ctrlX - CORE

- CNC Softmotion V00

rexroth
A Bosch Company

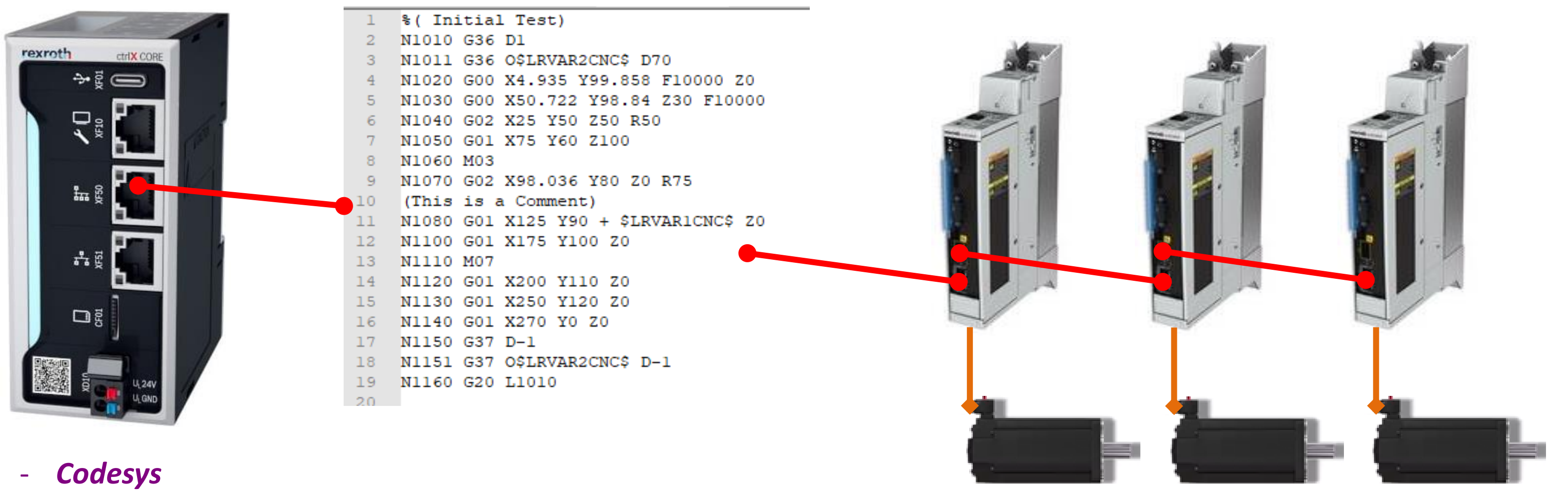
Jordi Laboria (DCET/SLF4-ES)
jordi.laboria@boschrexroth.es
646071813

In the manual:

- *Installation “Package SoftMotion + ctrlX PLC Adaption*
- *SoftMotion Libraries*
- *Activate SoftMotion in a Real axis*
- *Example of program with three virtual axes*
- *Machine program control from file on PC or ctrlX*
- *M functions*
- *CNC variables (in Online Program)*
- *"G" Command Table, Identifiers, Expressions, Functions*
- *Program jumps with G20*
- *Display of G code lines*
- *Licenses*
- *Sending external files to ctrlX*
- *StartUp of Parameters in Real axes*
- *Read / Write Modules EtherCat Parameters*
- *Task assignment*

ctrlX Core

CNC Program



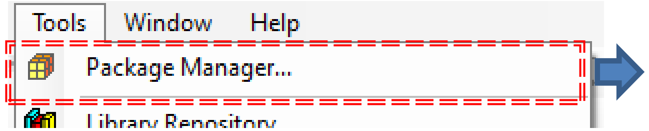
```
%( Initial Test)
N1010 G36 D1
N1011 G36 O$LRVAR2CNC$ D70
N1020 G00 X4.935 Y99.858 F10000 Z0
N1030 G00 X50.722 Y98.84 Z30 F10000
N1040 G02 X25 Y50 Z50 R50
N1050 G01 X75 Y60 Z100
N1060 M03
N1070 G02 X98.036 Y80 Z0 R75
(This is a Comment)
N1080 G01 X125 Y90 + $LRVAR1CNC$ Z0
N1100 G01 X175 Y100 Z0
N1110 M07
N1120 G01 X200 Y110 Z0
N1130 G01 X250 Y120 Z0
N1140 G01 X270 Y0 Z0
N1150 G37 D-1
N1151 G37 O$LRVAR2CNC$ D-1
N1160 G20 L1010
```

- *Codesys*
- *SoftMotion*
- *SoftMotion CNC*

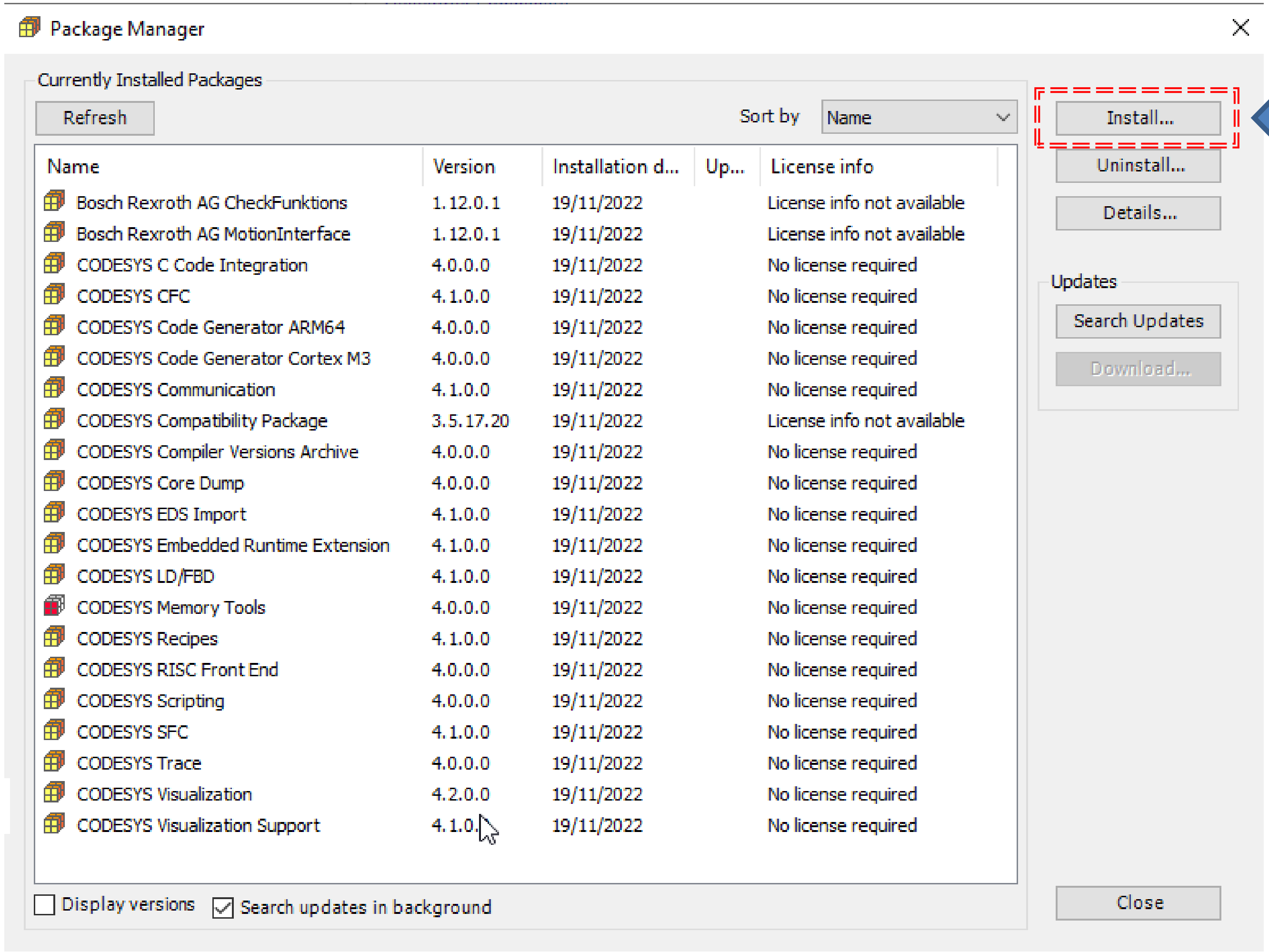
Installation “Softmotion Package”

ctrlX - *Installation of the “Softmotion Package”*

- *To access the installation option of the new "Package" we must enter "Tools" and select the "Package Manager" tab, after which we will be shown the list of "Packages" that we have installed*



The first time we enter the Package Manager, the functional packages currently installed appear.



Name	Version	Installation d...	Up...	License info
Bosch Rexroth AG CheckFunktions	1.12.0.1	19/11/2022		License info not available
Bosch Rexroth AG MotionInterface	1.12.0.1	19/11/2022		License info not available
CODESYS C Code Integration	4.0.0.0	19/11/2022		No license required
CODESYS CFC	4.1.0.0	19/11/2022		No license required
CODESYS Code Generator ARM64	4.0.0.0	19/11/2022		No license required
CODESYS Code Generator Cortex M3	4.0.0.0	19/11/2022		No license required
CODESYS Communication	4.1.0.0	19/11/2022		No license required
CODESYS Compatibility Package	3.5.17.20	19/11/2022		License info not available
CODESYS Compiler Versions Archive	4.0.0.0	19/11/2022		No license required
CODESYS Core Dump	4.0.0.0	19/11/2022		No license required
CODESYS EDS Import	4.1.0.0	19/11/2022		No license required
CODESYS Embedded Runtime Extension	4.1.0.0	19/11/2022		No license required
CODESYS LD/FBD	4.1.0.0	19/11/2022		No license required
CODESYS Memory Tools	4.0.0.0	19/11/2022		No license required
CODESYS Recipes	4.1.0.0	19/11/2022		No license required
CODESYS RISC Front End	4.0.0.0	19/11/2022		No license required
CODESYS Scripting	4.0.0.0	19/11/2022		No license required
CODESYS SFC	4.1.0.0	19/11/2022		No license required
CODESYS Trace	4.0.0.0	19/11/2022		No license required
CODESYS Visualization	4.2.0.0	19/11/2022		No license required
CODESYS Visualization Support	4.1.0.	19/11/2022		No license required

☐ Display versions ☑ Search updates in background

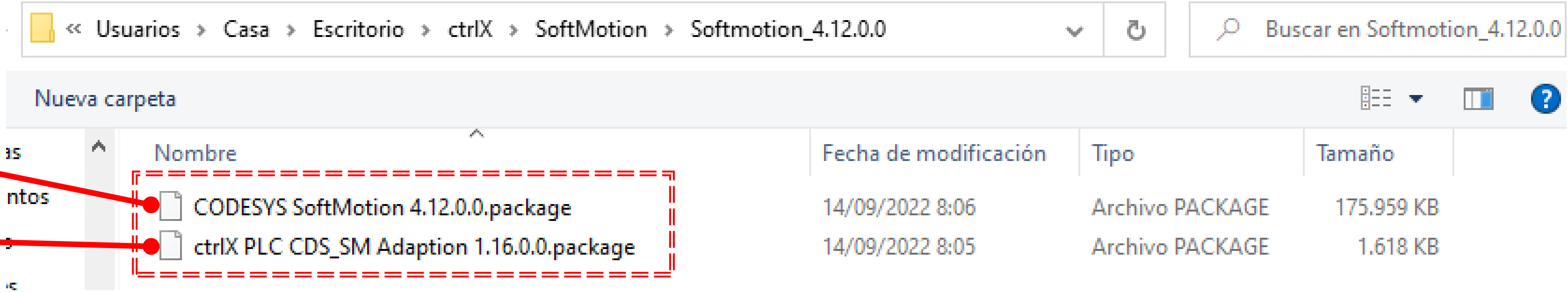
Then we must activate "Install" and look for the file path of the new "Package"

The install or uninstall process requires that all instances of the program be closed.

Package Standards

ctrlX - *Installation of the “Softmotion Package”*

- *In our case, two files should appear, the first from CodeSys SoftMotion and the second, ours, which adapts the system to SoftMotion*



« Usuarios › Casa › Escritorio › ctrlX › SoftMotion › Softmotion_4.12.0.0

Buscar en Softmotion_4.12.0.0

Nueva carpeta

Nombre	Fecha de modificación	Tipo	Tamaño
CODESYS SoftMotion 4.12.0.0.package	14/09/2022 8:06	Archivo PACKAGE	175.959 KB
ctrlX PLC CDS_SM Adaption 1.16.0.0.package	14/09/2022 8:05	Archivo PACKAGE	1.618 KB

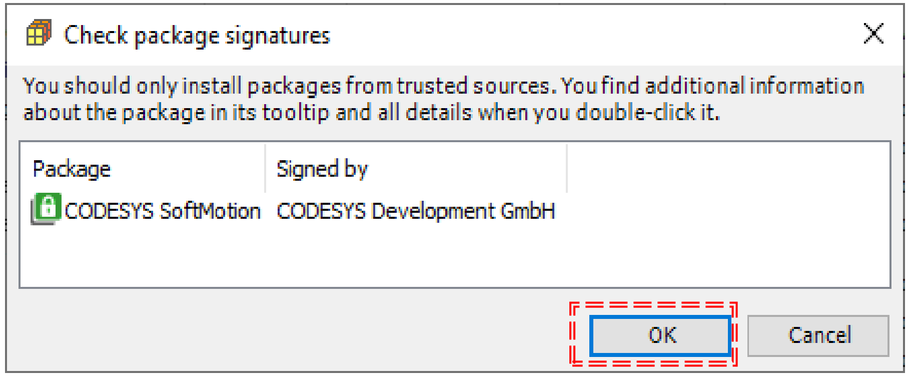
Package SoftMotion

Adaptation ctrlX PLC

I have tried to install them without following a specific order and I have apparently had no problems. However, it would be preferable to install first the "CodeSys SoftMotion" and then the "Adaptation of ctrlX PLC"

- *Installation of the SoftMotion Package*

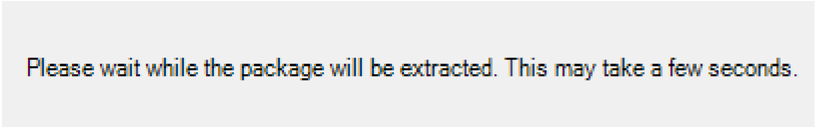
After selecting the file indicated above we must follow the following steps:



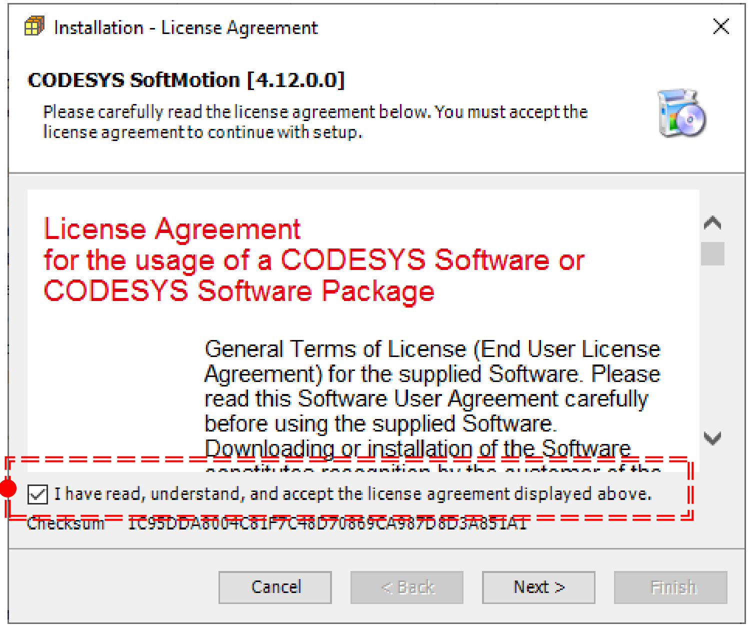
Check package signatures

You should only install packages from trusted sources. You find additional information about the package in its tooltip and all details when you double-click it.

Package	Signed by
CODESYS SoftMotion	CODESYS Development GmbH



Please wait while the package will be extracted. This may take a few seconds.



Installation - License Agreement

CODESYS SoftMotion [4.12.0.0]

Please carefully read the license agreement below. You must accept the license agreement to continue with setup.

License Agreement
for the usage of a CODESYS Software or CODESYS Software Package

General Terms of License (End User License Agreement) for the supplied Software. Please read this Software User Agreement carefully before using the supplied Software. Downloading or installation of the Software constitutes recognition by the customer of the

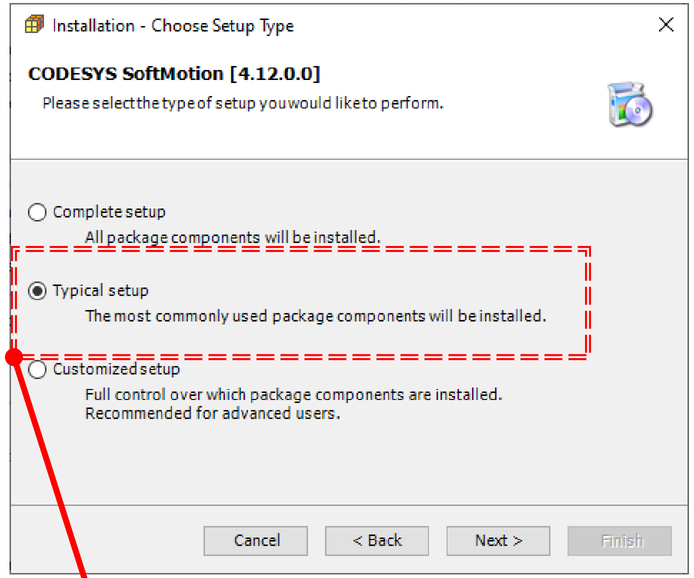
☑ I have read, understand, and accept the license agreement displayed above.

Checksum 1C95DDA8004C81F7C48D70869CA987D8D3A851A1

Enable tab and activate “Next”

ctrlX - *Installation of the "Softmotion Package"*

- *Then we go through these steps*



Installation - Choose Setup Type

CODESYS SoftMotion [4.12.0.0]

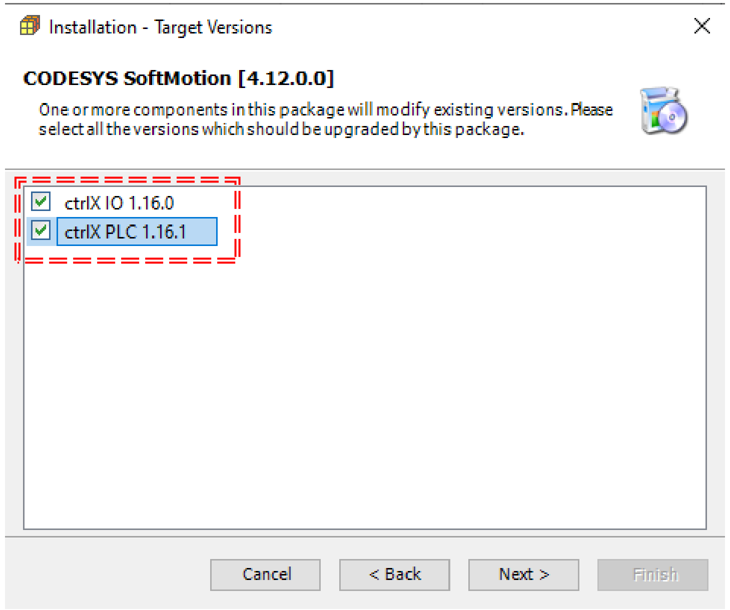
Please select the type of setup you would like to perform.

○ Complete setup
All package components will be installed.

● Typical setup
The most commonly used package components will be installed.

○ Customized setup
Full control over which package components are installed.
Recommended for advanced users.

With the "Typical Setup" should be enough.



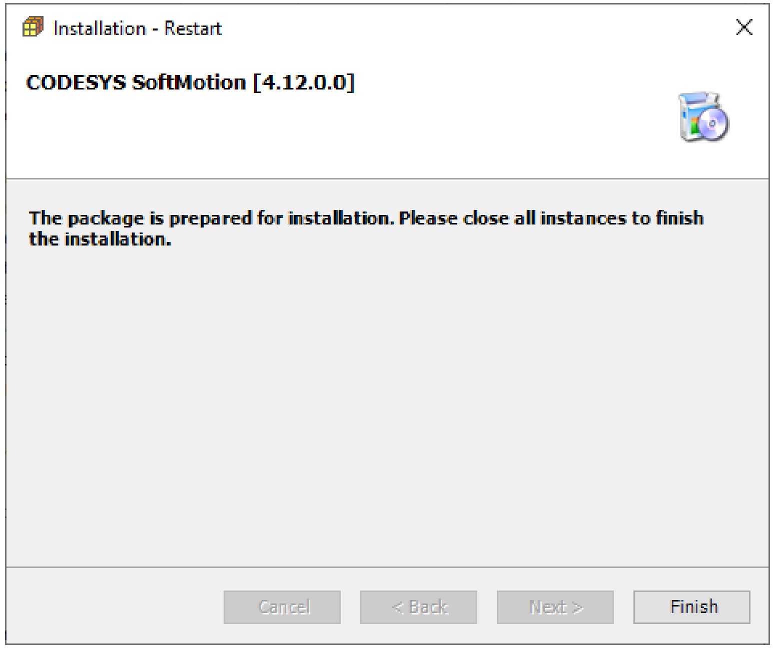
Installation - Target Versions

CODESYS SoftMotion [4.12.0.0]

One or more components in this package will modify existing versions. Please select all the versions which should be upgraded by this package.

☑ ctrlX IO 1.16.0
☑ ctrlX PLC 1.16.1

The installation of the "Package" SoftMotion warns us that it will make modifications in the following components



Installation - Restart

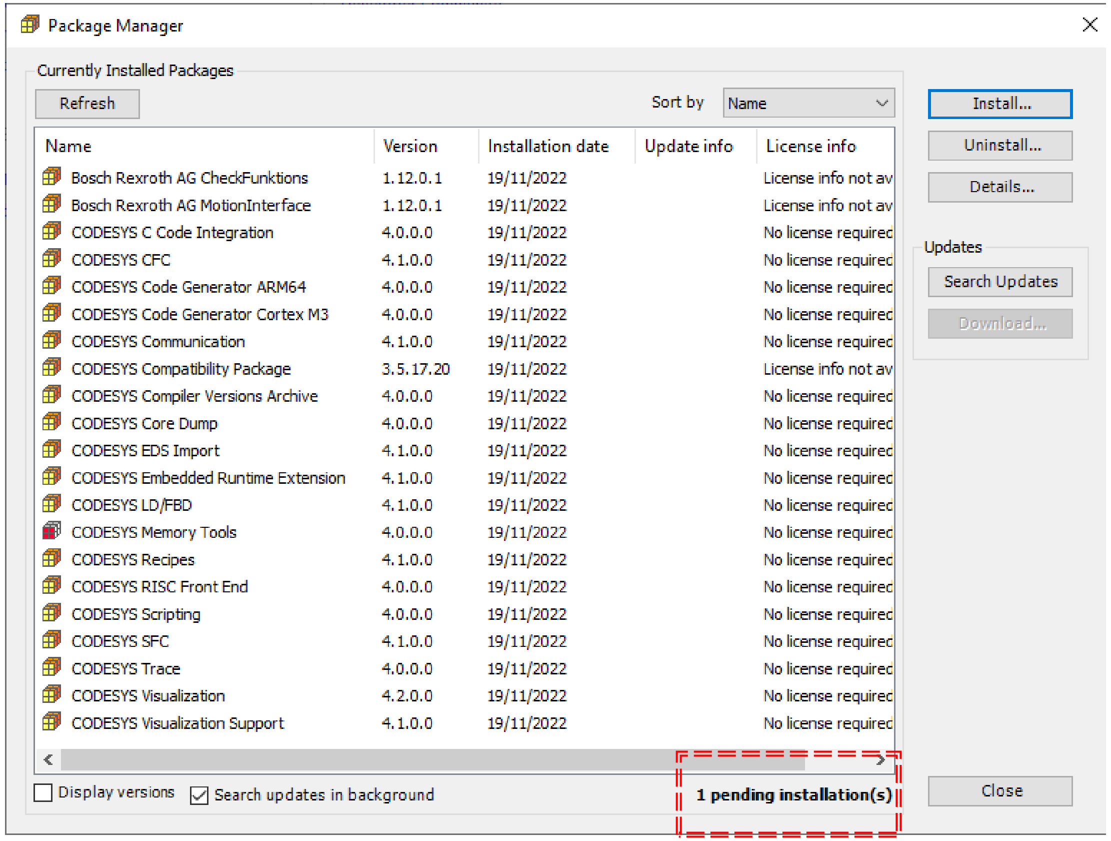
CODESYS SoftMotion [4.12.0.0]

The package is prepared for installation. Please close all instances to finish the installation.

Installation ready, we continue with "Finish"

ctrlX - *Installation of the “Softmotion Package”*

- ***The installation is not carried out automatically since, as we can see, it is signaled as an object pending installation.***



Package Manager

Currently Installed Packages

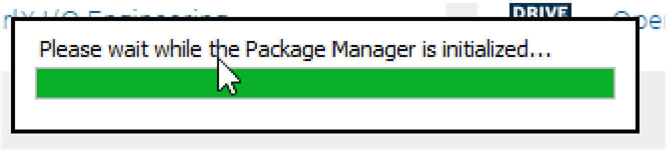
Refresh

Sort by Name

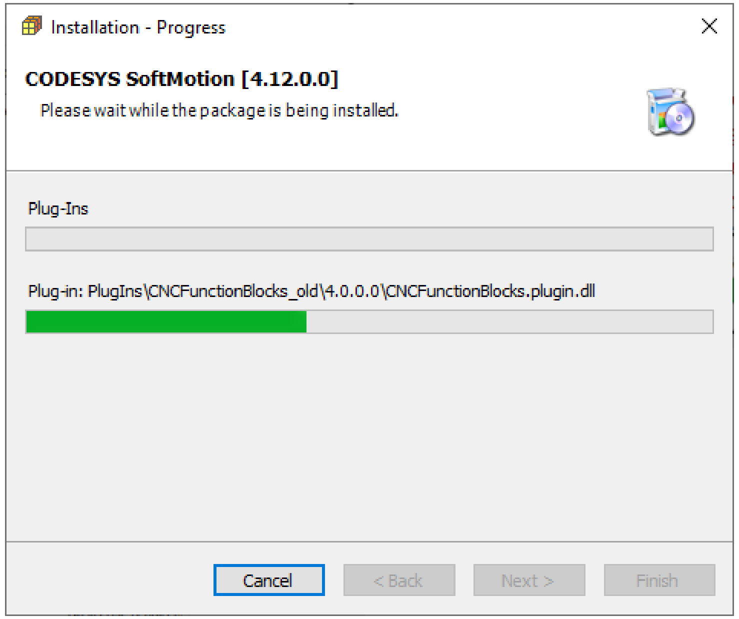
Name	Version	Installation date	Update info	License info
Bosch Rexroth AG CheckFunktions	1.12.0.1	19/11/2022		License info not av
Bosch Rexroth AG MotionInterface	1.12.0.1	19/11/2022		License info not av
CODESYS C Code Integration	4.0.0.0	19/11/2022		No license requirec
CODESYS CFC	4.1.0.0	19/11/2022		No license requirec
CODESYS Code Generator ARM64	4.0.0.0	19/11/2022		No license requirec
CODESYS Code Generator Cortex M3	4.0.0.0	19/11/2022		No license requirec
CODESYS Communication	4.1.0.0	19/11/2022		No license requirec
CODESYS Compatibility Package	3.5.17.20	19/11/2022		License info not av
CODESYS Compiler Versions Archive	4.0.0.0	19/11/2022		No license requirec
CODESYS Core Dump	4.0.0.0	19/11/2022		No license requirec
CODESYS EDS Import	4.1.0.0	19/11/2022		No license requirec
CODESYS Embedded Runtime Extension	4.1.0.0	19/11/2022		No license requirec
CODESYS LD/FBD	4.1.0.0	19/11/2022		No license requirec
CODESYS Memory Tools	4.0.0.0	19/11/2022		No license requirec
CODESYS Recipes	4.1.0.0	19/11/2022		No license requirec
CODESYS RISC Front End	4.0.0.0	19/11/2022		No license requirec
CODESYS Scripting	4.0.0.0	19/11/2022		No license requirec
CODESYS SFC	4.1.0.0	19/11/2022		No license requirec
CODESYS Trace	4.0.0.0	19/11/2022		No license requirec
CODESYS Visualization	4.2.0.0	19/11/2022		No license requirec
CODESYS Visualization Support	4.1.0.0	19/11/2022		No license requirec

□ Display versions ☑ Search updates in background **1 pending installation(s)**

For the installation to activate, we must close the “ctrlX PLC Enginnering”. On some occasions the installation is activated when it is closed, at other times it is activated when the PLC program is reopened



Please wait while the Package Manager is initialized...



Installation - Progress

CODESYS SoftMotion [4.12.0.0]

Please wait while the package is being installed.

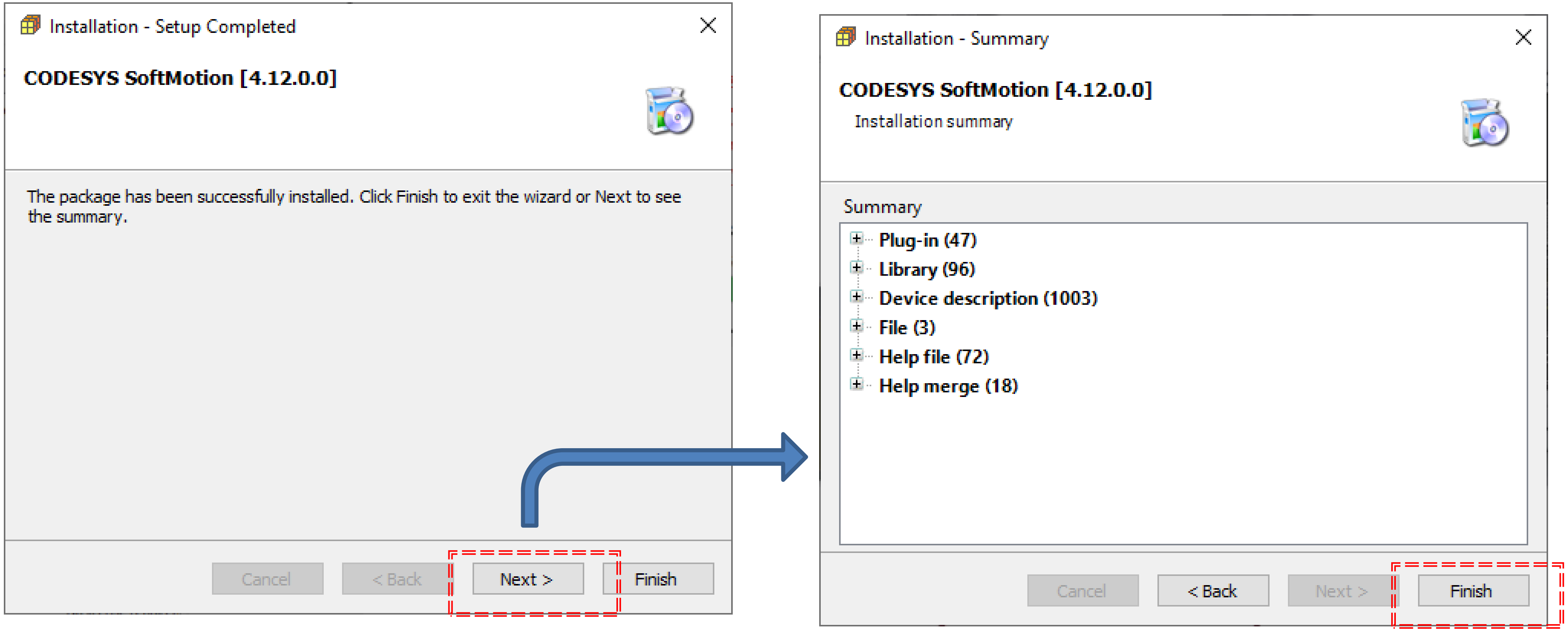
Plug-Ins

Plug-in: PlugIns\CNCFunctionBlocks_old\4.0.0.0\CNCFunctionBlocks.plugin.dll

Installation process. This can last a few seconds or minutes.

ctrlX - *Installation of the "Softmotion Package"*

- *Once the installation is finished we can consider it finished or with "Next" access the summary of the installed elements*



Installation - Setup Completed

CODESYS SoftMotion [4.12.0.0]

The package has been successfully installed. Click Finish to exit the wizard or Next to see the summary.

Installation - Summary

CODESYS SoftMotion [4.12.0.0]

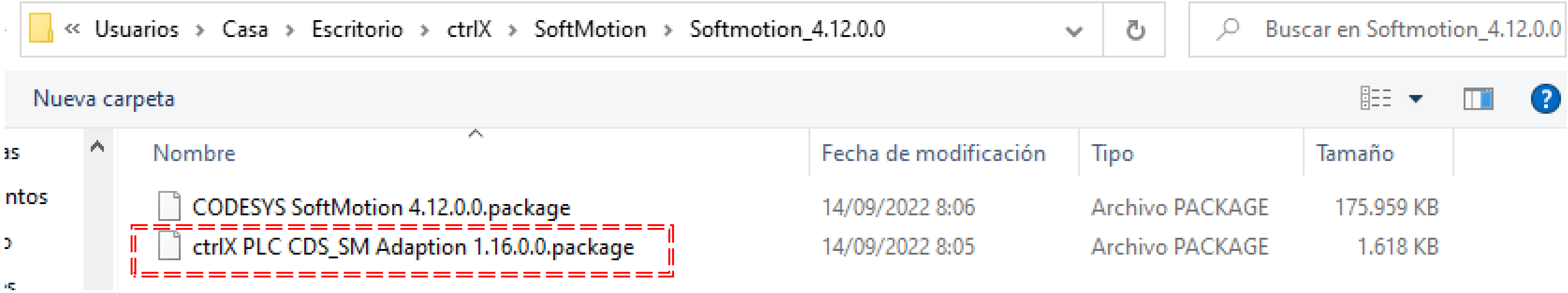
Installation summary

Summary

- **Plug-in (47)**
- **Library (96)**
- **Device description (1003)**
- **File (3)**
- **Help file (72)**
- **Help merge (18)**

ctrlX- *Installation of "ctrlX PLC Adaption"*

- *Next we will proceed to the installation of ctrlX PLC Adaption (1):*

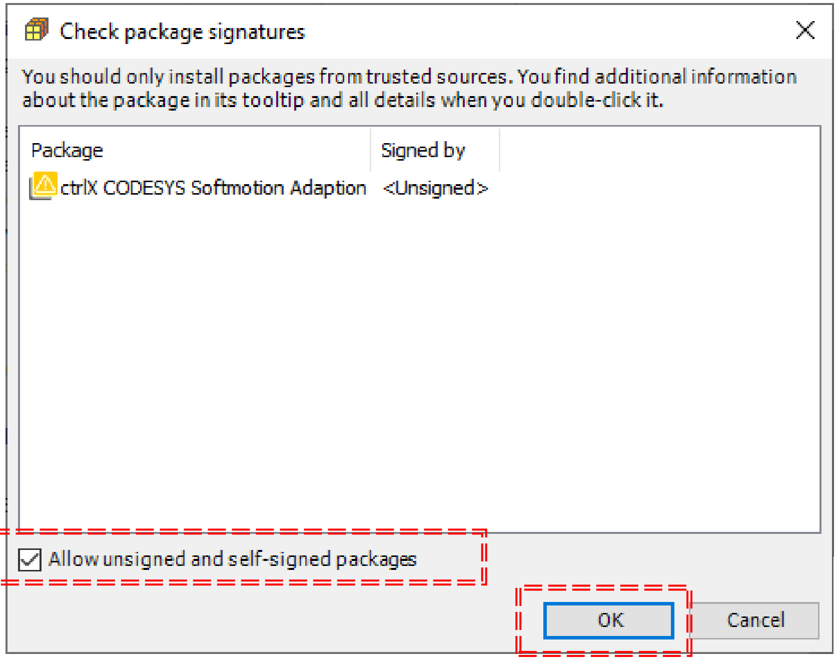


« Usuarios › Casa › Escritorio › ctrlX › SoftMotion › Softmotion_4.12.0.0

Buscar en Softmotion_4.12.0.0

Nueva carpeta

Nombre	Fecha de modificación	Tipo	Tamaño
CODESYS SoftMotion 4.12.0.0.package	14/09/2022 8:06	Archivo PACKAGE	175.959 KB
ctrlX PLC CDS_SM Adaption 1.16.0.0.package	14/09/2022 8:05	Archivo PACKAGE	1.618 KB



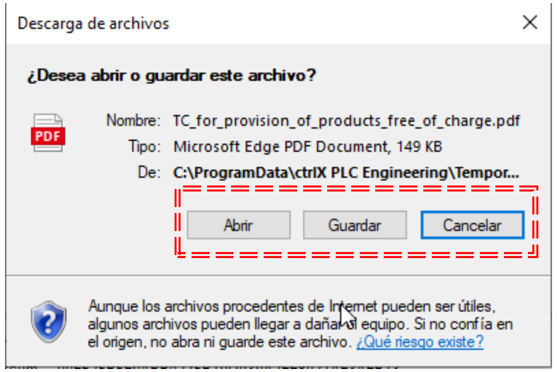
Check package signatures

You should only install packages from trusted sources. You find additional information about the package in its tooltip and all details when you double-click it.

Package	Signed by
ctrlX CODESYS Softmotion Adaption	<Unsigned>

☑ Allow unsigned and self-signed packages

Enable the tab and press Ok



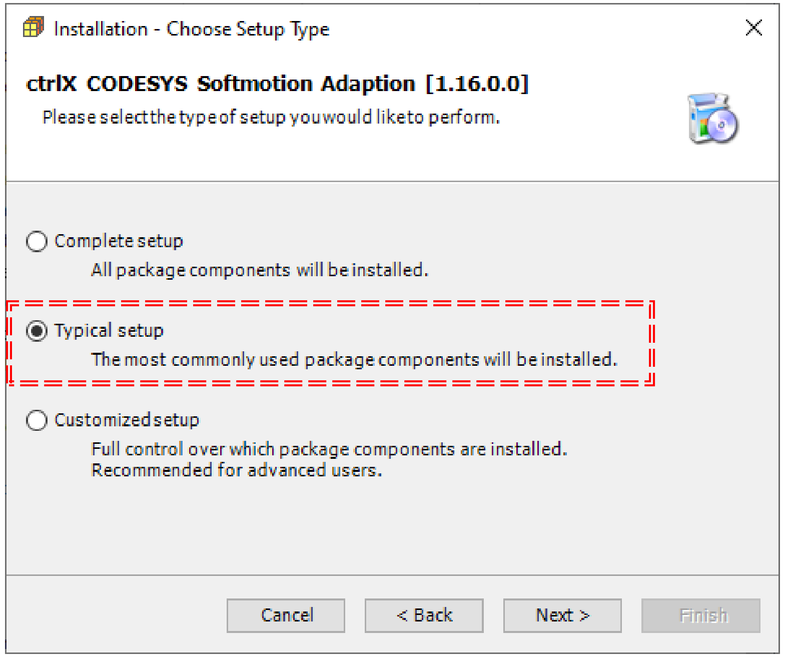
Descarga de archivos

¿Desea abrir o guardar este archivo?

Nombre: TC_for_provision_of_products_free_of_charge.pdf
Tipo: Microsoft Edge PDF Document, 149 KB
De: C:\ProgramData\ctrlX PLC Engineering\Tempor...

Aunque los archivos procedentes de Internet pueden ser útiles, algunos archivos pueden llegar a dañar el equipo. Si no confía en el origen, no abra ni guarde este archivo. ¿Qué riesgo existe?

We can use any of the three options.



Installation - Choose Setup Type

ctrlX CODESYS Softmotion Adaption [1.16.0.0]

Please select the type of setup you would like to perform.

○ Complete setup
All package components will be installed.

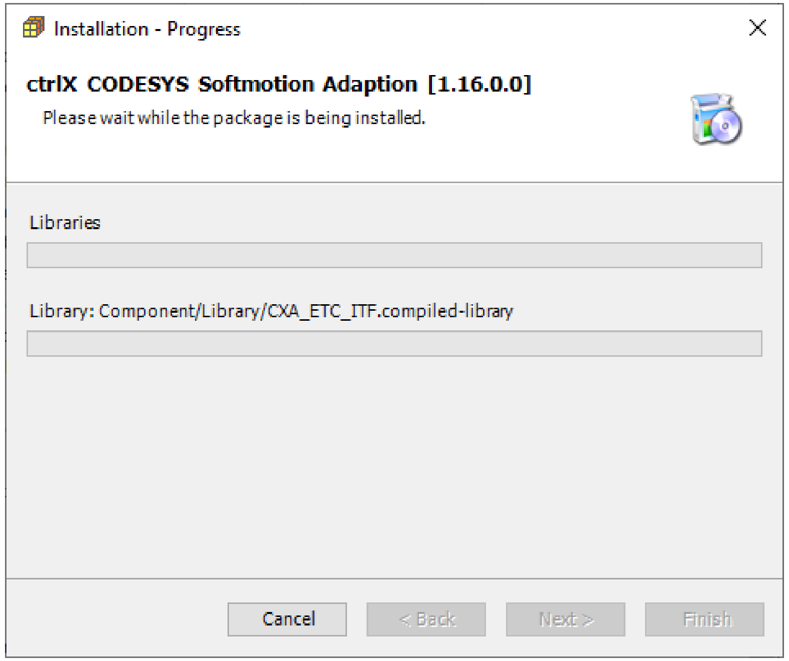
● Typical setup
The most commonly used package components will be installed.

○ Customized setup
Full control over which package components are installed. Recommended for advanced users.

Select "Typical Setup" and "Next"

ctrlX- Installation of “ctrlX PLC Adaption”

- *Next we will proceed to the installation of ctrlX PLC Adaption (2)*



Installation - Progress

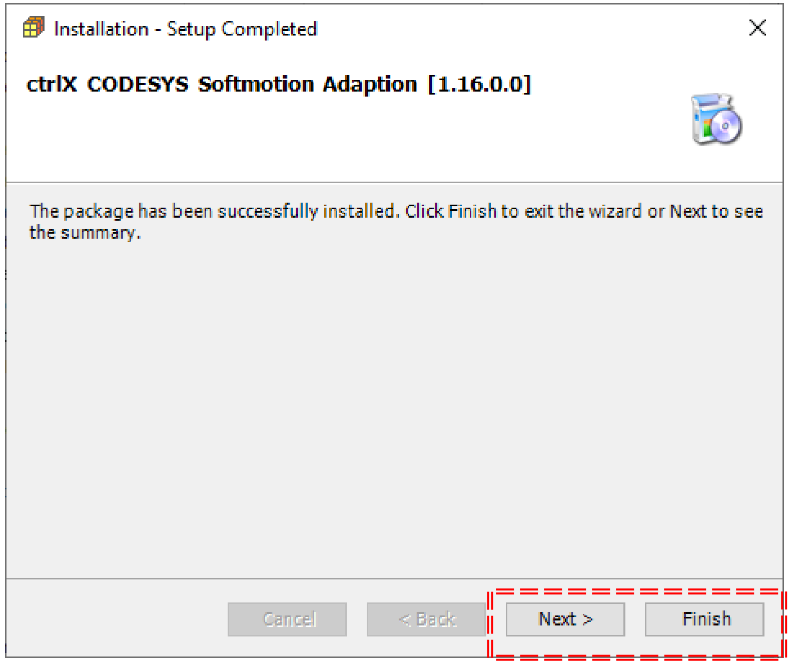
ctrlX CODESYS Softmotion Adaption [1.16.0.0]

Please wait while the package is being installed.

Libraries

Library: Component/Library/CXA_ETC_ITF.compiled-library

Installation process. This can last a few seconds or minutes.

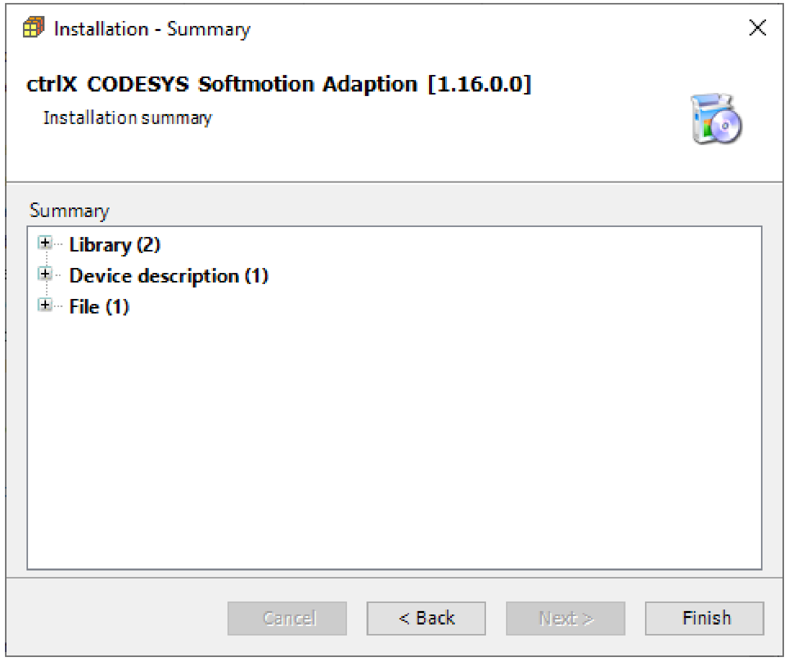


Installation - Setup Completed

ctrlX CODESYS Softmotion Adaption [1.16.0.0]

The package has been successfully installed. Click Finish to exit the wizard or Next to see the summary.

Installation process. This can last a few seconds or minutes.



