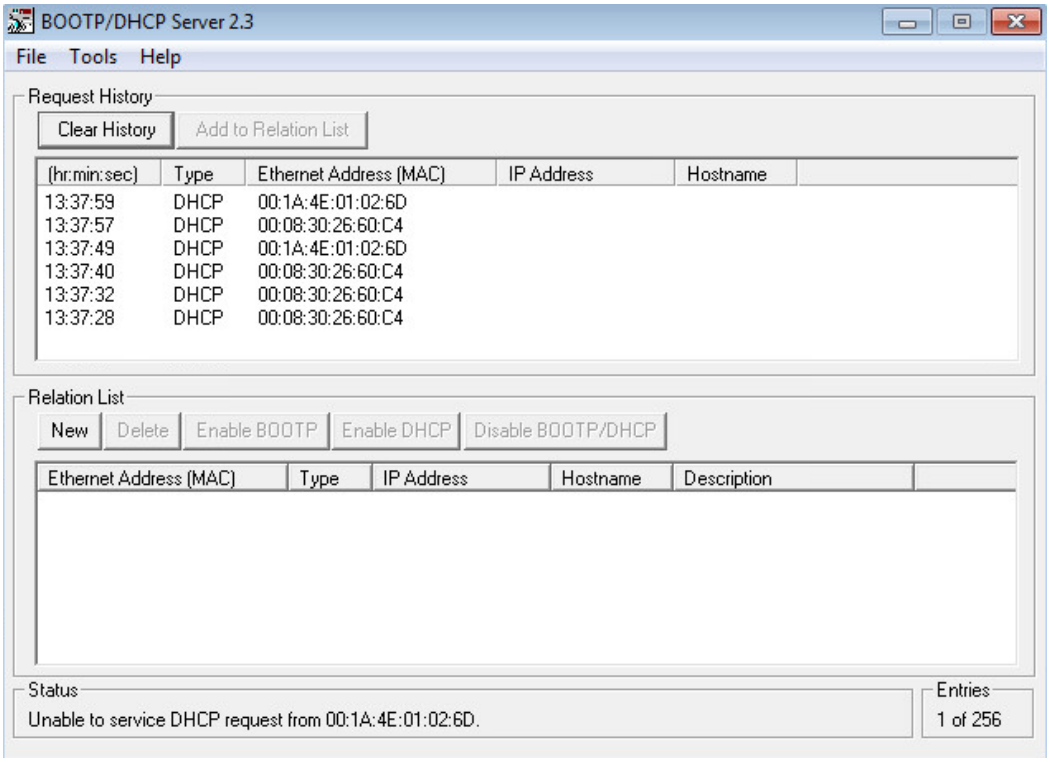
Identification	Description	Example
S1 – S2 	S1: (5..8) Bit 5 is LSB Bit 8 is MSB S2: (1...4) Bit 1 is LSB Bit 4 is MSB	S1 to OFF / S2 to OFF = All Dip Switches OFF = DHCP. 
X17 – X18 	X17 RT ETH In X18 RT ETH Out	The network cable is plugged into X17:  X17 RT ETH In

9.3.2 Using the Software BOOTP/ DHCP Server

The following settings are shown with BOOTP/DHCP Server Version 2.3.2.0 from Rockwell Automation.
The software shows all the MAC addresses in the Ethernet network. An IP address can be assigned to a LinMot drive with this software. Once the servo drive has an assigned IP address, the LinMot-Talk Software can connect to the drive over the Ethernet real-time port.