Installation - Summary

ctrlX CODESYS Softmotion Adaption [1.16.0.0]

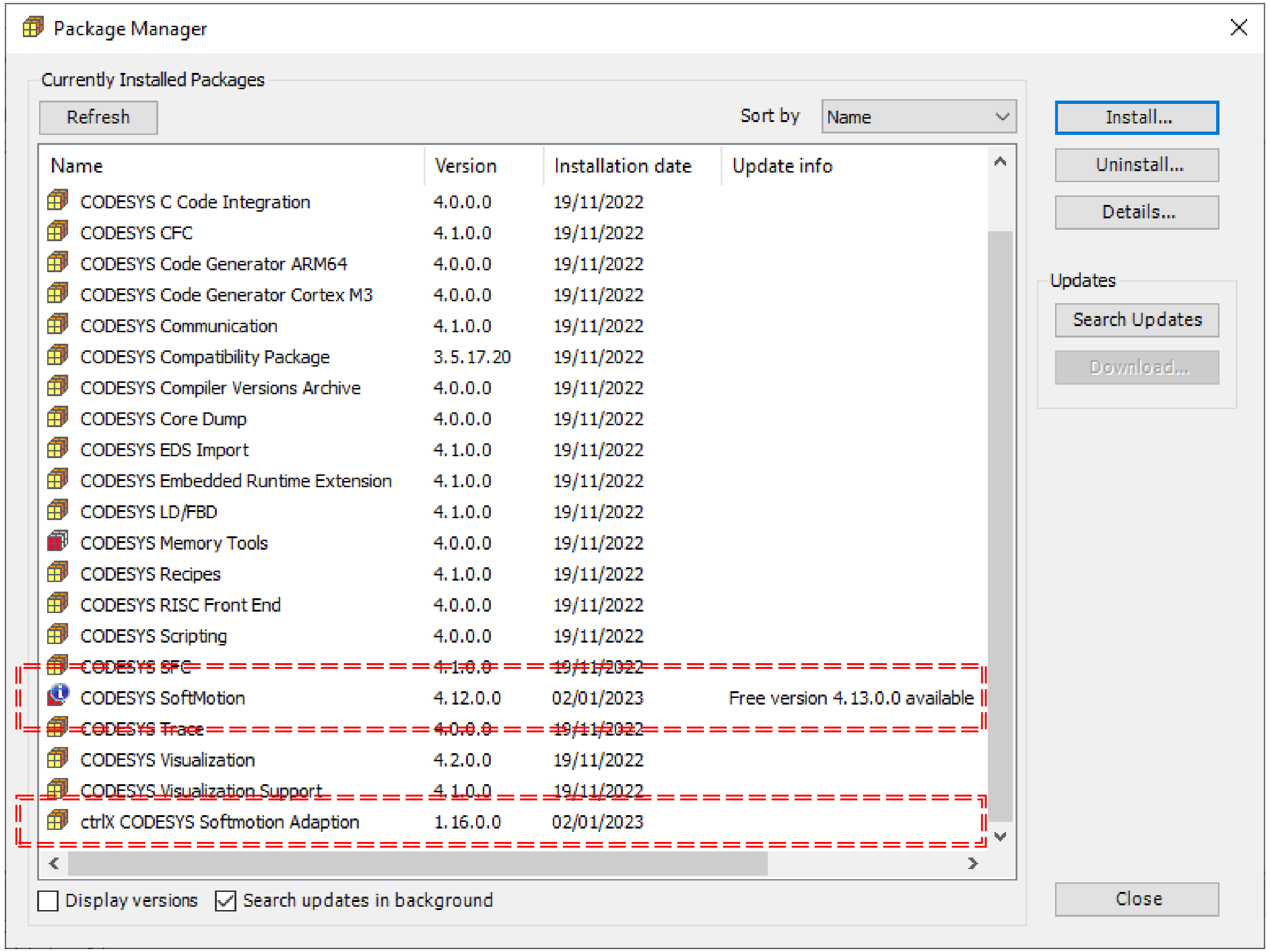
Installation summary

Summary

- Library (2)
- Device description (1)
- File (1)

ctrlX - *Visualization of the Packages*

- ***Overview of the installed SoftMotion and Adaption Packages:***



Package Manager

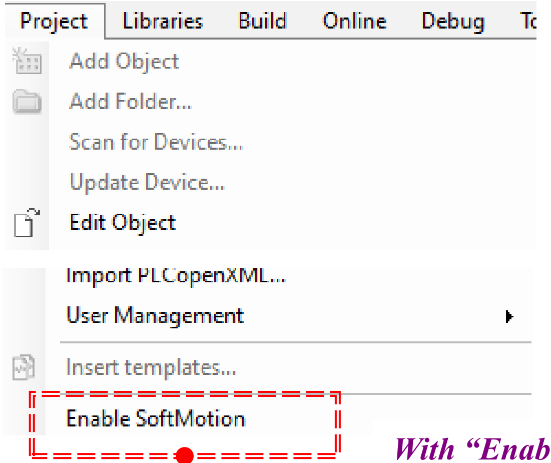
Currently Installed Packages

Name	Version	Installation date	Update info
CODESYS C Code Integration	4.0.0.0	19/11/2022	
CODESYS CFC	4.1.0.0	19/11/2022	
CODESYS Code Generator ARM64	4.0.0.0	19/11/2022	
CODESYS Code Generator Cortex M3	4.0.0.0	19/11/2022	
CODESYS Communication	4.1.0.0	19/11/2022	
CODESYS Compatibility Package	3.5.17.20	19/11/2022	
CODESYS Compiler Versions Archive	4.0.0.0	19/11/2022	
CODESYS Core Dump	4.0.0.0	19/11/2022	
CODESYS EDS Import	4.1.0.0	19/11/2022	
CODESYS Embedded Runtime Extension	4.1.0.0	19/11/2022	
CODESYS LD/FBD	4.1.0.0	19/11/2022	
CODESYS Memory Tools	4.0.0.0	19/11/2022	
CODESYS Recipes	4.1.0.0	19/11/2022	
CODESYS RISC Front End	4.0.0.0	19/11/2022	
CODESYS Scripting	4.0.0.0	19/11/2022	
CODESYS SFC	4.1.0.0	19/11/2022	
CODESYS SoftMotion	4.12.0.0	02/01/2023	Free version 4.13.0.0 available
CODESYS Trace	4.0.0.0	19/11/2022	
CODESYS Visualization	4.2.0.0	19/11/2022	
CODESYS Visualization Support	4.1.0.0	19/11/2022	
ctrlX CODESYS Softmotion Adaption	1.16.0.0	02/01/2023	

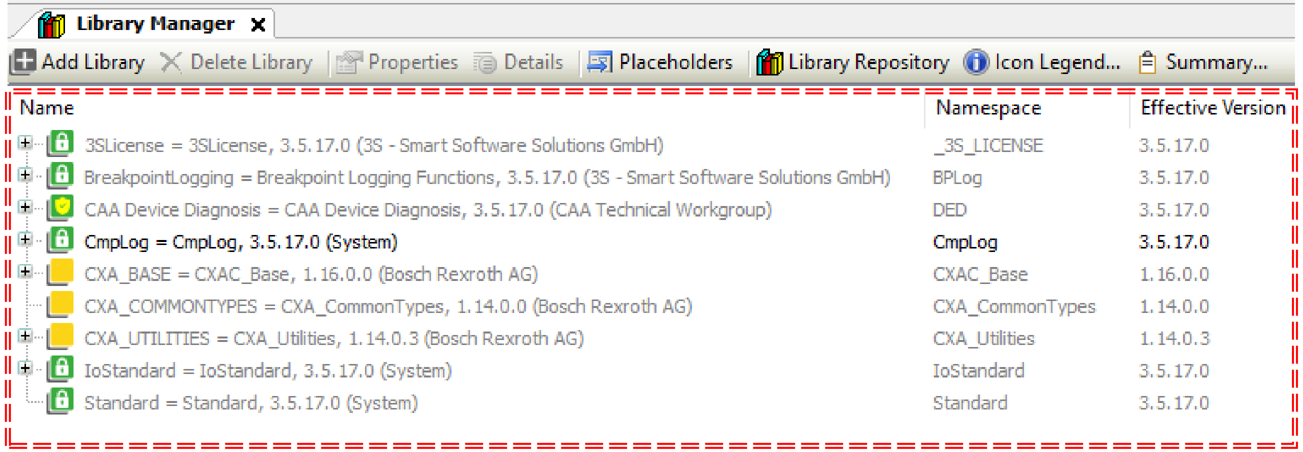
☐ Display versions ☑ Search updates in background

ctrlX - *Visualization of the Packages*

- *As long as the SoftMotion part is not activated, the libraries will not appear:*

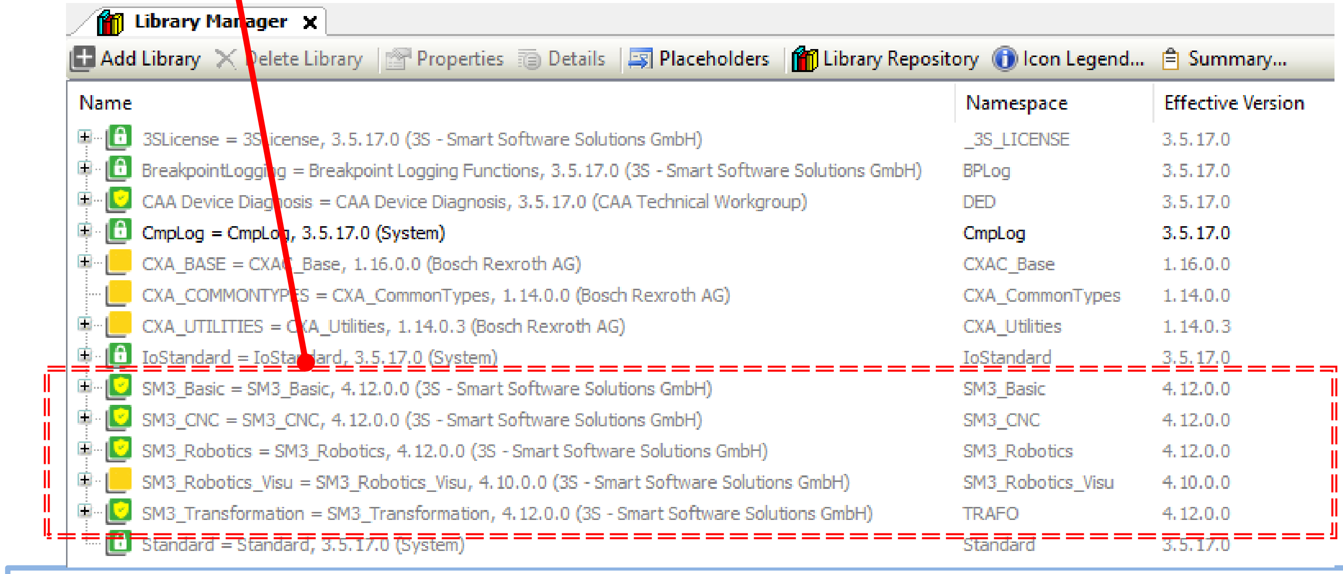


Without "Enable SoftMotion" and by default these libraries appear.



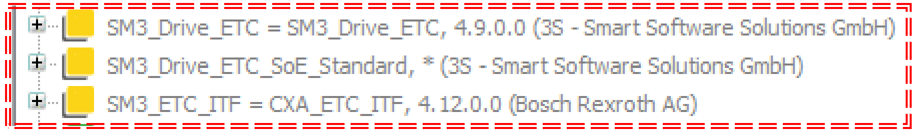
Name	Namespace	Effective Version
3SLicense = 3SLicense, 3.5.17.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.17.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.17.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.17.0
CAA Device Diagnosis = CAA Device Diagnosis, 3.5.17.0 (CAA Technical Workgroup)	DED	3.5.17.0
CmpLog = CmpLog, 3.5.17.0 (System)	CmpLog	3.5.17.0
CXA_BASE = CXAC_Base, 1.16.0.0 (Bosch Rexroth AG)	CXAC_Base	1.16.0.0
CXA_COMMONTYPES = CXA_CommonTypes, 1.14.0.0 (Bosch Rexroth AG)	CXA_CommonTypes	1.14.0.0
CXA_UTILITIES = CXA_Utilities, 1.14.0.3 (Bosch Rexroth AG)	CXA_Utilities	1.14.0.3
IoStandard = IoStandard, 3.5.17.0 (System)	IoStandard	3.5.17.0
Standard = Standard, 3.5.17.0 (System)	Standard	3.5.17.0

With "Enable SoftMotion" these libraries are added



Name	Namespace	Effective Version
3SLicense = 3SLicense, 3.5.17.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.17.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.17.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.17.0
CAA Device Diagnosis = CAA Device Diagnosis, 3.5.17.0 (CAA Technical Workgroup)	DED	3.5.17.0
CmpLog = CmpLog, 3.5.17.0 (System)	CmpLog	3.5.17.0
CXA_BASE = CXAC_Base, 1.16.0.0 (Bosch Rexroth AG)	CXAC_Base	1.16.0.0
CXA_COMMONTYPES = CXA_CommonTypes, 1.14.0.0 (Bosch Rexroth AG)	CXA_CommonTypes	1.14.0.0
CXA_UTILITIES = CXA_Utilities, 1.14.0.3 (Bosch Rexroth AG)	CXA_Utilities	1.14.0.3
IoStandard = IoStandard, 3.5.17.0 (System)	IoStandard	3.5.17.0
SM3_Basic = SM3_Basic, 4.12.0.0 (3S - Smart Software Solutions GmbH)	SM3_Basic	4.12.0.0
SM3_CNC = SM3_CNC, 4.12.0.0 (3S - Smart Software Solutions GmbH)	SM3_CNC	4.12.0.0
SM3_Robotics = SM3_Robotics, 4.12.0.0 (3S - Smart Software Solutions GmbH)	SM3_Robotics	4.12.0.0
SM3_Robotics_Visu = SM3_Robotics_Visu, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_Robotics_Visu	4.10.0.0
SM3_Transformation = SM3_Transformation, 4.12.0.0 (3S - Smart Software Solutions GmbH)	TRAFO	4.12.0.0
Standard = Standard, 3.5.17.0 (System)	Standard	3.5.17.0

With "Enable SoftMotion" and some real axis, active for SoftMotion in the system, these three libraries are added

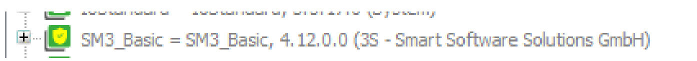


SM3_Drive_ETC = SM3_Drive_ETC, 4.9.0.0 (3S - Smart Software Solutions GmbH)
SM3_Drive_ETC_SoE_Standard, * (3S - Smart Software Solutions GmbH)
SM3_ETC_ITF = CXA_ETC_ITF, 4.12.0.0 (Bosch Rexroth AG)

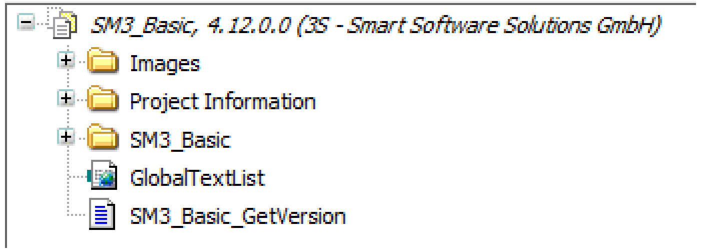
SoftMotion Libraries

ctrlX - *SoftMotion Libraries – SMC_Basic:*

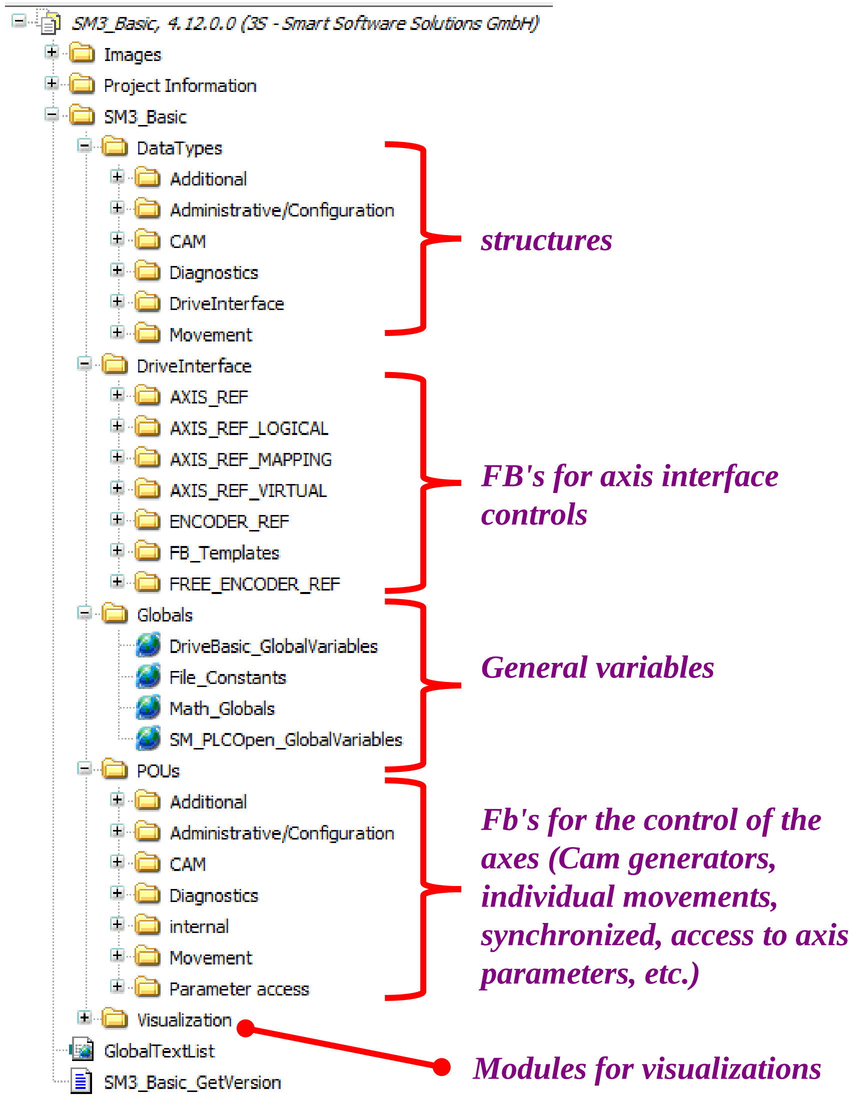
- *SMC_Basic Basic Library:*
- *- This library is the basic one for all SoftMotion applications:*
  - *- PLCOpen function blocks*
  - *- Movement controls for an axis (Positioning / Speed, etc)*
  - *- Master / Slave controls (Cam, cam generator, etc)*
  - *- Additional Fb's (Trigger, TouchProbe, etc)*
  - *- Help functions for handling files or error messages*



SM3_Basic = SM3_Basic, 4.12.0.0 (3S - Smart Software Solutions GmbH)



- *SM3_Basic, 4.12.0.0 (3S - Smart Software Solutions GmbH)*
  - Images
  - Project Information
  - SM3_Basic
  - GlobalTextList
  - SM3_Basic_GetVersion



- *SM3_Basic, 4.12.0.0 (3S - Smart Software Solutions GmbH)*
  - Images
  - Project Information
  - SM3_Basic
    - DataTypes
      - Additional
      - Administrative/Configuration
      - CAM
      - Diagnostics
      - DriveInterface
      - Movement
    - DriveInterface
      - AXIS_REF
      - AXIS_REF_LOGICAL
      - AXIS_REF_MAPPING
      - AXIS_REF_VIRTUAL
      - ENCODER_REF
      - FB_Templates
      - FREE_ENCODER_REF
    - Globals
      - DriveBasic_GlobalVariables
      - File_Constants
      - Math_Globals
      - SM_PLCOpen_GlobalVariables
    - POUs
      - Additional
      - Administrative/Configuration
      - CAM
      - Diagnostics
      - internal
      - Movement
      - Parameter access
    - Visualization
  - GlobalTextList
  - SM3_Basic_GetVersion

structures

FB's for axis interface controls

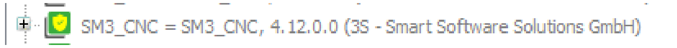
General variables

Fb's for the control of the axes (Cam generators, individual movements, synchronized, access to axis parameters, etc.)

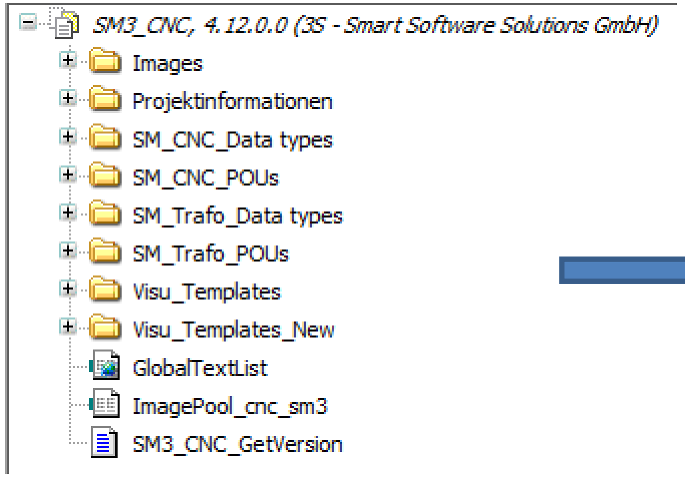
Modules for visualizations

ctrlX - SoftMotion Libraries – SM3_CNC:

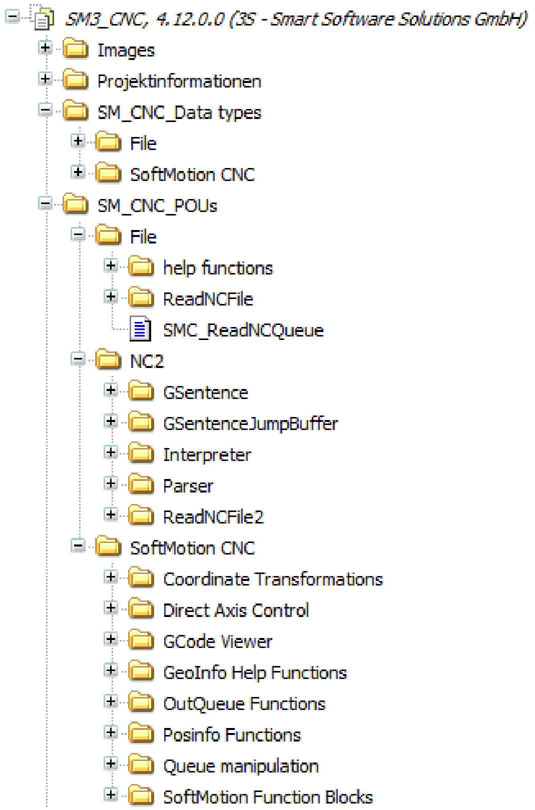
- *SM3_CNC library:*
  - *- This library is based on the SM3_BASIC:*
    - *- Blocks for kinematic transformations*
    - *- Control blocks to generate, execute and display the movements generated by the CNC*
    - *- It also has blocks for preprocessing and reconstruction of the movement path*



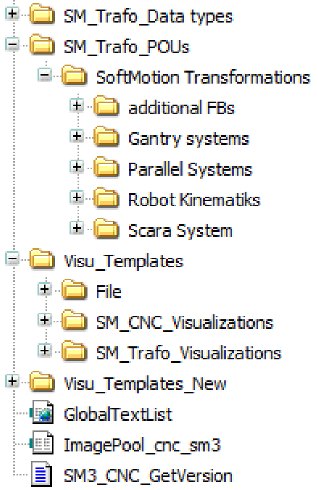
- SM3_CNC = SM3_CNC, 4.12.0.0 (3S - Smart Software Solutions GmbH)



- *SM3_CNC, 4.12.0.0 (3S - Smart Software Solutions GmbH)*
  - Images
  - Projektinformationen
  - SM_CNC_Data types
  - SM_CNC_POUs
  - SM_Trafo_Data types
  - SM_Trafo_POUs
  - Visu_Templates
  - Visu_Templates_New
  - GlobalTextList
  - ImagePool_cnc_sm3
  - SM3_CNC_GetVersion

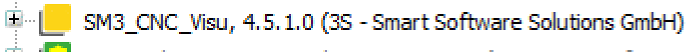


- *SM3_CNC, 4.12.0.0 (3S - Smart Software Solutions GmbH)*
  - Images
  - Projektinformationen
  - SM_CNC_Data types
    - File
    - SoftMotion CNC
  - SM_CNC_POUs
    - File
      - help functions
      - ReadNCFile
      - SMC_ReadNCQueue
    - NC2
      - GSentence
      - GSentenceJumpBuffer
      - Interpreter
      - Parser
      - ReadNCFile2
    - SoftMotion CNC
      - Coordinate Transformations
      - Direct Axis Control
      - GCode Viewer
      - GeoInfo Help Functions
      - OutQueue Functions
      - Posinfo Functions
      - Queue manipulation
      - SoftMotion Function Blocks

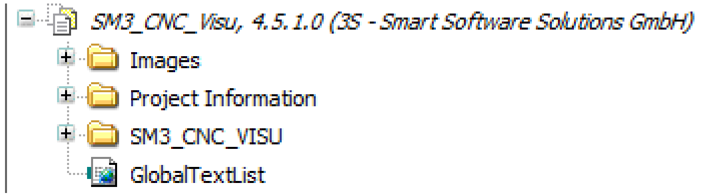


- SM_Trafo_Data types
- SM_Trafo_POUs
  - SoftMotion Transformations
    - additional FBs
    - Gantry systems
    - Parallel Systems
    - Robot Kinematiks
    - Scara System
- Visu_Templates
  - File
  - SM_CNC_Visualizations
  - SM_Trafo_Visualizations
- Visu_Templates_New
- GlobalTextList
- ImagePool_cnc_sm3
- SM3_CNC_GetVersion

Additionally, the SM3_CNC_Visu library is also available for the visualization of elements



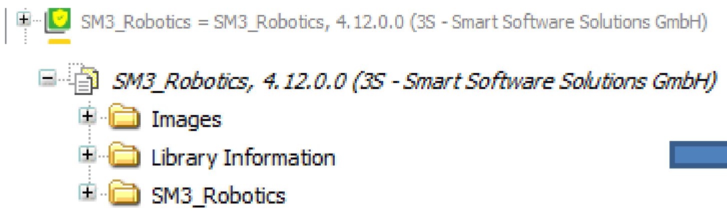
- SM3_CNC_Visu, 4.5.1.0 (3S - Smart Software Solutions GmbH)



- *SM3_CNC_Visu, 4.5.1.0 (3S - Smart Software Solutions GmbH)*
  - Images
  - Project Information
  - SM3_CNC_VISU
  - GlobalTextList

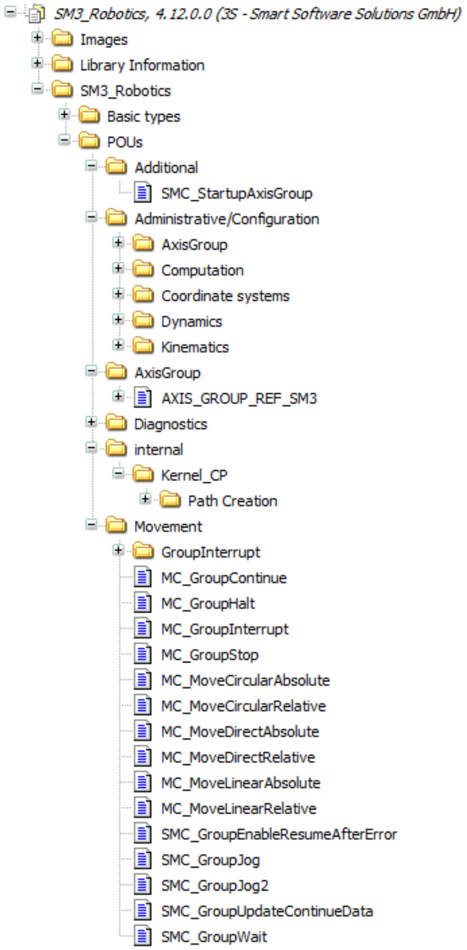
ctrlX - SoftMotion Libraries – SM3_Robotics:

- *SM3_Robotics Library:*
  - *This library contains function blocks according to PLCOpen Part 4 for robotics:*
    - *Additional functionality blocks are included*
    - *And it also has the SM3_Transformations library that contains all the kinematics supported*



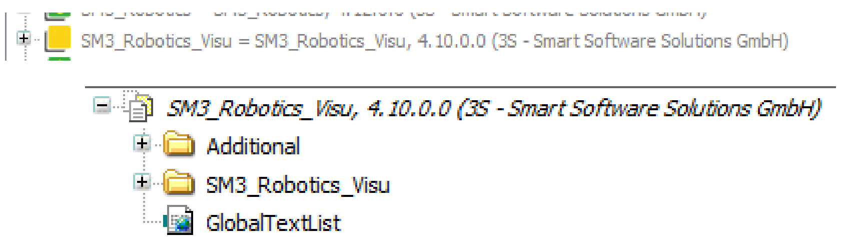
SM3_Robotics = SM3_Robotics, 4.12.0.0 (3S - Smart Software Solutions GmbH)

- *SM3_Robotics, 4.12.0.0 (3S - Smart Software Solutions GmbH)*
  - Images
  - Library Information
  - SM3_Robotics



- *SM3_Robotics, 4.12.0.0 (3S - Smart Software Solutions GmbH)*
  - Images
  - Library Information
  - SM3_Robotics
    - Basic types
    - POUs
      - Additional
        - SMC_StartupAxisGroup
      - Administrative/Configuration
        - AxisGroup
        - Computation
        - Coordinate systems
        - Dynamics
        - Kinematics
      - AxisGroup
        - AXIS_GROUP_REF_SM3
      - Diagnostics
      - internal
        - Kernel_CP
          - Path Creation
      - Movement
        - GroupInterrupt
        - MC_GroupContinue
        - MC_GroupHalt
        - MC_GroupInterrupt
        - MC_GroupStop
        - MC_MoveCircularAbsolute
        - MC_MoveCircularRelative
        - MC_MoveDirectAbsolute
        - MC_MoveDirectRelative
        - MC_MoveLinearAbsolute
        - MC_MoveLinearRelative
        - SMC_GroupEnableResumeAfterError
        - SMC_GroupJog
        - SMC_GroupJog2
        - SMC_GroupUpdateContinueData
        - SMC_GroupWait

Additionally, the SM3_Robotics_Visu library is also available for displaying elements

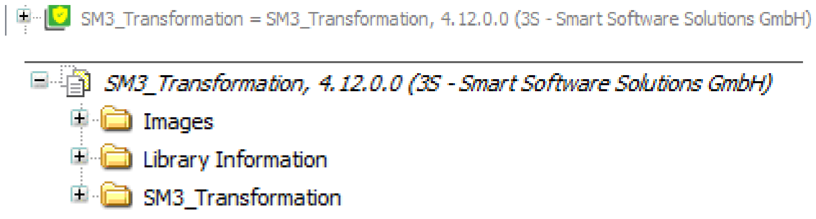


SM3_Robotics_Visu = SM3_Robotics_Visu, 4.10.0.0 (3S - Smart Software Solutions GmbH)

- *SM3_Robotics_Visu, 4.10.0.0 (3S - Smart Software Solutions GmbH)*
  - Additional
  - SM3_Robotics_Visu
  - GlobalTextList

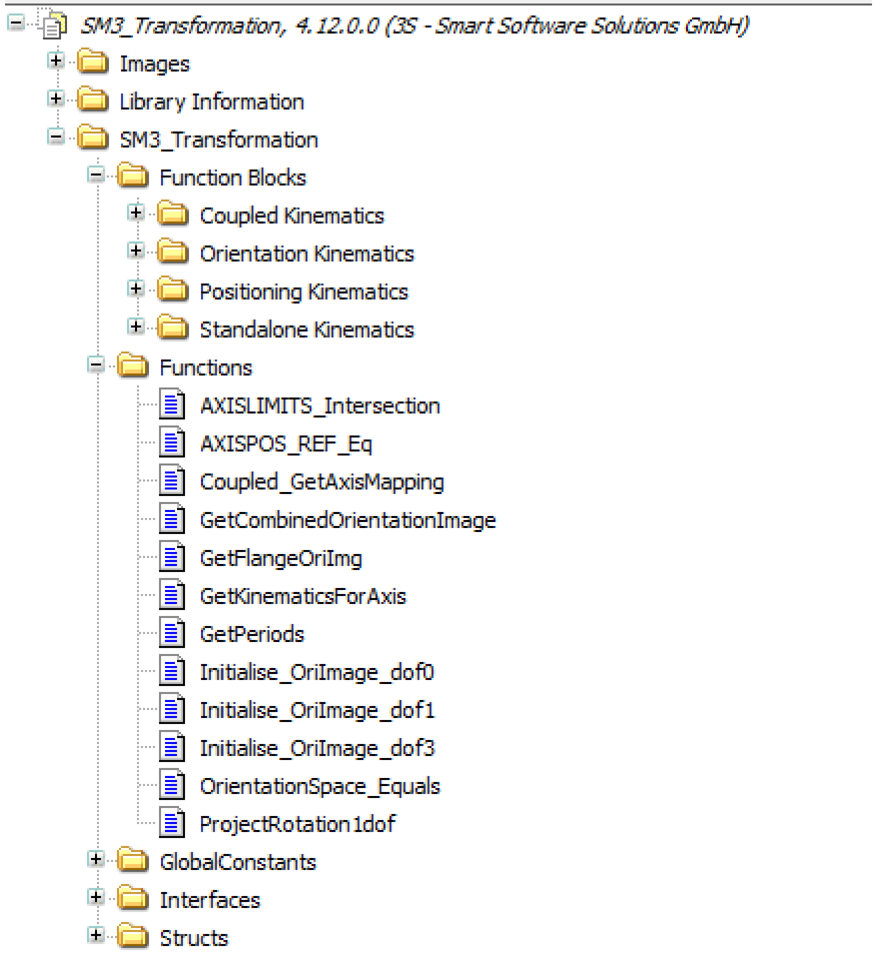
ctrlX - SoftMotion Libraries – SM3_Transformation:

- *SM3_Transformation CNC library:*
  - *This library contains the control blocks of the various supported kinematics*



- SM3_Transformation = SM3_Transformation, 4.12.0.0 (3S - Smart Software Solutions GmbH)

- *SM3_Transformation, 4.12.0.0 (3S - Smart Software Solutions GmbH)*
  - Images
  - Library Information
  - SM3_Transformation

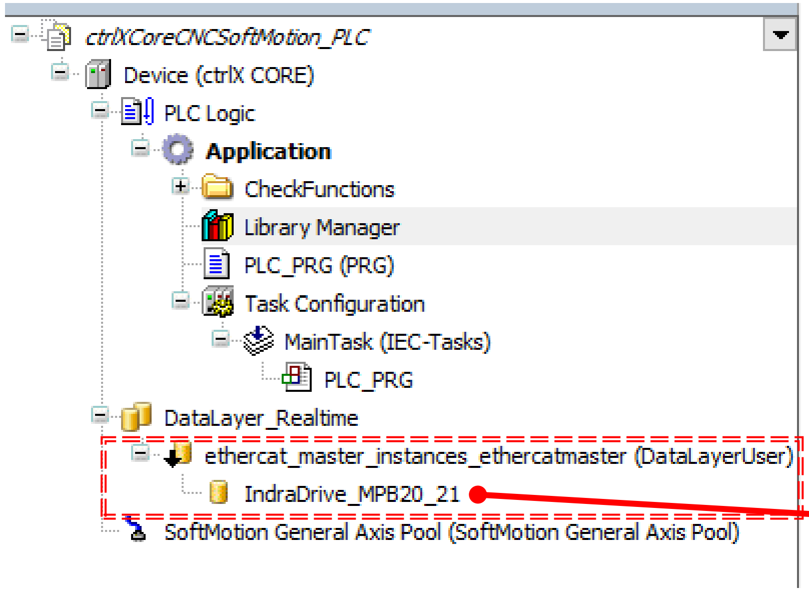


- *SM3_Transformation, 4.12.0.0 (3S - Smart Software Solutions GmbH)*
  - Images
  - Library Information
  - SM3_Transformation
    - Function Blocks
      - Coupled Kinematics
      - Orientation Kinematics
      - Positioning Kinematics
      - Standalone Kinematics
    - Functions
      - AXISLIMITS_Intersection
      - AXISPOS_REF_Eq
      - Coupled_GetAxisMapping
      - GetCombinedOrientationImage
      - GetFlangeOriImg
      - GetKinematicsForAxis
      - GetPeriods
      - Initialise_OriImage_dof0
      - Initialise_OriImage_dof1
      - Initialise_OriImage_dof3
      - OrientationSpace_Equals
      - ProjectRotation1dof
    - GlobalConstants
    - Interfaces
    - Structs

Activate SoftMotion on a real axis

ctrlX - Activation of SoftMotion on a real axis

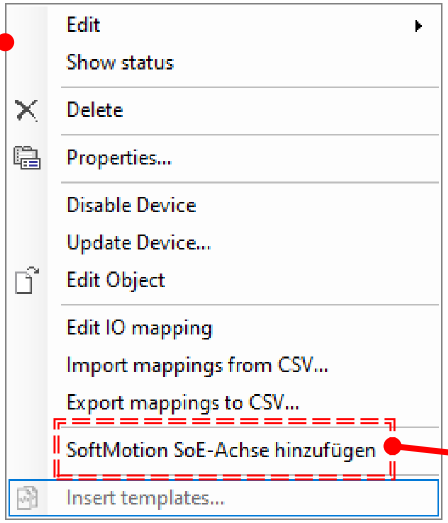
- *Activate SoftMotion on a real axis:*

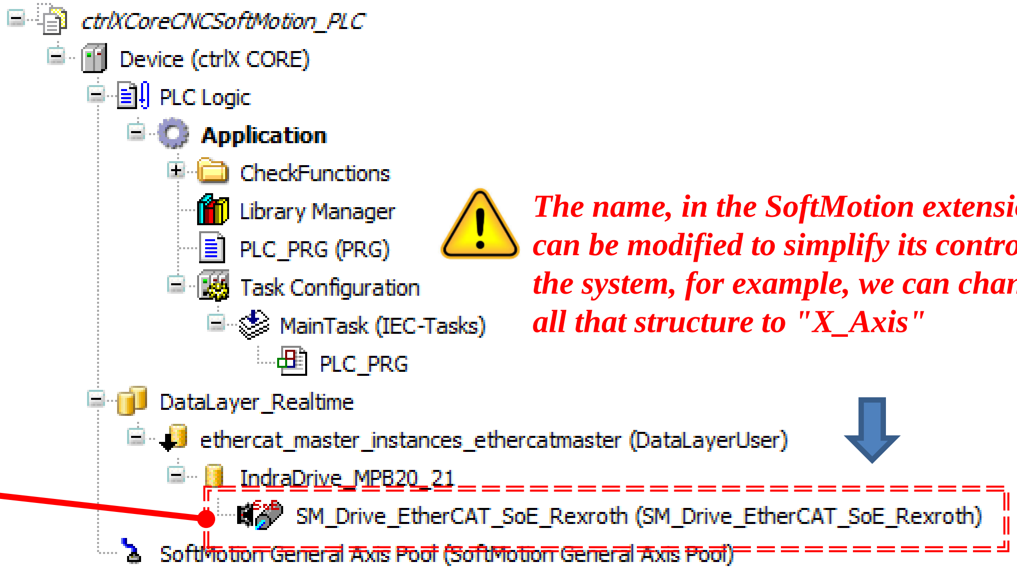


ctrlXCoreCNCSoftMotion_PLC
- Device (ctrlX CORE)
  - PLC Logic
    - Application
      - CheckFunctions
      - Library Manager
      - PLC_PRG (PRG)
      - Task Configuration
        - MainTask (IEC-Tasks)
          - PLC_PRG
  - DataLayer_Realtime
    - ethercat_master_instances_ethercatmaster (DataLayerUser)
      - IndraDrive_MPB20_21
  - SoftMotion General Axis Pool (SoftMotion General Axis Pool)

The inclusion of the axis within the system must be done in the usual way and initially from the part of ctrlX IO

The activation of SoftMotion generates a new module in the axis inserted in EtherCat



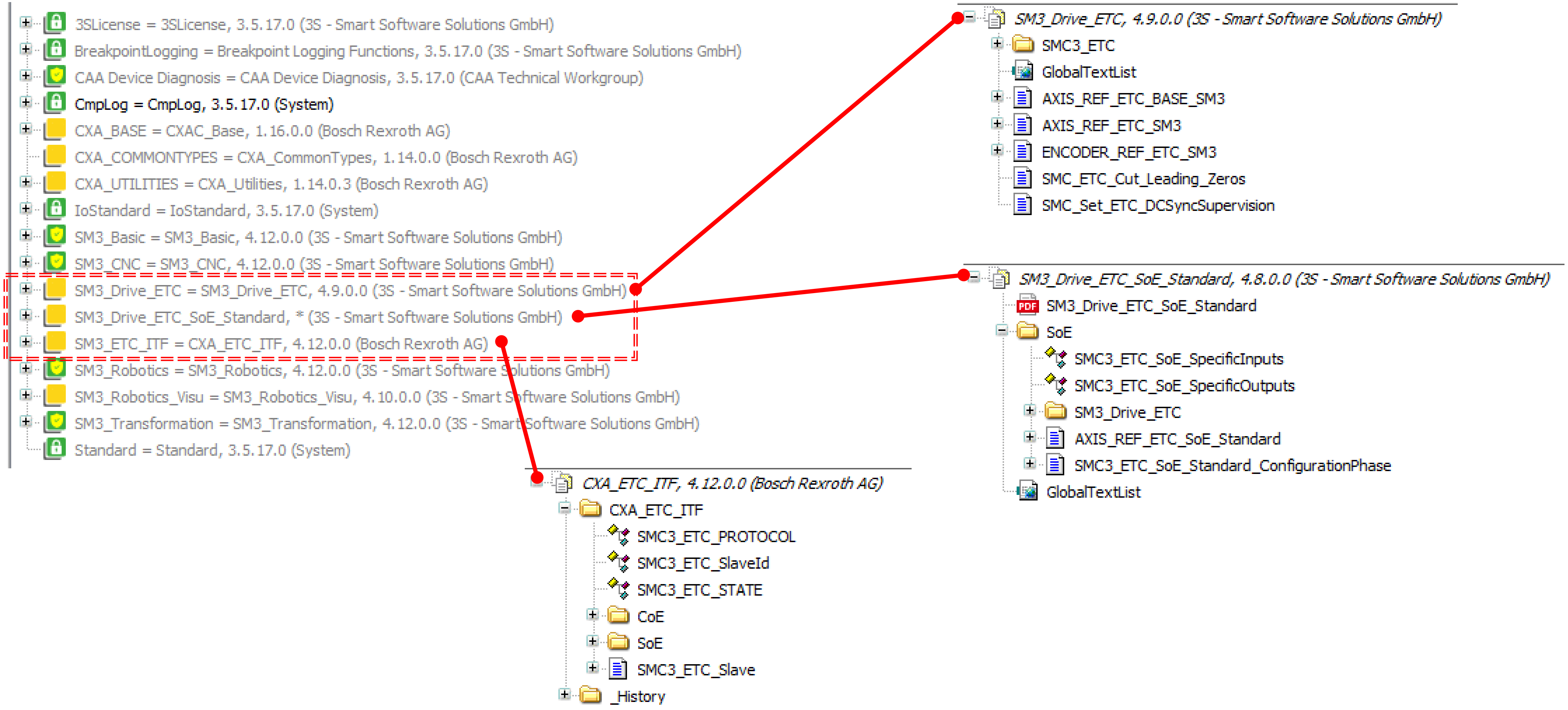


ctrlXCoreCNCSoftMotion_PLC
- Device (ctrlX CORE)
  - PLC Logic
    - Application
      - CheckFunctions
      - Library Manager
      - PLC_PRG (PRG)
      - Task Configuration
        - MainTask (IEC-Tasks)
          - PLC_PRG
  - DataLayer_Realtime
    - ethercat_master_instances_ethercatmaster (DataLayerUser)
      - IndraDrive_MPB20_21
        - SM_Drive_EtherCAT_SoE_Rexroth (SM_Drive_EtherCAT_SoE_Rexroth)
  - SoftMotion General Axis Pool (SoftMotion General Axis Pool)

The name, in the SoftMotion extension, can be modified to simplify its control in the system, for example, we can change all that structure to "X_Axis"

ctrlX - *Activation of SoftMotion on a real axis*

- ***With the inclusion of the real axis, some libraries are also inserted to control the axes:***



```
3SLicense = 3SLicense, 3.5.17.0 (3S - Smart Software Solutions GmbH)
BreakpointLogging = Breakpoint Logging Functions, 3.5.17.0 (3S - Smart Software Solutions GmbH)
CAA Device Diagnosis = CAA Device Diagnosis, 3.5.17.0 (CAA Technical Workgroup)
CmpLog = CmpLog, 3.5.17.0 (System)
CXA_BASE = CXAC_Base, 1.16.0.0 (Bosch Rexroth AG)
CXA_COMMONTYPES = CXA_CommonTypes, 1.14.0.0 (Bosch Rexroth AG)
CXA_UTILITIES = CXA_Utilities, 1.14.0.3 (Bosch Rexroth AG)
IoStandard = IoStandard, 3.5.17.0 (System)
SM3_Basic = SM3_Basic, 4.12.0.0 (3S - Smart Software Solutions GmbH)
SM3_CNC = SM3_CNC, 4.12.0.0 (3S - Smart Software Solutions GmbH)
SM3_Drive_ETC = SM3_Drive_ETC, 4.9.0.0 (3S - Smart Software Solutions GmbH)
SM3_Drive_ETC_SoE_Standard, * (3S - Smart Software Solutions GmbH)
SM3_ETC_ITF = CXA_ETC_ITF, 4.12.0.0 (Bosch Rexroth AG)
SM3_Robotics = SM3_Robotics, 4.12.0.0 (3S - Smart Software Solutions GmbH)
SM3_Robotics_Visu = SM3_Robotics_Visu, 4.10.0.0 (3S - Smart Software Solutions GmbH)
SM3_Transformation = SM3_Transformation, 4.12.0.0 (3S - Smart Software Solutions GmbH)
Standard = Standard, 3.5.17.0 (System)
```

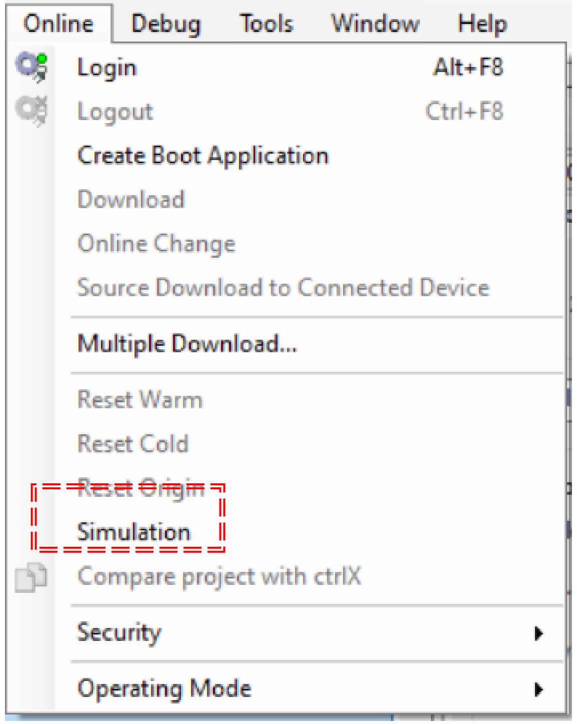
```
SM3_Drive_ETC, 4.9.0.0 (3S - Smart Software Solutions GmbH)
    SMC3_ETC
    GlobalTextList
    AXIS_REF_ETC_BASE_SM3
    AXIS_REF_ETC_SM3
    ENCODER_REF_ETC_SM3
    SMC_ETC_Cut_Leading_Zeros
    SMC_Set_ETC_DCSyncSupervision
```

```
SM3_Drive_ETC_SoE_Standard, 4.8.0.0 (3S - Smart Software Solutions GmbH)
    SM3_Drive_ETC_SoE_Standard
    SoE
        SMC3_ETC_SoE_SpecificInputs
        SMC3_ETC_SoE_SpecificOutputs
        SM3_Drive_ETC
        AXIS_REF_ETC_SoE_Standard
        SMC3_ETC_SoE_Standard_ConfigurationPhase
    GlobalTextList
```

```
CXA_ETC_ITF, 4.12.0.0 (Bosch Rexroth AG)
    CXA_ETC_ITF
        SMC3_ETC_PROTOCOL
        SMC3_ETC_SlaveId
        SMC3_ETC_STATE
        CoE
        SoE
        SMC3_ETC_Slave
    _History
```

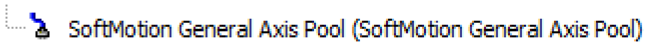
Example with three virtual axes in simulation mode

⚠ ***The program modules have been generated in Ladder format so that they are more understandable in the presentation environment, but it can be used, as is obvious, any other programming language in which we feel more comfortable.***

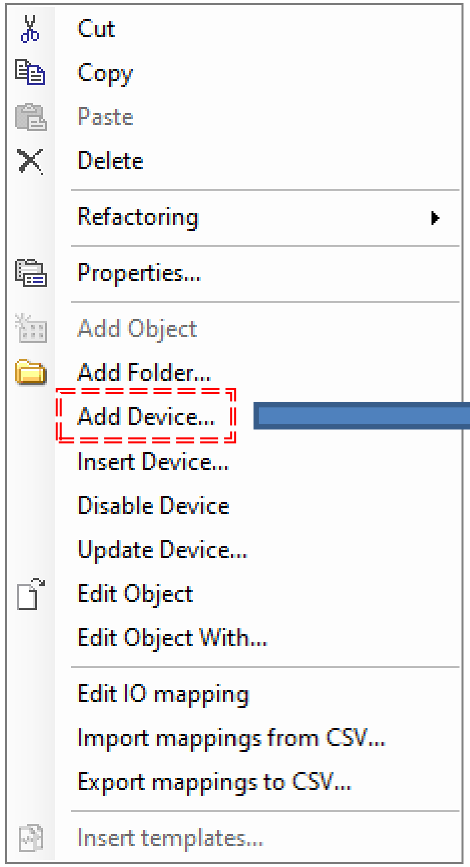


ctrlX - *Example of CNC program – Insert virtual axes*

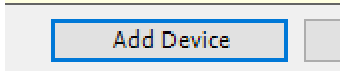
- ***For the example we are going to use three virtual axes that are incorporated as follows:***



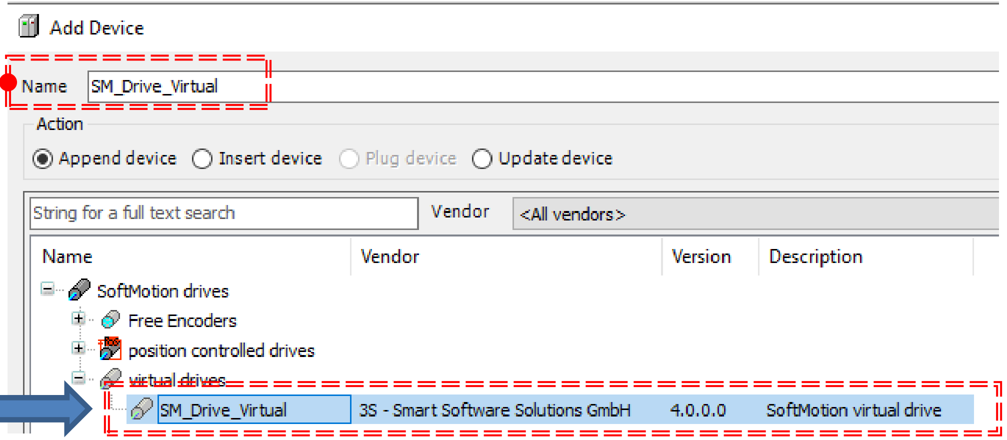
SoftMotion General Axis Pool (SoftMotion General Axis Pool)



Modify Name and insert 3 virtual axes with "Add Device"



Select "SM_Drive_Virtual"



Add Device

Name: SM_Drive_Virtual

Action

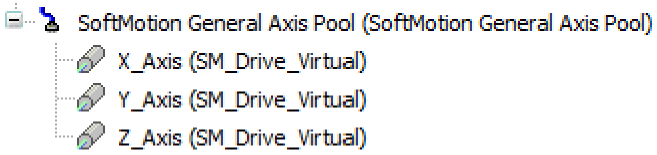
● Append device ○ Insert device ○ Plug device ○ Update device

String for a full text search

Vendor: <All vendors>

Name	Vendor	Version	Description
SoftMotion drives			
Free Encoders			
position controlled drives			
virtual drives			
SM_Drive_Virtual	3S - Smart Software Solutions GmbH	4.0.0.0	SoftMotion virtual drive

- ***The SoftMotion structure for the virtual axes should look like this:***

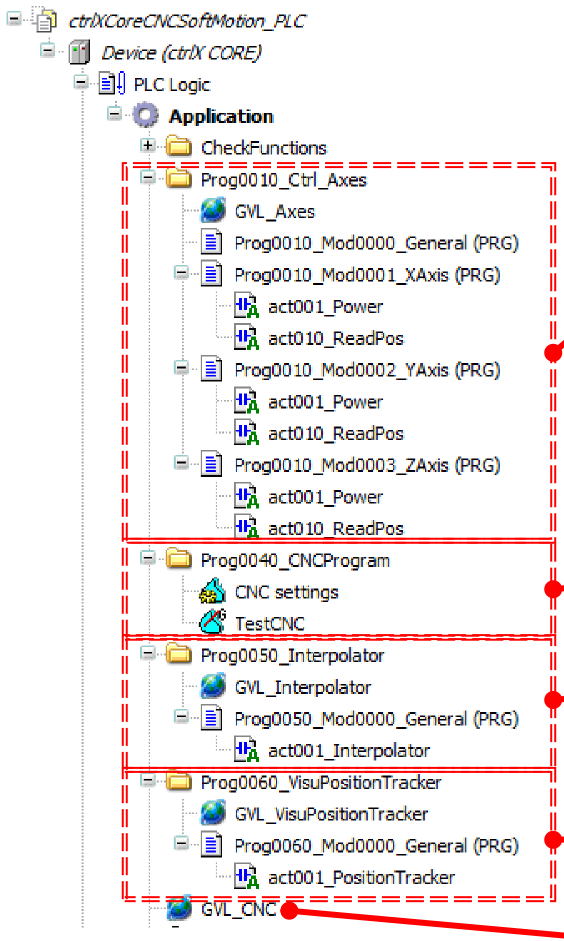


SoftMotion General Axis Pool (SoftMotion General Axis Pool)
X_Axis (SM_Drive_Virtual)
Y_Axis (SM_Drive_Virtual)
Z_Axis (SM_Drive_Virtual)

⚠ ***The use of virtual axes allows us to use the program in "Simulation" mode and execute it without having to establish a connection to any equipment. This allows debugging the program, without the need for hardware components.***

ctrlX - *Example of CNC program - Folder structure and Initial elements for the basic program*

- ***The sample program is broken down into several folders in each of which the necessary modules for project control will be placed.***



```
ctrlXCoreCNCSoftMotion_PLC
  Device (ctrlX CORE)
    PLC Logic
      Application
        CheckFunctions
        Prog0010_Ctrl_Axes
          GVL_Axes
          Prog0010_Mod0000_General (PRG)
          Prog0010_Mod0001_XAxis (PRG)
            act001_Power
            act010_ReadPos
          Prog0010_Mod0002_YAxis (PRG)
            act001_Power
            act010_ReadPos
          Prog0010_Mod0003_ZAxis (PRG)
            act001_Power
            act010_ReadPos
        Prog0040_CNCProgram
          CNC settings
          TestCNC
        Prog0050_Interpolator
          GVL_Interpolator
          Prog0050_Mod0000_General (PRG)
            act001_Interpolator
        Prog0060_VisuPositionTracker
          GVL_VisuPositionTracker
          Prog0060_Mod0000_General (PRG)
            act001_PositionTracker
        GVL_CNC
```

X, Y, Z axes control

In the first part of the example, we will only need the power activation of the axes and the reading of the current position of each of them. Obviously it will be missing, among other things, the fault reset manager, the enabling of manual movements, etc., which will be integrated later.

Folder with the CNC Programs generated internally in ctrlX PLC Engineering

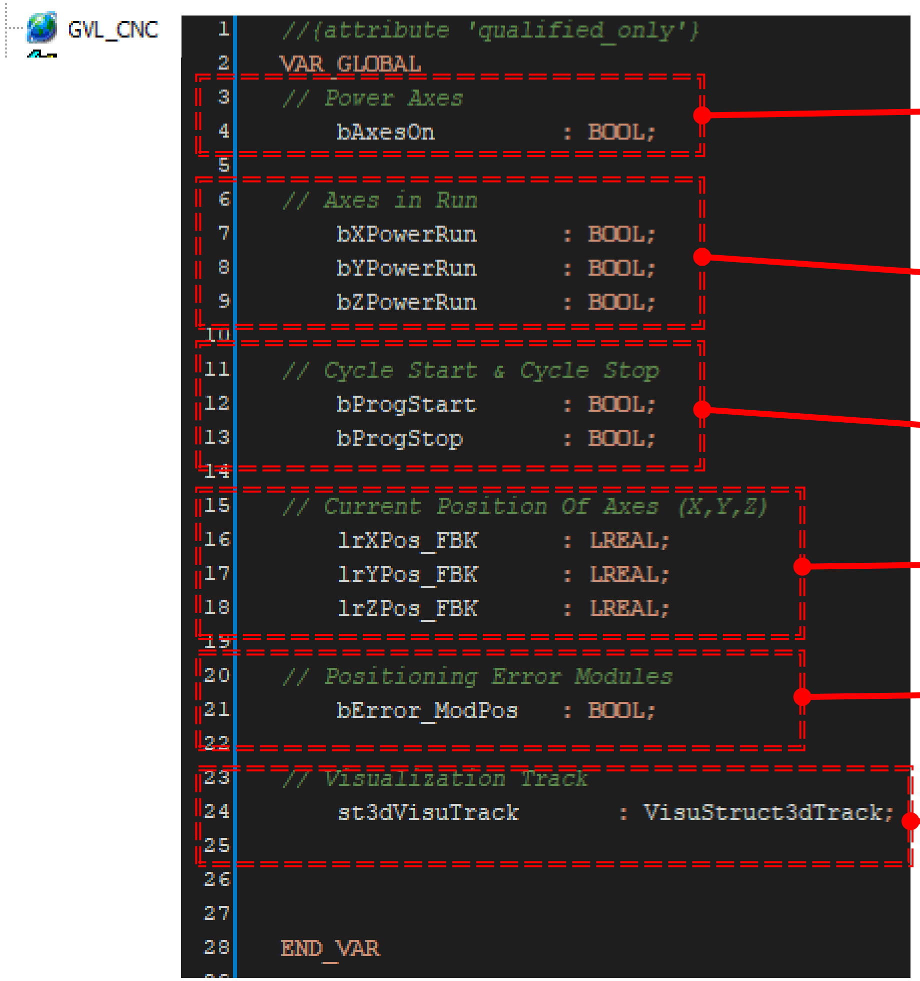
Folder with the Motion Interpolator control

Folder with the Position Viewer control for the screen

Table of general variables used to control the elements

ctrlX - *Example of CNC program - Global Variables used in the program*

- *First of all, we are going to see the folder with the general variables that will be used both in the control modules and in the screen displays.:*



```
//{attribute 'qualified_only'}
VAR_GLOBAL
    // Power Axes
        bAxesOn          : BOOL;

    // Axes in Run
        bXPowerRun       : BOOL;
        bYPowerRun       : BOOL;
        bZPowerRun       : BOOL;

    // Cycle Start & Cycle Stop
        bProgStart       : BOOL;
        bProgStop        : BOOL;

    // Current Position Of Axes (X,Y,Z)
        lrXPos_FBK       : LREAL;
        lrYPos_FBK       : LREAL;
        lrZPos_FBK       : LREAL;

    // Positioning Error Modules
        bError_ModPos    : BOOL;

    // Visualization Track
        st3dVisuTrack        : VisuStruct3dTrack;



END_VAR
```

Activation bit for the power modules of the axes.

As it is a 3-axis operating system, the power will be activated simultaneously on all of them.

Signaling of axes with power activated.

Start and stop activation bits of the CNC program.

Current positions of the axes, extracted from the position reading modules.

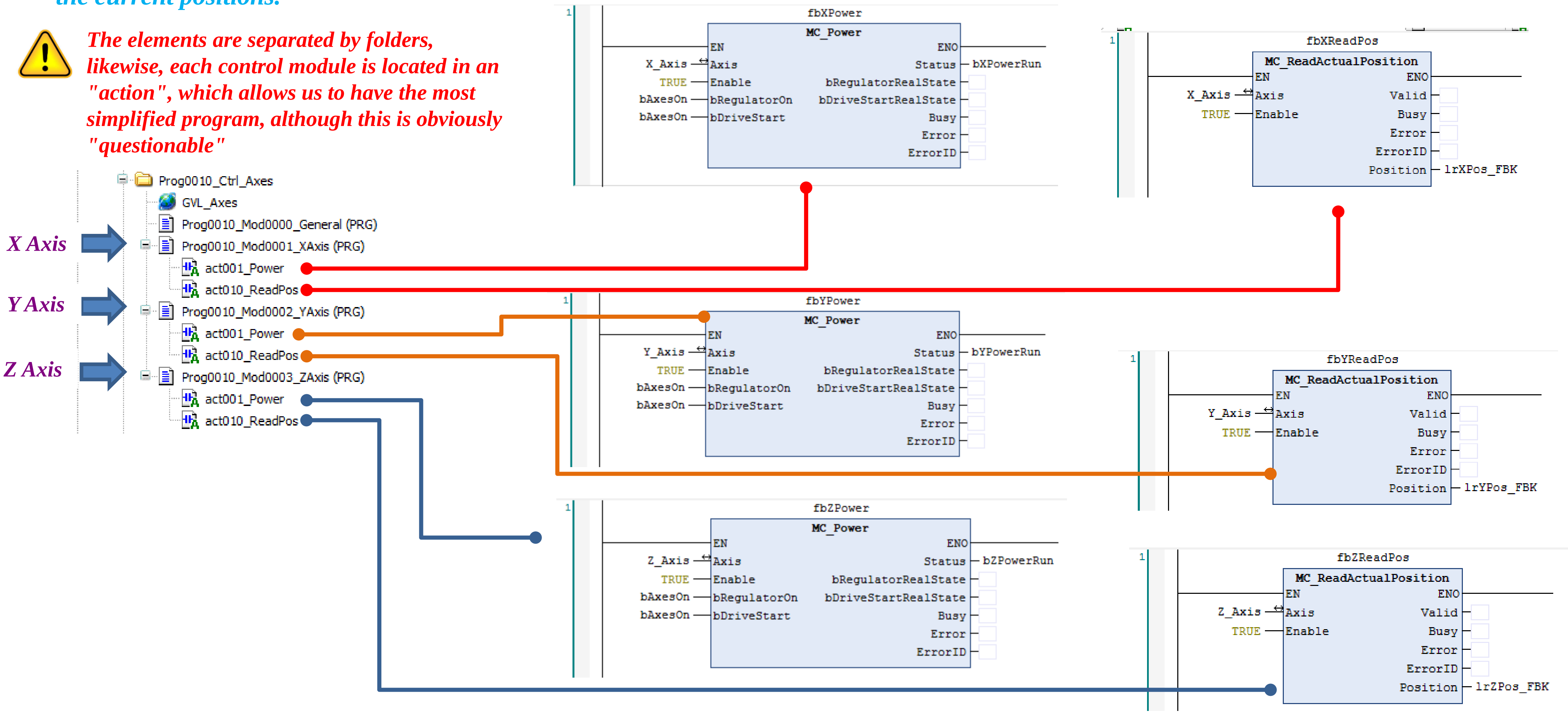
Error control bit in the modules for automatic positioning of the axes.

Fb with structure for the control of the positions on the track of the visualization on the screen

ctrlX - *Example of CNC program - Folders and control modules of the axes*

- *In the axis control folders, as we have already mentioned, at the moment we will only have the control to activate the power of the axis and the reading of the current positions.*

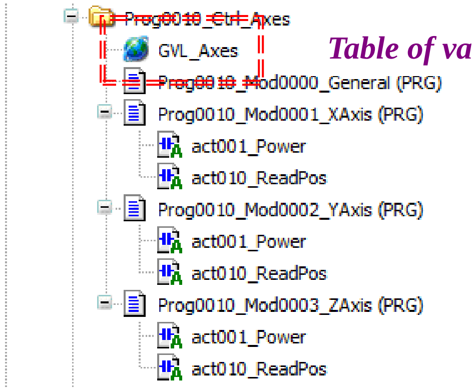
The elements are separated by folders, likewise, each control module is located in an "action", which allows us to have the most simplified program, although this is obviously "questionable"



ctrlX - *Example of CNC program - Libraries of the axis control modules*

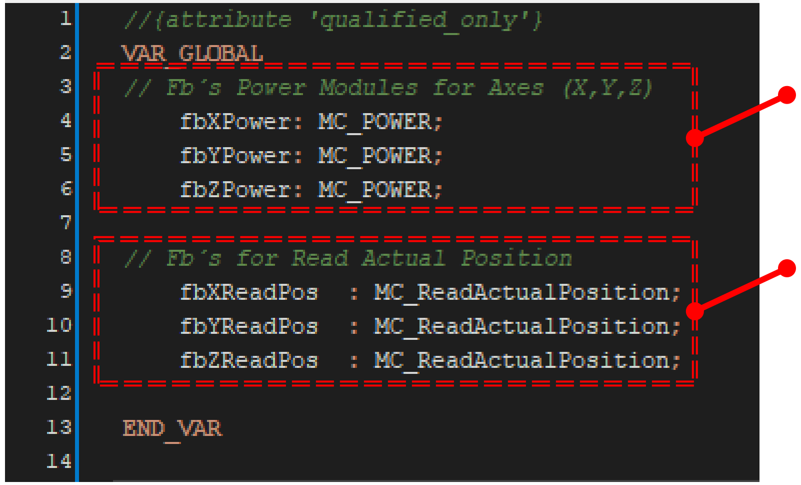
- *The two modules used previously are located in the following library:*

SM3_Basic Library



```
Prog0010_Ctrl_Axes
    GVL_Axes
    Prog0010_Mod0000_General (PRG)
    Prog0010_Mod0001_XAxis (PRG)
        act001_Power
        act010_ReadPos
    Prog0010_Mod0002_YAxis (PRG)
        act001_Power
        act010_ReadPos
    Prog0010_Mod0003_ZAxis (PRG)
        act001_Power
        act010_ReadPos
```

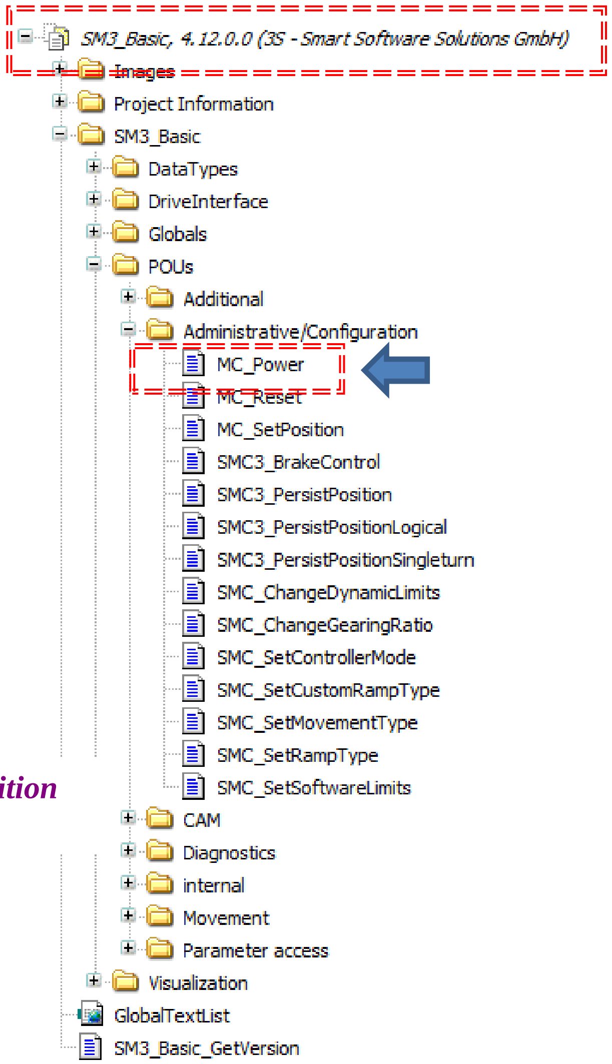
Table of variables used



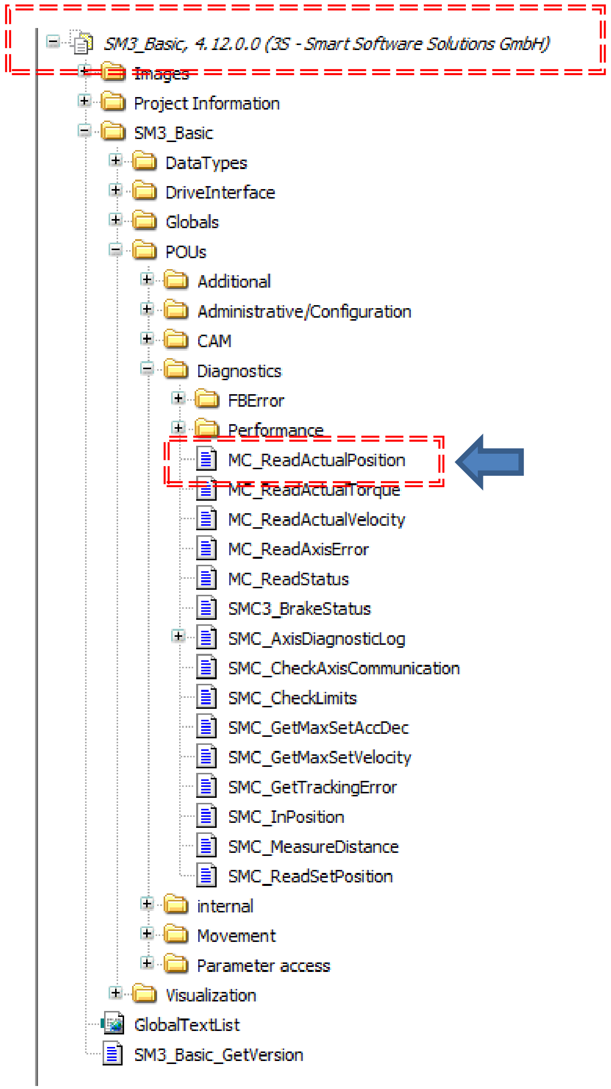
```
//{attribute 'qualified_only'}
VAR_GLOBAL
// Fb´s Power Modules for Axes (X,Y,Z)
    fbXPower: MC_POWER;
    fbYPower: MC_POWER;
    fbZPower: MC_POWER;

// Fb´s for Read Actual Position
    fbXReadPos  : MC_ReadActualPosition;
    fbYReadPos  : MC_ReadActualPosition;
    fbZReadPos  : MC_ReadActualPosition;

END_VAR

```

Modules to activate power.

Modules for position reading.



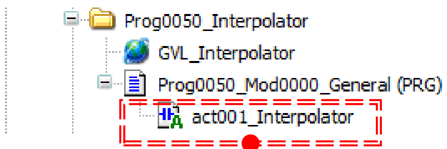
```
SM3_Basic, 4.12.0.0 (3S - Smart Software Solutions GmbH)
    Images
    Project Information
    SM3_Basic
        DataTypes
        DriveInterface
        Globals
        POUs
            Additional
            Administrative/Configuration
                MC_Power
                MC_Reset
                MC_SetPosition
                SMC3_BrakeControl
                SMC3_PersistPosition
                SMC3_PersistPositionLogical
                SMC3_PersistPositionSingleturn
                SMC_ChangeDynamicLimits
                SMC_ChangeGearingRatio
                SMC_SetControllerMode
                SMC_SetCustomRampType
                SMC_SetMovementType
                SMC_SetRampType
                SMC_SetSoftwareLimits
            CAM
            Diagnostics
            internal
            Movement
            Parameter access
        Visualization
    GlobalTextList
    SM3_Basic_GetVersion
```



```
SM3_Basic, 4.12.0.0 (3S - Smart Software Solutions GmbH)
    Images
    Project Information
    SM3_Basic
        DataTypes
        DriveInterface
        Globals
        POUs
            Additional
            Administrative/Configuration
            CAM
            Diagnostics
                FBError
                Performance
                MC_ReadActualPosition
                MC_ReadActualTorque
                MC_ReadActualVelocity
                MC_ReadAxisError
                MC_ReadStatus
                SMC3_BrakeStatus
                SMC_AxisDiagnosticLog
                SMC_CheckAxisCommunication
                SMC_CheckLimits
                SMC_GetMaxSetAccDec
                SMC_GetMaxSetVelocity
                SMC_GetTrackingError
                SMC_InPosition
                SMC_MeasureDistance
                SMC_ReadSetPosition
            internal
            Movement
            Parameter access
        Visualization
    GlobalTextList
    SM3_Basic_GetVersion
```

ctrlX - *Example of CNC program - Libraries used in the Motion Interpolator*

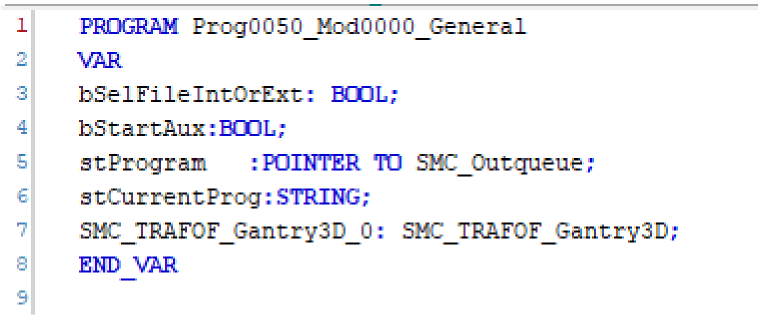
- *In the following explanation of the example program, we will see the configuration of the Motion Interpolator*



```
Prog0050_Interpolator
    GVL_Interpolator
    Prog0050_Mod0000_General (PRG)
        act001_Interpolator
```

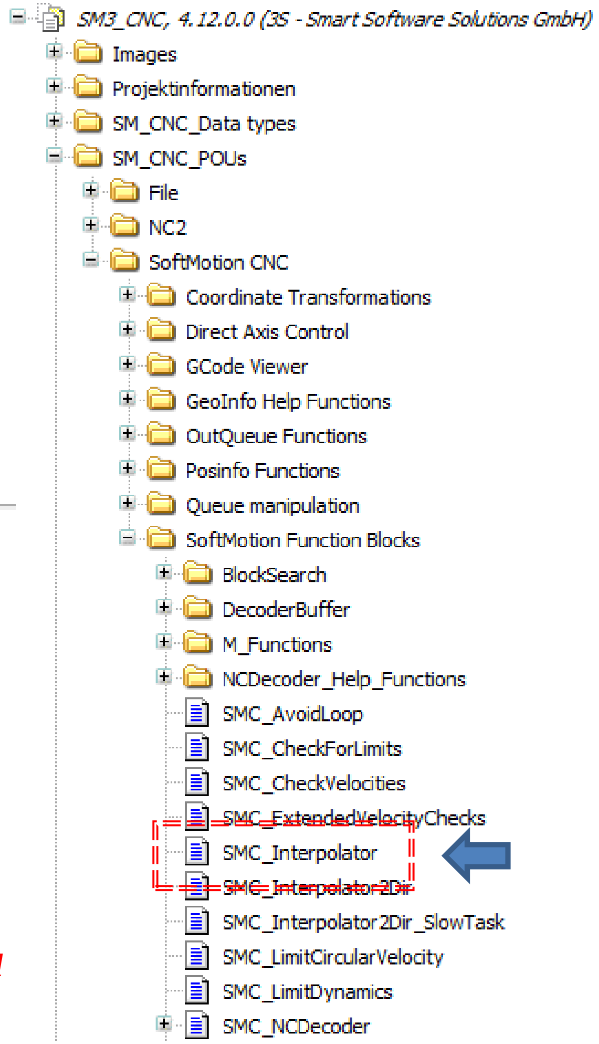
Control module for the interpolation of the axes.

Variables used

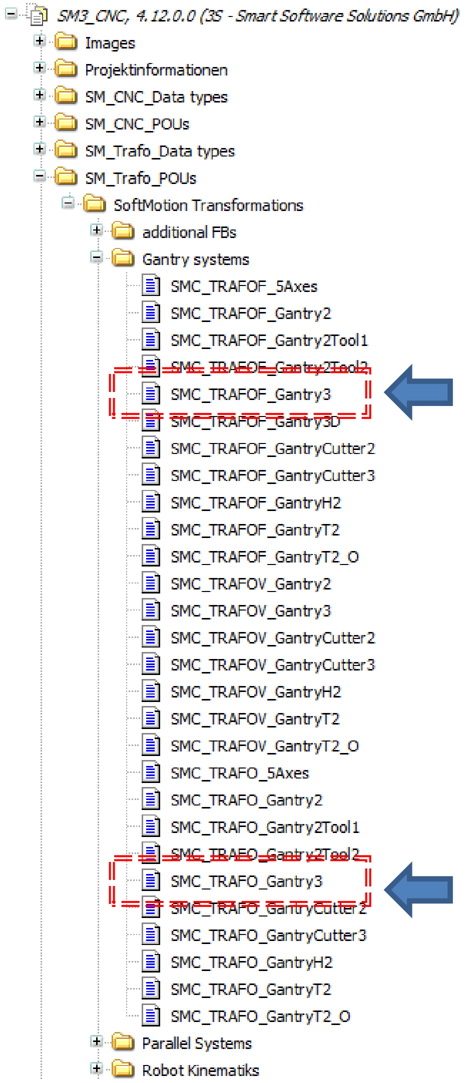


```
PROGRAM Prog0050_Mod0000_General
VAR
bSelFileIntOrExt: BOOL;
bStartAux:BOOL;
stProgram    :POINTER TO SMC_Outqueue;
stCurrentProg:STRING;
SMC_TRAFOF_Gantry3D_0: SMC_TRAFOF_Gantry3D;
END_VAR
```

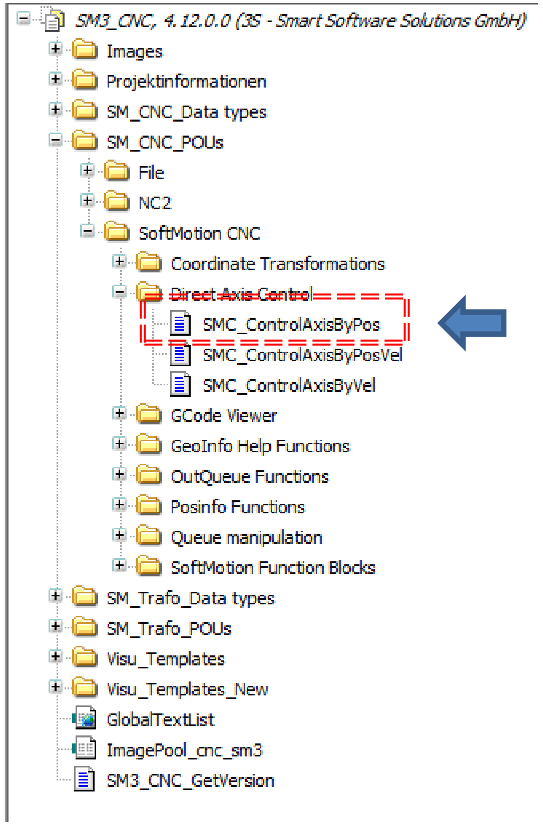
Modules used to control the Interpolator



```
SM3_CNC, 4.12.0.0 (3S - Smart Software Solutions GmbH)
    Images
    Projektinformationen
    SM_CNC_Data types
    SM_CNC_POUs
        File
        NC2
        SoftMotion CNC
            Coordinate Transformations
            Direct Axis Control
            GCode Viewer
            GeoInfo Help Functions
            OutQueue Functions
            Posinfo Functions
            Queue manipulation
            SoftMotion Function Blocks
                BlockSearch
                DecoderBuffer
                M_Functions
                NCDecoder_Help_Functions
                SMC_AvoidLoop
                SMC_CheckForLimits
                SMC_CheckVelocities
                SMC_ExtendedVelocityChecks
                SMC_Interpolator
                SMC_Interpolator2Dir
                SMC_Interpolator2Dir_SlowTask
                SMC_LimitCircularVelocity
                SMC_LimitDynamics
                SMC_NCDecoder
```



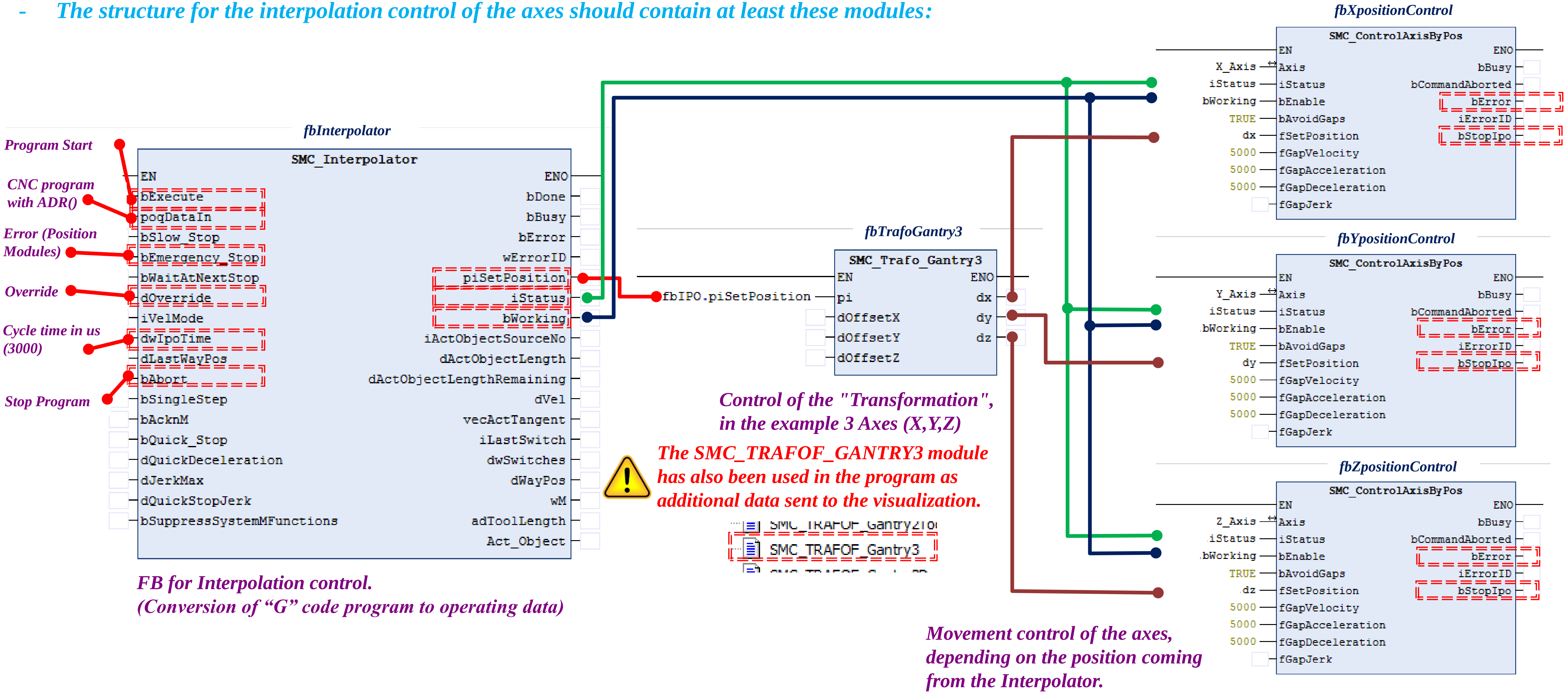
```
SM3_CNC, 4.12.0.0 (3S - Smart Software Solutions GmbH)
    Images
    Projektinformationen
    SM_CNC_Data types
    SM_CNC_POUs
    SM_Trafo_Data types
    SM_Trafo_POUs
        SoftMotion Transformations
            additional FBs
            Gantry systems
                SMC_TRAFOF_5Axes
                SMC_TRAFOF_Gantry2
                SMC_TRAFOF_Gantry2Tool1
                SMC_TRAFOF_Gantry2Tool2
                SMC_TRAFOF_Gantry3
                SMC_TRAFOF_Gantry3D
                SMC_TRAFOF_GantryCutter2
                SMC_TRAFOF_GantryCutter3
                SMC_TRAFOF_GantryH2
                SMC_TRAFOF_GantryT2
                SMC_TRAFOF_GantryT2_O
                SMC_TRAFOV_Gantry2
                SMC_TRAFOV_Gantry3
                SMC_TRAFOV_GantryCutter2
                SMC_TRAFOV_GantryCutter3
                SMC_TRAFOV_GantryH2
                SMC_TRAFOV_GantryT2
                SMC_TRAFOV_GantryT2_O
                SMC_TRAFO_5Axes
                SMC_TRAFO_Gantry2
                SMC_TRAFO_Gantry2Tool1
                SMC_TRAFO_Gantry2Tool2
                SMC_TRAFO_Gantry3
                SMC_TRAFO_GantryCutter2
                SMC_TRAFO_GantryCutter3
                SMC_TRAFO_GantryH2
                SMC_TRAFO_GantryT2
                SMC_TRAFO_GantryT2_O
            Parallel Systems
            Robot Kinematiks
```



```
SM3_CNC, 4.12.0.0 (3S - Smart Software Solutions GmbH)
    Images
    Projektinformationen
    SM_CNC_Data types
    SM_CNC_POUs
        File
        NC2
        SoftMotion CNC
            Coordinate Transformations
            Direct Axis Control
                SMC_ControlAxisByPos
                SMC_ControlAxisByPosVel
                SMC_ControlAxisByVel
            GCode Viewer
            GeoInfo Help Functions
            OutQueue Functions
            Posinfo Functions
            Queue manipulation
            SoftMotion Function Blocks
    SM_Trafo_Data types
    SM_Trafo_POUs
    Visu_Templates
    Visu_Templates_New
    GlobalTextList
    ImagePool_cnc_sm3
    SM3_CNC_GetVersion
```

ctrlX - Example of CNC program – Minimum structure for Interpolation

- *The structure for the interpolation control of the axes should contain at least these modules:*



Program Start

CNC program with ADR()

Error (Position Modules)

Override

Cycle time in us (3000)

Stop Program

FB for Interpolation control.
(Conversion of "G" code program to operating data)

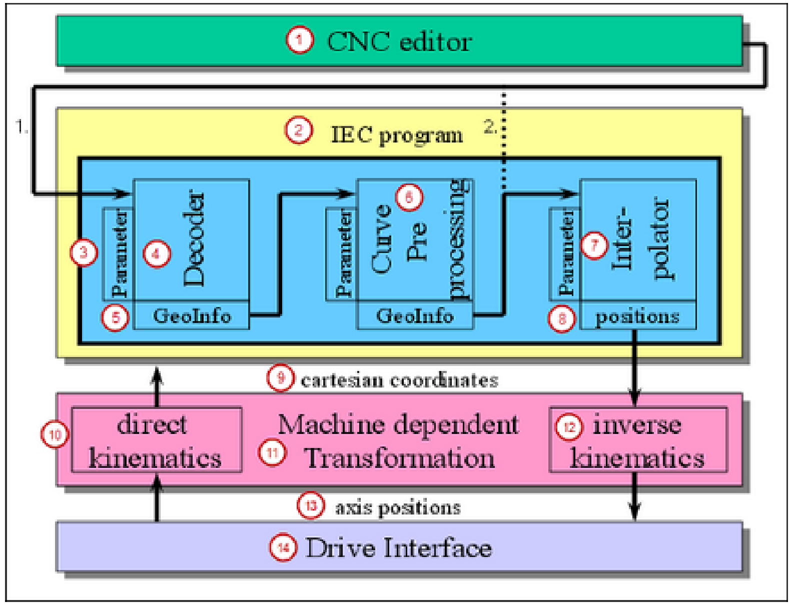
Control of the "Transformation", in the example 3 Axes (X,Y,Z)

The SMC_TRAFOF_GANTRY3 module has also been used in the program as additional data sent to the visualization.

Movement control of the axes, depending on the position coming from the Interpolator.

ctrlX - *CNC program example – SoftMotion Software components for CNC control*

- *Components of the SoftMotion Software of the CNC editor:*



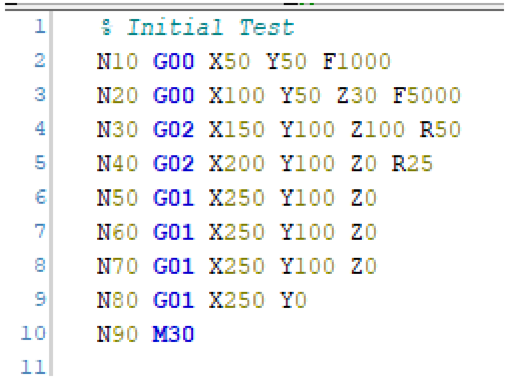
(1) *CNC Editor or CNC program*	(2) *IEC Program*	(3) *Parameter*
(4) *Decoder*	(5) *GeoInfo*	(6) *Pre-Procesing Path*
(7) *Interpolator*	(8) *Positions*	(9) *Cartesian Coordinates*
(10) *Direct Kinematics*	(11) *Machine Transformation*	(12) *Invers Kinematics*
(13) *Axis Positions*	(14) *Drive Interface*	

ctrlX - Example of CNC program - Description of the inputs of the interpolation module (1)

- *Interpolator Module (Description of inputs (1)):*

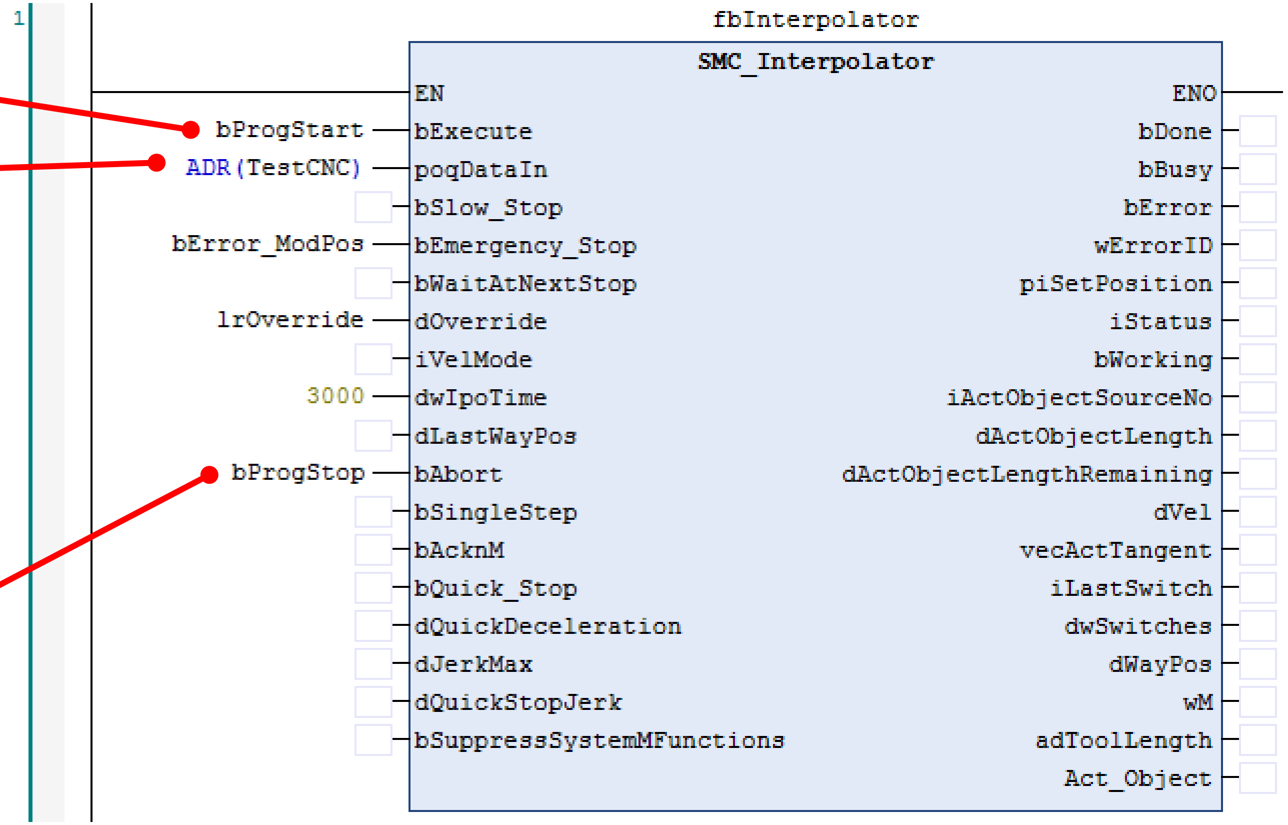
Start Program

CNC program Name or dta send for the “Interpreter Module” with ADR()*



```
% Initial Test
N10 G00 X50 Y50 F1000
N20 G00 X100 Y50 Z30 F5000
N30 G02 X150 Y100 Z100 R50
N40 G02 X200 Y100 Z0 R25
N50 G01 X250 Y100 Z0
N60 G01 X250 Y100 Z0
N70 G01 X250 Y100 Z0
N80 G01 X250 Y0
N90 M30
```

Stop Program



```
fbInterpolator
SMC_Interpolator
                EN                                  ENO
bProgStart    — bExecute                          bDone
ADR(TestCNC)  — poqDataIn                         bBusy
                bSlow_Stop                        bError
bError_ModPos — bEmergency_Stop                   wErrorID
                bWaitAtNextStop                   piSetPosition
lrOverride    — dOverride                         iStatus
                iVelMode                          bWorking
3000          — dwIpoTime                         iActObjectSourceNo
                dLastWayPos                       dActObjectLength
bProgStop     — bAbort                            dActObjectLengthRemaining
                bSingleStep                       dVel
                bAcknM                            vecActTangent
                bQuick_Stop                       iLastSwitch
                dQuickDeceleration                dwSwitches
                dJerkMax                          dWayPos
                dQuickStopJerk                    wM
                bSuppressSystemMFunctions         adToolLength
                                                  Act_Object
```

⚠ *Next we will see the use of the "Interpreter" module*

ctrlX - *Example of CNC program - Description of the inputs of the interpolation module (2)*

- *Interpolator Module (Description of inputs (2)):*

In TRUE state the SMC_Interpolator will be made to slow down to 0 according to the velocity profile defined in (iVelMode) and the maximum delay of the current SMC_GEOINFO object (dDecel, see below) and wait until bSlow_Stop is reset to FALSE

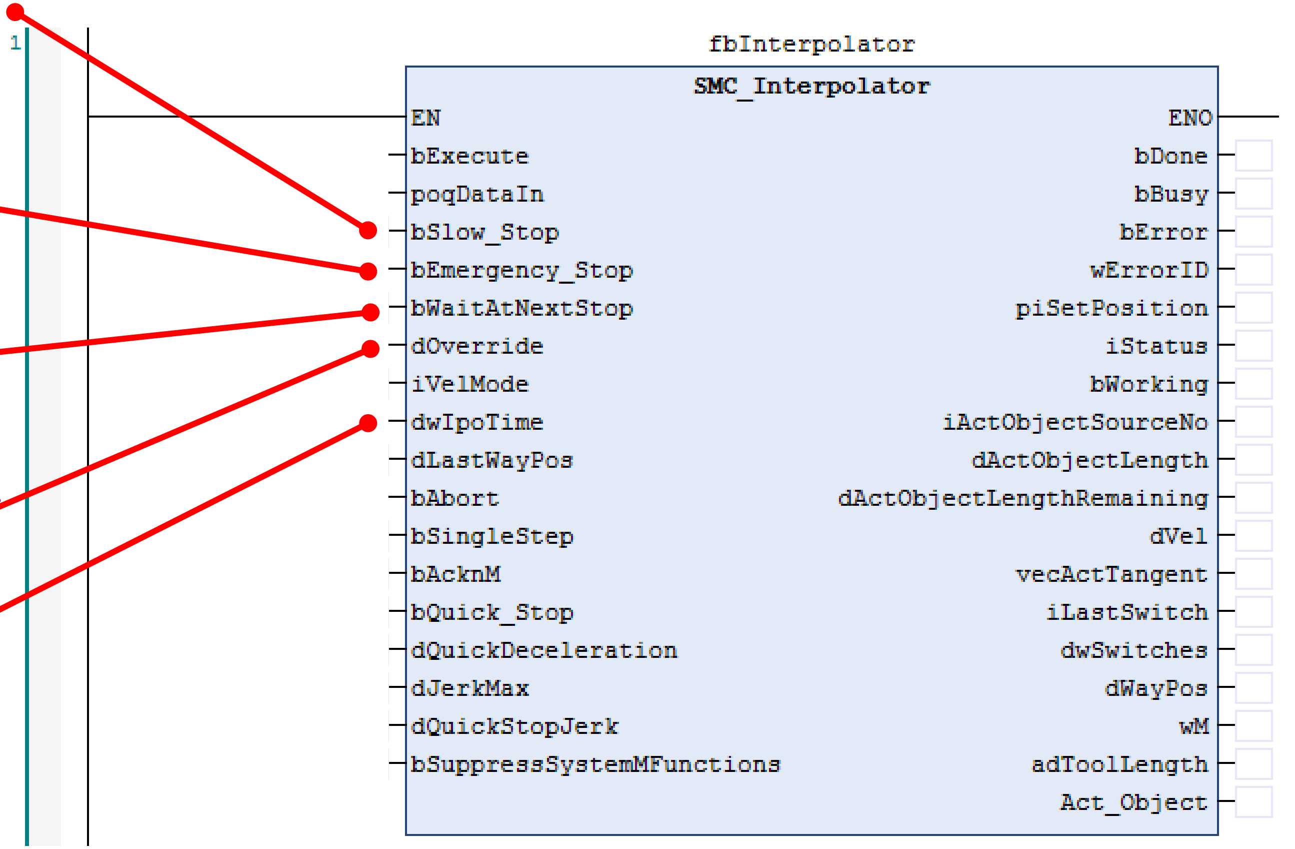
As soon as this input becomes TRUE, the SMC_Interpolator will cause an immediate stop, this means that the position will be held. Therefore, the speed will be set to 0 immediately.