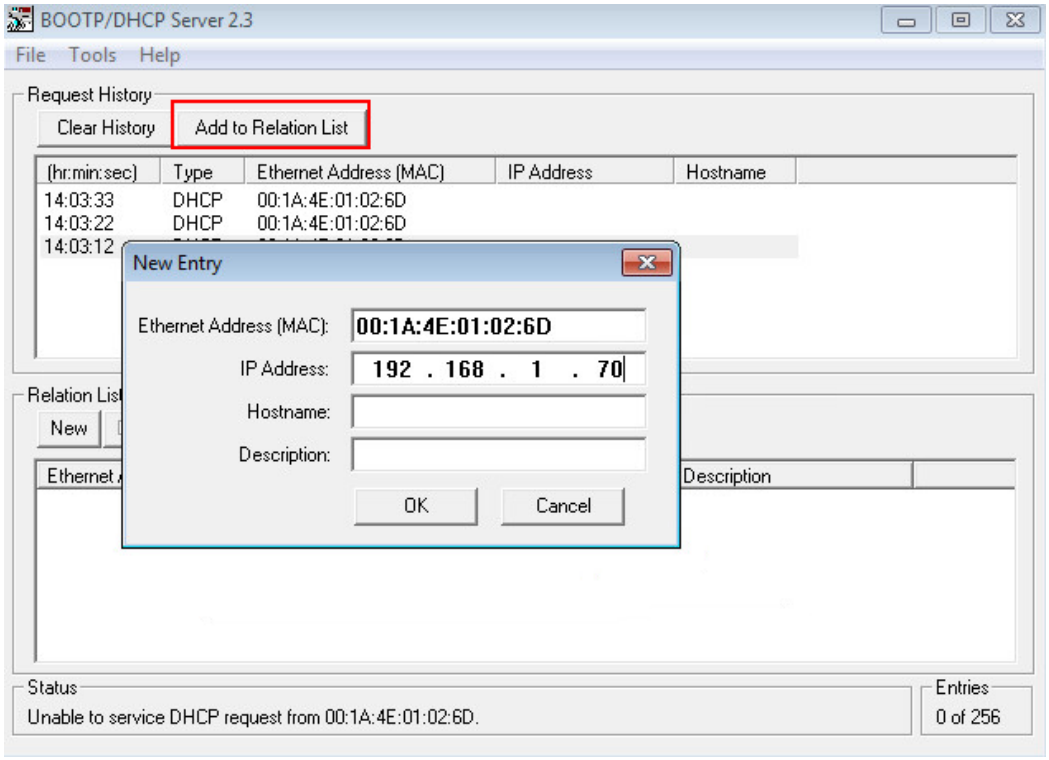
9.3.2.1 Search for the LinMot Drive in the Network

The BOOTP Software shows all the slaves in the Network.

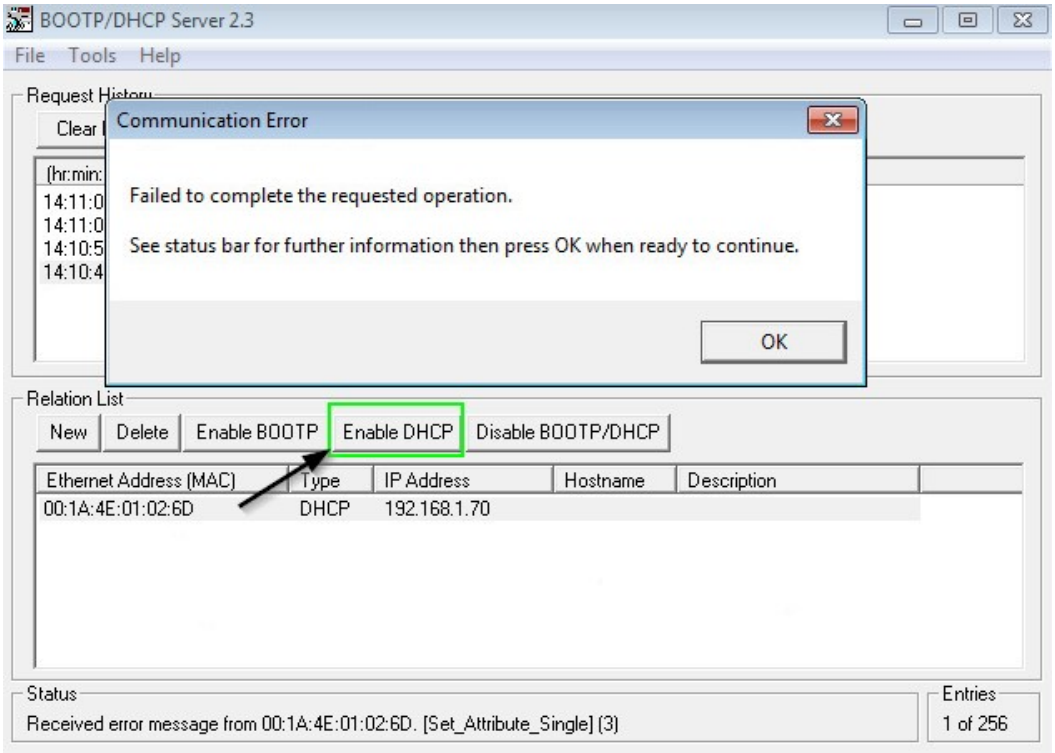


Compare the MAC-Address in the list with the MAC on the Servo Drive label.