As long as this variable is FALSE (default), the path is passed non-stop. Otherwise, the SMC_Interpolator will be made to hold position at the next regular stop, that is, at position points where velocity is 0, usually at path angles, and stop until bWaitAtNextStop is reset to FALSE.

This variable can be used to control the override. dOverride is not allowed to be less than 0.01. The programmed speed of particular objects will be scaled by dOverride; Therefore, the set speed can be increased or decreased in online mode. For example, dOverride=1 (default) causes scheduled speeds to run, while dOverride=2 would double them.

⚠ ***Be careful with this value if we are working in "Simulation" mode or with real axes. In general, in simulation we should have it much higher than with the real axes.***

This variable must be set for each call. It represents the cycle time in µsec.
By default we are using a “3000”



fbInterpolator

SMC_Interpolator

EN	ENO
bExecute	bDone
poqDataIn	bBusy
bSlow_Stop	bError
bEmergency_Stop	wErrorID
bWaitAtNextStop	piSetPosition
dOverride	iStatus
iVelMode	bWorking
dwIpoTime	iActObjectSourceNo
dLastWayPos	dActObjectLength
bAbort	dActObjectLengthRemaining
bSingleStep	dVel
bAcknM	vecActTangent
bQuick_Stop	iLastSwitch
dQuickDeceleration	dwSwitches
dJerkMax	dWayPos
dQuickStopJerk	wM
bSuppressSystemMFunctions	adToolLength
	Act_Object

ctrlX - Example of CNC program - Description of the inputs of the interpolation module (3)

- Interpolator Module (Description of inputs (3)):

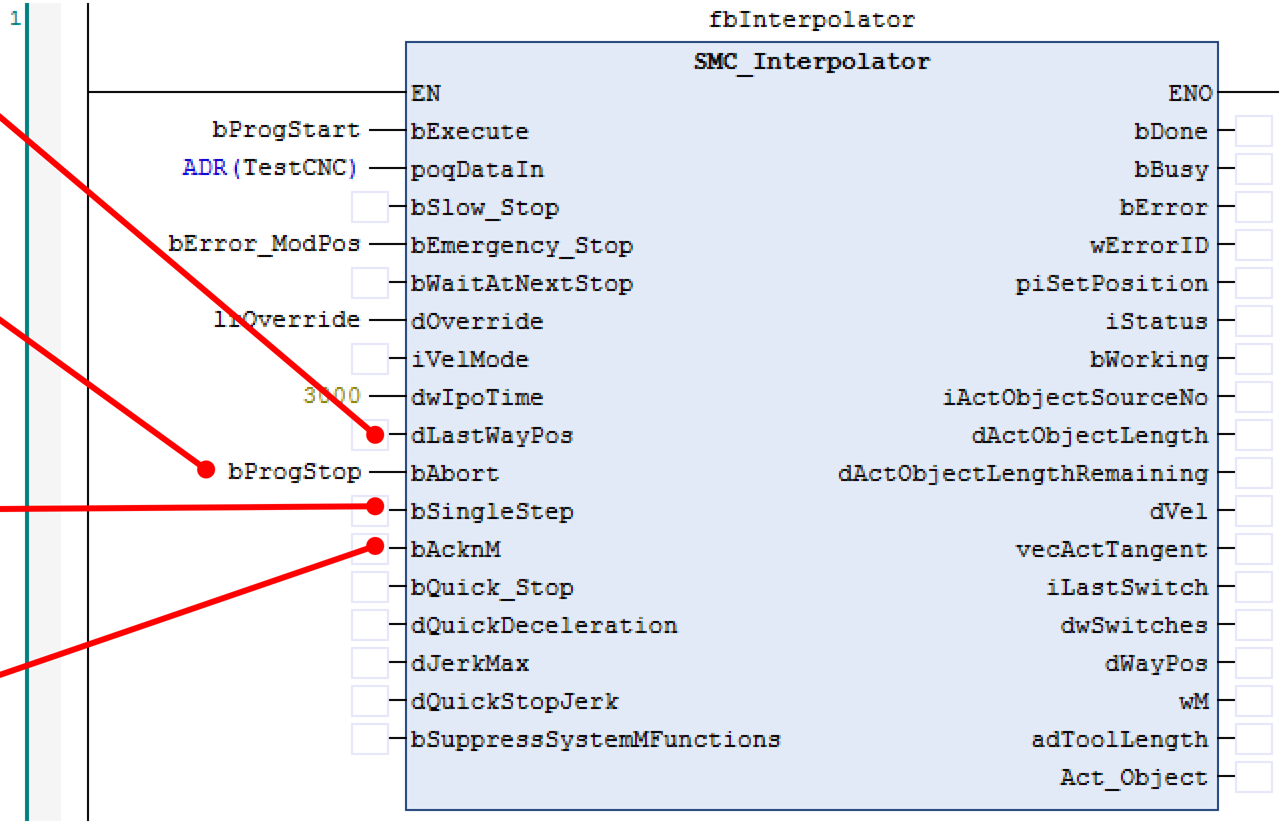
This input allows the user to measure the leg of the route that is being pulled by the interpolator. The output dWayPos is the sum of dLastWayPos and the distance traveled within the current cycle. If dLastWayPos is set equal to output dWayPos, dWayPos will always be incremented by the current path segment, resulting in the total length of the path traveled. dLastWayPos can be (re)set to 0 or a different value at any time.

This input set to TRUE will abort the function block and reset the outputs. A rising edge of bExecute is required to start the interpolator again after aborting.

This input causes the interpolator to stop at the transition between two path objects (also on transitions with identical tangents) for the duration of one cycle. If bSingleStep is set to TRUE during motion, the interpolator will stop at the end of that object, which can be reached without exceeding the programmed deceleration value.

If the interpolator must stop at the next possible stop position (ie, at points where velocity is 0), then bWaitAtNextStop must be used.

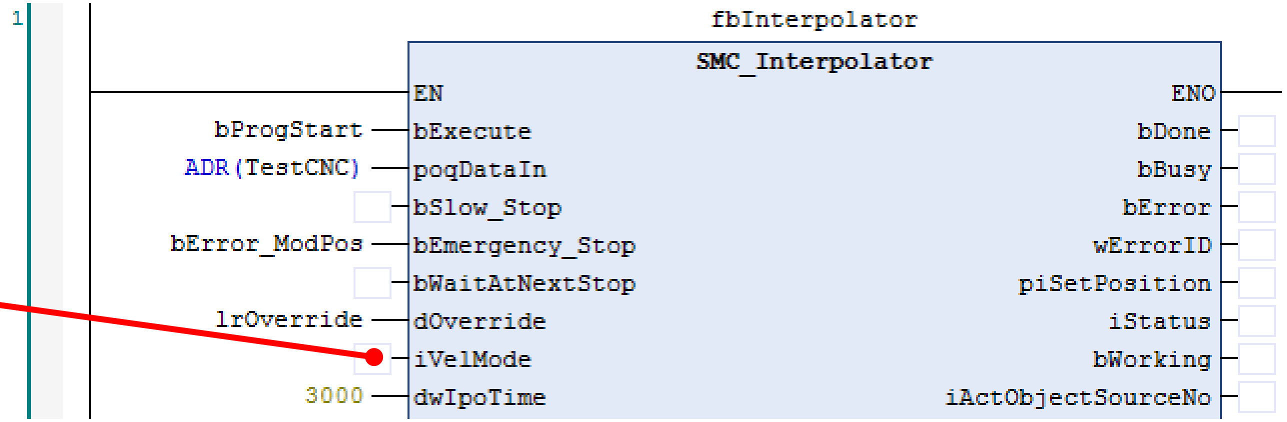
This input can be used to recognize an M function. If the input is TRUE, the wM output will be cleared and path processing will continue.

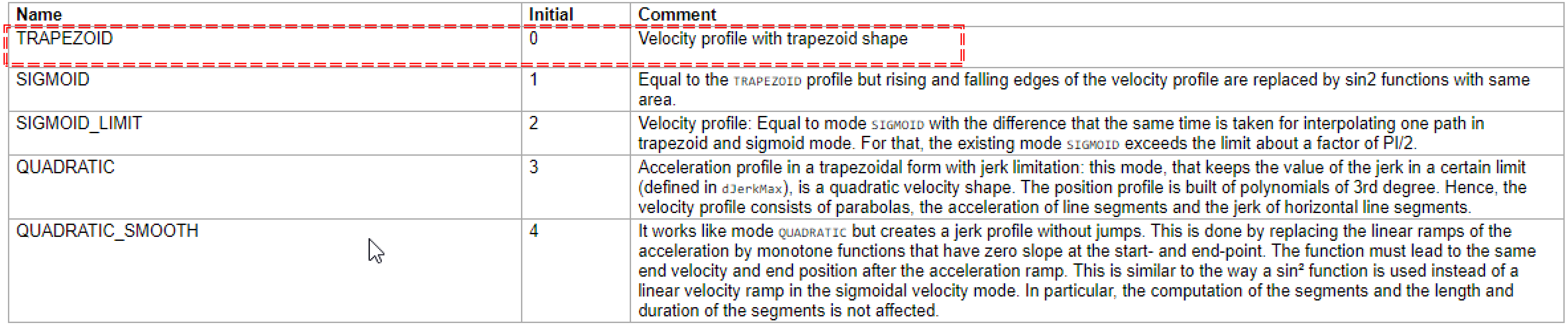


ctrlX - *Example of CNC program - Description of the inputs of the interpolation module (4)*

- *Interpolator Module (Description of inputs (4)):*

This input defines the velocity profile defined in SMC_INT_VELMODE.





Name	Initial	Comment
TRAPEZOID	0	Velocity profile with trapezoid shape
SIGMOID	1	Equal to the TRAPEZOID profile but rising and falling edges of the velocity profile are replaced by sin2 functions with same area.
SIGMOID_LIMIT	2	Velocity profile: Equal to mode SIGMOID with the difference that the same time is taken for interpolating one path in trapezoid and sigmoid mode. For that, the existing mode SIGMOID exceeds the limit about a factor of PI/2.
QUADRATIC	3	Acceleration profile in a trapezoidal form with jerk limitation: this mode, that keeps the value of the jerk in a certain limit (defined in dJerkMax), is a quadratic velocity shape. The position profile is built of polynomials of 3rd degree. Hence, the velocity profile consists of parabolas, the acceleration of line segments and the jerk of horizontal line segments.
QUADRATIC_SMOOTH	4	It works like mode QUADRATIC but creates a jerk profile without jumps. This is done by replacing the linear ramps of the acceleration by monotone functions that have zero slope at the start- and end-point. The function must lead to the same end velocity and end position after the acceleration ramp. This is similar to the way a sin² function is used instead of a linear velocity ramp in the sigmoidal velocity mode. In particular, the computation of the segments and the length and duration of the segments is not affected.

⚠ *By default the interpolator uses the “Trapezoid” mode.*

ctrlX - *Example of CNC program - Description of the inputs of the interpolation module (5)*

- ***Interpolator Module (Description of inputs (5)):***

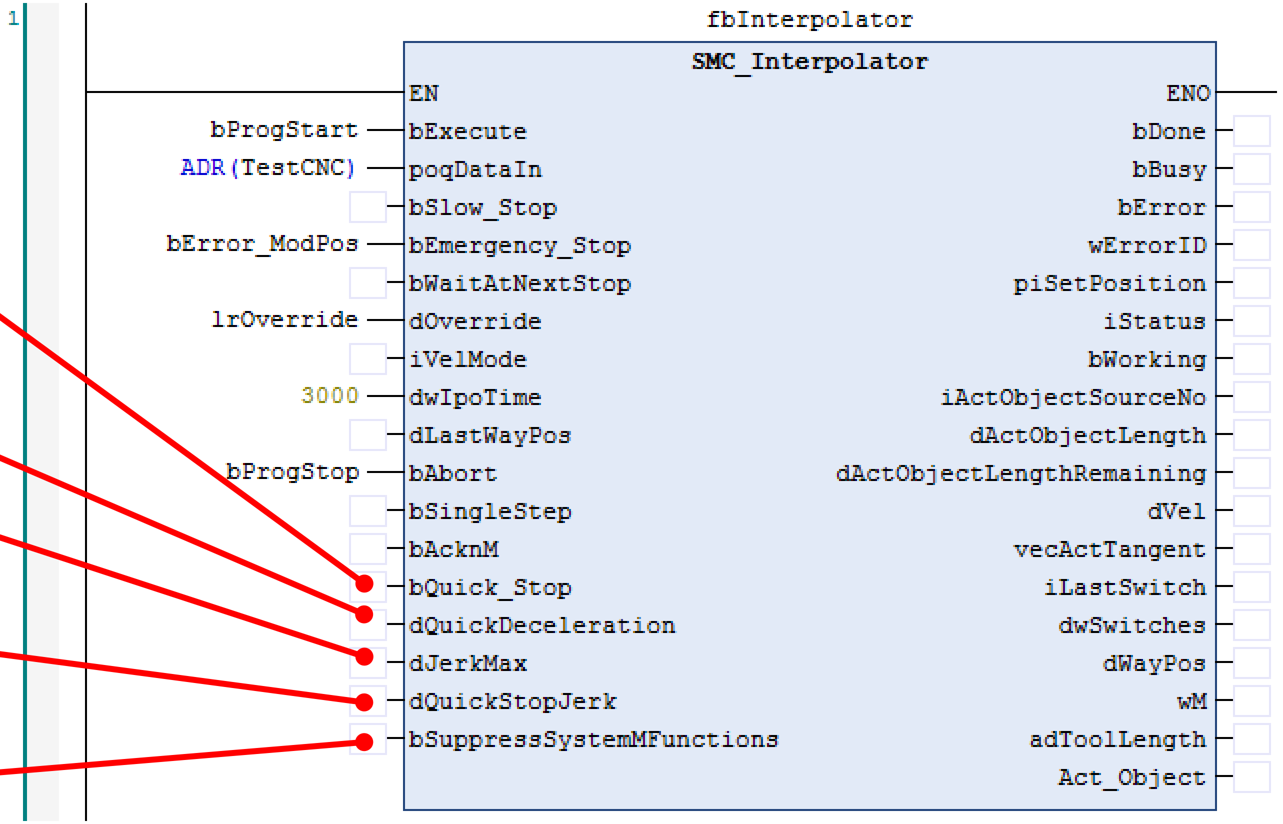
If this input is TRUE, the interpolator will slow down to zero, until bQuick_Stop is reset to FALSE. The reduction is done according to the defined speed profile (iVelMode) and the deceleration given by the maximum of dQuickDeceleration and the delay currently programmed in the route. If a square rate mode is used, then the jerk is limited by max(dJerkMax, dQuickStopJerk).

Deceleration value used for bQuick_Stop

Maximum Allowable Jerk Magnitude: Only used for Quadratic Rate modes. It must be positive and cannot be changed while the interpolator is running.

The value of "dQuickStopJerk" is used by a quick stop to reduce acceleration if one of the quadratic velocity modes is selected.

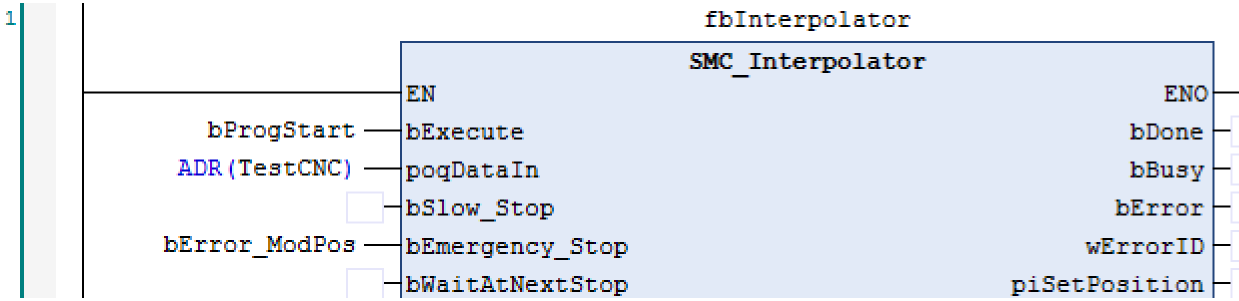
If this option is set, the wM output will not be set for internal M functions created by the G75 or G4 commands.



```
1
                         fbInterpolator
                        SMC_Interpolator
                 EN                                        ENO
bProgStart     — bExecute                                  bDone —
ADR(TestCNC)   — poqDataIn                                 bBusy —
               — bSlow_Stop                               bError —
bError_ModPos  — bEmergency_Stop                        wErrorID —
               — bWaitAtNextStop                    piSetPosition —
lrOverride     — dOverride                               iStatus —
               — iVelMode                               bWorking —
3000           — dwIpoTime                     iActObjectSourceNo —
               — dLastWayPos                     dActObjectLength —
bProgStop      — bAbort                 dActObjectLengthRemaining —
               — bSingleStep                                dVel —
               — bAcknM                            vecActTangent —
               — bQuick_Stop                         iLastSwitch —
               — dQuickDeceleration                   dwSwitches —
               — dJerkMax                                dWayPos —
               — dQuickStopJerk                               wM —
               — bSuppressSystemMFunctions          adToolLength —
                                                     Act_Object —
```

ctrlX - *Example of CNC program - Description of the outputs of the interpolation module (1)*

- ***Interpolator Module (Description of outputs (1)):***



This variable will be set to TRUE as soon as the input data (poqDataIn) has been fully processed. The function block will not take any further action until a reset is performed. If the bExecute input is FALSE, then bDone will be reset to FALSE.

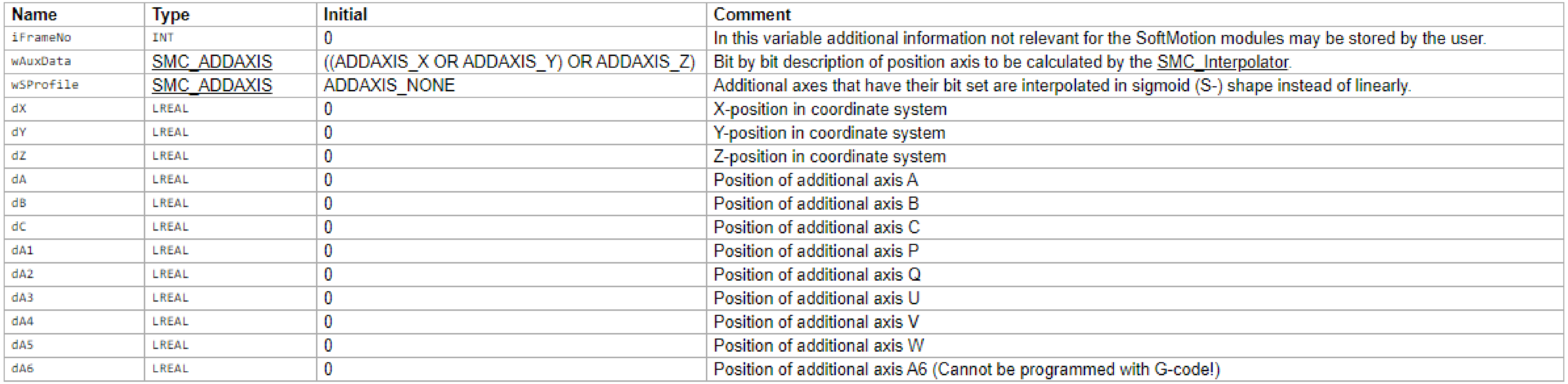
TRUE while the execution of the function block is not finished

TRUE while the execution of the function block is not finished

Error

It reflects the calculated set position and contains the Cartesian coordinates of the next position, as well as the state of the additional axis. SMC_POSINFO

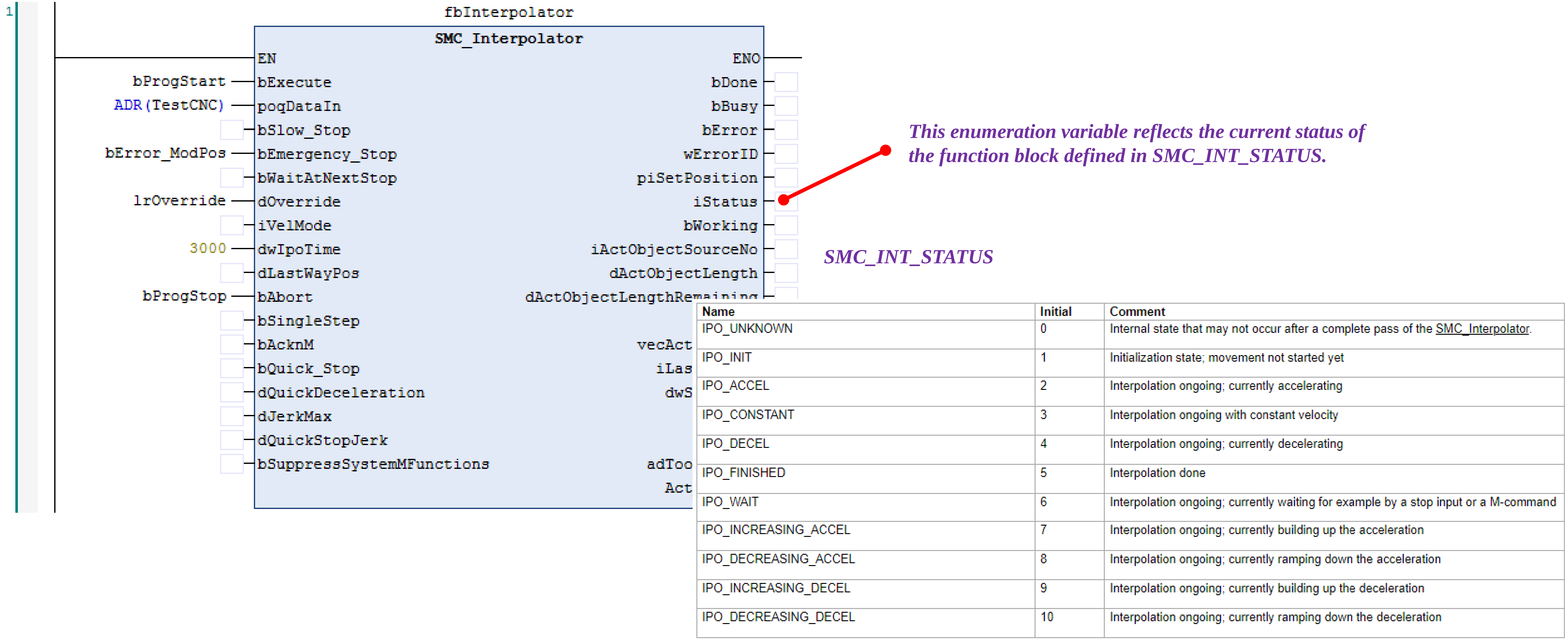
SMC_POSINFO



Name	Type	Initial	Comment
iFrameNo	INT	0	In this variable additional information not relevant for the SoftMotion modules may be stored by the user.
wAuxData	SMC_ADDAXIS	((ADDAXIS_X OR ADDAXIS_Y) OR ADDAXIS_Z)	Bit by bit description of position axis to be calculated by the SMC_Interpolator.
wSProfile	SMC_ADDAXIS	ADDAXIS_NONE	Additional axes that have their bit set are interpolated in sigmoid (S-) shape instead of linearly.
dX	LREAL	0	X-position in coordinate system
dY	LREAL	0	Y-position in coordinate system
dZ	LREAL	0	Z-position in coordinate system
dA	LREAL	0	Position of additional axis A
dB	LREAL	0	Position of additional axis B
dC	LREAL	0	Position of additional axis C
dA1	LREAL	0	Position of additional axis P
dA2	LREAL	0	Position of additional axis Q
dA3	LREAL	0	Position of additional axis U
dA4	LREAL	0	Position of additional axis V
dA5	LREAL	0	Position of additional axis W
dA6	LREAL	0	Position of additional axis A6 (Cannot be programmed with G-code!)

ctrlX - *Example of CNC program - Description of the outputs of the interpolation module (2)*

- ***Interpolator Module (Description of outputs (2)):***



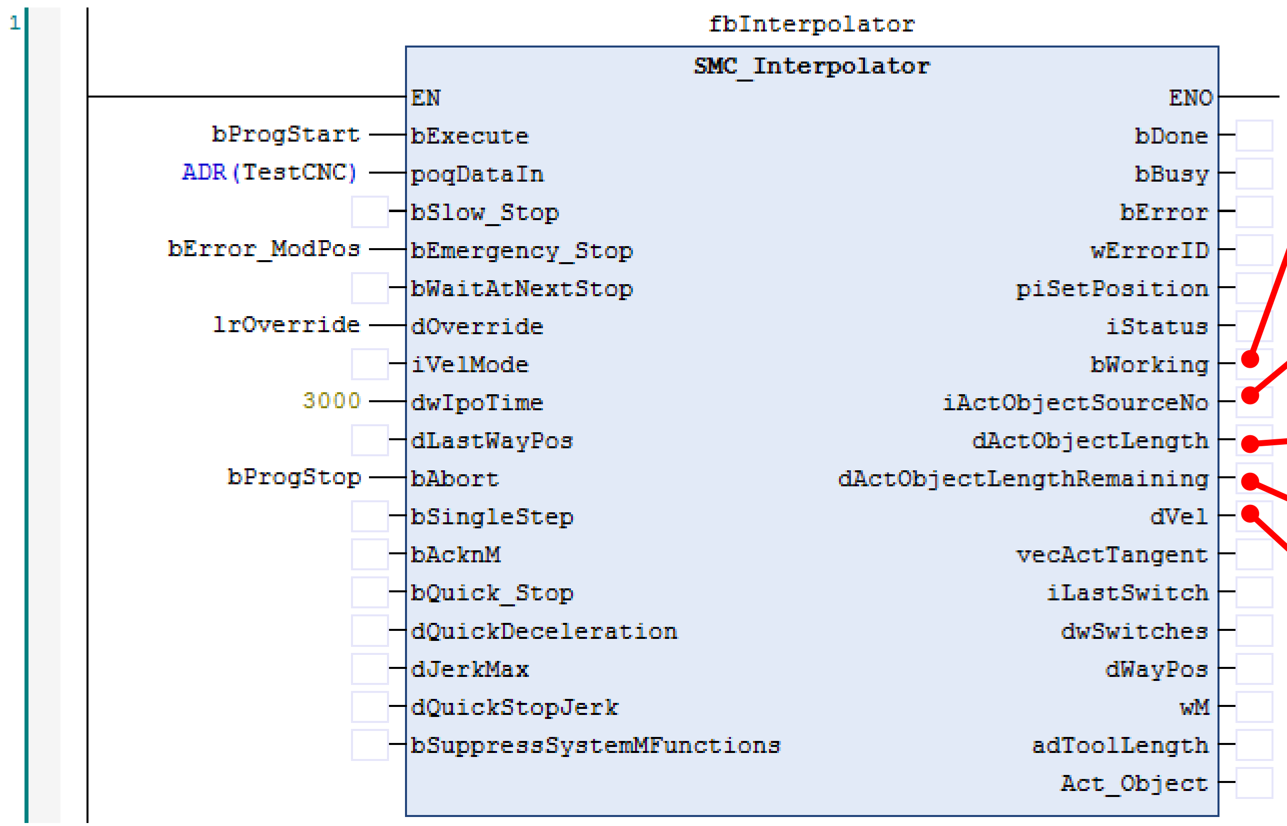
This enumeration variable reflects the current status of the function block defined in SMC_INT_STATUS.

SMC_INT_STATUS

Name	Initial	Comment
IPO_UNKNOWN	0	Internal state that may not occur after a complete pass of the SMC_Interpolator.
IPO_INIT	1	Initialization state; movement not started yet
IPO_ACCEL	2	Interpolation ongoing; currently accelerating
IPO_CONSTANT	3	Interpolation ongoing with constant velocity
IPO_DECEL	4	Interpolation ongoing; currently decelerating
IPO_FINISHED	5	Interpolation done
IPO_WAIT	6	Interpolation ongoing; currently waiting for example by a stop input or a M-command
IPO_INCREASING_ACCEL	7	Interpolation ongoing; currently building up the acceleration
IPO_DECREASING_ACCEL	8	Interpolation ongoing; currently ramping down the acceleration
IPO_INCREASING_DECEL	9	Interpolation ongoing; currently building up the deceleration
IPO_DECREASING_DECEL	10	Interpolation ongoing; currently ramping down the deceleration

ctrlX - *Example of CNC program - Description of the outputs of the interpolation module (3)*

- ***Interpolator Module (Description of outputs (3)):***



This output should be connected to the bEnable input of SMC_ControlAxisByPos. Will be TRUE if (and only if) list processing has started and has not yet finished

Value of iSourceLine_No member of the active SMC_GEOINFO object of "poqDataIn-queue". If (bWorking = FALSE), the value is set to "-1".

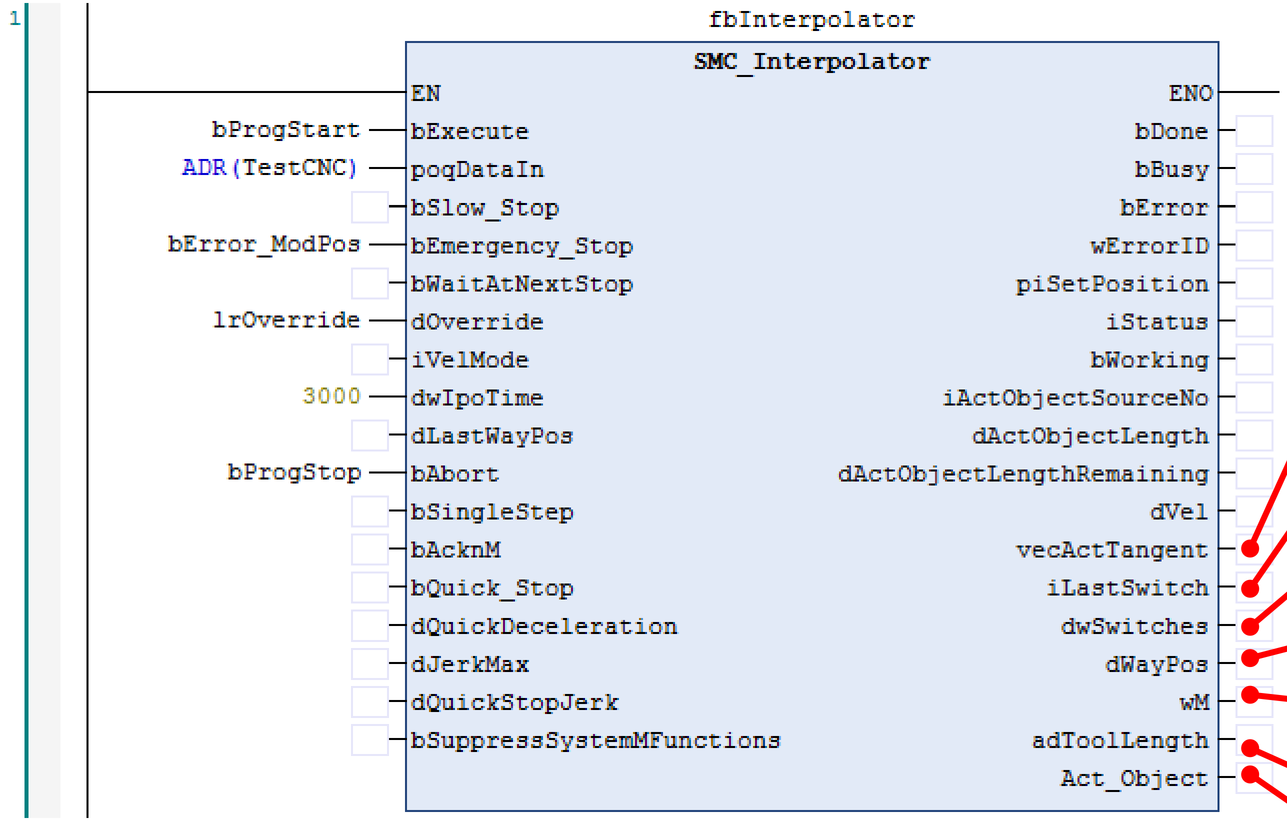
The length of the current object; valid if bWorking = TRUE.

The remaining length of the current object; valid if bWorking = TRUE.

This variable contains the current trajectory speed.

ctrlX - *Example of CNC program - Description of the outputs of the interpolation module (4)*

- *Interpolator Module (Description of outputs (4)):*



```
1                              fbInterpolator
                             SMC_Interpolator
                    EN                                   ENO
bProgStart       ── bExecute                             bDone
ADR(TestCNC)     ── poqDataIn                            bBusy
                    bSlow_Stop                           bError
bError_ModPos    ── bEmergency_Stop                      wErrorID
                    bWaitAtNextStop                      piSetPosition
lrOverride       ── dOverride                            iStatus
                    iVelMode                             bWorking
3000             ── dwIpoTime                            iActObjectSourceNo
                    dLastWayPos                          dActObjectLength
bProgStop        ── bAbort                               dActObjectLengthRemaining
                    bSingleStep                          dVel
                    bAcknM                               vecActTangent
                    bQuick_Stop                          iLastSwitch
                    dQuickDeceleration                   dwSwitches
                    dJerkMax                             dWayPos
                    dQuickStopJerk                       wM
                    bSuppressSystemMFunctions            adToolLength
                                                         Act_Object
```

This structure contains the tangent of the trajectory, vector

This output contains the number of the last switch passed.
Note: If multiple switches have been passed in one cycle, only the last one will be mentioned.

This DWORD describes the current switching state of all switches 1 32.
Bit0 of the DWORD represents switch 1, Bit31 represents switch 32.
Unlike iLastSwitch, this bit field will also have multiple switches in a cycle.

See the dLastWAyPos entry.

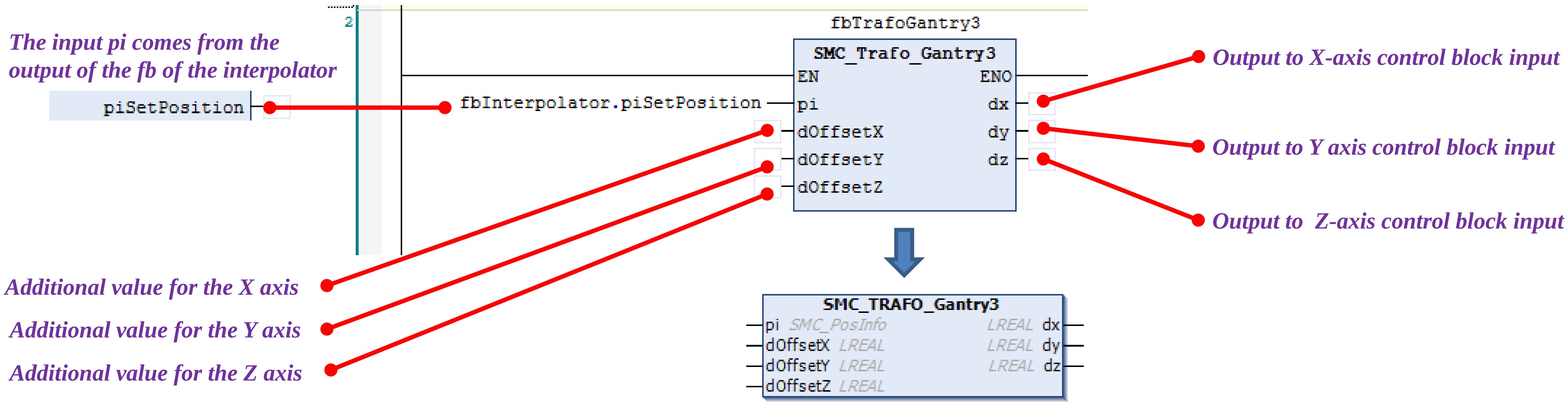
If the interpolator passes an M function, this output will be set to the value associated with the M function. The interpolator will stop until the M function is acknowledged by the bAcknM input.

Parameters for tool length compensation (set by G43 I/J/K)

Pointer to the currently interpolated path element. Possibly 0.

ctrlX - *Example of CNC program - Description of the inputs / outputs of the transformation module*

- *The next module linked after the interpolator and that we will use in the example is the following, SMC_Trafo_Gantry3*



The input pi comes from the output of the fb of the interpolator

Output to X-axis control block input

Output to Y axis control block input

Output to Z-axis control block input

Additional value for the X axis

Additional value for the Y axis

Additional value for the Z axis

On page 28 you can see the connectivity of the different elements

ctrlX- Example of CNC program - Description of the inputs of the motion control module

- *The last of the modules is in charge of generating the movement of the axis (Inputs Module):*

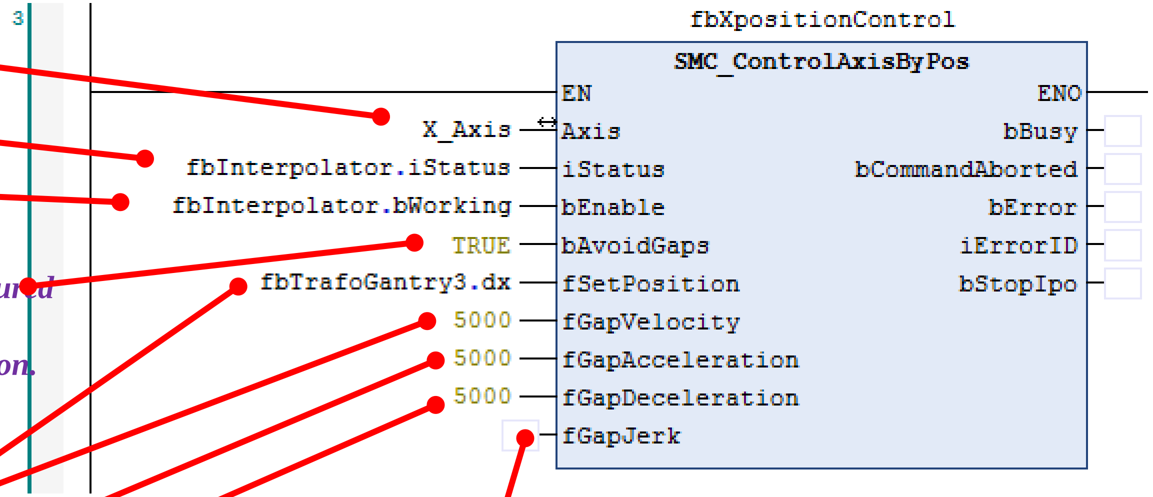
Name of the axis to use

SMC_Interpolator instance status

Interpolator in “Working” state that activates the “Enable” of the module

TRUE: start position tracking
If the velocity exceeds the limit value set in AXIS_REF_SM3.fSWMaxVelocity and configured in the drive dialog with the Maximum Values setting, the bStopIpo output is set. The axis moves to the position with the values fGapVelocity, fGapAcceleration, and fGapDeceleration. Upon reaching the set position, bStopIpo is set to FALSE.

Position value that should normally come from the transform block



```
                                        fbXpositionControl
                                     SMC_ControlAxisByPos
                               EN                             ENO
                     X_Axis -- Axis                         bBusy
    fbInterpolator.iStatus -- iStatus              bCommandAborted
   fbInterpolator.bWorking -- bEnable                      bError
                      TRUE -- bAvoidGaps                  iErrorID
        fbTrafoGantry3.dx -- fSetPosition                bStopIpo
                      5000 -- fGapVelocity
                      5000 -- fGapAcceleration
                      5000 -- fGapDeceleration
                            -- fGapJerk
```

Be careful with this values if we are working in "Simulation" mode or with real axes. In general, in simulation we should have it much higher than with the real axes.

Speed in [u/s]

Acceleration in [u/s2]

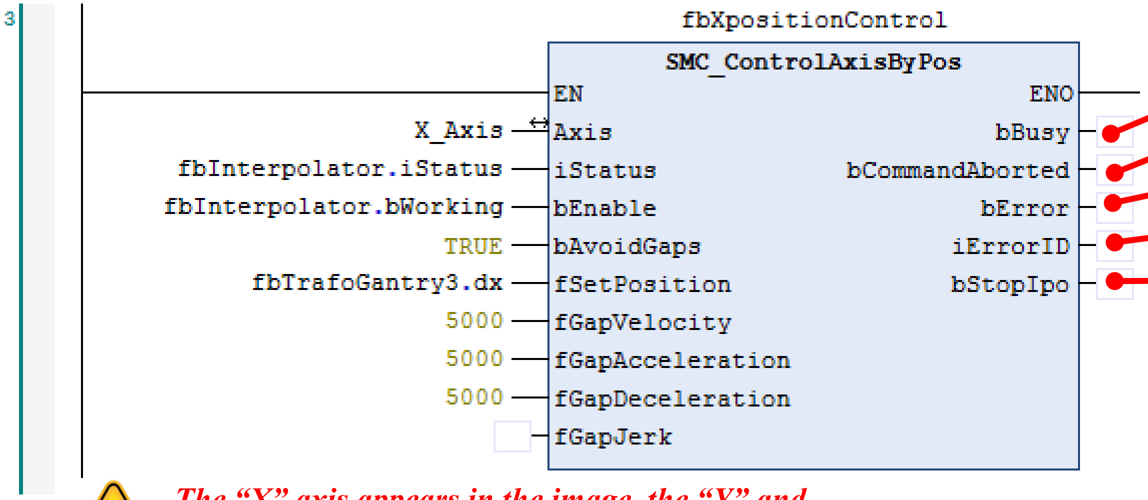
Deceleration in [u/s2]
Note: Also used if bAvoidGaps is FALSE, to stop when disabled

Jerk by jump bypass on [u/s3]
Note: also used if bAvoidGaps is FALSE, to stop when disabled

The “X” axis appears in the image, the “Y” and “Z” axes work in the same way and initially they should maintain the same operating values.

ctrlX - Example of CNC program - Description of the outputs of the motion control module

- *The last of the modules is in charge of generating the movement of the axis (Outputs Module):*



Module Activated

Command aborted by the execution of another axis control module

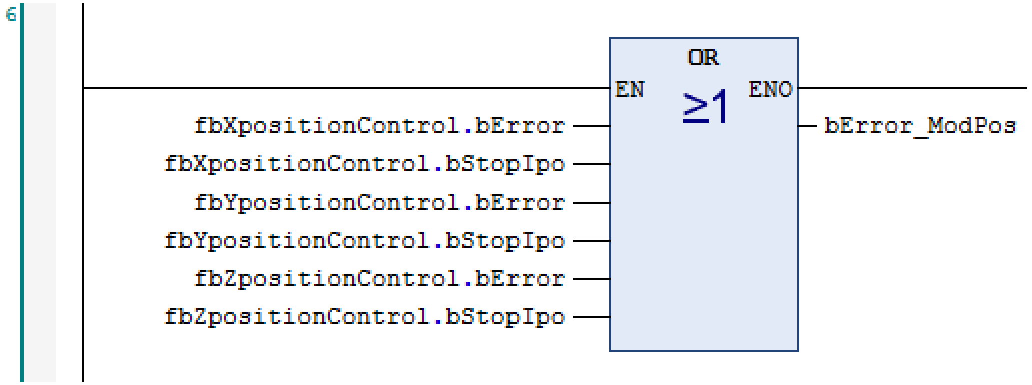
Error activated

Error Ident

If TRUE: A jump in velocity or position has occurred and adaptation to the new position is being executed.
If bStopIpo is connected to the EmergencyStop input of the SMC_Interpolator instance, the interpolator waits until the axis is correctly positioned.
See also: SMC_ERROR (iErrorID)

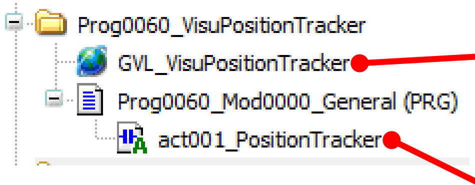
The "X" axis appears in the image, the "Y" and "Z" axes work in the same way and initially they should maintain the same operating values.

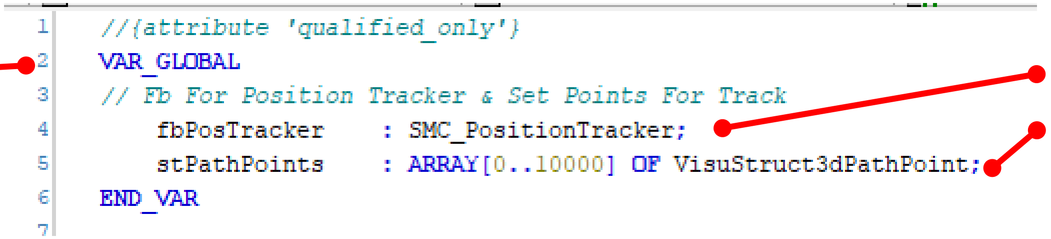
- *At the end of the interpolation module, an "Or" is performed on all the errors of the positioning modules and the errors of the block to carry out a wait in the interpolator, as detailed in the bStopIpo output of the previous block.*


6
OR
EN
≥1
ENO
fbXpositionControl.bError
fbXpositionControl.bStopIpo
fbYpositionControl.bError
fbYpositionControl.bStopIpo
fbZpositionControl.bError
fbZpositionControl.bStopIpo
bError_ModPos

ctrlX - Example of CNC program – Position Track To visualize the CNC program on the screen

- *In the following folder we will establish the data for the display of the values on the screen:*





```
//{attribute 'qualified_only'}
VAR_GLOBAL
// Fb For Position Tracker & Set Points For Track
     fbPosTracker      : SMC_PositionTracker;
     stPathPoints      : ARRAY[0..10000] OF VisuStruct3dPathPoint;
END_VAR
```

Module fb for the control of the Track of Positions

Array with the structure of points

Output value for the display. This output is a fb and is created in the general variables folder

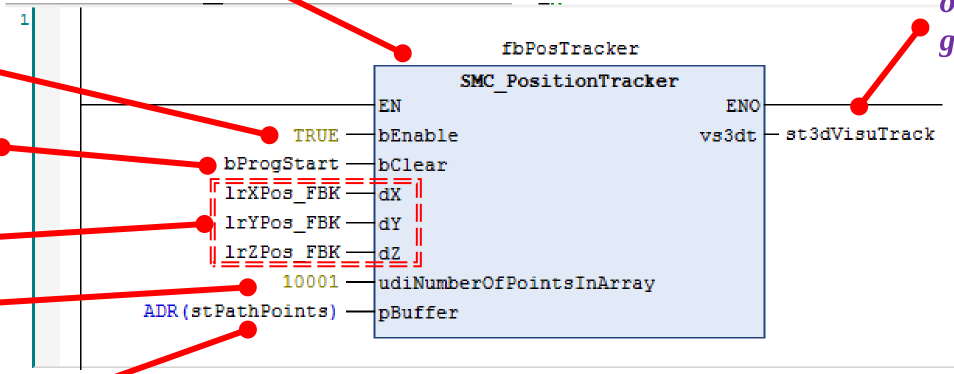
Activate Module (Always active with “TRUE”)

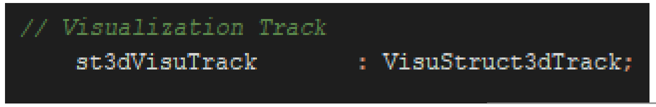
Clear block with program start mark (Cycle)

Current positions of the axes (X,Y,Z)

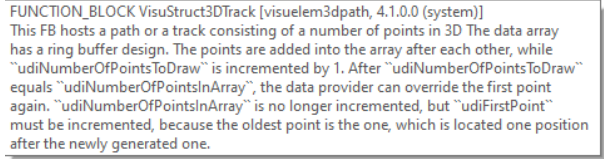
Number of points used in the array

Point structure from the system variable “VisuStruct3dPathPoint”



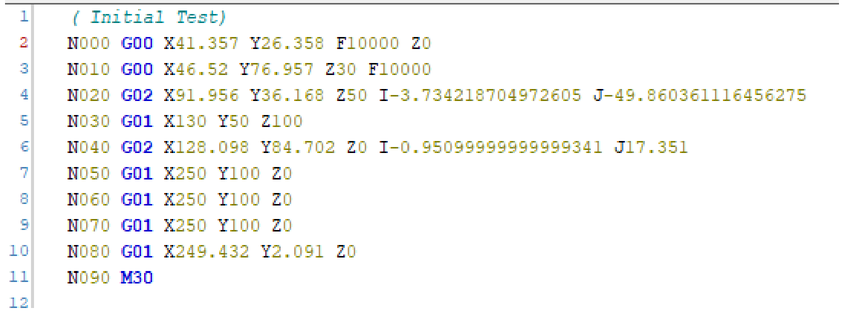


```
// Visualization Track
    st3dVisuTrack         : VisuStruct3dTrack;
```

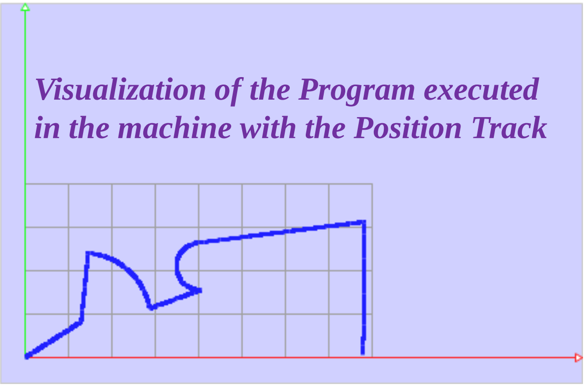


FUNCTION_BLOCK VisuStruct3DTrack [visuelem3dpath, 4.1.0.0 (system)]
This FB hosts a path or a track consisting of a number of points in 3D The data array has a ring buffer design. The points are added into the array after each other, while ``udiNumberOfPointsToDraw`` is incremented by 1. After ``udiNumberOfPointsToDraw`` equals ``udiNumberOfPointsInArray``, the data provider can override the first point again. ``udiNumberOfPointsInArray`` is no longer incremented, but ``udiFirstPoint`` must be incremented, because the oldest point is the one, which is located one position after the newly generated one.