Select the correct MAC and assign the IP address to the drive.



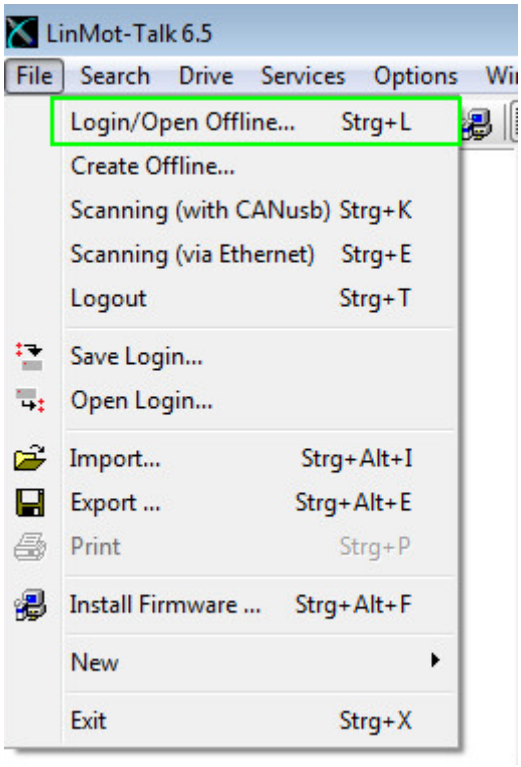
Add the IP address in the Relation List and press “Enable DHCP”. Normally a Communication Error is displayed, but the message can be ignored. Press OK.

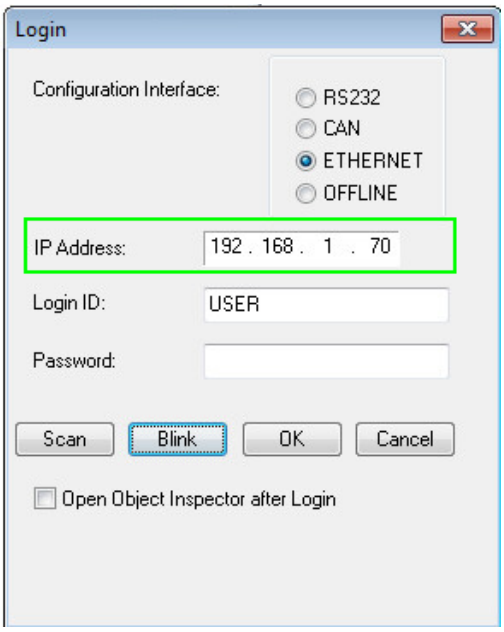


Note: After the IP address assignment, the BOOTP/DHCP Server Software must be closed. After that, restart the LinMot-Talk Software to login in the drive.

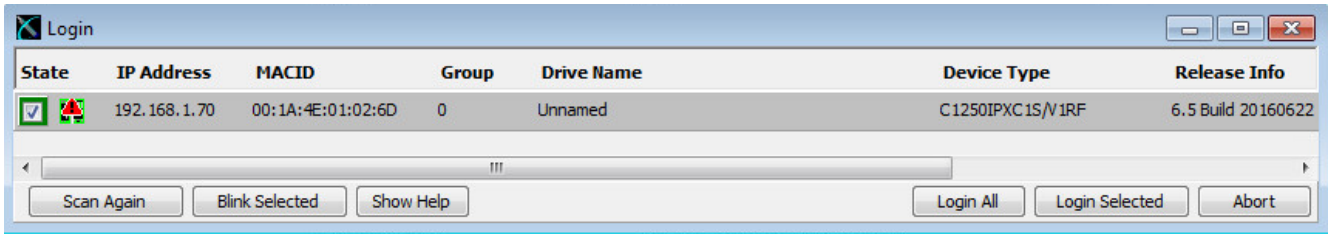
9.3.3 Login with LinMot-Talk

In the LinMot-Talk Software ***File -> Login/Open Offline*** and enter the IP address of the Servo Drive.