Programa CNC



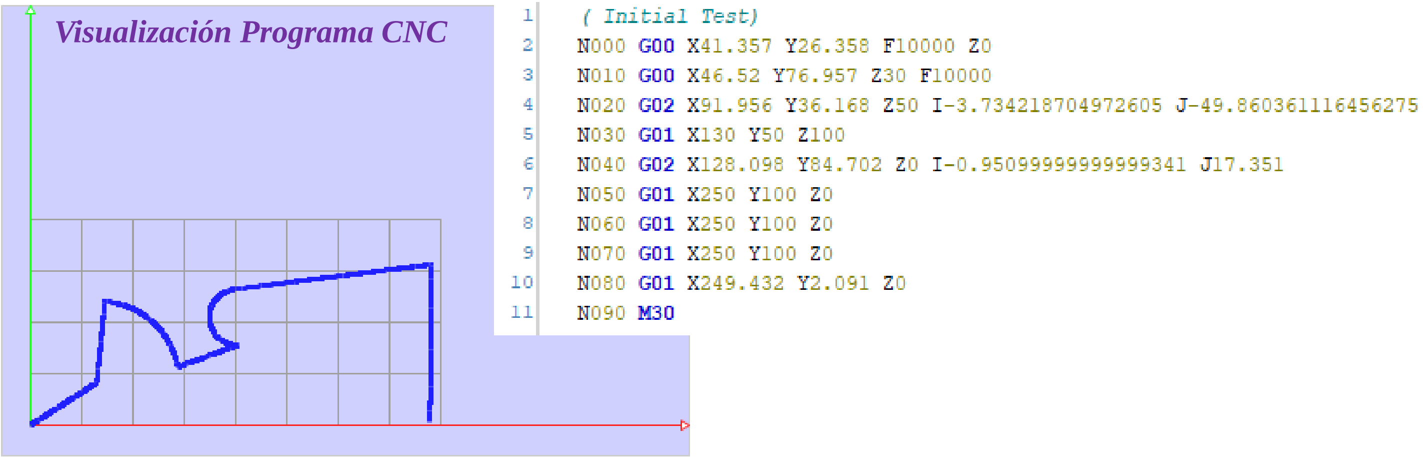
```
( Initial Test)
N000 G00 X41.357 Y26.358 F10000 Z0
N010 G00 X46.52 Y76.957 Z30 F10000
N020 G02 X91.956 Y36.168 Z50 I-3.734218704972605 J-49.860361116456275
N030 G01 X130 Y50 Z100
N040 G02 X128.098 Y84.702 Z0 I-0.9509999999999341 J17.351
N050 G01 X250 Y100 Z0
N060 G01 X250 Y100 Z0
N070 G01 X250 Y100 Z0
N080 G01 X249.432 Y2.091 Z0
N090 M30
```



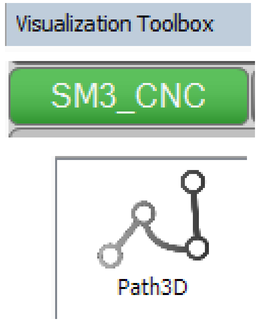
Visualization of the Program executed in the machine with the Position Track

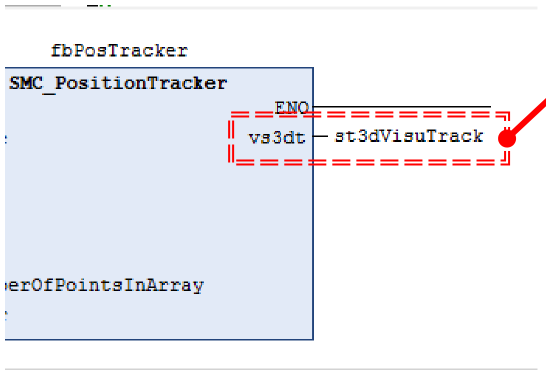
ctrlX - *Example of CNC program – Position Track To visualize the CNC program on the screen*

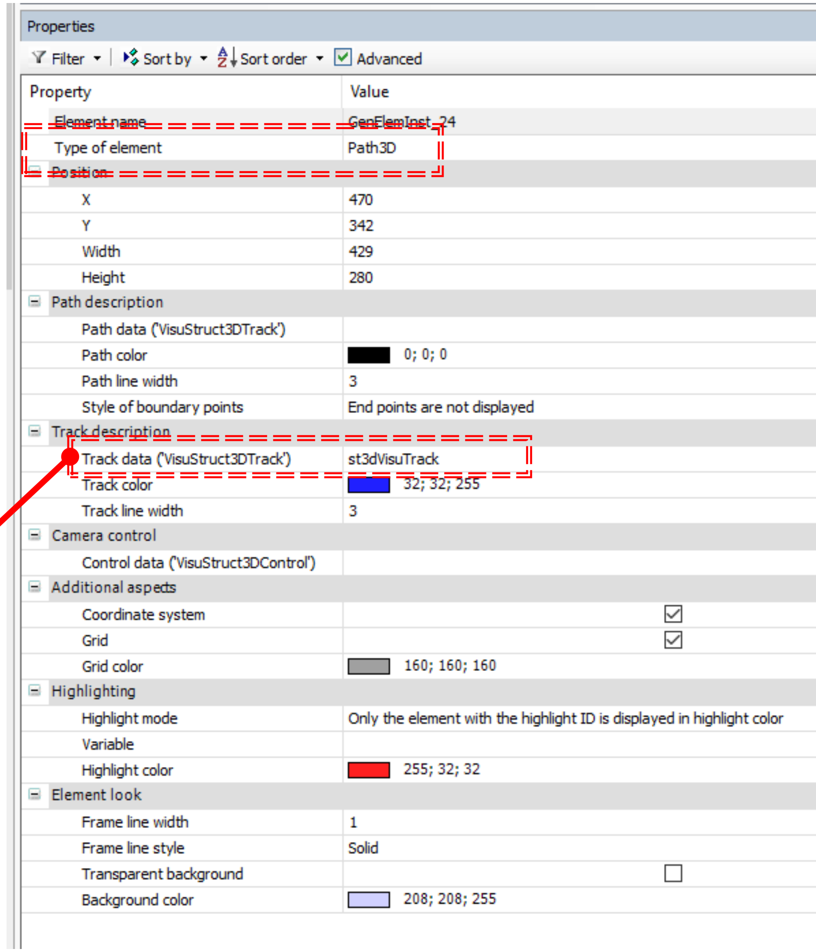
- ***The path visualization module is the Path3D located in "Visualization ToolBox" and specifically in "SM3_CNC"***



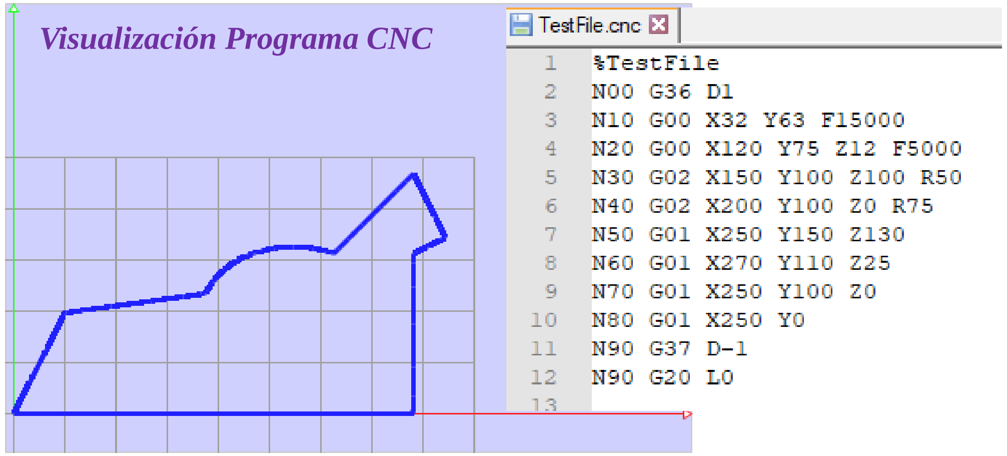
```
( Initial Test)
N000 G00 X41.357 Y26.358 F10000 Z0
N010 G00 X46.52 Y76.957 Z30 F10000
N020 G02 X91.956 Y36.168 Z50 I-3.734218704972605 J-49.860361116456275
N030 G01 X130 Y50 Z100
N040 G02 X128.098 Y84.702 Z0 I-0.95099999999999341 J17.351
N050 G01 X250 Y100 Z0
N060 G01 X250 Y100 Z0
N070 G01 X250 Y100 Z0
N080 G01 X249.432 Y2.091 Z0
N090 M30
```







Property	Value
Element name	GenElemInst_24
Type of element	Path3D
⊟ Position	
X	470
Y	342
Width	429
Height	280
⊟ Path description	
Path data ('VisuStruct3DTrack')	
Path color	■ 0; 0; 0
Path line width	3
Style of boundary points	End points are not displayed
⊟ Track description	
Track data ('VisuStruct3DTrack')	st3dVisuTrack
Track color	■ 32; 32; 255
Track line width	3
⊟ Camera control	
Control data ('VisuStruct3DControl')	
⊟ Additional aspects	
Coordinate system	☑
Grid	☑
Grid color	■ 160; 160; 160
⊟ Highlighting	
Highlight mode	Only the element with the highlight ID is displayed in highlight color
Variable	
Highlight color	■ 255; 32; 32
⊟ Element look	
Frame line width	1
Frame line style	Solid
Transparent background	☐
Background color	■ 208; 208; 255

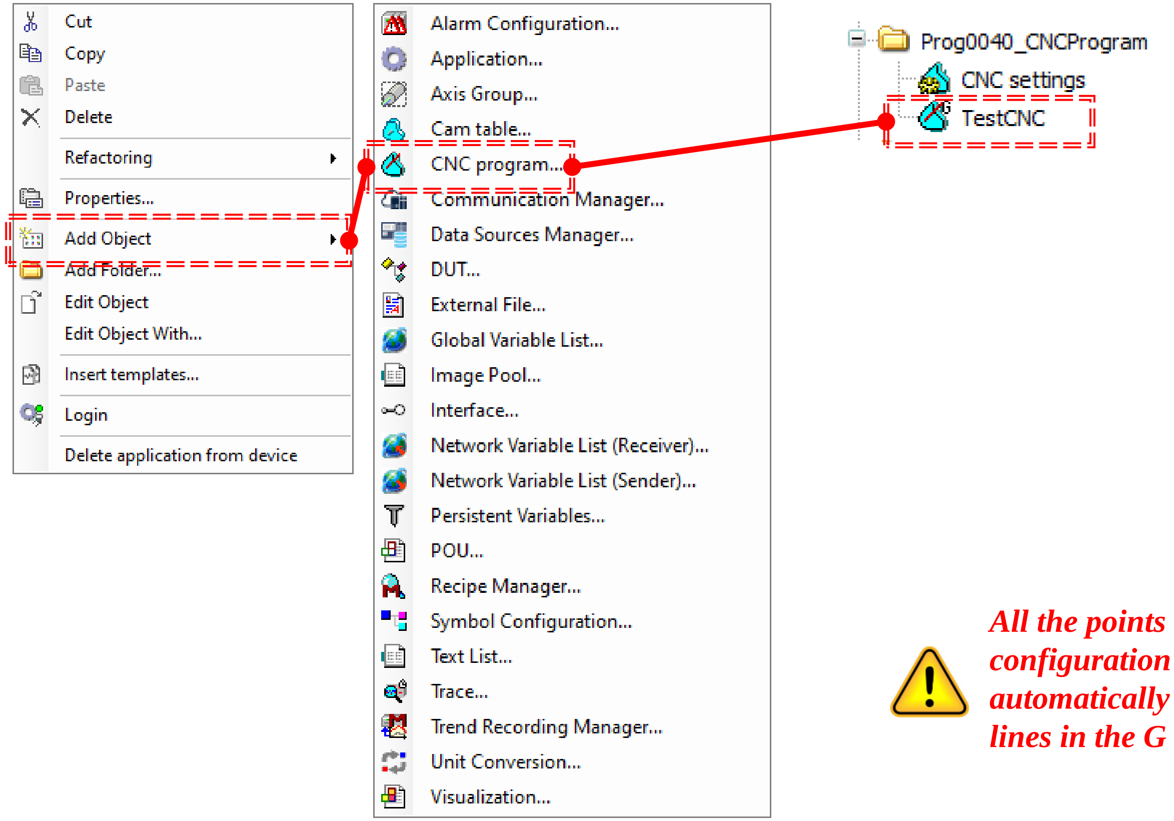


```
%TestFile
N00 G36 D1
N10 G00 X32 Y63 F15000
N20 G00 X120 Y75 Z12 F5000
N30 G02 X150 Y100 Z100 R50
N40 G02 X200 Y100 Z0 R75
N50 G01 X250 Y150 Z130
N60 G01 X270 Y110 Z25
N70 G01 X250 Y100 Z0
N80 G01 X250 Y0
N90 G37 D-1
N90 G20 L0
```

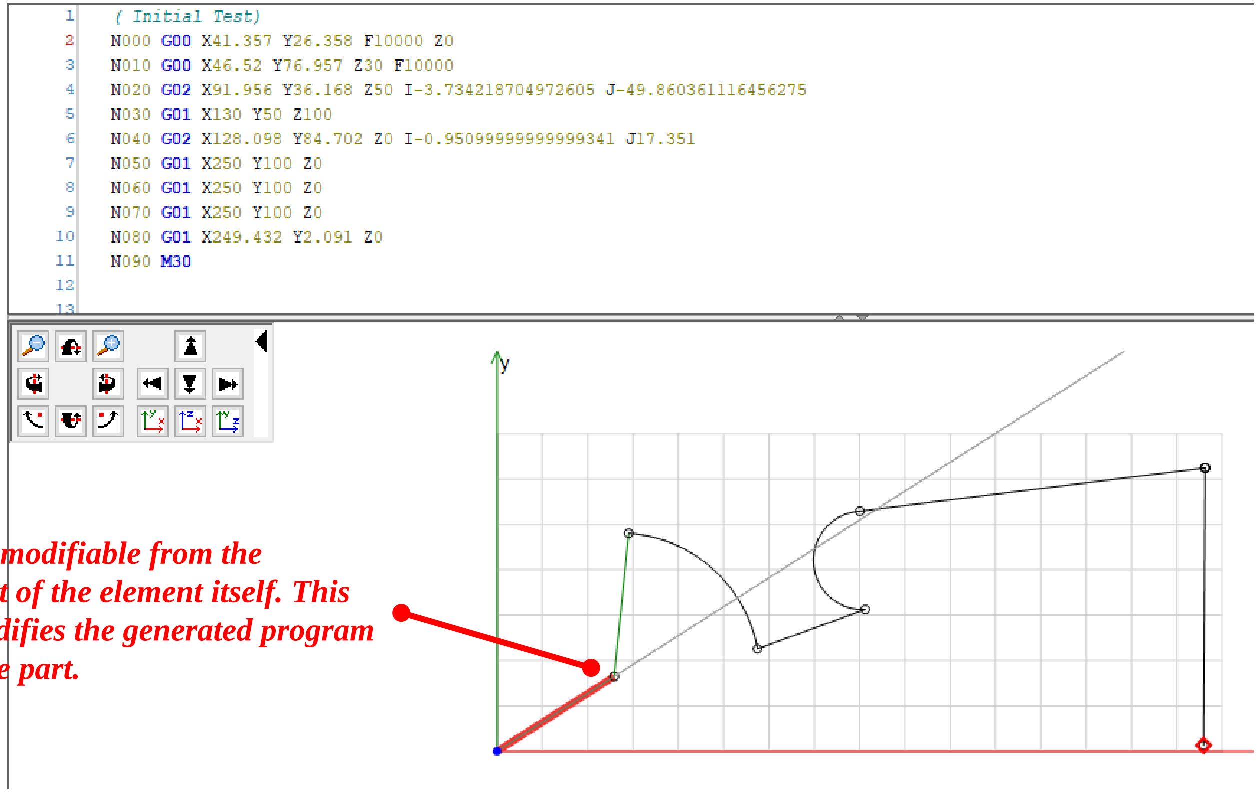
ctrlX - *Example of CNC program - Generation of CNC Programs*

- *The machine program can be managed in several ways:*
  - *Program located in ctrlX PLC Engineering*
  - *Program located on the PC*

- *We will start by using a machine program, generated from the software itself. First of all we will create a folder that we will call Prog0040_CNCProgram or whatever we want to call it and insert a program of the type "CNC Program"*



CNC program

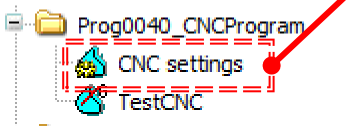


```
( Initial Test)
N000 G00 X41.357 Y26.358 F10000 Z0
N010 G00 X46.52 Y76.957 Z30 F10000
N020 G02 X91.956 Y36.168 Z50 I-3.734218704972605 J-49.860361116456275
N030 G01 X130 Y50 Z100
N040 G02 X128.098 Y84.702 Z0 I-0.9509999999999341 J17.351
N050 G01 X250 Y100 Z0
N060 G01 X250 Y100 Z0
N070 G01 X250 Y100 Z0
N080 G01 X249.432 Y2.091 Z0
N090 M30
```

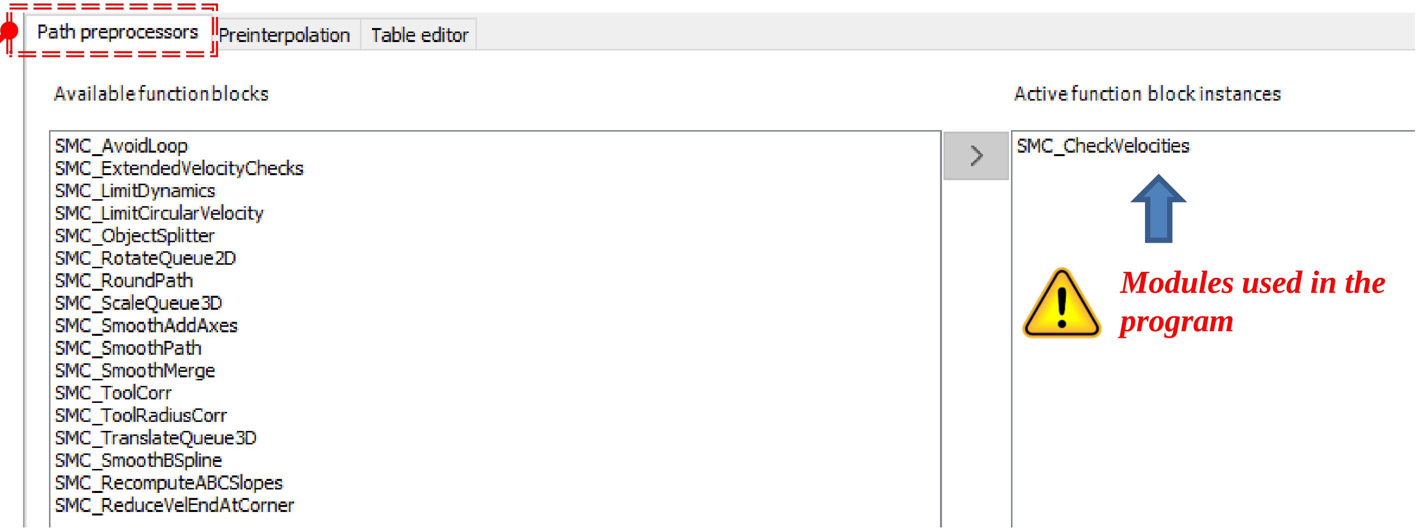
All the points are modifiable from the configuration part of the element itself. This automatically modifies the generated program lines in the G code part.

ctrlX - *Example of CNC program - Generation of CNC Programs*

- *We also have another module, apart from the created program, in which the CNC "Settings" appear*



Prog0040_CNCProgram
- CNC settings
- TestCNC



Path preprocessors | Preinterpolation | Table editor

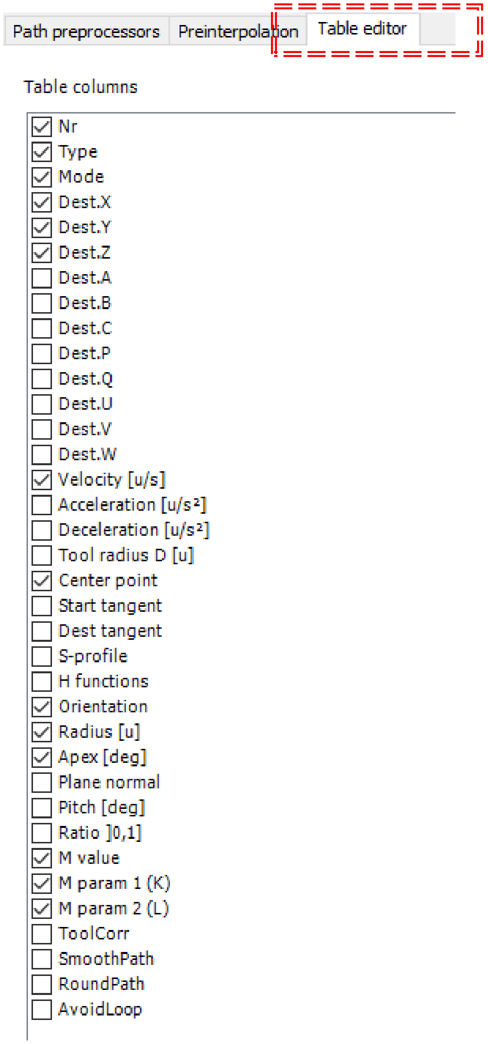
Available function blocks

SMC_AvoidLoop
SMC_ExtendedVelocityChecks
SMC_LimitDynamics
SMC_LimitCircularVelocity
SMC_ObjectSplitter
SMC_RotateQueue2D
SMC_RoundPath
SMC_ScaleQueue3D
SMC_SmoothAddAxes
SMC_SmoothPath
SMC_SmoothMerge
SMC_ToolCorr
SMC_ToolRadiusCorr
SMC_TranslateQueue3D
SMC_SmoothBSpline
SMC_RecomputeABCSlopes
SMC_ReduceVelEndAtCorner

Active function block instances

SMC_CheckVelocities

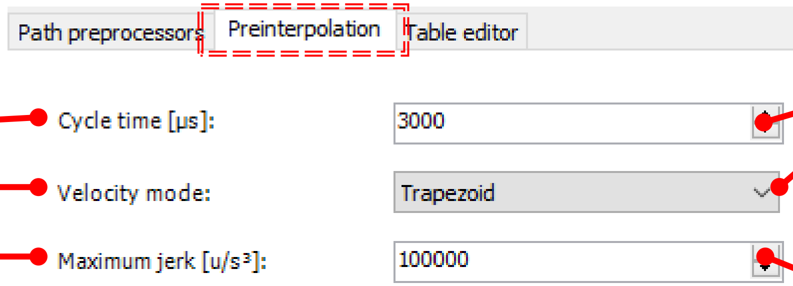
Modules used in the program



Path preprocessors | Preinterpolation | Table editor

Table columns

☑ Nr
☑ Type
☑ Mode
☑ Dest.X
☑ Dest.Y
☑ Dest.Z
☐ Dest.A
☐ Dest.B
☐ Dest.C
☐ Dest.P
☐ Dest.Q
☐ Dest.U
☐ Dest.V
☐ Dest.W
☑ Velocity [u/s]
☐ Acceleration [u/s²]
☐ Deceleration [u/s²]
☐ Tool radius D [u]
☑ Center point
☐ Start tangent
☐ Dest tangent
☐ S-profile
☐ H functions
☑ Orientation
☑ Radius [u]
☑ Apex [deg]
☐ Plane normal
☐ Pitch [deg]
☐ Ratio]0,1]
☑ M value
☑ M param 1 (K)
☑ M param 2 (L)
☐ ToolCorr
☐ SmoothPath
☐ RoundPath
☐ AvoidLoop

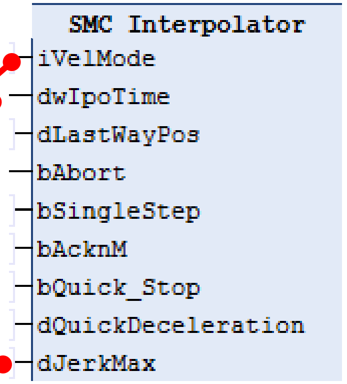


Path preprocessors | Preinterpolation | Table editor

Tiempo de ciclo — Cycle time [µs]: 3000

Tipo de rampa — Velocity mode: Trapezoid

Máximo Jerk — Maximum jerk [u/s³]: 100000



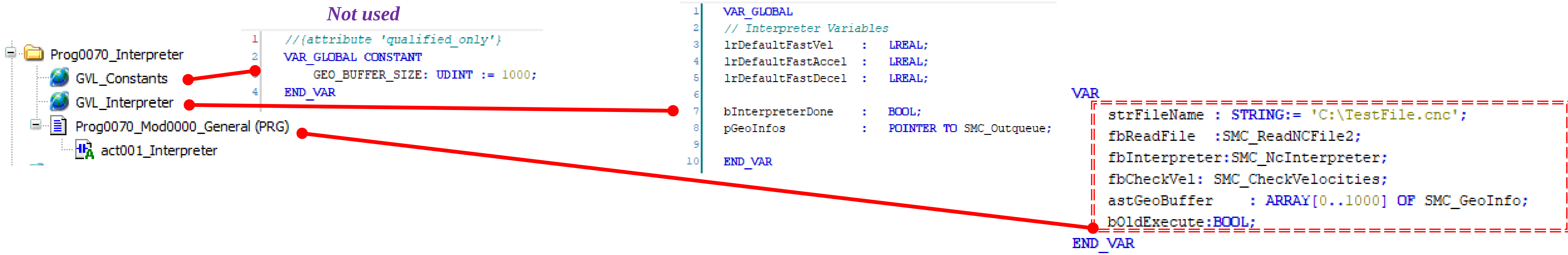
```
SMC_Interpolator
iVelMode
dwIpoTime
dLastWayPos
bAbort
bSingleStep
bAcknM
bQuick_Stop
dQuickDeceleration
dJerkMax
```

Some of these data are subject to the Interpolator

Control machine program from file on PC Or ctrl X

ctrlX - Example of CNC program - Use of CNC program from the PC or ctrlX

- *To control programs from the PC we are going to generate a new control module, in which the instruction interpreter will be used.*

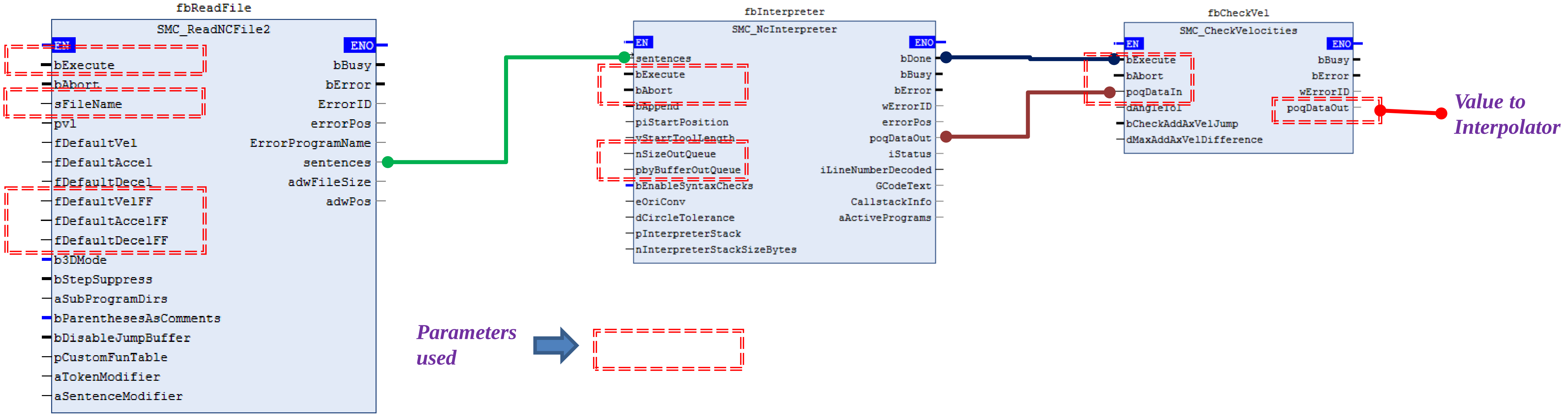


```
//{attribute 'qualified_only'}
VAR_GLOBAL CONSTANT
    GEO_BUFFER_SIZE: UDINT := 1000;
END_VAR
```

```
VAR_GLOBAL
// Interpreter Variables
lrDefaultFastVel    :   LREAL;
lrDefaultFastAccel  :   LREAL;
lrDefaultFastDecel  :   LREAL;

bInterpreterDone    :   BOOL;
pGeoInfos           :   POINTER TO SMC_Outqueue;

END_VAR
```

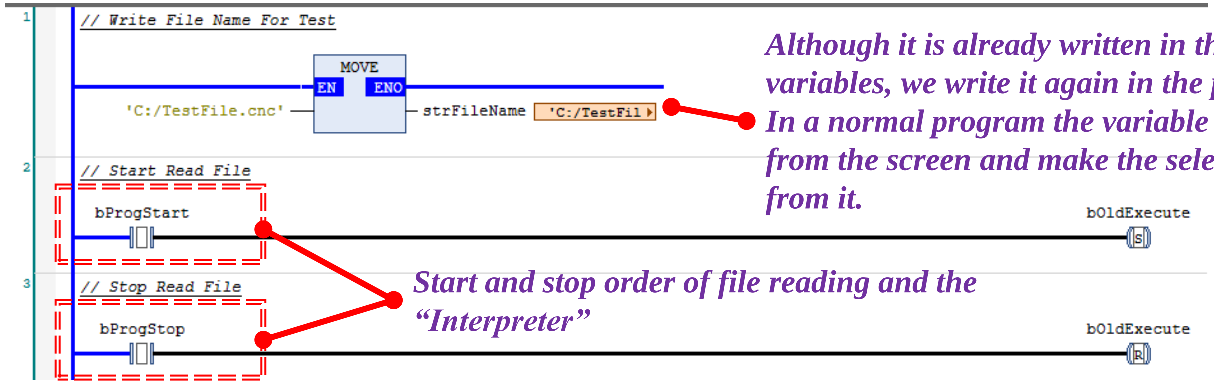
```
VAR
  strFileName : STRING:= 'C:\TestFile.cnc';
  fbReadFile  :SMC_ReadNCFile2;
  fbInterpreter:SMC_NcInterpreter;
  fbCheckVel: SMC_CheckVelocities;
  astGeoBuffer     : ARRAY[0..1000] OF SMC_GeoInfo;
  bOldExecute:BOOL;
END_VAR
```

- *Generic structure for the assignment of the parameters used in the various modules of the file reading*



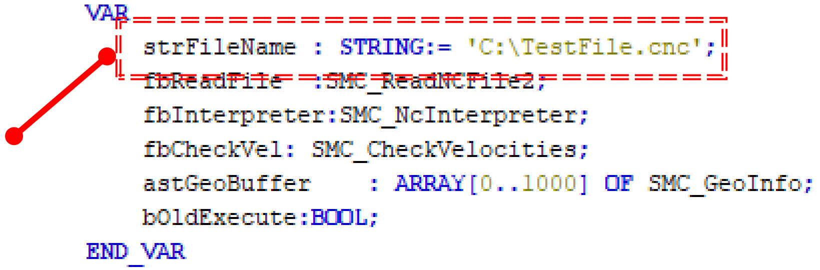
ctrlX - Example of CNC program - Use of CNC program from the PC or ctrlX

- *As we have commented, the program is in Ladder to facilitate its "understanding"*

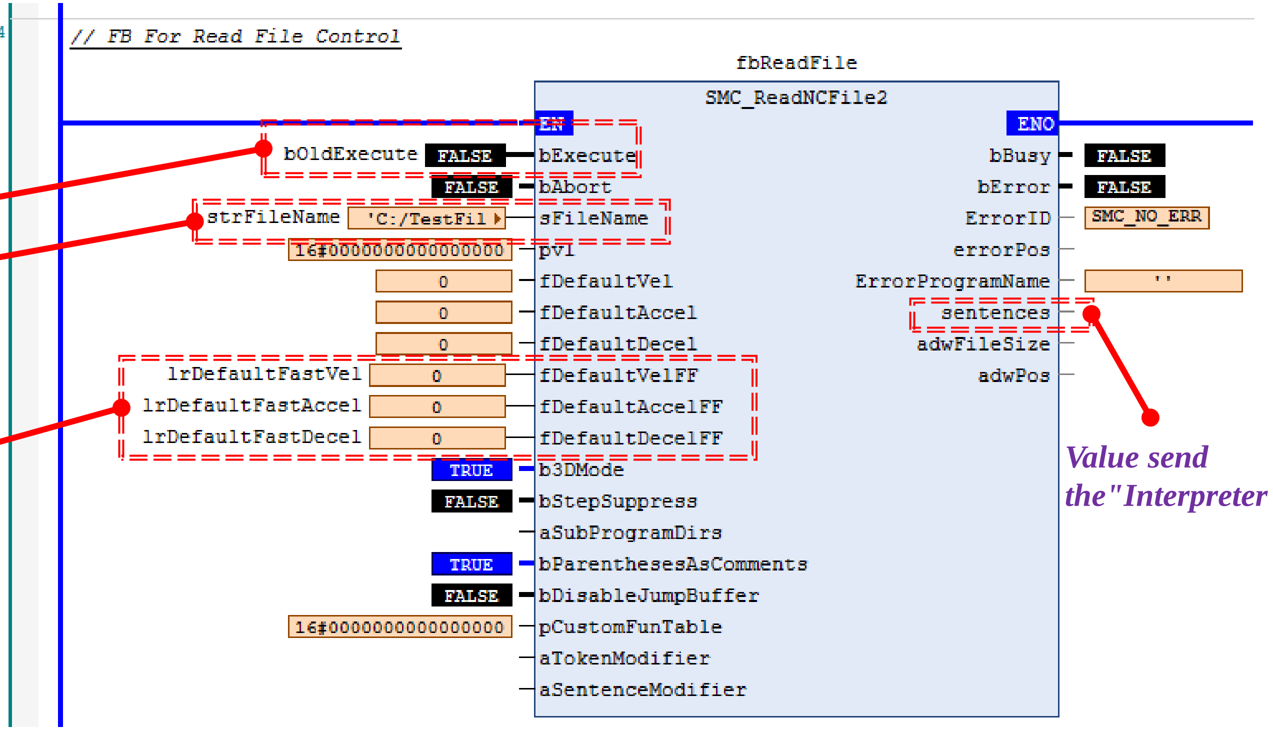


Although it is already written in the declaration of variables, we write it again in the program itself. In a normal program the variable should be controlled from the screen and make the selection of the program from it.

Start and stop order of file reading and the "Interpreter"



```
VAR
    strFileName : STRING:= 'C:\TestFile.cnc';
    fbReadFile   :SMC_ReadNCFile2;
    fbInterpreter:SMC_NcInterpreter;
    fbCheckVel: SMC_CheckVelocities;
    astGeoBuffer     : ARRAY[0..1000] OF SMC_GeoInfo;
    bOldExecute:BOOL;
END_VAR
```



```
4   // FB For Read File Control
                                        fbReadFile
                                     SMC_ReadNCFile2
                              EN                               ENO
bOldExecute  FALSE          - bExecute                   bBusy -  FALSE
             FALSE          - bAbort                    bError -  FALSE
strFileName  'C:/TestFil    - sFileName                ErrorID -  SMC_NO_ERR
16#0000000000000000         - pvl                     errorPos -
             0              - fDefaultVel     ErrorProgramName -  ''
             0              - fDefaultAccel          sentences -
             0              - fDefaultDecel         adwFileSize -
lrDefaultFastVel    0       - fDefaultVelFF             adwPos -
lrDefaultFastAccel  0       - fDefaultAccelFF
lrDefaultFastDecel  0       - fDefaultDecelFF
             TRUE           - b3DMode
             FALSE          - bStepSuppress
                            - aSubProgramDirs
             TRUE           - bParenthesesAsComments
             FALSE          - bDisableJumpBuffer
16#0000000000000000         - pCustomFunTable
                            - aTokenModifier
                            - aSentenceModifier
```

Activate Reading

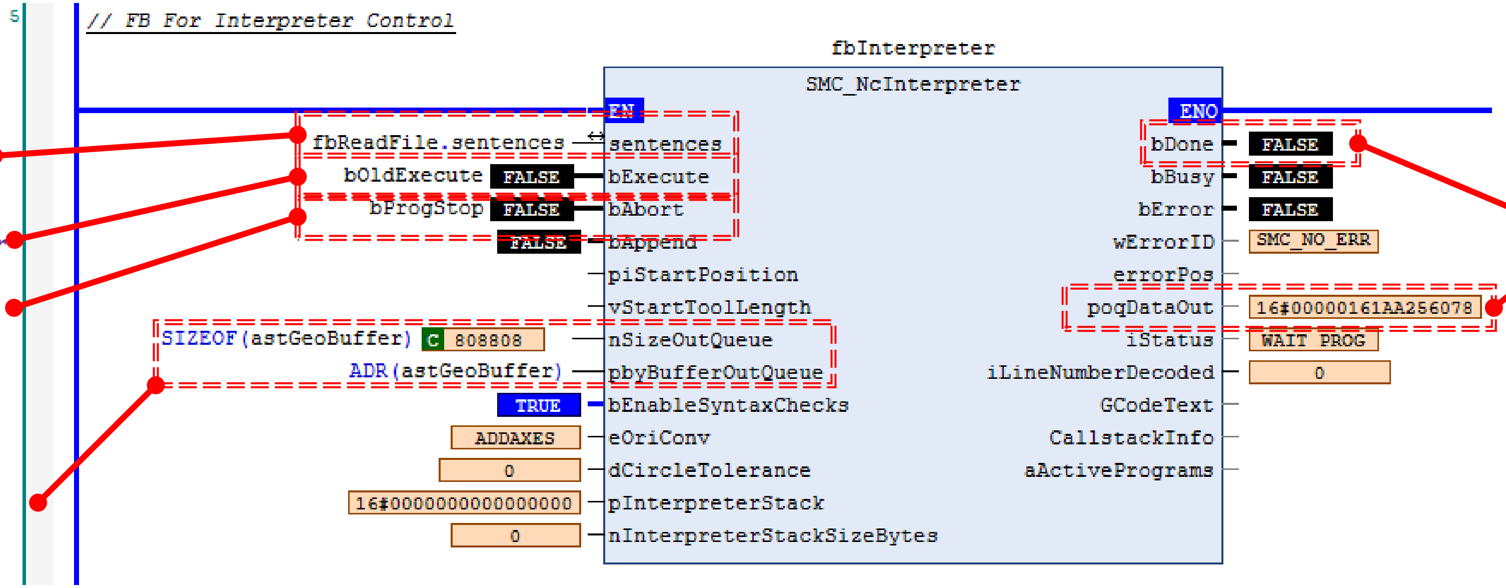
File to read

Default values if the FF variable is not in the CNC program

Value send the "Interpreter"

ctrlX - Example of CNC program - Use of CNC program from the PC or ctrlX

- *The interpreter manages the data received from the reading module and converts it into data understandable by the system:*



From the reading module

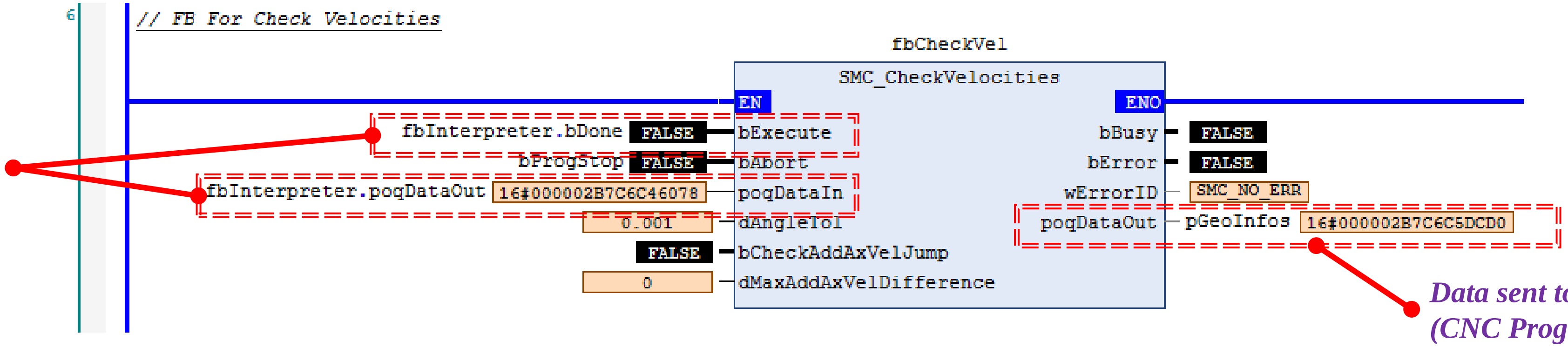
Activate interpreter

Stop Interpreter

Array for the data buffer

To speed control module

```
5   // FB For Interpreter Control

                                   fbInterpreter
                                 SMC_NcInterpreter
                        EN                                    ENO
fbReadFile.sentences ↔ sentences                         bDone   FALSE
bOldExecute  FALSE     bExecute                          bBusy   FALSE
bProgStop    FALSE     bAbort                            bError  FALSE
             FALSE     bAppend                        wErrorID   SMC_NO_ERR
                       piStartPosition                errorPos
                       vStartToolLength               poqDataOut 16#00000161AA256078
SIZEOF(astGeoBuffer) C 808808  nSizeOutQueue          iStatus    WAIT_PROG
ADR(astGeoBuffer)      pbyBufferOutQueue       iLineNumberDecoded 0
             TRUE      bEnableSyntaxChecks            GCodeText
             ADDAXES   eOriConv                       CallstackInfo
             0         dCircleTolerance               aActivePrograms
16#0000000000000000    pInterpreterStack
             0         nInterpreterStackSizeBytes
```



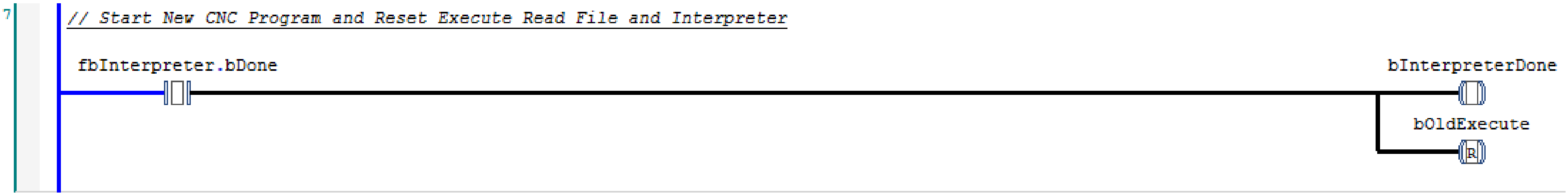
Data from the "Interpreter"

Data sent to the Interpolator (CNC Program)

```
6   // FB For Check Velocities

                                   fbCheckVel
                               SMC_CheckVelocities
                                 EN                          ENO
fbInterpreter.bDone  FALSE       bExecute               bBusy   FALSE
bProgStop            FALSE       bAbort                 bError  FALSE
fbInterpreter.poqDataOut 16#000002B7C6C46078 poqDataIn  wErrorID  SMC_NO_ERR
                     0.001       dAngleTol              poqDataOut  pGeoInfos 16#000002B7C6C5DCD0
                     FALSE       bCheckAddAxVelJump
                     0           dMaxAddAxVelDifference
```

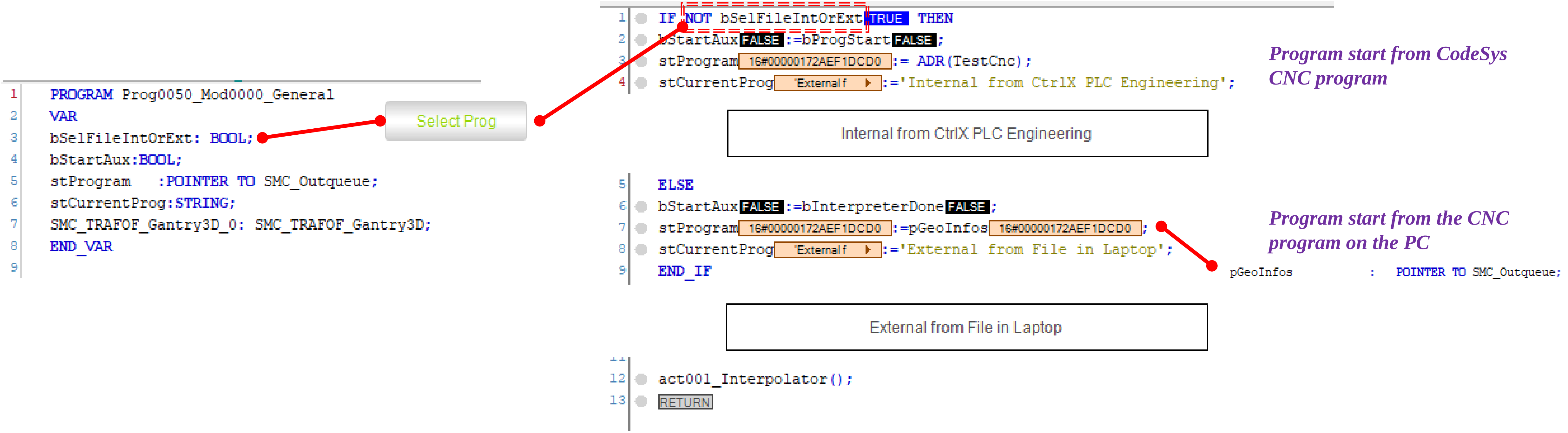
ctrlX - *Example of CNC program - Use of CNC program from the PC or ctrlX*