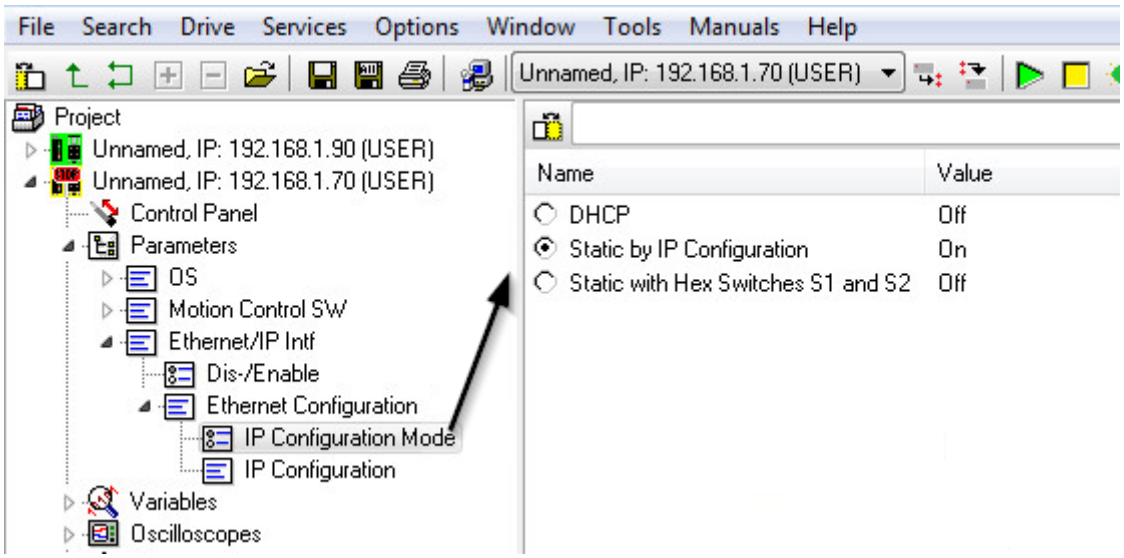


Use the Scan function if the IP address is unknown.

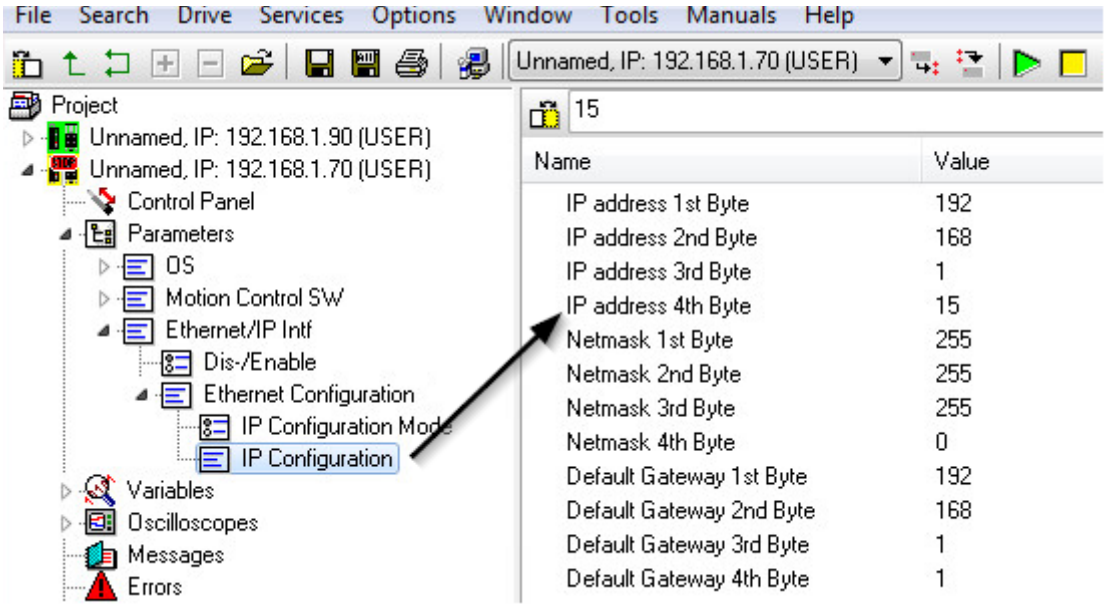


9.3.3.1 Change the IP settings in the LinMot-Talk Software

Go to *Parameters/ Ethernet/IP Intf/ Ethernet Configuration/ IP Configuration Mode* and set the mode to “Static by IP Configuration”.