- *Once the reading has been carried out and the interpreter has transformed the information of the read program, the module activation signals are reset and the program start order is activated.*



```
// Start New CNC Program and Reset Execute Read File and Interpreter

fbInterpreter.bDone                                    bInterpreterDone
--| |--------------------------------------------------+--( )
                                                       |  bOldExecute
                                                       +--(R)
```

- *In the example program and to verify the operation of the internal CNC program from CodeSys and the CNC program located on the PC, the following program control lines have been added*



```
PROGRAM Prog0050_Mod0000_General
VAR
bSelFileIntOrExt: BOOL;
bStartAux:BOOL;
stProgram    :POINTER TO SMC_Outqueue;
stCurrentProg:STRING;
SMC_TRAFOF_Gantry3D_0: SMC_TRAFOF_Gantry3D;
END_VAR
```

```
IF NOT bSelFileIntOrExt TRUE THEN
bStartAux FALSE :=bProgStart FALSE ;
stProgram 16#00000172AEF1DCD0 := ADR(TestCnc);
stCurrentProg 'External f ▸ :='Internal from CtrlX PLC Engineering';
```

Internal from CtrlX PLC Engineering

Program start from CodeSys CNC program

```
ELSE
bStartAux FALSE :=bInterpreterDone FALSE ;
stProgram 16#00000172AEF1DCD0 :=pGeoInfos 16#00000172AEF1DCD0 ;
stCurrentProg 'External f ▸ :='External from File in Laptop';
END_IF
```

Program start from the CNC program on the PC

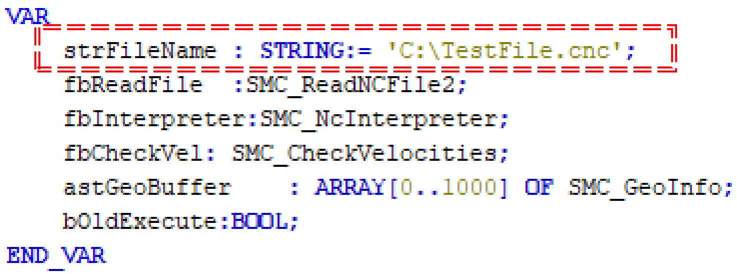
```
pGeoInfos          :    POINTER TO SMC_Outqueue;
```

External from File in Laptop

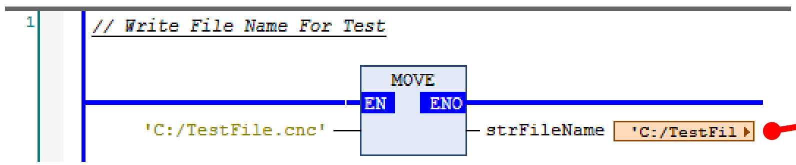
```
act001_Interpolator();
RETURN
```

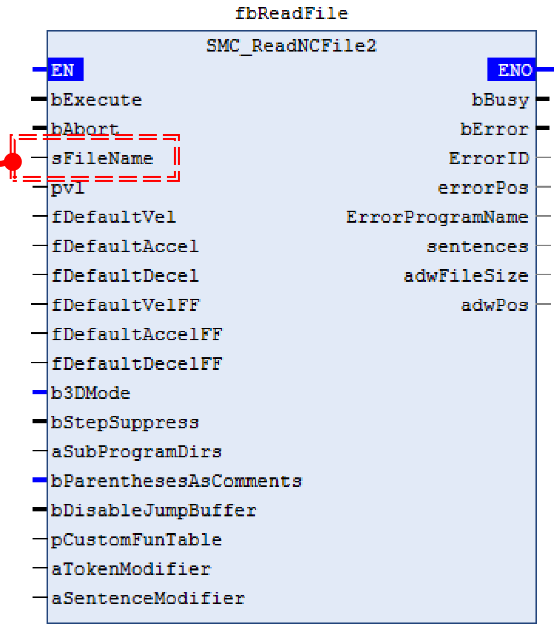
ctrlX - *Example of CNC program - Use of CNC program from the PC or ctrlX*

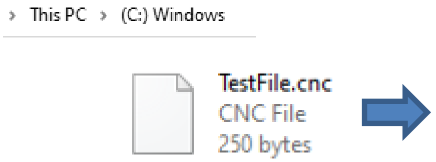
- *Program used for the example. The programs used must be in a specific path on the PC that allows access from the screen in the form of a selection based on the extension. The extension of the programs is .cnc, as can be seen in the attached images and the path and the name of the file must be placed on the sFileName parameter of the "SMC_ReadFile2" function.*

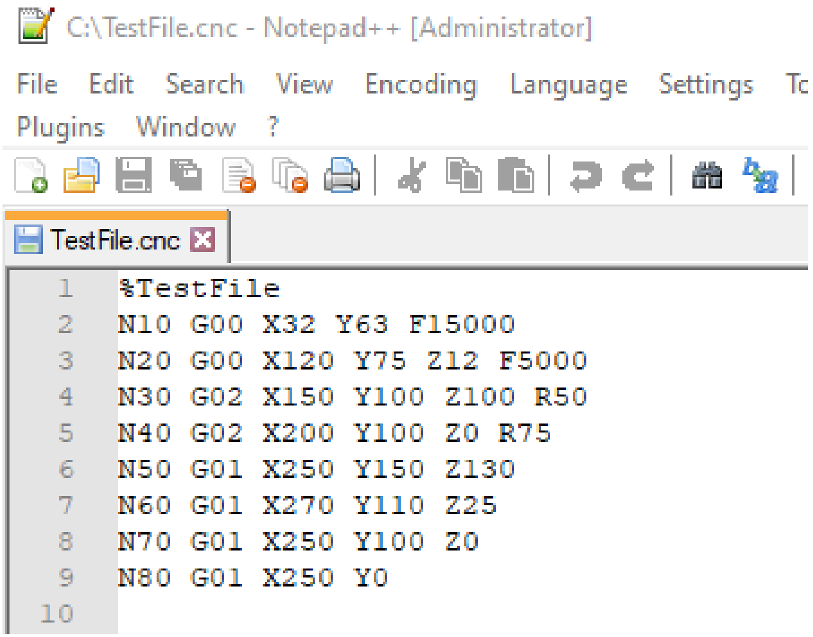


```
VAR
    strFileName : STRING:= 'C:\TestFile.cnc';
    fbReadFile  :SMC_ReadNCFile2;
    fbInterpreter:SMC_NcInterpreter;
    fbCheckVel: SMC_CheckVelocities;
    astGeoBuffer    : ARRAY[0..1000] OF SMC_GeoInfo;
    bOldExecute:BOOL;
END_VAR
```









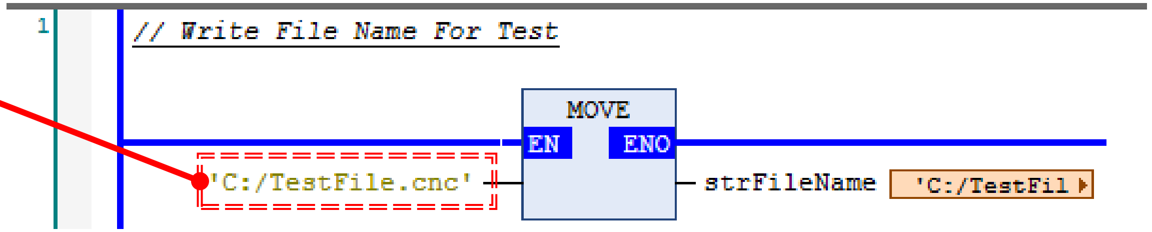
C:\TestFile.cnc - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tc
Plugins Window ?
TestFile.cnc

```
%TestFile
N10 G00 X32 Y63 F15000
N20 G00 X120 Y75 Z12 F5000
N30 G02 X150 Y100 Z100 R50
N40 G02 X200 Y100 Z0 R75
N50 G01 X250 Y150 Z130
N60 G01 X270 Y110 Z25
N70 G01 X250 Y100 Z0
N80 G01 X250 Y0
```

ctrlX - *Example of CNC program - Use of CNC program from the PC or ctrlX*

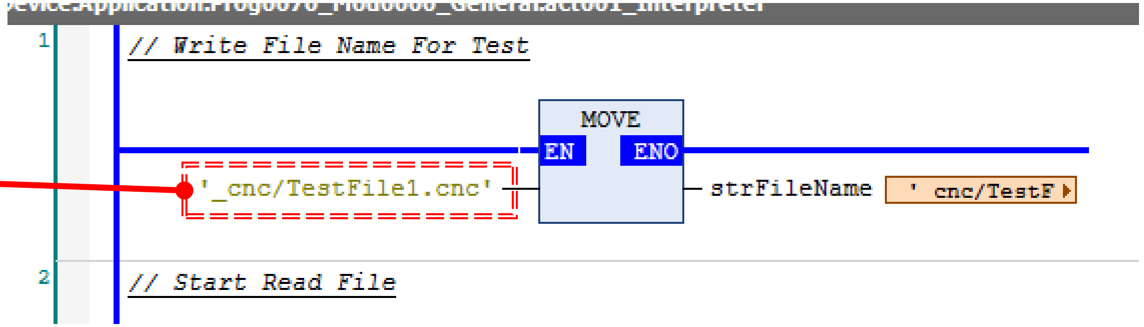
- *In the example we have been using the simulation mode, this supposes that the system works on the PC, at the moment in which we start working with the control, in this case the ctrlX path to use will not be on the PC but on control, so it must be taken into account*

Program path in simulation mode, using CNC programs located on the PC



```
// Write File Name For Test

'C:/TestFile.cnc' --[MOVE EN ENO]-- strFileName 'C:/TestFil
```

In Online mode the route is different and is located in the control itself

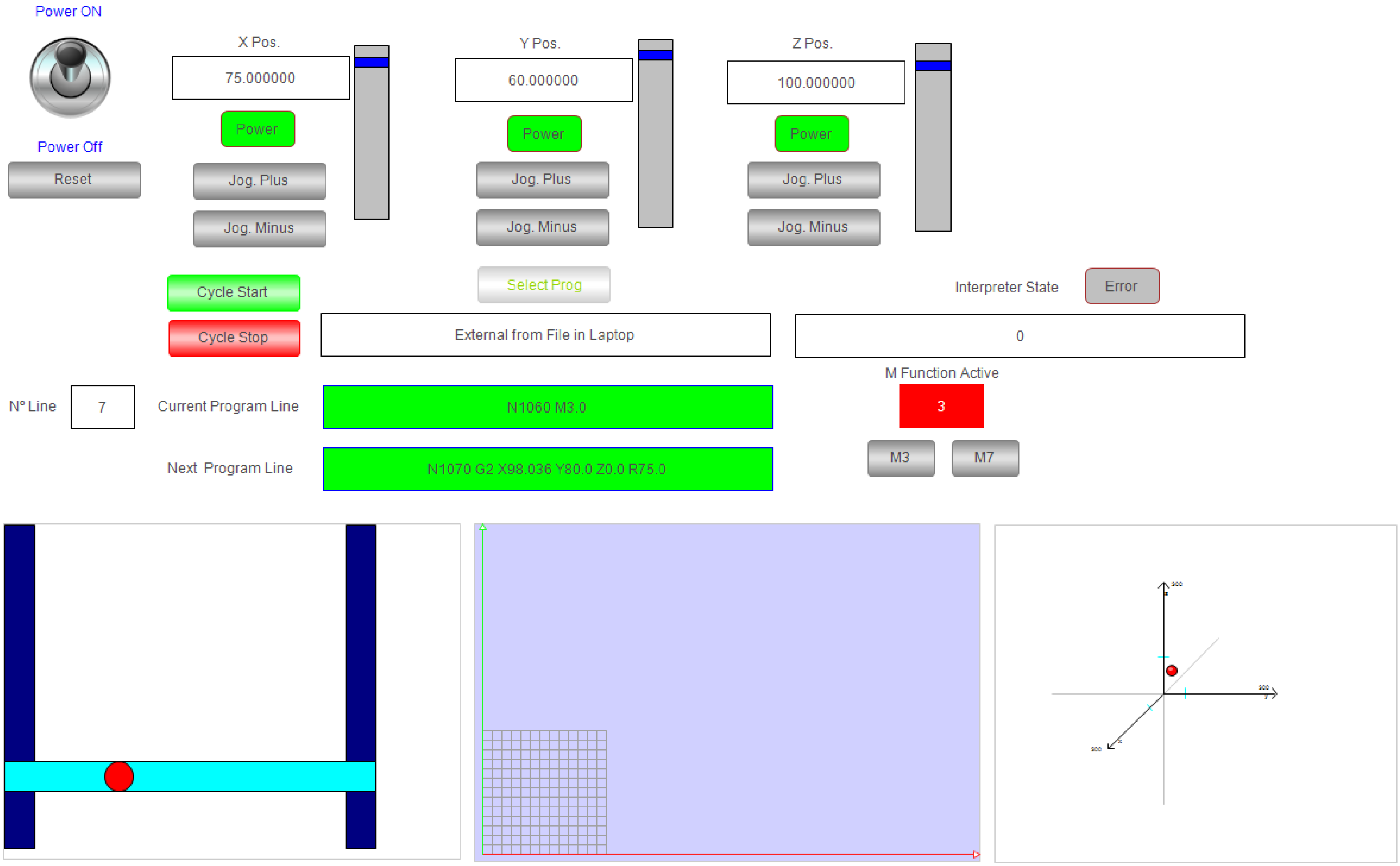


```
// Write File Name For Test

'_cnc/TestFile1.cnc' --[MOVE EN ENO]-- strFileName ' cnc/TestF

// Start Read File
```

Sending files to ctrlX are detailed later

ctrlX - *Example of CNC program - Simple example of control screen*

- ***Test Screen:***



X Pos. 75.000000

Y Pos. 60.000000

Z Pos. 100.000000

External from File in Laptop

Interpreter State 0

N° Line 7

Current Program Line N1060 M3.0

Next Program Line N1070 G2 X98.036 Y80.0 Z0.0 R75.0

M Function Active 3

M-functions

ctrlX - *Example of CNC program - Use of M Functions*

- *The M function is an additional functionality that allows starting and controlling certain actions within the G-Code interpolation.*

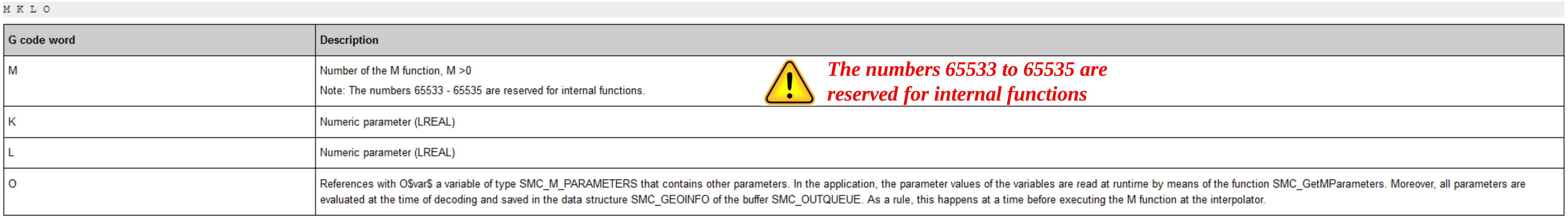
  Syntax:

  - The interpolator decelerates to speed 0, sending the number of the M function activated through the "vM" output and waits for confirmation within the interpolator module itself or with the use of "SMC_PreAcknowlefgeM"

  - The M functions are generated from the application and are not defined by the system

  - M functions can send up to a total of two values in Lreal format

  - These parameters can be "displayed" in the plc program using the SMC_GetMParameters module.



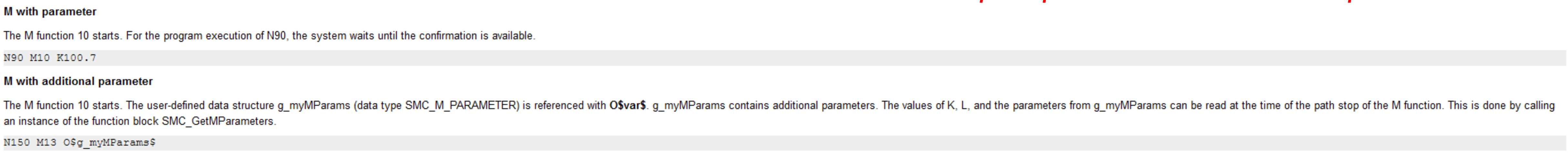
```
M K L O
```

G code word	Description
M	Number of the M function, M >0 Note: The numbers 65533 - 65535 are reserved for internal functions.
K	Numeric parameter (LREAL)
L	Numeric parameter (LREAL)
O	References with Ovar a variable of type SMC_M_PARAMETERS that contains other parameters. In the application, the parameter values of the variables are read at runtime by means of the function SMC_GetMParameters. Moreover, all parameters are evaluated at the time of decoding and saved in the data structure SMC_GEOINFO of the buffer SMC_OUTQUEUE. As a rule, this happens at a time before executing the M function at the interpolator.

⚠ *The numbers 65533 to 65535 are reserved for internal functions*

⚠ *The maximum number of functions in a CNC program is 64. This restriction can be circumvented by inserting a G75 Code before the first M function. G4 codes also count as M functions.*

Examples:



M with parameter

The M function 10 starts. For the program execution of N90, the system waits until the confirmation is available.

```
N90 M10 K100.7
```

M with additional parameter

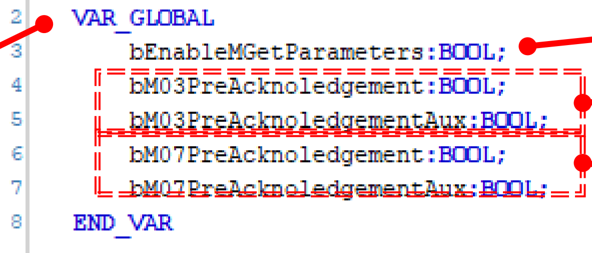
The M function 10 starts. The user-defined data structure g_myMParams (data type SMC_M_PARAMETER) is referenced with **Ovar**. g_myMParams contains additional parameters. The values of K, L, and the parameters from g_myMParams can be read at the time of the path stop of the M function. This is done by calling an instance of the function block SMC_GetMParameters.

```
N150 M13 O$g_myMParams$
```

ctrlX - *Example of CNC program - Use of M Functions*

- *The M functions must be configured as follows (Example of operation)*

Following the example of what we have seen so far, we are going to generate a folder to implement the control of the M functions.

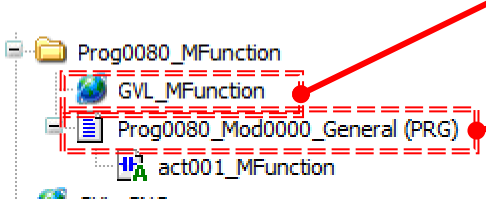


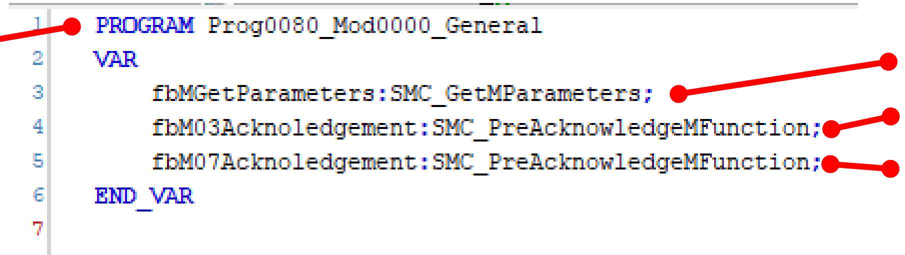
```
VAR_GLOBAL
    bEnableMGetParameters:BOOL;
    bM03PreAcknoledgement:BOOL;
    bM03PreAcknoledgementAux:BOOL;
    bM07PreAcknoledgement:BOOL;
    bM07PreAcknoledgementAux:BOOL;
END_VAR
```

Activate reading for the control of Marks with Parameters

Control bits for recognition of the M03 function

Control bits for recognition of the M07 function





```
PROGRAM Prog0080_Mod0000_General
VAR
    fbMGetParameters:SMC_GetMParameters;
    fbM03Acknoledgement:SMC_PreAcknowledgeMFunction;
    fbM07Acknoledgement:SMC_PreAcknowledgeMFunction;
END_VAR
```

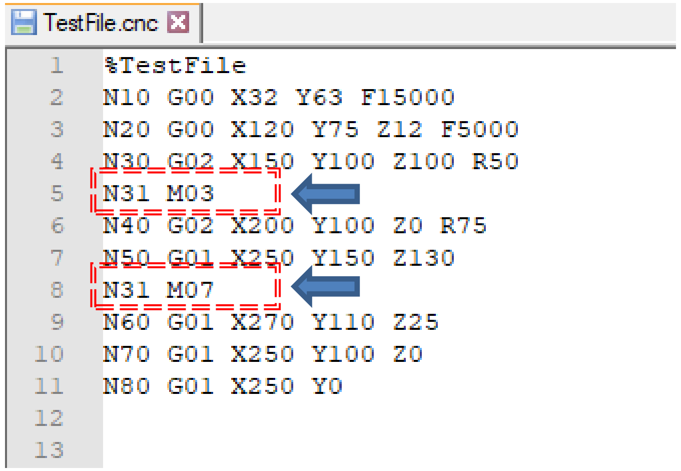
Control of M marks activated with Parameters

Recognition of function M03 (Example)

Recognition of the M07 function (Example)

- *The CNC program must be modified to test the M functions:*

The CNC program located in the PC is modified, incorporating two M functions, "M03" and "M07

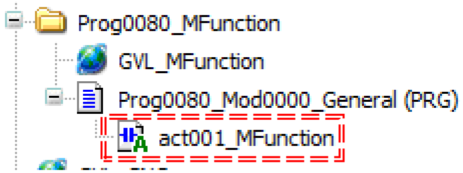


```
%TestFile
N10 G00 X32 Y63 F15000
N20 G00 X120 Y75 Z12 F5000
N30 G02 X150 Y100 Z100 R50
N31 M03
N40 G02 X200 Y100 Z0 R75
N50 G01 X250 Y150 Z130
N31 M07
N60 G01 X270 Y110 Z25
N70 G01 X250 Y100 Z0
N80 G01 X250 Y0
```

⚠ *The "M" functions can be used to control counters, cycles, etc.*

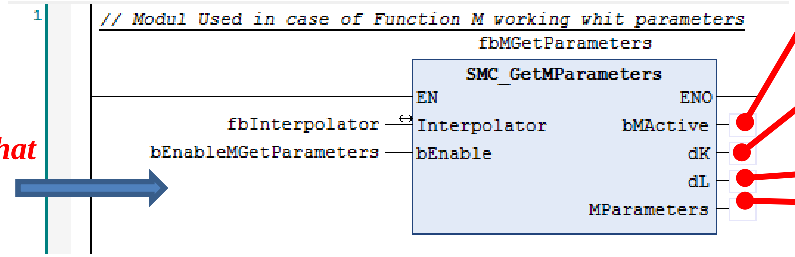
ctrlX - *Example of CNC program - Use of M Functions*

- *In the associated action we will control the brands used in the example.*



Prog0080_MFunction
GVL_MFunction
Prog0080_Mod0000_General (PRG)
act001_MFunction

General brand control block



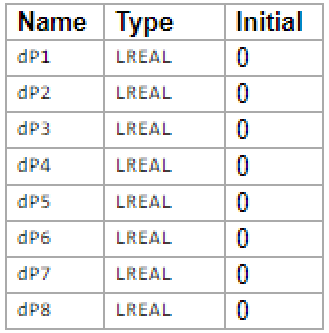
```
// Modul Used in case of Function M working whit parameters
                         fbMGetParameters
                        SMC_GetMParameters
                      -EN                      ENO-
       fbInterpolator -Interpolator        bMActive-
 bEnableMGetParameters -bEnable                  dK-
                                                 dL-
                                         MParameters-
```

Some active M function

K value associated with the active function

L value associated with the active function

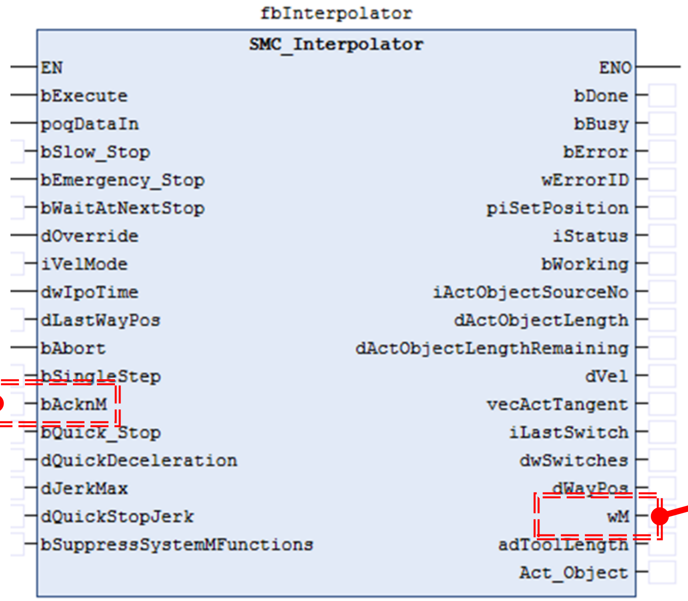
Parameters associated with the brand



Name	Type	Initial
dP1	LREAL	0
dP2	LREAL	0
dP3	LREAL	0
dP4	LREAL	0
dP5	LREAL	0
dP6	LREAL	0
dP7	LREAL	0
dP8	LREAL	0

Although this block indicates that there is an active brand, it does not show which brand is activated.

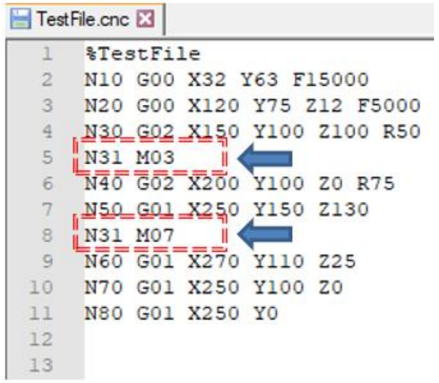
- *Remember that the interpolator can indicate the number of the M function activated*



```
                         fbInterpolator
                         SMC_Interpolator
-EN                                                  ENO-
-bExecute                                          bDone-
-poqDataIn                                         bBusy-
-bSlow_Stop                                       bError-
-bEmergency_Stop                                wErrorID-
-bWaitAtNextStop                            piSetPosition-
-dOverride                                       iStatus-
-iVelMode                                       bWorking-
-dwIpoTime                            iActObjectSourceNo-
-dLastWayPos                             dActObjectLength-
-bAbort                         dActObjectLengthRemaining-
-bSingleStep                                        dVel-
-bAcknM                                    vecActTangent-
-bQuick_Stop                                iLastSwitch-
-dQuickDeceleration                          dwSwitches-
-dJerkMax                                        dWayPos-
-dQuickStopJerk                                       wM-
-bSuppressSystemMFunctions                   adToolLength-
                                              Act_Object-
```

Recognition of the M function (any of them)

Possibly the best thing would be to perform the "Recognition" of the function with the individual control block.



TestFile.cnc

```
%TestFile
N10 G00 X32 Y63 F15000
N20 G00 X120 Y75 Z12 F5000
N30 G02 X150 Y100 Z100 R50
N31 M03
N40 G02 X200 Y100 Z0 R75
N50 G01 X250 Y150 Z130
N31 M07
N60 G01 X270 Y110 Z25
N70 G01 X250 Y100 Z0
N80 G01 X250 Y0
```

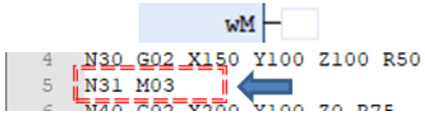
Function number M activated

ctrlX - *Example of CNC program - Use of M Functions*

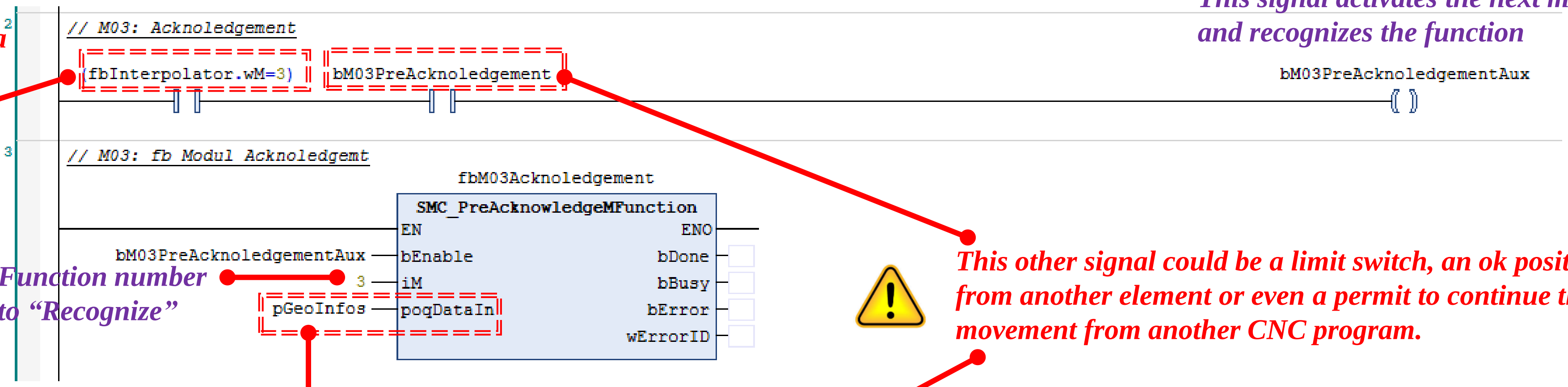
- *For the system to work in a controlled way we can use something similar to this:*

Function Recognition M03

Como el interpolador nos entrega el numero de función activada podemos realizar una comprobación del estado de la misma.



```
wM
N30 G02 X150 Y100 Z100 R50
N31 M03
N40 G02 X200 Y100 Z0 R75
```



```
// M03: Acknoledgement
(fbInterpolator.wM=3)   bM03PreAcknoledgement                bM03PreAcknoledgementAux
---| |---------------------| |-------------------------------------( )

// M03: fb Modul Acknoledgemt
                                  fbM03Acknoledgement
                          SMC_PreAcknowledgeMFunction
                         EN                         ENO
bM03PreAcknoledgementAux -bEnable                 bDone -
                       3 -iM                      bBusy -
               pGeoInfos -poqDataIn              bError -
                                              wErrorID -
```

This signal activates the next module and recognizes the function

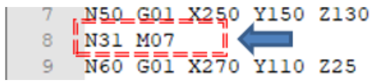
Function number to "Recognize"

Codes received by the CNC program

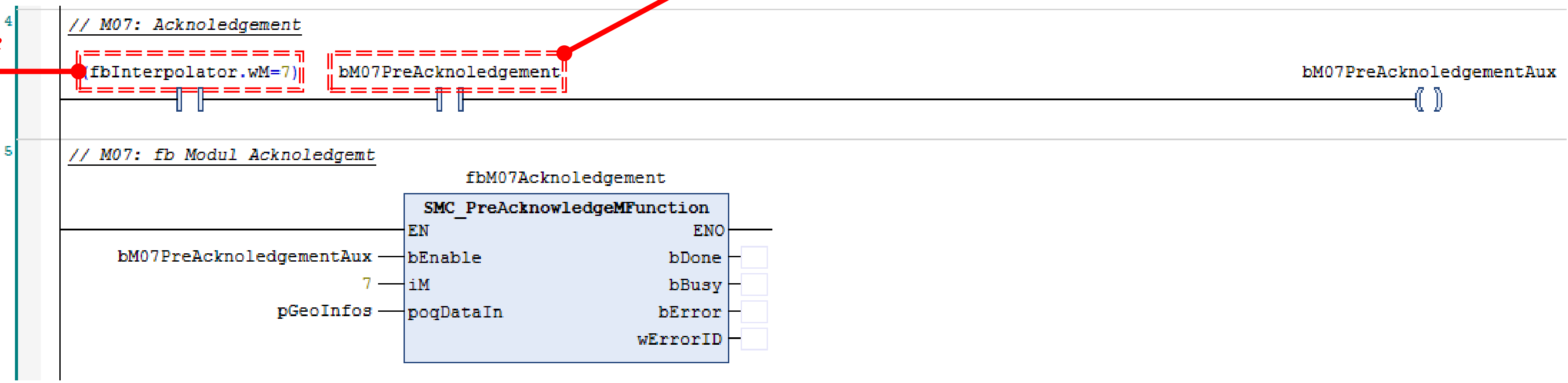
This other signal could be a limit switch, an ok position signal from another element or even a permit to continue the program movement from another CNC program.

Function Recognition M07

Function M07 is recognized in the same way as M03.



```
N50 G01 X250 Y150 Z130
N31 M07
N60 G01 X270 Y110 Z25
```

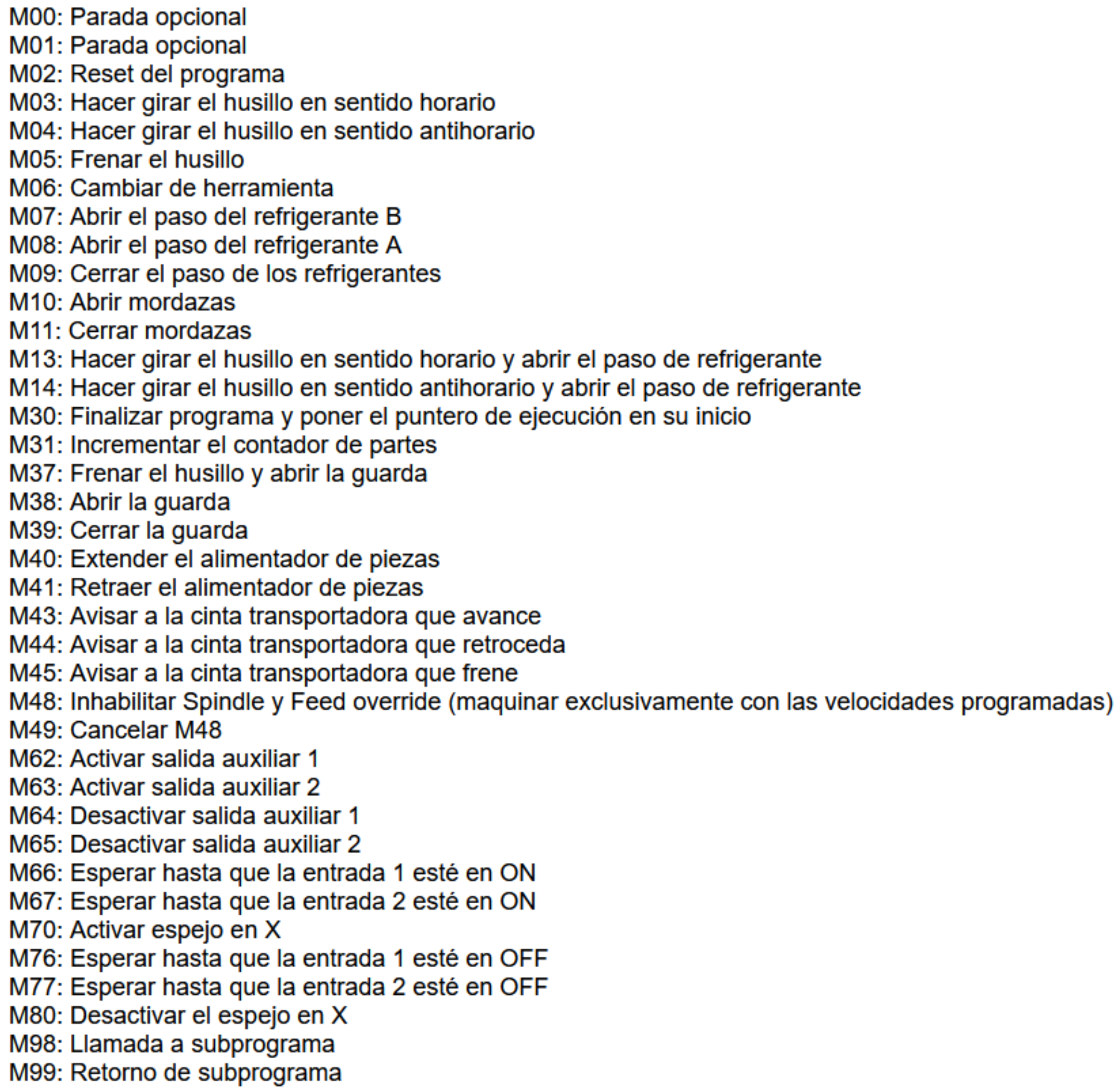


```
// M07: Acknoledgement
(fbInterpolator.wM=7)   bM07PreAcknoledgement                bM07PreAcknoledgementAux
---| |---------------------| |-------------------------------------( )

// M07: fb Modul Acknoledgemt
                                  fbM07Acknoledgement
                          SMC_PreAcknowledgeMFunction
                         EN                         ENO
bM07PreAcknoledgementAux -bEnable                 bDone -
                       7 -iM                      bBusy -
               pGeoInfos -poqDataIn              bError -
                                              wErrorID -
```

ctrlX - *Example of CNC program - Use of M Functions*

- *Table of standardized M functions*

As we have already said, the use of M functions is completely free and is not associated with standard uses.

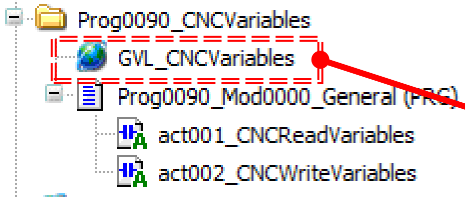


M00: Parada opcional
M01: Parada opcional
M02: Reset del programa
M03: Hacer girar el husillo en sentido horario
M04: Hacer girar el husillo en sentido antihorario
M05: Frenar el husillo
M06: Cambiar de herramienta
M07: Abrir el paso del refrigerante B
M08: Abrir el paso del refrigerante A
M09: Cerrar el paso de los refrigerantes
M10: Abrir mordazas
M11: Cerrar mordazas
M13: Hacer girar el husillo en sentido horario y abrir el paso de refrigerante
M14: Hacer girar el husillo en sentido antihorario y abrir el paso de refrigerante
M30: Finalizar programa y poner el puntero de ejecución en su inicio
M31: Incrementar el contador de partes
M37: Frenar el husillo y abrir la guarda
M38: Abrir la guarda
M39: Cerrar la guarda
M40: Extender el alimentador de piezas
M41: Retraer el alimentador de piezas
M43: Avisar a la cinta transportadora que avance
M44: Avisar a la cinta transportadora que retroceda
M45: Avisar a la cinta transportadora que frene
M48: Inhabilitar Spindle y Feed override (maquinar exclusivamente con las velocidades programadas)
M49: Cancelar M48
M62: Activar salida auxiliar 1
M63: Activar salida auxiliar 2
M64: Desactivar salida auxiliar 1
M65: Desactivar salida auxiliar 2
M66: Esperar hasta que la entrada 1 esté en ON
M67: Esperar hasta que la entrada 2 esté en ON
M70: Activar espejo en X
M76: Esperar hasta que la entrada 1 esté en OFF
M77: Esperar hasta que la entrada 2 esté en OFF
M80: Desactivar el espejo en X
M98: Llamada a subprograma
M99: Retorno de subprograma

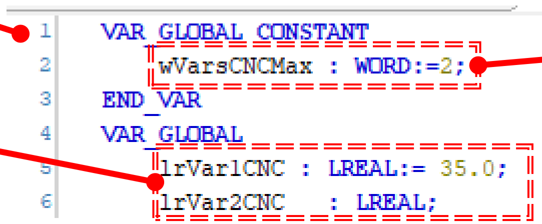
CNC variables

ctrlX - *Example of CNC program - Variables in CNC Program*

- *The creation of variables to be introduced in the CNC program will be carried out in the following way:*



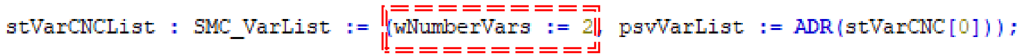
Variables folder



```
VAR_GLOBAL CONSTANT
    wVarsCNCMax : WORD:=2;
END_VAR
VAR_GLOBAL
    lrVar1CNC  : LREAL:= 35.0;
    lrVar2CNC   : LREAL;
```

Variables created to control program variables

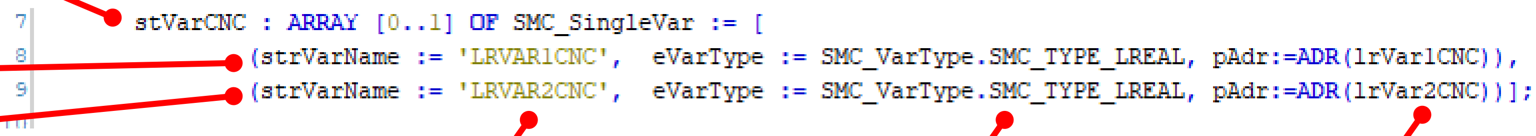
Variable not used. We can use it to determine the number of variables used in the "structure" stVarCNCList -> wNumbersVars



```
stVarCNCList : SMC_VarList := (wNumberVars := 2, psvVarList := ADR(stVarCNC[0]));
```

- *In the same variables folder we must create the module using SMC_SingleVar where the variables that we want to use to access variables in the CNC program will be introduced.*

General array for the insertion of variables in the system.



```
stVarCNC : ARRAY [0..1] OF SMC_SingleVar := [
        (strVarName := 'LRVAR1CNC',   eVarType := SMC_VarType.SMC_TYPE_LREAL, pAdr:=ADR(lrVar1CNC)),
        (strVarName := 'LRVAR2CNC',   eVarType := SMC_VarType.SMC_TYPE_LREAL, pAdr:=ADR(lrVar2CNC))];
```

Variable 1

Variable 2

Variable Name Used in the CNC program

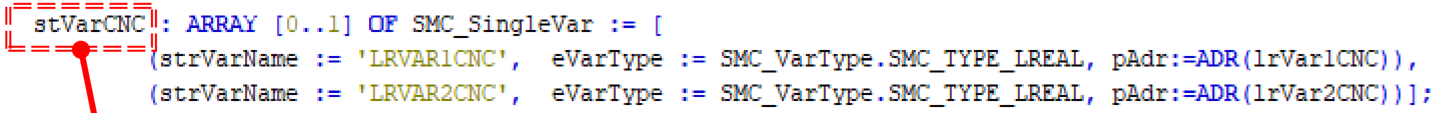
Variable Type Definition

Addressing of the variable used

⚠ *Use of uppercase names in "strVarName" assignment is recommended*

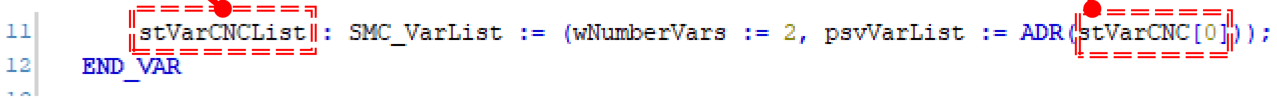
ctrlX - *Example of CNC program - Variables in CNC Program*

- *Next, we must create the variable, referenced on SMC_VarList, which will be used to load variables in the CNC program interpreter.*



```
stVarCNC : ARRAY [0..1] OF SMC_SingleVar := [
    (strVarName := 'LRVAR1CNC',  eVarType := SMC_VarType.SMC_TYPE_LREAL, pAdr:=ADR(lrVar1CNC)),
    (strVarName := 'LRVAR2CNC',  eVarType := SMC_VarType.SMC_TYPE_LREAL, pAdr:=ADR(lrVar2CNC))];
```

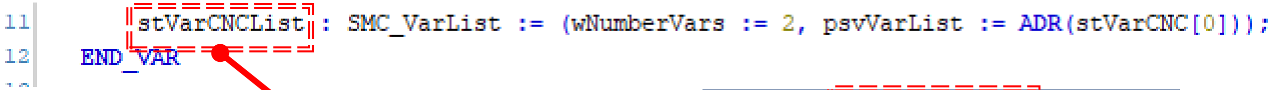
List of variables for use in the CNC program



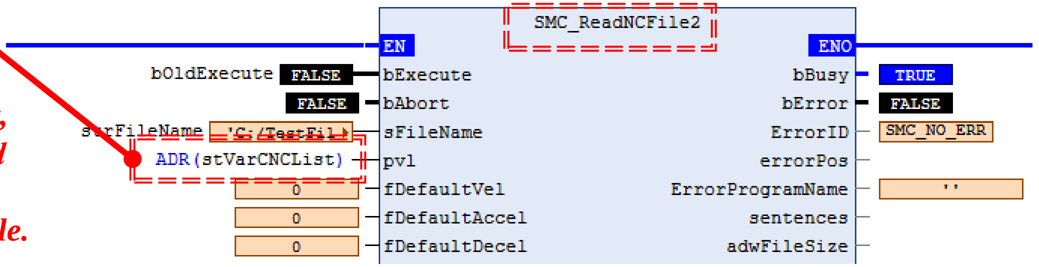
```
    stVarCNCList : SMC_VarList := (wNumberVars := 2, psvVarList := ADR(stVarCNC[0]));
END_VAR
```

If we have more than one variable, we must incorporate the initial array number of the array of variables.

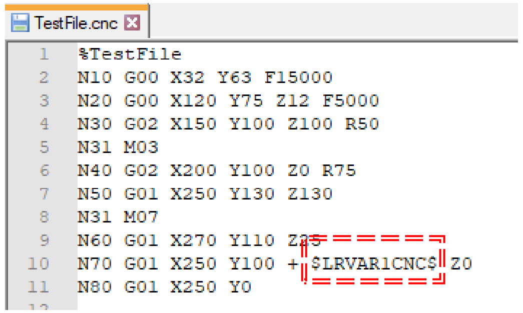
- *The list of variables must be entered in the fb file reading module:*



```
    stVarCNCList : SMC_VarList := (wNumberVars := 2, psvVarList := ADR(stVarCNC[0]));
END_VAR
```



The variable from the variable list, "stVarCNCList", must be assigned to the pv1 parameter of the fb module for reading the external file.



TestFile.cnc

```
%TestFile
N10 G00 X32 Y63 F15000
N20 G00 X120 Y75 Z12 F5000
N30 G02 X150 Y100 Z100 R50
N31 M03
N40 G02 X200 Y100 Z0 R75
N50 G01 X250 Y130 Z130
N31 M07
N60 G01 X270 Y110 Z25
N70 G01 X250 Y100 + $LRVAR1CNC$ Z0
N80 G01 X250 Y0
```

This type of variables and due to the type of system configuration are only updated after a new start of the program

** "G" command table*

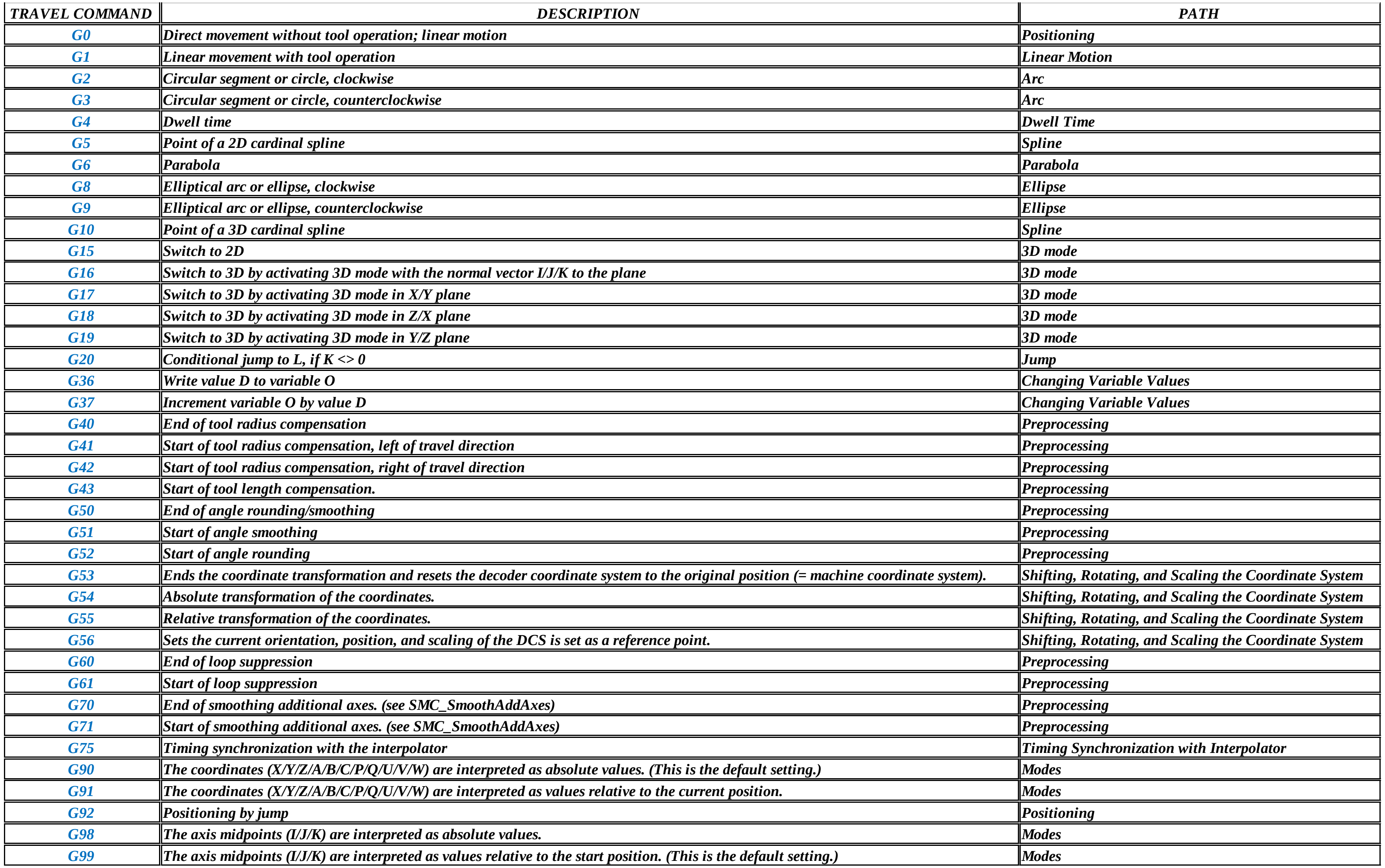
** Identifiers*

** Expressions*

** Mathematical Functions*

ctrlX - *List of G Gommands*

- *G Command Table:*

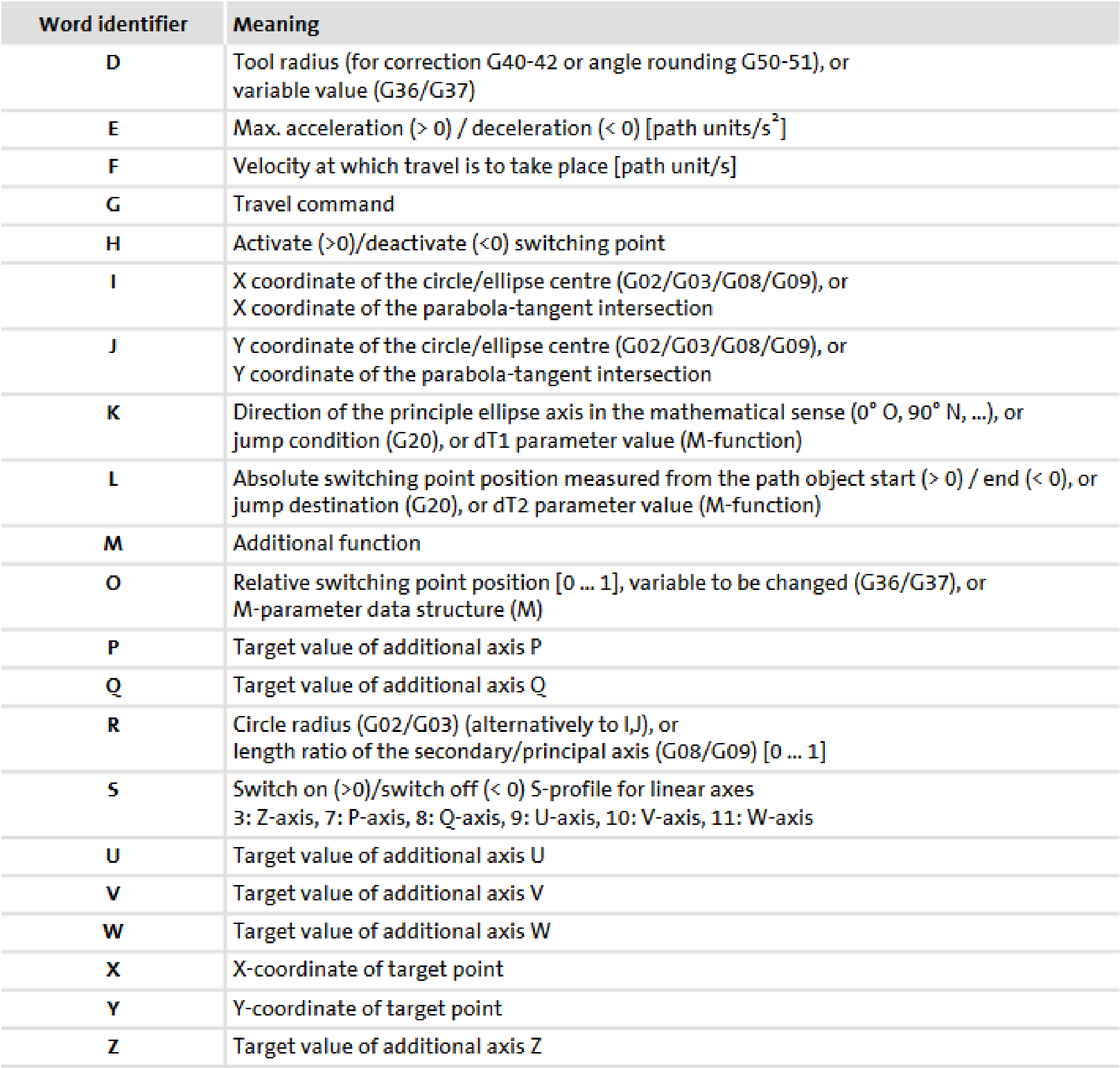


TRAVEL COMMAND	DESCRIPTION	PATH
G0	Direct movement without tool operation; linear motion	Positioning
G1	Linear movement with tool operation	Linear Motion
G2	Circular segment or circle, clockwise	Arc
G3	Circular segment or circle, counterclockwise	Arc
G4	Dwell time	Dwell Time
G5	Point of a 2D cardinal spline	Spline
G6	Parabola	Parabola
G8	Elliptical arc or ellipse, clockwise	Ellipse
G9	Elliptical arc or ellipse, counterclockwise	Ellipse
G10	Point of a 3D cardinal spline	Spline
G15	Switch to 2D	3D mode
G16	Switch to 3D by activating 3D mode with the normal vector I/J/K to the plane	3D mode
G17	Switch to 3D by activating 3D mode in X/Y plane	3D mode
G18	Switch to 3D by activating 3D mode in Z/X plane	3D mode
G19	Switch to 3D by activating 3D mode in Y/Z plane	3D mode
G20	Conditional jump to L, if K <> 0	Jump
G36	Write value D to variable O	Changing Variable Values
G37	Increment variable O by value D	Changing Variable Values
G40	End of tool radius compensation	Preprocessing
G41	Start of tool radius compensation, left of travel direction	Preprocessing
G42	Start of tool radius compensation, right of travel direction	Preprocessing
G43	Start of tool length compensation.	Preprocessing
G50	End of angle rounding/smoothing	Preprocessing
G51	Start of angle smoothing	Preprocessing
G52	Start of angle rounding	Preprocessing
G53	Ends the coordinate transformation and resets the decoder coordinate system to the original position (= machine coordinate system).	Shifting, Rotating, and Scaling the Coordinate System
G54	Absolute transformation of the coordinates.	Shifting, Rotating, and Scaling the Coordinate System
G55	Relative transformation of the coordinates.	Shifting, Rotating, and Scaling the Coordinate System
G56	Sets the current orientation, position, and scaling of the DCS is set as a reference point.	Shifting, Rotating, and Scaling the Coordinate System
G60	End of loop suppression	Preprocessing
G61	Start of loop suppression	Preprocessing
G70	End of smoothing additional axes. (see SMC_SmoothAddAxes)	Preprocessing
G71	Start of smoothing additional axes. (see SMC_SmoothAddAxes)	Preprocessing
G75	Timing synchronization with the interpolator	Timing Synchronization with Interpolator
G90	The coordinates (X/Y/Z/A/B/C/P/Q/U/V/W) are interpreted as absolute values. (This is the default setting.)	Modes
G91	The coordinates (X/Y/Z/A/B/C/P/Q/U/V/W) are interpreted as values relative to the current position.	Modes
G92	Positioning by jump	Positioning
G98	The axis midpoints (I/J/K) are interpreted as absolute values.	Modes
G99	The axis midpoints (I/J/K) are interpreted as values relative to the start position. (This is the default setting.)	Modes

Table of commands extracted from the help of SoftMotion. However, some may not be included.

ctrlX - *Identifiers*

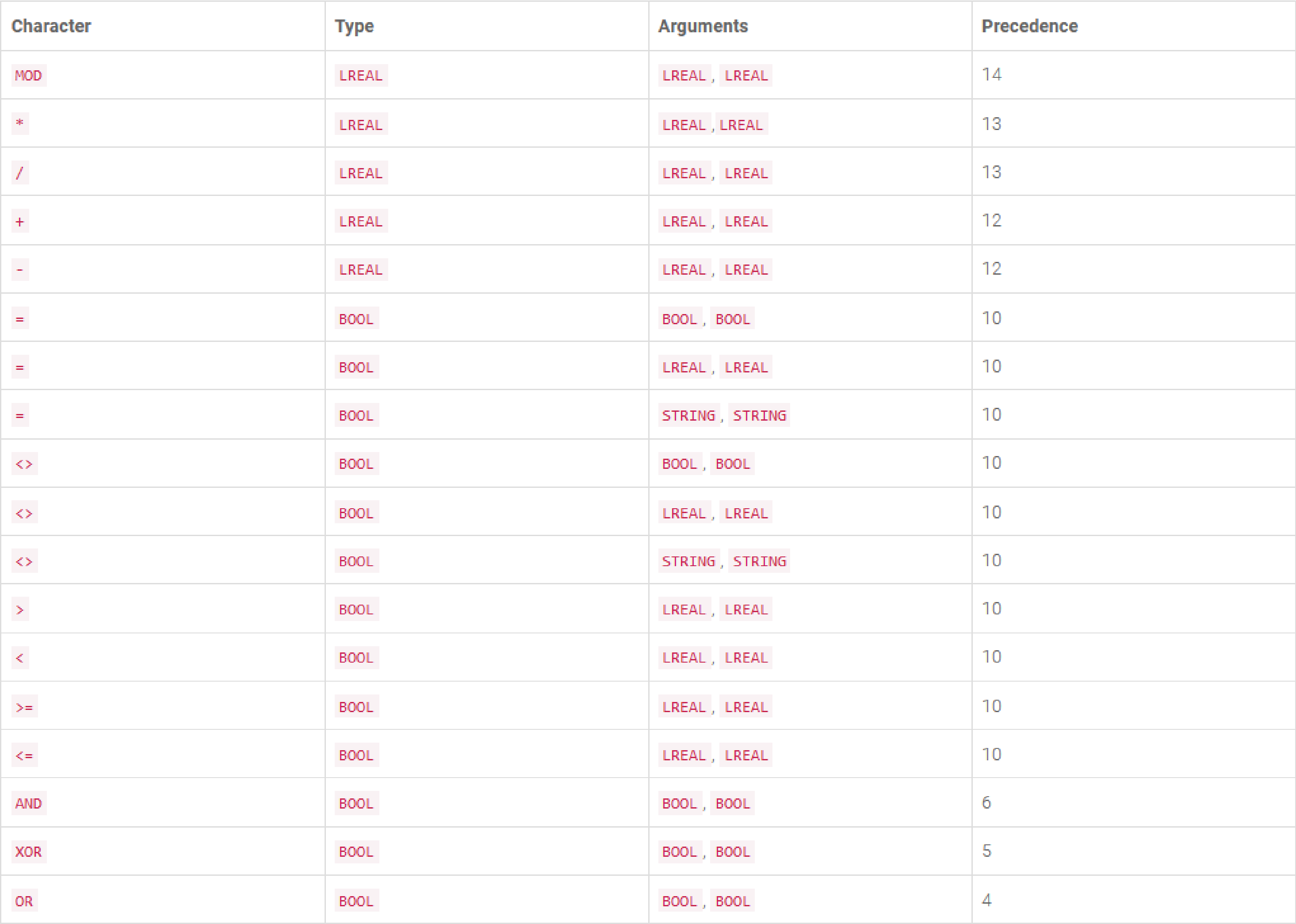
- ***Word type identifiers:***



Word identifier	Meaning
D	Tool radius (for correction G40-42 or angle rounding G50-51), or variable value (G36/G37)
E	Max. acceleration (> 0) / deceleration (< 0) [path units/s^2]
F	Velocity at which travel is to take place [path unit/s]
G	Travel command
H	Activate (>0)/deactivate (<0) switching point
I	X coordinate of the circle/ellipse centre (G02/G03/G08/G09), or X coordinate of the parabola-tangent intersection
J	Y coordinate of the circle/ellipse centre (G02/G03/G08/G09), or Y coordinate of the parabola-tangent intersection
K	Direction of the principle ellipse axis in the mathematical sense (0° O, 90° N, ...), or jump condition (G20), or dT1 parameter value (M-function)
L	Absolute switching point position measured from the path object start (> 0) / end (< 0), or jump destination (G20), or dT2 parameter value (M-function)
M	Additional function
O	Relative switching point position [0 ... 1], variable to be changed (G36/G37), or M-parameter data structure (M)
P	Target value of additional axis P
Q	Target value of additional axis Q
R	Circle radius (G02/G03) (alternatively to I,J), or length ratio of the secondary/principal axis (G08/G09) [0 ... 1]
S	Switch on (>0)/switch off (< 0) S-profile for linear axes 3: Z-axis, 7: P-axis, 8: Q-axis, 9: U-axis, 10: V-axis, 11: W-axis
U	Target value of additional axis U
V	Target value of additional axis V
W	Target value of additional axis W
X	X-coordinate of target point
Y	Y-coordinate of target point
Z	Target value of additional axis Z

ctrlX - Operators

- *Operators:*

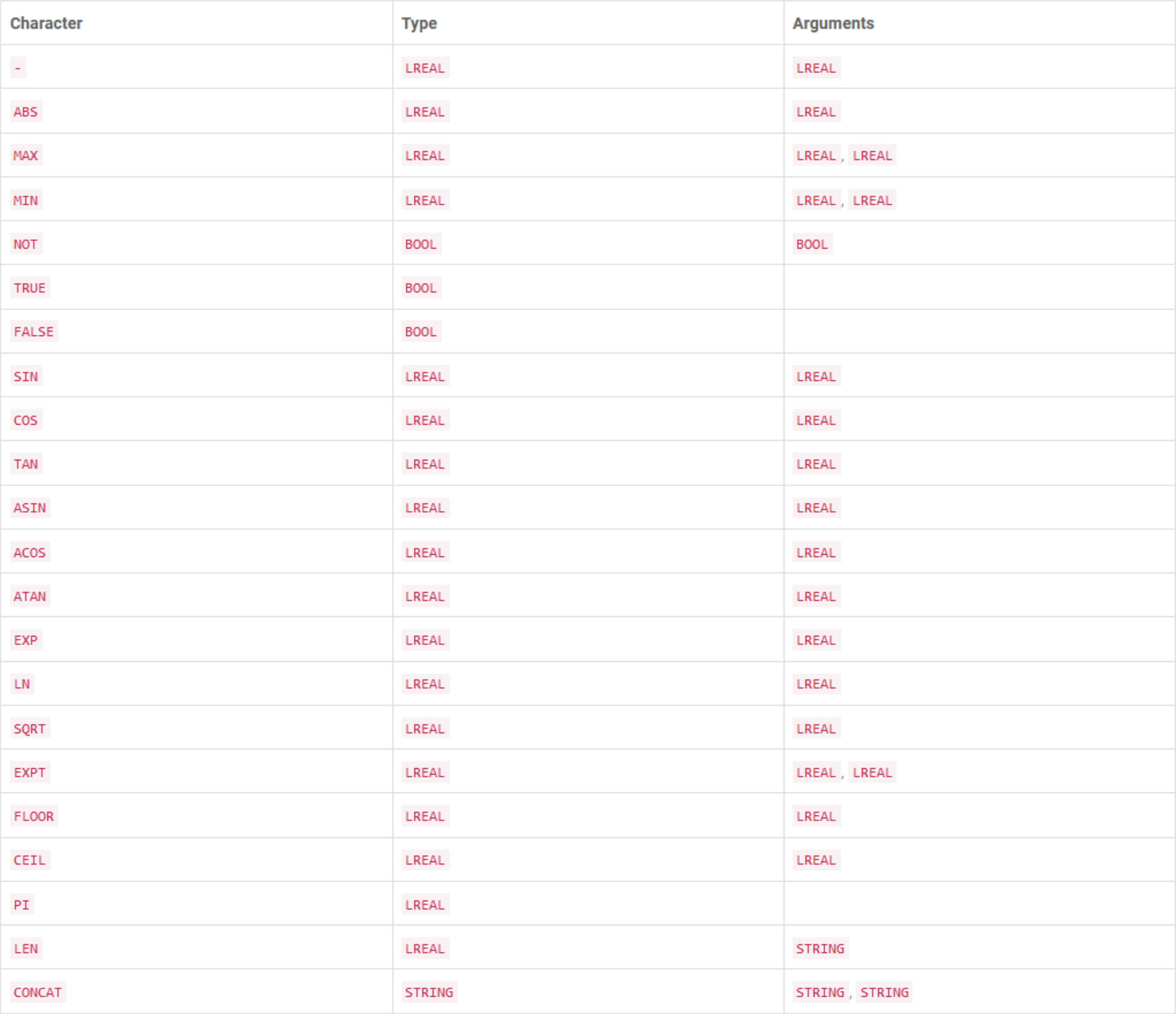


Character	Type	Arguments	Precedence
`MOD`	`LREAL`	`LREAL` , `LREAL`	14
`*`	`LREAL`	`LREAL` , `LREAL`	13
`/`	`LREAL`	`LREAL` , `LREAL`	13
`+`	`LREAL`	`LREAL` , `LREAL`	12
`-`	`LREAL`	`LREAL` , `LREAL`	12
`=`	`BOOL`	`BOOL` , `BOOL`	10
`=`	`BOOL`	`LREAL` , `LREAL`	10
`=`	`BOOL`	`STRING` , `STRING`	10
`<>`	`BOOL`	`BOOL` , `BOOL`	10
`<>`	`BOOL`	`LREAL` , `LREAL`	10
`<>`	`BOOL`	`STRING` , `STRING`	10
`>`	`BOOL`	`LREAL` , `LREAL`	10
`<`	`BOOL`	`LREAL` , `LREAL`	10
`>=`	`BOOL`	`LREAL` , `LREAL`	10
`<=`	`BOOL`	`LREAL` , `LREAL`	10
`AND`	`BOOL`	`BOOL` , `BOOL`	6
`XOR`	`BOOL`	`BOOL` , `BOOL`	5
`OR`	`BOOL`	`BOOL` , `BOOL`	4

⚠ *Expressions only work in the CNC Online program and not in the CNC program editor*

ctrlX - *Math Functions*

- *Math Functions:*



Character	Type	Arguments
-	LREAL	LREAL
ABS	LREAL	LREAL
MAX	LREAL	LREAL , LREAL
MIN	LREAL	LREAL , LREAL
NOT	BOOL	BOOL
TRUE	BOOL	
FALSE	BOOL	
SIN	LREAL	LREAL
COS	LREAL	LREAL
TAN	LREAL	LREAL
ASIN	LREAL	LREAL
ACOS	LREAL	LREAL
ATAN	LREAL	LREAL
EXP	LREAL	LREAL
LN	LREAL	LREAL
SQRT	LREAL	LREAL
EXPT	LREAL	LREAL , LREAL
FLOOR	LREAL	LREAL
CEIL	LREAL	LREAL
PI	LREAL	
LEN	LREAL	STRING
CONCAT	STRING	STRING , STRING

Program Jumps with G20

ctrlX - *Program jumps with G20 code*

- *Program jumps are generated from the use of the G20 command:*

G20 L K

L – Parameter target for jump

- Line number defined for the jump, for example a L1020 would jump to line 1020

- Jump to a label:

These jumps are defined with a:

“?” like jump, for example L?2

“!” as the target of the jump L!4

These types of jumps only work with the online program and not with the CNC editor.

This type of jumps cannot be used to jump back

K – jump conditions:

It is used with K< > 0, the jump is executed.

If K is not defined, an internal decoder variable will be used.

- *Example of program jump in continuous motion:*

In general, CNC programs are "terminated" when they reach the end of the code. In some equipment the formula is used: one .

1 .BEGINNING

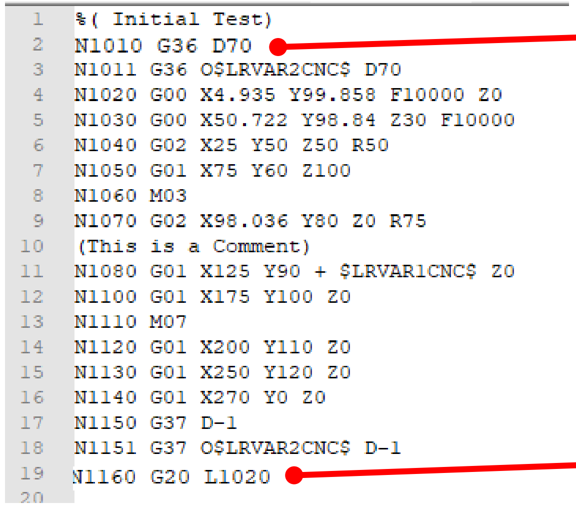
// Program Line

// Program Line

// Program Line

1 GOTO . BEGINNING

To generate an endless loop.



```
%( Initial Test)
N1010 G36 D70
N1011 G36 O$LRVAR2CNC$ D70
N1020 G00 X4.935 Y99.858 F10000 Z0
N1030 G00 X50.722 Y98.84 Z30 F10000
N1040 G02 X25 Y50 Z50 R50
N1050 G01 X75 Y60 Z100
N1060 M03
N1070 G02 X98.036 Y80 Z0 R75
 (This is a Comment)
N1080 G01 X125 Y90 + $LRVAR1CNC$ Z0
N1100 G01 X175 Y100 Z0
N1110 M07
N1120 G01 X200 Y110 Z0
N1130 G01 X250 Y120 Z0
N1140 G01 X270 Y0 Z0
N1150 G37 D-1
N1151 G37 O$LRVAR2CNC$ D-1
N1160 G20 L1020
```

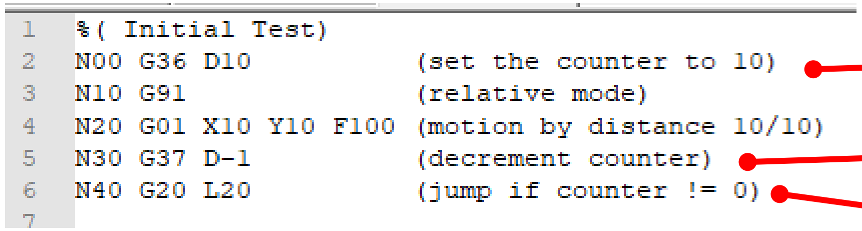
Line N1010 activates the writing of a variable, in this case internal, since there is no one defined, with a value of 70 (Parameter D)

The operation loop is not infinite and will only be executed the 70 times that it has been arranged in the internal variable. This also causes a delay in starting the program as the system seems to be generating that loop internally before the program is activated.

Line N1160 with the use of the G20 command generates the jump to line N1020 and the program continues without problems (For 70 cycles)

ctrlX - Program jumps with G20 code

- - Program jumps with G20 code



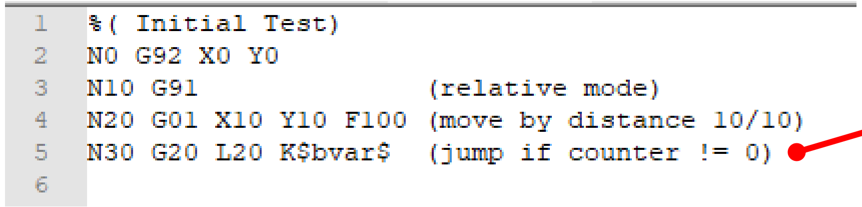
```
%( Initial Test)
N00 G36 D10           (set the counter to 10)
N10 G91               (relative mode)
N20 G01 X10 Y10 F100  (motion by distance 10/10)
N30 G37 D-1           (decrement counter)
N40 G20 L20           (jump if counter != 0)

```

Number of cycles (G36 writes the variable)

G37 increments the variable defined in G36. In this case a 1 is subtracted

If the counter reaches "0" the program ends

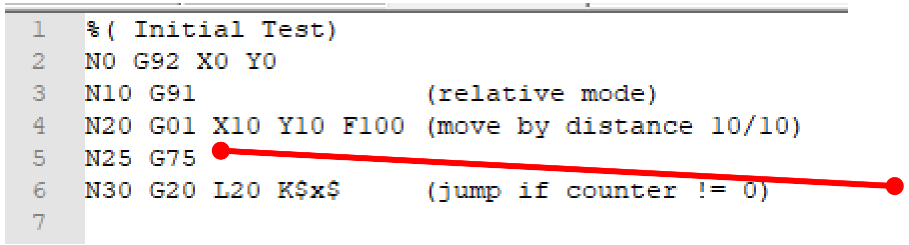
- *In this other example, it is the variable "bVar" that controls whether the program should end, while it is above "0" the program will continue its course increasing the values of the X and Y axes by 10*



```
%( Initial Test)
N0 G92 X0 Y0
N10 G91               (relative mode)
N20 G01 X10 Y10 F100  (move by distance 10/10)
N30 G20 L20 K$bvar$   (jump if counter != 0)

```

If the counter reaches "0" the program ends

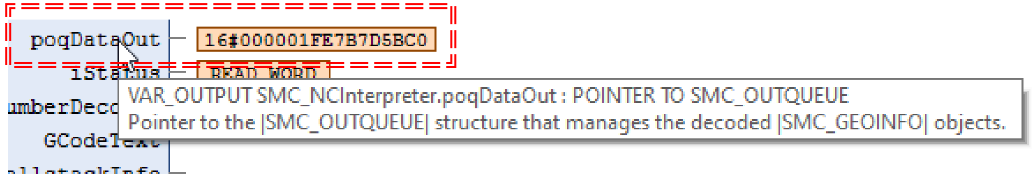
- *This other example is similar, however in this case the G75 command is being used*



```
%( Initial Test)
N0 G92 X0 Y0
N10 G91               (relative mode)
N20 G01 X10 Y10 F100  (move by distance 10/10)
N25 G75
N30 G20 L20 K$x$      (jump if counter != 0)

```

With G75 it is ensured that the axes have reached the position before evaluating the status of “x”. G75 synchronizes the time with the interpolator.

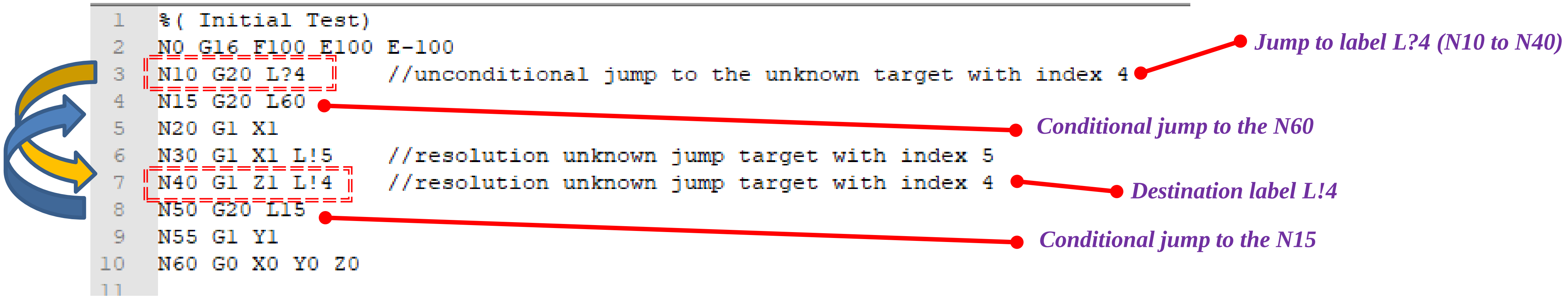
⚠ ***G75 CANNOT be used in programs generated with SMC_OutQueue, which is what the NC_Interpreter is using to generate the structures sent to the interpolator.***



VAR_OUTPUT SMC_NCInterpreter.poqDataOut : POINTER TO SMC_OUTQUEUE
Pointer to the |SMC_OUTQUEUE| structure that manages the decoded |SMC_GEOINFO| objects.

ctrlX - *Program jumps with G20 code*

- ***Jumps with Tags:***



```
%( Initial Test)
N0 G16 F100 E100 E-100
N10 G20 L?4      //unconditional jump to the unknown target with index 4
N15 G20 L60
N20 G1 X1
N30 G1 X1 L!5    //resolution unknown jump target with index 5
N40 G1 Z1 L!4    //resolution unknown jump target with index 4
N50 G20 L15
N55 G1 Y1
N60 G0 X0 Y0 Z0

```

Jump to label L?4 (N10 to N40)

Conditional jump to the N60

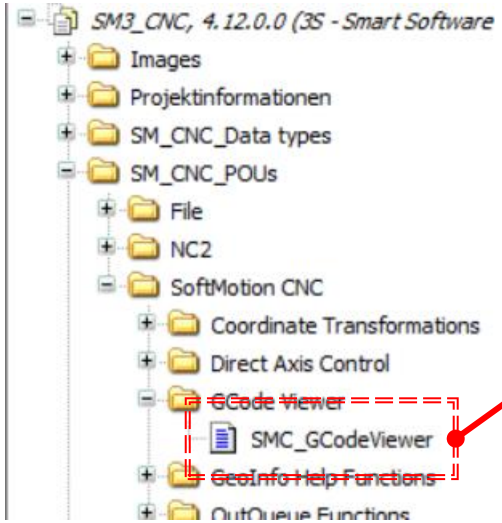
Destination label L!4

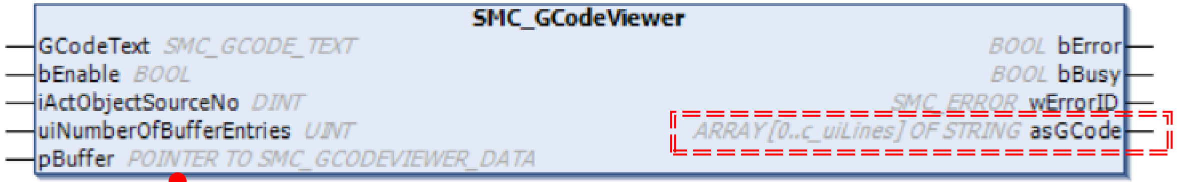
Conditional jump to the N15

Display of G code lines

ctrlX - *Display of G code program lines*

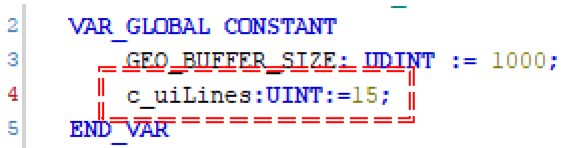
- *The G code lines can be displayed on the screen and for this we should use the SMC_GCodeViewer module, however this module does not seem to work correctly and either it always indicates that the buffer is of very small size or an error ends up being generated that causes the step from the CPU to Stop.*





```
                               SMC_GCodeViewer
GCodeText  SMC_GCODE_TEXT                                  BOOL  bError
bEnable  BOOL                                              BOOL  bBusy
iActObjectSourceNo  DINT                            SMC_ERROR  wErrorID
uiNumberOfBufferEntries  UINT      ARRAY [0..c_uiLines] OF STRING  asGCode
pBuffer  POINTER TO SMC_GCODEVIEWER_DATA
```

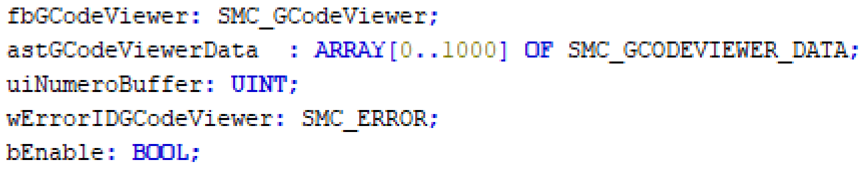
The array type output variable, asGcode, uses the "c_uiLines" variable, which does not appear externally, so it cannot be used directly. The search for it has determined that it is a UINT variable with a fixed value of "15"



```
VAR_GLOBAL CONSTANT
    GEO_BUFFER_SIZE: UDINT := 1000;
    c_uiLines:UINT:=15;
END_VAR
```

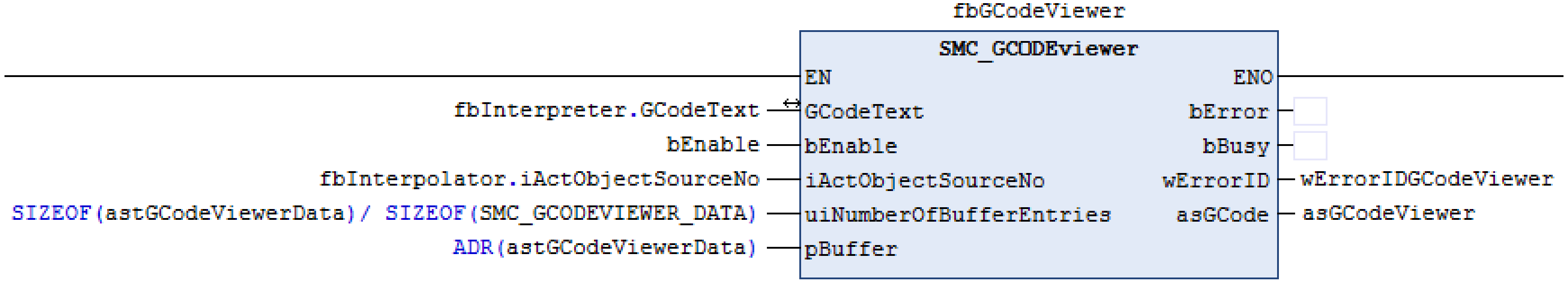
Variable c_uiLines created in a folder of constant values

Variables for the control of the SMC_GCodeViewer module



```
fbGCodeViewer: SMC_GCodeViewer;
astGCodeViewerData   : ARRAY[0..1000] OF SMC_GCODEVIEWER_DATA;
uiNumeroBuffer: UINT;
wErrorIDGCodeViewer: SMC_ERROR;
bEnable: BOOL;
```

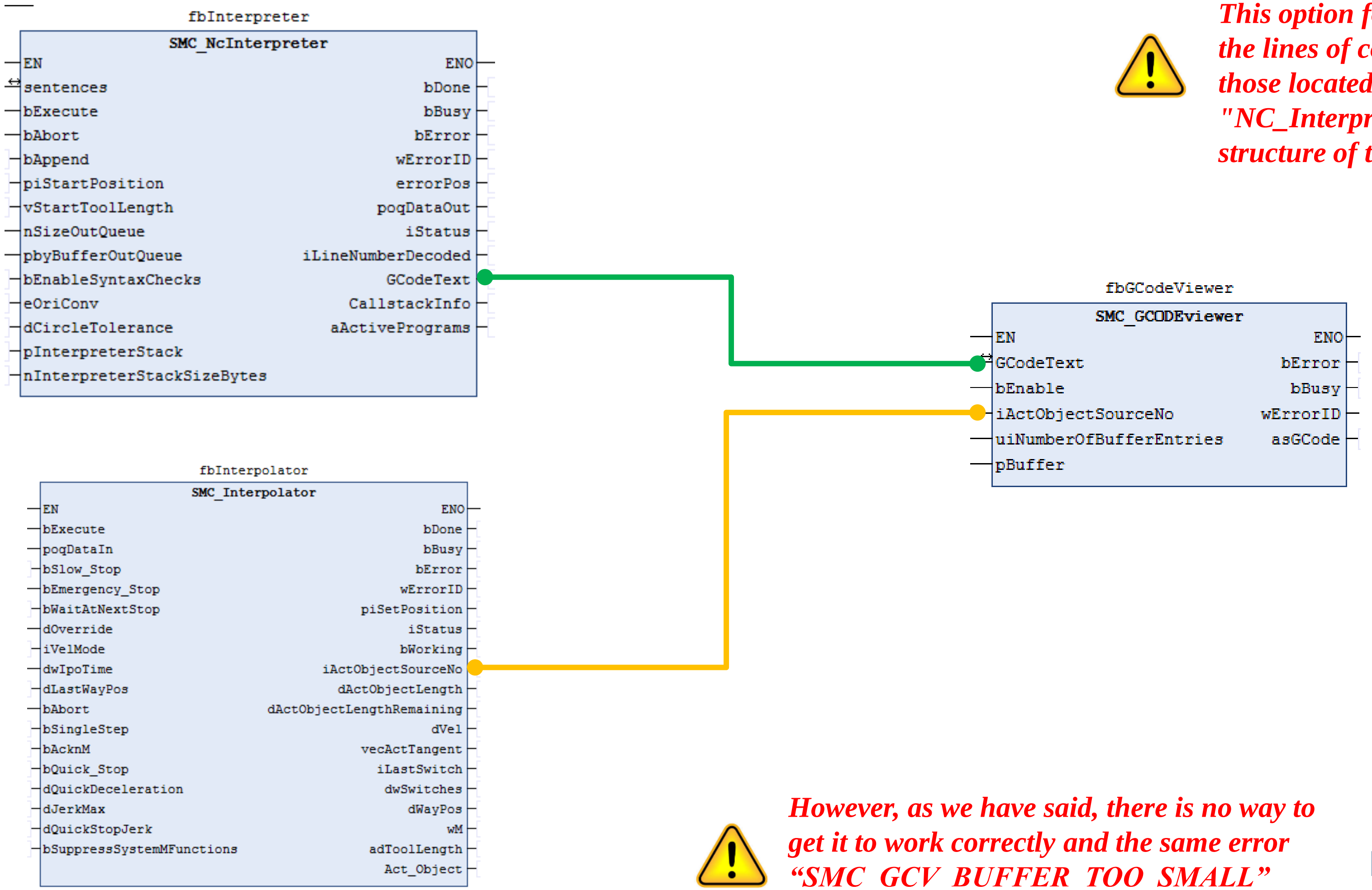
In this case, the number of bytes of the "pBuffer" structure IS NOT USED and instead the elements must be considered, 1000, in the example.



```
                                                          fbGCodeViewer
                                                        SMC_GCODEviewer
                                                 EN                                ENO
                    fbInterpreter.GCodeText ─ GCodeText                         bError ─
                                     bEnable ─ bEnable                            bBusy ─
           fbInterpolator.iActObjectSourceNo ─ iActObjectSourceNo               wErrorID ─ wErrorIDGCodeViewer
SIZEOF(astGCodeViewerData)/ SIZEOF(SMC_GCODEVIEWER_DATA) ─ uiNumberOfBufferEntries   asGCode ─ asGCodeViewer
                     ADR(astGCodeViewerData) ─ pBuffer
```

ctrlX - *Display of G code program lines*

- ***Structurally this should work like this:***



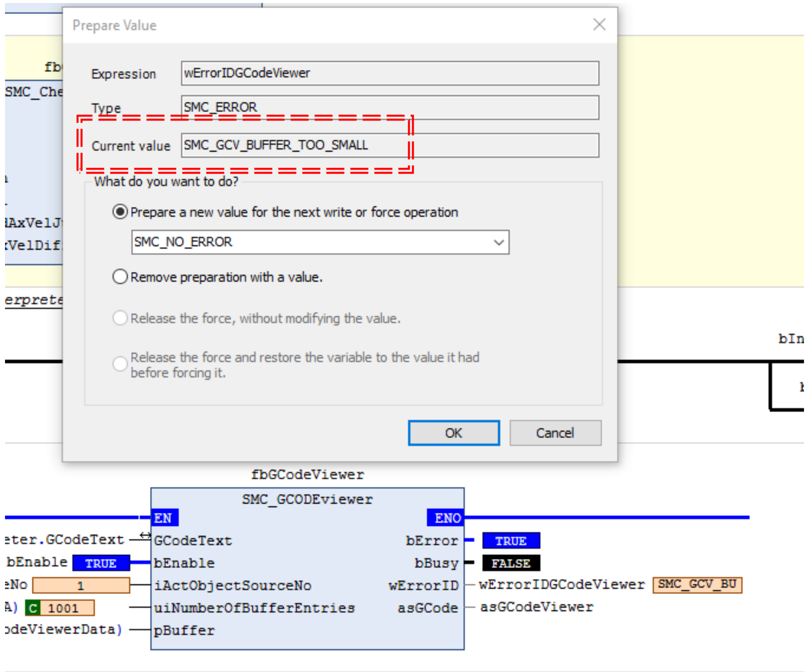
```
fbInterpreter
SMC_NcInterpreter
EN                                      ENO
sentences                             bDone
bExecute                              bBusy
bAbort                               bError
bAppend                            wErrorID
piStartPosition                    errorPos
vStartToolLength                 poqDataOut
nSizeOutQueue                       iStatus
pbyBufferOutQueue        iLineNumberDecoded
bEnableSyntaxChecks               GCodeText
eOriConv                      CallstackInfo
dCircleTolerance            aActivePrograms
pInterpreterStack
nInterpreterStackSizeBytes
```

```
fbGCodeViewer
SMC_GCODEviewer
EN                                ENO
GCodeText                      bError
bEnable                         bBusy
iActObjectSourceNo           wErrorID
uiNumberOfBufferEntries        asGCode
pBuffer
```

```
fbInterpolator
SMC_Interpolator
EN                                         ENO
bExecute                                 bDone
poqDataIn                                bBusy
bSlow_Stop                              bError
bEmergency_Stop                       wErrorID
bWaitAtNextStop                  piSetPosition
dOverride                              iStatus
iVelMode                              bWorking
dwIpoTime                   iActObjectSourceNo
dLastWayPos                 dActObjectLength
bAbort             dActObjectLengthRemaining
bSingleStep                               dVel
bAcknM                          vecActTangent
bQuick_Stop                        iLastSwitch
dQuickDeceleration                  dwSwitches
dJerkMax                               dWayPos
dQuickStopJerk                              wM
bSuppressSystemMFunctions          adToolLength
                                    Act_Object
```

This option for the extraction of the visualization of the lines of code, is only valid for the PC programs or those located in the control, since these do use the "NC_Interpreter" module to generate the instruction structure of the CNC program.

However, as we have said, there is no way to get it to work correctly and the same error "SMC_GCV_BUFFER_TOO_SMALL" always appears.



Prepare Value

Expression: wErrorIDGCodeViewer

Type: SMC_ERROR

Current value: SMC_GCV_BUFFER_TOO_SMALL

What do you want to do?

● Prepare a new value for the next write or force operation

SMC_NO_ERROR

○ Remove preparation with a value.

○ Release the force, without modifying the value.