Change the settings of the IP address if the IP Configuration Mode is Static by IP Configuration.
Parameters/ Ethernet/IP Intf/ Ethernet Configuration/ IP Configuration



Attention: Modifications to the IP address will be activated only after a restart of the firmware. Logging back into the drive with the new IP address will be necessary to reestablish the LinMot-Talk connection.

10 Appendix II: Master / Gantry System

In some applications like a portal system often a master gantry setup is required where a slave axis follows a master axis.

This can be achieved with the Motion Axis Gear (MAG) instruction.

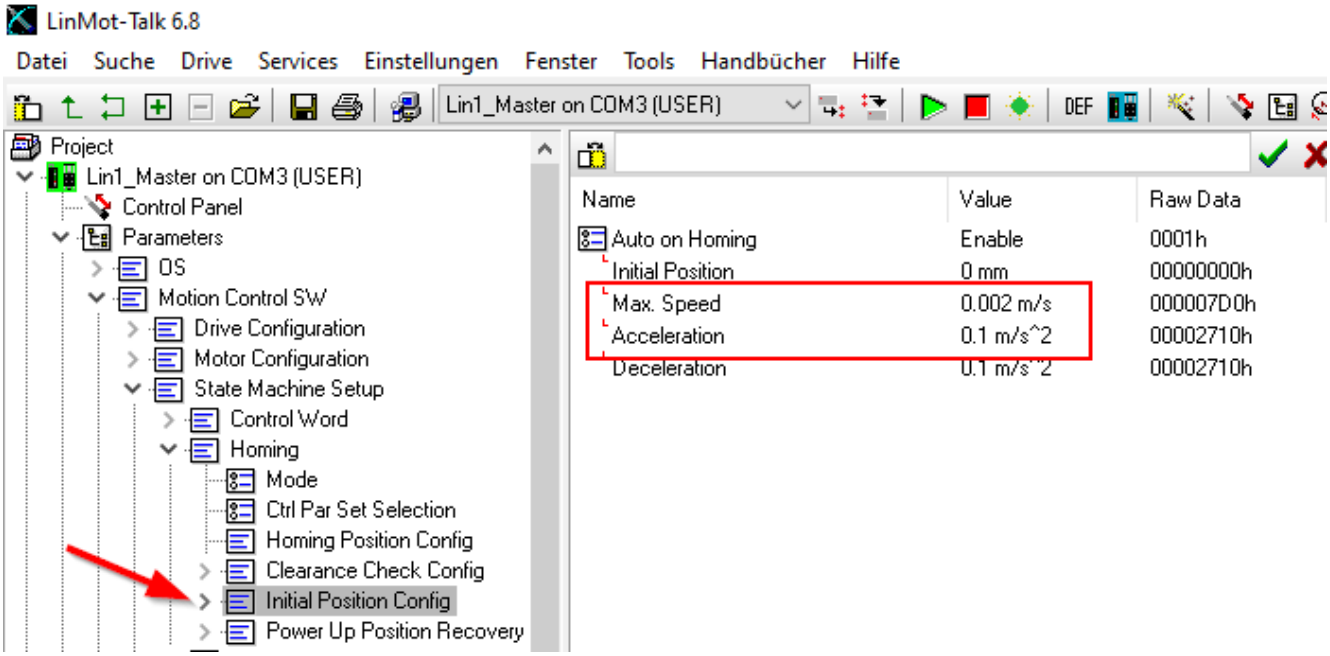
There are some conditions that must be fulfilled:

- The axes must not be rigidly coupled mechanically as they are both position-controlled. They must have some mechanical play, or they will otherwise heat up.
- The Going to Initial position move must be setup slower than the reference (home) search move.

10.1 Drive Settings

10.1.1 Decrease Go to Initial Position Speed and Acceleration

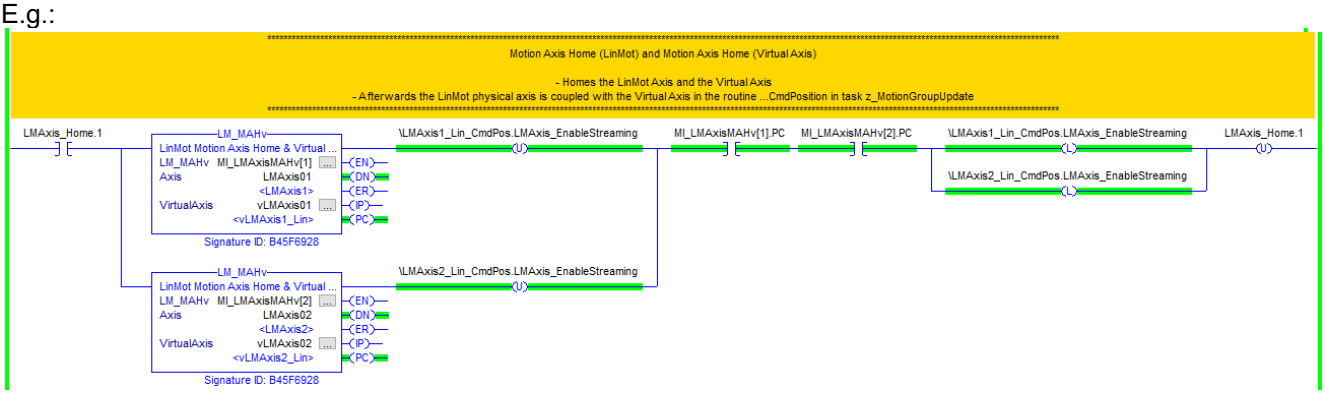
Make sure that the *Max. Speed* and *Acceleration* values are lower than in the *Homing Position Config*:



10.2 Controller Logic

10.2.1 Homing

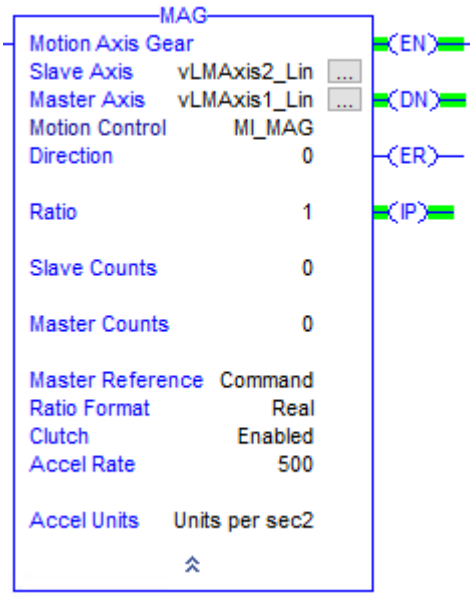
The homing of both axes must be initiated at the same time. Using the LM_MAHv instruction the homing routine defined in the LinMot drive is initiated. After the homing is done, the physical axes are coupled to the virtual axes.
a
E.g.:



10.2.2 Gear

The slave axis is coupled to the master axis using the Motion Axis Gear (MAG) instruction.

E.g.:



Notes

Contact & Support

Europe / Asia Headquarters

NTI AG - LinMot & MagSpring
Bodenaeckerstrasse 2
CH-8957 Spreitenbach
Switzerland

Sales / Administration: +41 56 419 91 91
office@linmot.com

Tech. Support: +41 56 544 71 00
support@linmot.com

+49 201 858940 10
support@linmot.de

Web: https://www.linmot.com

North / South America Headquarters

LinMot USA Inc.
N1922 State Road 120, Unit 1
Lake Geneva, WI 53147
USA

Sales / Administration: 262.743.2555
usasales@linmot.com

Tech. Support: 262.743.2555
usasupport@linmot.com

Web: https://www.linmot-usa.com

Visit https://www.linmot.com/contact to find a distributor next to you.

Smart solutions are…