○ Release the force and restore the variable to the value it had before forcing it.

```
                              fbGCodeViewer
                              SMC_GCODEviewer
                              EN                              ENO
eter.GCodeText                GCodeText                    bError  TRUE
bEnable  TRUE                 bEnable                       bBusy  FALSE
No  1                         iActObjectSourceNo         wErrorID  wErrorIDGCodeViewer  SMC_GCV_BU
A)  C 1001                    uiNumberOfBufferEntries     asGCode  asGCodeViewer
odeViewerData)                pBuffer
```

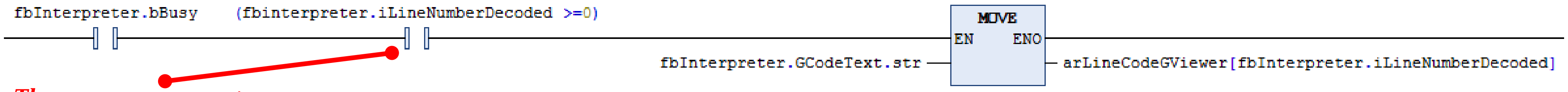
ctrlX - *Display of G code program lines*

- *One way to do this without the SMC_GCODEViewer module is as follows:*

  Generate variable to extract the arrays from the lines of the CNC program

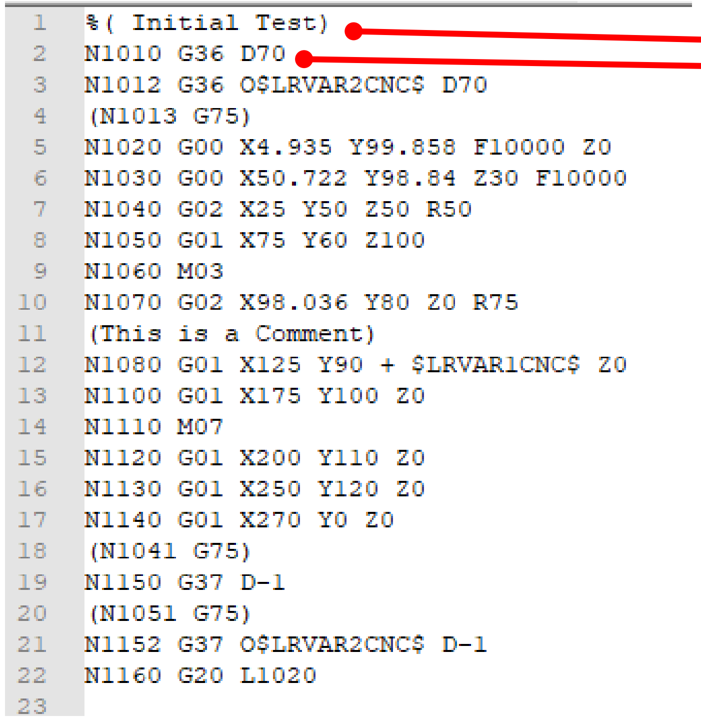
```
arLineCodeGViewer: ARRAY[0..1000] OF STRING;
```

- *Add the following line in the control module of the "interpreter", which will allow us to extract the structure of the CNC program in the corresponding Strings*

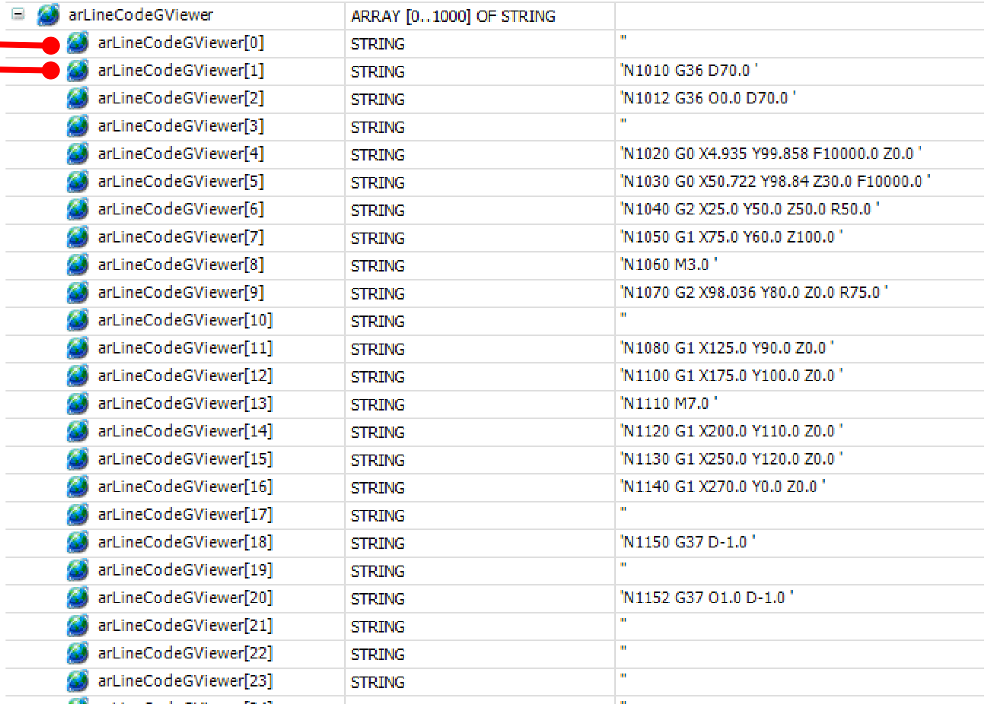

fbInterpreter.bBusy
(fbinterpreter.iLineNumberDecoded >=0)
MOVE
EN
ENO
fbInterpreter.GCodeText.str
arLineCodeGViewer[fbInterpreter.iLineNumberDecoded]

The comparer prevents an error from being produced in the ctrlX by sometimes containing the variables negative values and therefore they are outside the Array

The lines of type comment () remain empty but with this we can obtain the entire program loaded by the "interpreter"



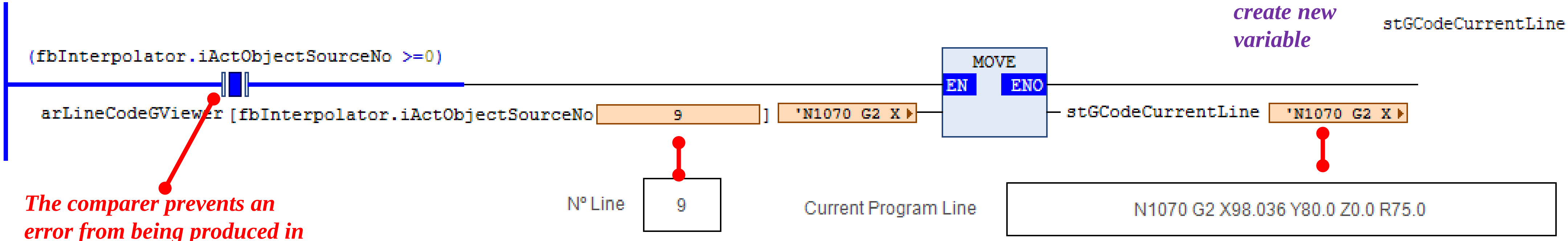
```
%( Initial Test)
N1010 G36 D70
N1012 G36 O$LRVAR2CNC$ D70
(N1013 G75)
N1020 G00 X4.935 Y99.858 F10000 Z0
N1030 G00 X50.722 Y98.84 Z30 F10000
N1040 G02 X25 Y50 Z50 R50
N1050 G01 X75 Y60 Z100
N1060 M03
N1070 G02 X98.036 Y80 Z0 R75
(This is a Comment)
N1080 G01 X125 Y90 + $LRVAR1CNC$ Z0
N1100 G01 X175 Y100 Z0
N1110 M07
N1120 G01 X200 Y110 Z0
N1130 G01 X250 Y120 Z0
N1140 G01 X270 Y0 Z0
(N1041 G75)
N1150 G37 D-1
(N1051 G75)
N1152 G37 O$LRVAR2CNC$ D-1
N1160 G20 L1020

```



arLineCodeGViewer	ARRAY [0..1000] OF STRING	
arLineCodeGViewer[0]	STRING	''
arLineCodeGViewer[1]	STRING	'N1010 G36 D70.0 '
arLineCodeGViewer[2]	STRING	'N1012 G36 O0.0 D70.0 '
arLineCodeGViewer[3]	STRING	''
arLineCodeGViewer[4]	STRING	'N1020 G0 X4.935 Y99.858 F10000.0 Z0.0 '
arLineCodeGViewer[5]	STRING	'N1030 G0 X50.722 Y98.84 Z30.0 F10000.0 '
arLineCodeGViewer[6]	STRING	'N1040 G2 X25.0 Y50.0 Z50.0 R50.0 '
arLineCodeGViewer[7]	STRING	'N1050 G1 X75.0 Y60.0 Z100.0 '
arLineCodeGViewer[8]	STRING	'N1060 M3.0 '
arLineCodeGViewer[9]	STRING	'N1070 G2 X98.036 Y80.0 Z0.0 R75.0 '
arLineCodeGViewer[10]	STRING	''
arLineCodeGViewer[11]	STRING	'N1080 G1 X125.0 Y90.0 Z0.0 '
arLineCodeGViewer[12]	STRING	'N1100 G1 X175.0 Y100.0 Z0.0 '
arLineCodeGViewer[13]	STRING	'N1110 M7.0 '
arLineCodeGViewer[14]	STRING	'N1120 G1 X200.0 Y110.0 Z0.0 '
arLineCodeGViewer[15]	STRING	'N1130 G1 X250.0 Y120.0 Z0.0 '
arLineCodeGViewer[16]	STRING	'N1140 G1 X270.0 Y0.0 Z0.0 '
arLineCodeGViewer[17]	STRING	''
arLineCodeGViewer[18]	STRING	'N1150 G37 D-1.0 '
arLineCodeGViewer[19]	STRING	''
arLineCodeGViewer[20]	STRING	'N1152 G37 O1.0 D-1.0 '
arLineCodeGViewer[21]	STRING	''
arLineCodeGViewer[22]	STRING	''
arLineCodeGViewer[23]	STRING	''

ctrlX - *Display of G code program lines*

- *We must also add the following line, which will allow us to display the current line running on the screen:*



```
(fbInterpolator.iActObjectSourceNo >=0)
arLineCodeGViewer [fbInterpolator.iActObjectSourceNo  9 ]  'N1070 G2 X   MOVE EN ENO   stGCodeCurrentLine  'N1070 G2 X
```

create new variable

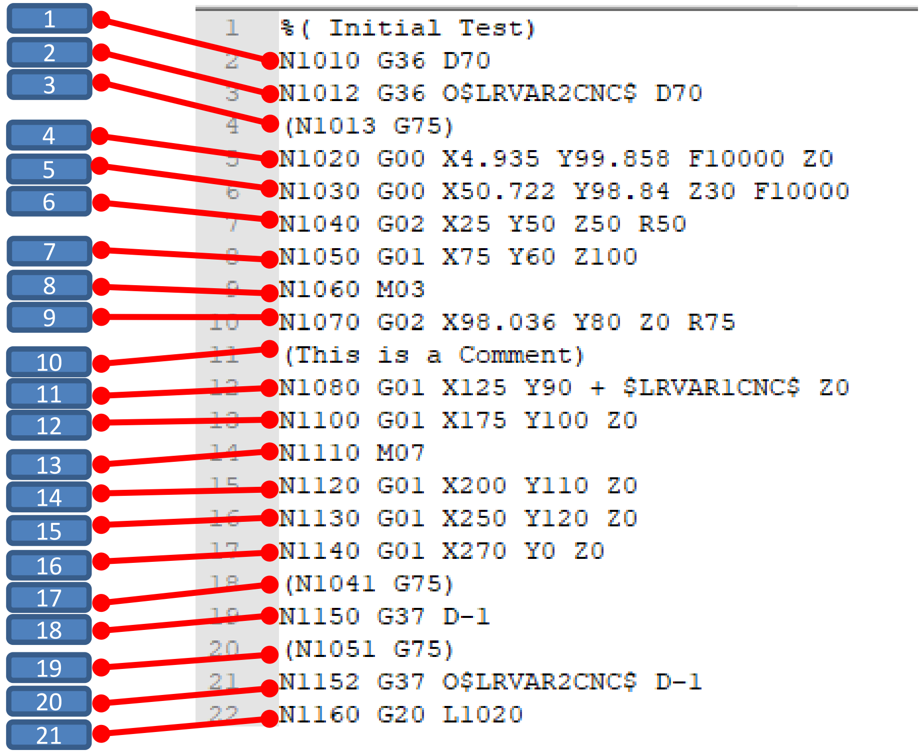
```
stGCodeCurrentLine: STRING;
```

N° Line 9

Current Program Line N1070 G2 X98.036 Y80.0 Z0.0 R75.0

The comparer prevents an error from being produced in the ctrlX by sometimes containing the variables negative values and therefore they are outside the Array

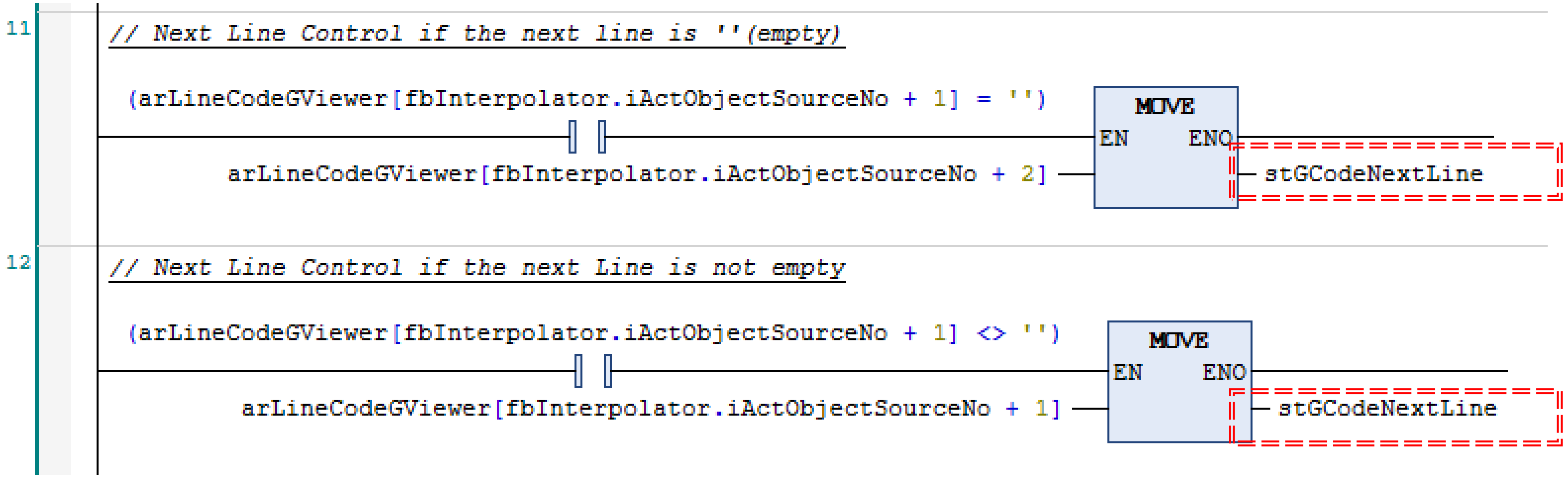
For the "interpreter", the line that follows the initial title will always be the first of the program, although this could be of the comment type. The system uses the line number for the selection of the text on the screen.



```
%( Initial Test)
N1010 G36 D70
N1012 G36 O$LRVAR2CNC$ D70
(N1013 G75)
N1020 G00 X4.935 Y99.858 F10000 Z0
N1030 G00 X50.722 Y98.84 Z30 F10000
N1040 G02 X25 Y50 Z50 R50
N1050 G01 X75 Y60 Z100
N1060 M03
N1070 G02 X98.036 Y80 Z0 R75
(This is a Comment)
N1080 G01 X125 Y90 + $LRVAR1CNC$ Z0
N1100 G01 X175 Y100 Z0
N1110 M07
N1120 G01 X200 Y110 Z0
N1130 G01 X250 Y120 Z0
N1140 G01 X270 Y0 Z0
(N1041 G75)
N1150 G37 D-1
(N1051 G75)
N1152 G37 O$LRVAR2CNC$ D-1
N1160 G20 L1020
```

ctrlX - Display of G code program lines

- *With these other two lines we can carry out an initial control to display the next line to be executed on the screen:*

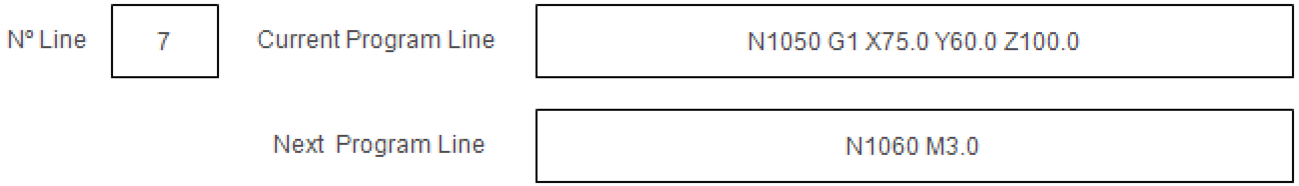


```
// Next Line Control if the next line is ''(empty)
(arLineCodeGViewer[fbInterpolator.iActObjectSourceNo + 1] = '')
arLineCodeGViewer[fbInterpolator.iActObjectSourceNo + 2]  --MOVE (EN, ENO)--  stGCodeNextLine

// Next Line Control if the next Line is not empty
(arLineCodeGViewer[fbInterpolator.iActObjectSourceNo + 1] <> '')
arLineCodeGViewer[fbInterpolator.iActObjectSourceNo + 1]  --MOVE (EN, ENO)--  stGCodeNextLine
```

Create New Variable

```
stGCodeNextLine: STRING;
```

⚠ *In the example, it has been assumed that there are no two lines of type comment in a row in the program. If there are more lines, some modification should be made in the program to control the existing comment lines.*



N° Line: 7

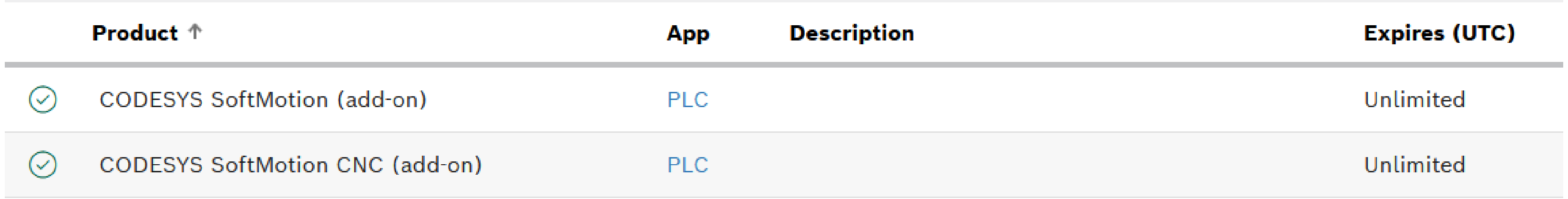
Current Program Line: N1050 G1 X75.0 Y60.0 Z100.0

Next Program Line: N1060 M3.0

Licenses

ctrlX - *Licences*

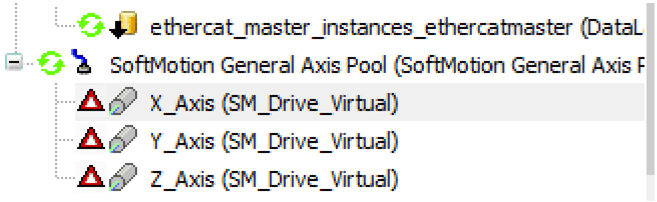
- *The equipment (ctrlX) must be licensed in the usual way and we must have the two SoftMotion options*



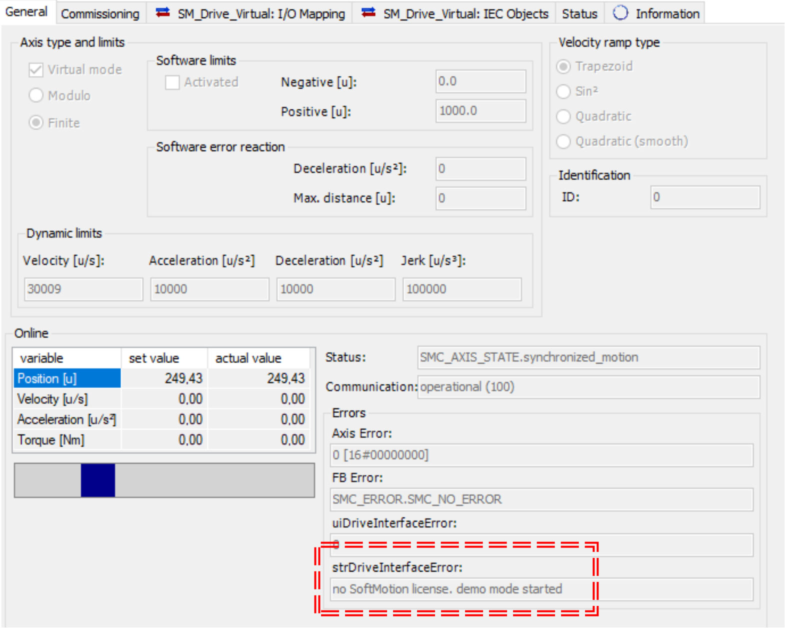
	Product ↑	App	Description	Expires (UTC)
✓	CODESYS SoftMotion (add-on)	PLC		Unlimited
✓	CODESYS SoftMotion CNC (add-on)	PLC		Unlimited

- *If they are not licensed, the system works for approximately one hour and we can see this status on the general screen of the axes.*

Error en los ejes virtuales



ethercat_master_instances_ethercatmaster (DataL
SoftMotion General Axis Pool (SoftMotion General Axis F
X_Axis (SM_Drive_Virtual)
Y_Axis (SM_Drive_Virtual)
Z_Axis (SM_Drive_Virtual)



Axis type and limits
☑ Virtual mode
○ Modulo
● Finite

Software limits
□ Activated
Negative [u]: 0.0
Positive [u]: 1000.0

Software error reaction
Deceleration [u/s²]: 0
Max. distance [u]: 0

Velocity ramp type
● Trapezoid
○ Sin²
○ Quadratic
○ Quadratic (smooth)

Identification
ID: 0

Dynamic limits

Velocity [u/s]:	Acceleration [u/s²]	Deceleration [u/s²]	Jerk [u/s³]:
30009	10000	10000	100000

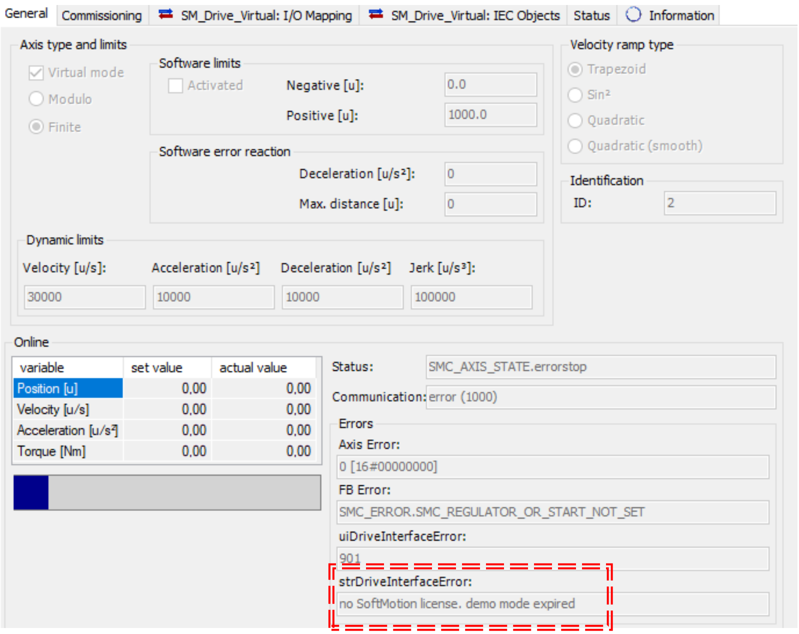
Online

variable	set value	actual value
Position [u]	249,43	249,43
Velocity [u/s]	0,00	0,00
Acceleration [u/s²]	0,00	0,00
Torque [Nm]	0,00	0,00

Status: SMC_AXIS_STATE.synchronized_motion
Communication: operational (100)

Errors
Axis Error: 0 [16#00000000]
FB Error: SMC_ERROR.SMC_NO_ERROR
uiDriveInterfaceError: 0
strDriveInterfaceError: no SoftMotion license. demo mode started

Softmotion in demo mode



Axis type and limits
☑ Virtual mode
○ Modulo
● Finite

Software limits
□ Activated
Negative [u]: 0.0
Positive [u]: 1000.0

Software error reaction
Deceleration [u/s²]: 0
Max. distance [u]: 0

Velocity ramp type
● Trapezoid
○ Sin²
○ Quadratic
○ Quadratic (smooth)

Identification
ID: 2

Dynamic limits

Velocity [u/s]:	Acceleration [u/s²]	Deceleration [u/s²]	Jerk [u/s³]:
30000	10000	10000	100000

Online

variable	set value	actual value
Position [u]	0,00	0,00
Velocity [u/s]	0,00	0,00
Acceleration [u/s²]	0,00	0,00
Torque [Nm]	0,00	0,00

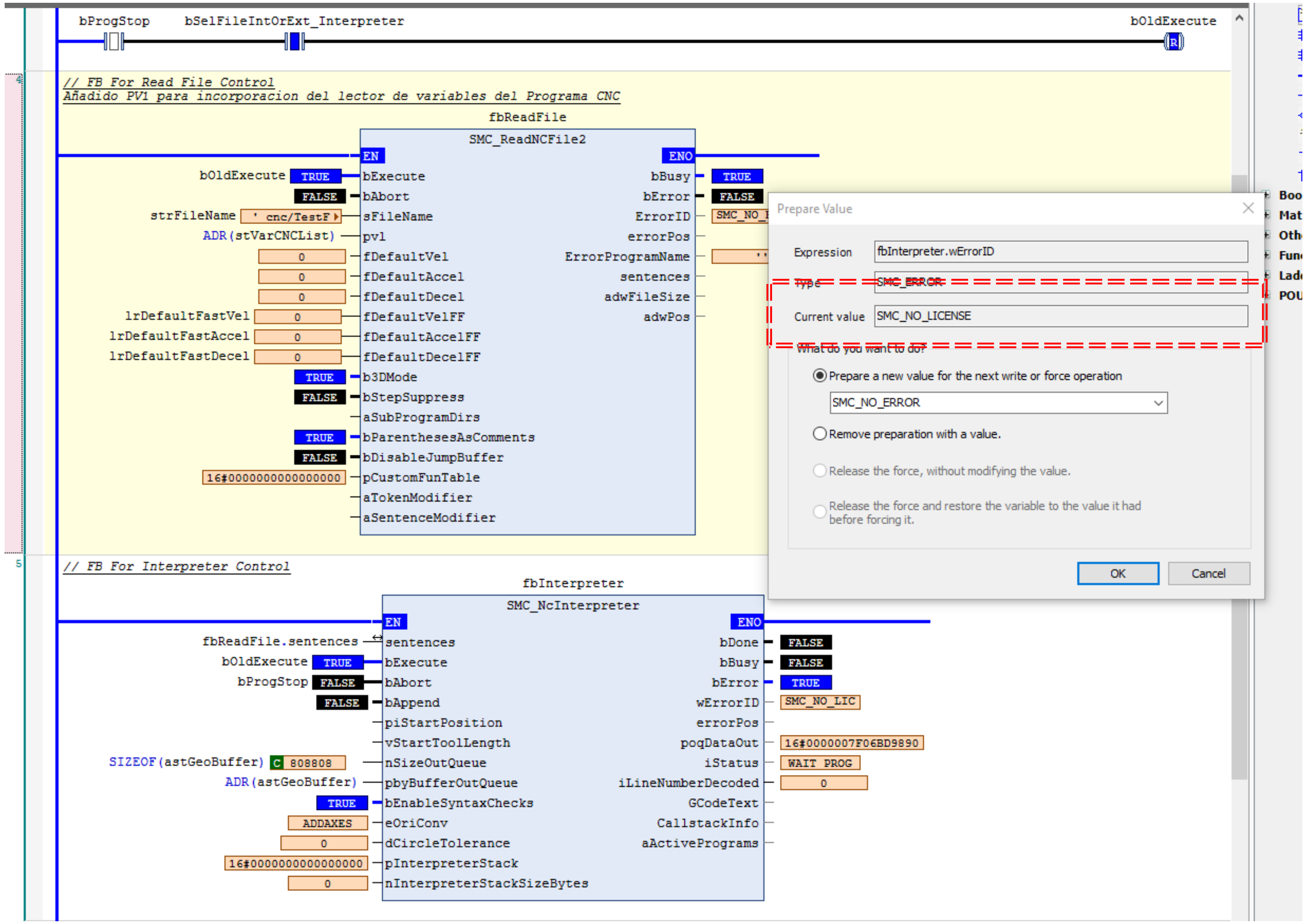
Status: SMC_AXIS_STATE.errorstop
Communication: error (1000)

Errors
Axis Error: 0 [16#00000000]
FB Error: SMC_ERROR.SMC_REGULATOR_OR_START_NOT_SET
uiDriveInterfaceError: 901
strDriveInterfaceError: no SoftMotion license. demo mode expired

Softmotion with expired license

ctrlX - Licences

- *A possible error due to lack of license is the following*



bProgStop bSelFileIntOrExt_Interpreter bOldExecute

// FB For Read File Control
Añadido PV1 para incorporacion del lector de variables del Programa CNC

fbReadFile
SMC_ReadNCFile2

Input value	Input	Output	Output value
	EN	ENO	
bOldExecute TRUE	bExecute	bBusy	TRUE
FALSE	bAbort	bError	FALSE
strFileName ' cnc/TestF	sFileName	ErrorID	SMC_NO_
ADR(stVarCNCList)	pv1	errorPos	
0	fDefaultVel	ErrorProgramName	''
0	fDefaultAccel	sentences	
0	fDefaultDecel	adwFileSize	
lrDefaultFastVel 0	fDefaultVelFF	adwPos	
lrDefaultFastAccel 0	fDefaultAccelFF		
lrDefaultFastDecel 0	fDefaultDecelFF		
TRUE	b3DMode		
FALSE	bStepSuppress		
	aSubProgramDirs		
TRUE	bParenthesesAsComments		
FALSE	bDisableJumpBuffer		
16#0000000000000000	pCustomFunTable		
	aTokenModifier		
	aSentenceModifier		

// FB For Interpreter Control

fbInterpreter
SMC_NcInterpreter

Input value	Input	Output	Output value
	EN	ENO	
fbReadFile.sentences	sentences	bDone	FALSE
bOldExecute TRUE	bExecute	bBusy	FALSE
bProgStop FALSE	bAbort	bError	TRUE
FALSE	bAppend	wErrorID	SMC_NO_LIC
	piStartPosition	errorPos	
	vStartToolLength	poqDataOut	16#0000007F06BD9890
SIZEOF(astGeoBuffer) C 808808	nSizeOutQueue	iStatus	WAIT_PROG
ADR(astGeoBuffer)	pbyBufferOutQueue	iLineNumberDecoded	0
TRUE	bEnableSyntaxChecks	GCodeText	
ADDAXES	eOriConv	CallstackInfo	
0	dCircleTolerance	aActivePrograms	
16#0000000000000000	pInterpreterStack		
0	nInterpreterStackSizeBytes		

Prepare Value

Expression: fbInterpreter.wErrorID

Type: SMC_ERROR

Current value: SMC_NO_LICENSE

What do you want to do?

● Prepare a new value for the next write or force operation

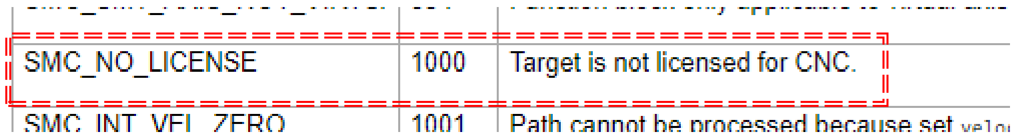
SMC_NO_ERROR

○ Remove preparation with a value.

○ Release the force, without modifying the value.

○ Release the force and restore the variable to the value it had before forcing it.

OK
Cancel
Boo
Mat
Oth
Fun
Lad
POU



SMC_NO_LICENSE	1000	Target is not licensed for CNC.
SMC_INT_VEL_ZERO	1001	Path cannot be processed because set velo

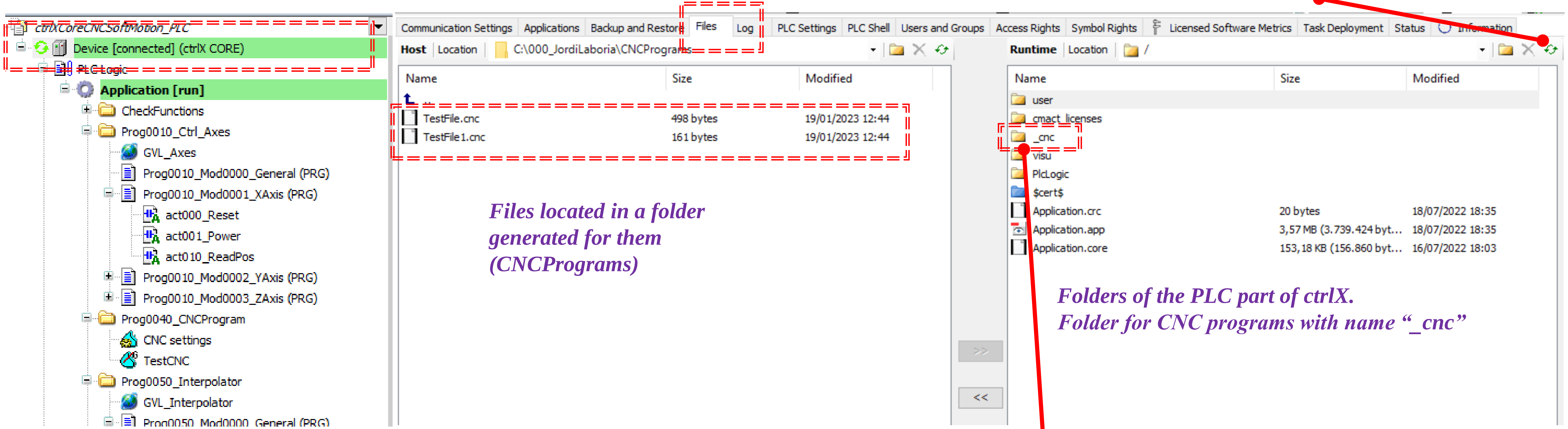
Sending external files to ctrlX

ctrlX - Sending external files to ctrlX

- *Files can be sent from the PC to ctrlX in various ways:*

In Online from the PLC it is possible to make the transfer from the "Files" option

If no folder appears at the time of access, we can activate the button ↻ to refresh the image



Host Location C:\000_JordiLaboria\CNCPrograms

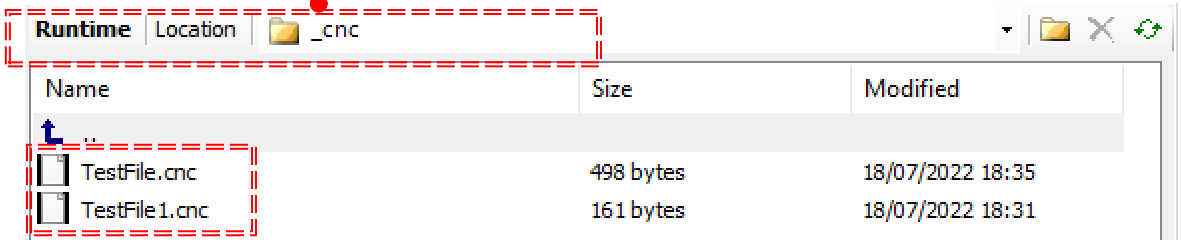
Name	Size	Modified
..		
TestFile.cnc	498 bytes	19/01/2023 12:44
TestFile1.cnc	161 bytes	19/01/2023 12:44

Runtime Location /

Name	Size	Modified
user		
cmact_licenses		
_cnc		
visu		
PlcLogic		
$cert$		
Application.crc	20 bytes	18/07/2022 18:35
Application.app	3,57 MB (3.739.424 byt...	18/07/2022 18:35
Application.core	153,18 KB (156.860 byt...	16/07/2022 18:03

Files located in a folder generated for them (CNCPrograms)

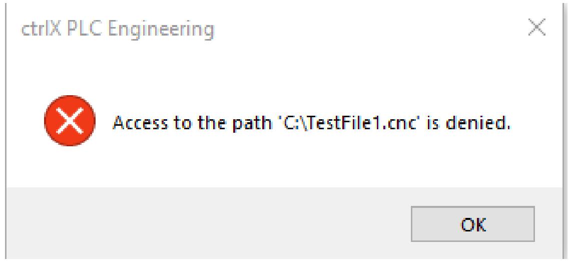
Folders of the PLC part of ctrlX.
Folder for CNC programs with name "_cnc"



Runtime Location _cnc

Name	Size	Modified
..		
TestFile.cnc	498 bytes	18/07/2022 18:35
TestFile1.cnc	161 bytes	18/07/2022 18:31

Some folders do not allow transfer back and forth and the following notice appears



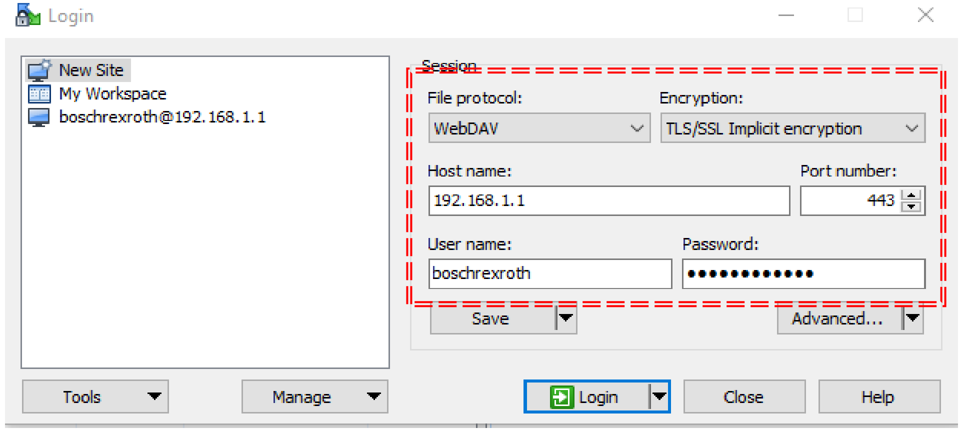
ctrlX PLC Engineering

Access to the path 'C:\TestFile1.cnc' is denied.

ctrlX - Sending external files to ctrlX

- *The transfer can also be sent using WinSCP FTP Client:*

Configuration used for communication



File protocol: WebDAV

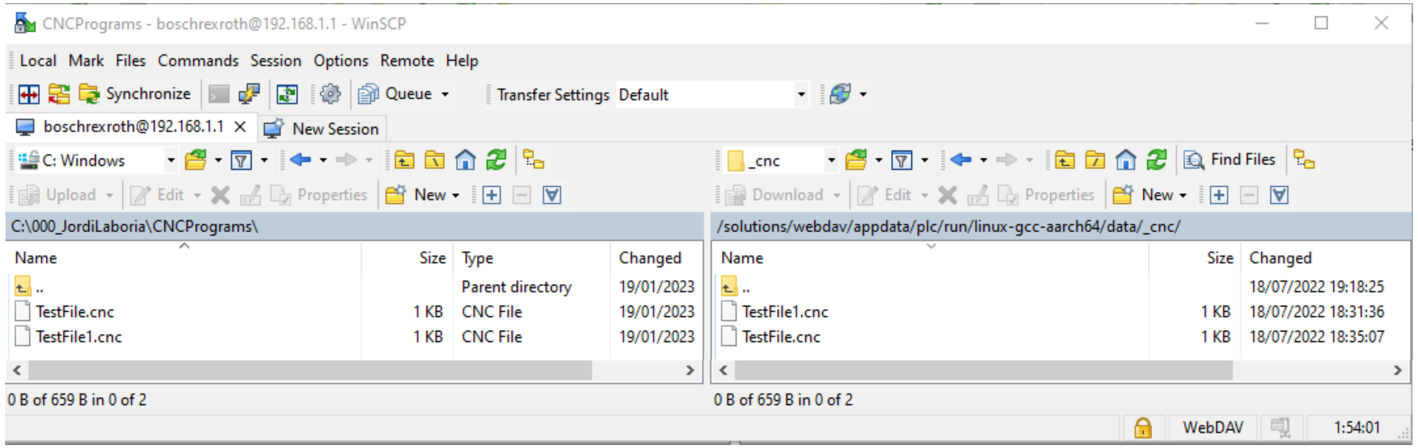
Encryption: TLS/SSL Implicit encryption

Host name: 192.168.1.1

Port number: 443

User name: boschrexroth

Password: ●●●●●●●●●●●●●


CNCPrograms - boschrexroth@192.168.1.1 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
boschrexroth@192.168.1.1
New Session
C: Windows
_cnc
Find Files
Upload
Edit
Properties
New
Download
WebDAV
1:54:01

C:\000_JordiLaboria\CNCPrograms\

Name	Size	Type	Changed
..		Parent directory	19/01/2023
TestFile.cnc	1 KB	CNC File	19/01/2023
TestFile1.cnc	1 KB	CNC File	19/01/2023

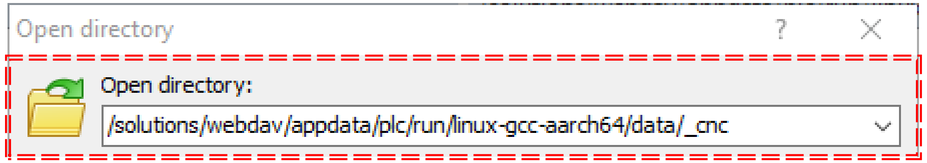
0 B of 659 B in 0 of 2

/solutions/webdav/appdata/plc/run/linux-gcc-aarch64/data/_cnc/

Name	Size	Changed
..		18/07/2022 19:18:25
TestFile1.cnc	1 KB	18/07/2022 18:31:36
TestFile.cnc	1 KB	18/07/2022 18:35:07

0 B of 659 B in 0 of 2

This is the route we must look for to access the CNC programs



Open directory: /solutions/webdav/appdata/plc/run/linux-gcc-aarch64/data/_cnc

StartUp Parameters
Real Axes

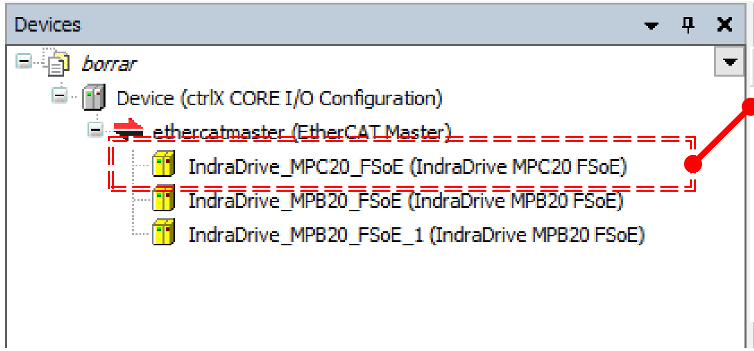
|

ctrlX - *Startup Real axes parameters*

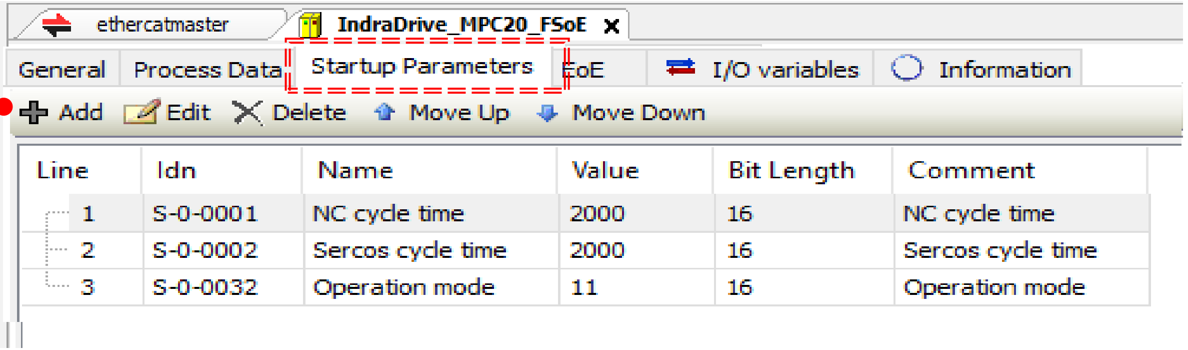
- *For the real axes we have an option in the IO configuration that allows us to insert a series of parameters that will be modified during the startup of the equipment.*

Use this software option for configuration of Startup parameters

Open ctrlX I/O Engineering


Devices

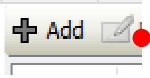
- borrar
  - Device (ctrlX CORE I/O Configuration)
    - ethercatmaster (EtherCAT Master)
      - IndraDrive_MPC20_FSoE (IndraDrive MPC20 FSoE)
      - IndraDrive_MPB20_FSoE (IndraDrive MPB20 FSoE)
      - IndraDrive_MPB20_FSoE_1 (IndraDrive MPB20 FSoE)



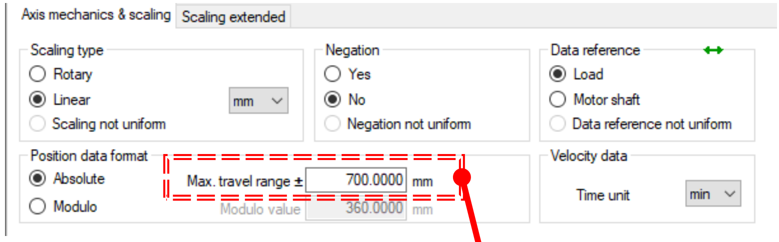
Line	Idn	Name	Value	Bit Length	Comment
1	S-0-0001	NC cycle time	2000	16	NC cycle time
2	S-0-0002	Sercos cycle time	2000	16	Sercos cycle time
3	S-0-0032	Operation mode	11	16	Operation mode

By default these are the parameters that are assigned in the startup of the axes

- *We can add a parameter using*



In the example we are going to modify the parameter S-0-0278, (Max. Travel Range)


Axis mechanics & scaling | Scaling extended

Scaling type: ○ Rotary ● Linear mm ○ Scaling not uniform

Negation: ○ Yes ● No ○ Negation not uniform

Data reference: ● Load ○ Motor shaft ○ Data reference not uniform

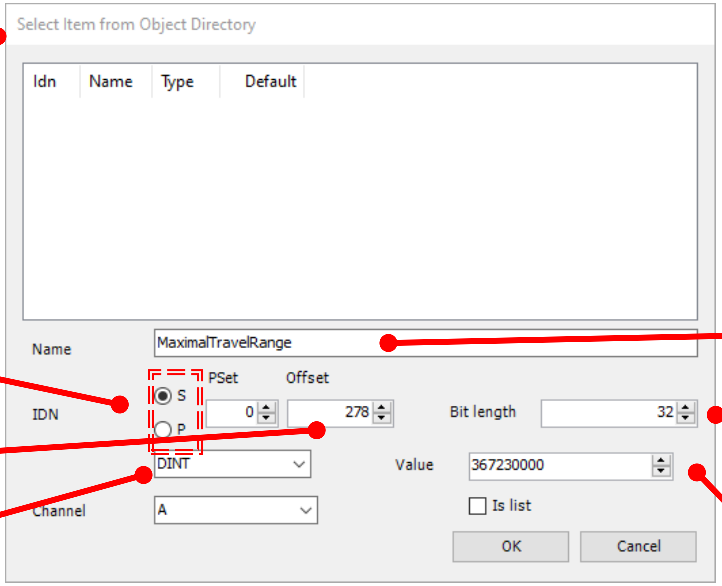
Position data format: ● Absolute ○ Modulo

Max. travel range ± 700.0000 mm

Modulo value 360.0000 mm

Velocity data: Time unit min

S-0-0278


Select Item from Object Directory

Idn	Name	Type	Default

Name: MaximalTravelRange

IDN: ● S ○ P PSet 0 Offset 278

Bit length 32

DINT

Value 367230000

□ Is list

Channel A

OK Cancel

Parameter type

IDN Number

Value Type

Attention to the values to be applied and to the factors generated by the decimals. In the example the number 36723 is multiplied by 10000

Parameter name (as an indicator only)

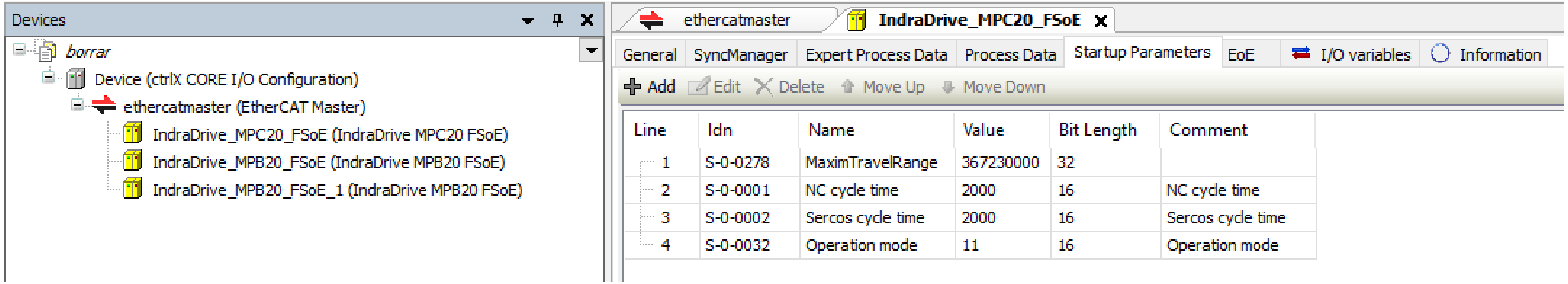
Parameter length (look in the docu)

Value to write in the parameter

rexroth
A Bosch Company

ctrlX - *Startup Real axes parameters*

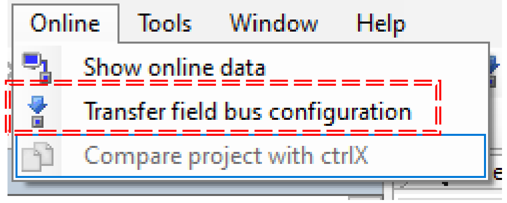
- *Once "ok" is activated, the parameter is inserted into the StartUp structure*



borrar
Device (ctrlX CORE I/O Configuration)
ethercatmaster (EtherCAT Master)
IndraDrive_MPC20_FSoE (IndraDrive MPC20 FSoE)
IndraDrive_MPB20_FSoE (IndraDrive MPB20 FSoE)
IndraDrive_MPB20_FSoE_1 (IndraDrive MPB20 FSoE)

Line	Idn	Name	Value	Bit Length	Comment
1	S-0-0278	MaximTravelRange	367230000	32	
2	S-0-0001	NC cycle time	2000	16	NC cycle time
3	S-0-0002	Sercos cycle time	2000	16	Sercos cycle time
4	S-0-0032	Operation mode	11	16	Operation mode

- *The structure or any modification carried out in the equipment must be activated using the “Transfer Field bus configuration” button.*

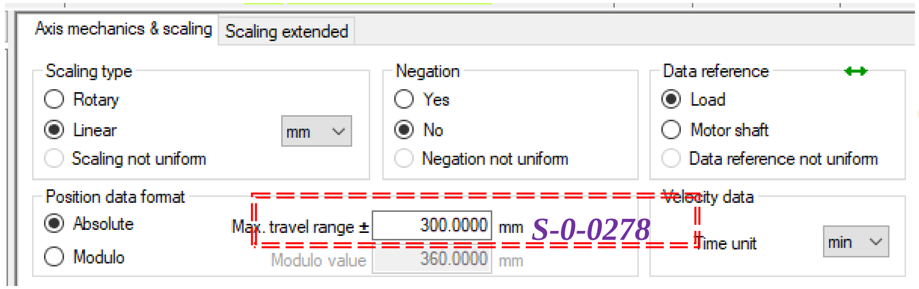


For the startup to activate, we must remove power from the system and start again

ctrlX - *Startup Real axes parameters*

- *An example of this would be the following*

Changed the value of S-0-0278 to 300 in Online



Axis mechanics & scaling | Scaling extended

Scaling type: ○ Rotary ● Linear (mm) ○ Scaling not uniform

Negation: ○ Yes ● No ○ Negation not uniform

Data reference: ● Load ○ Motor shaft ○ Data reference not uniform

Position data format: ● Absolute ○ Modulo

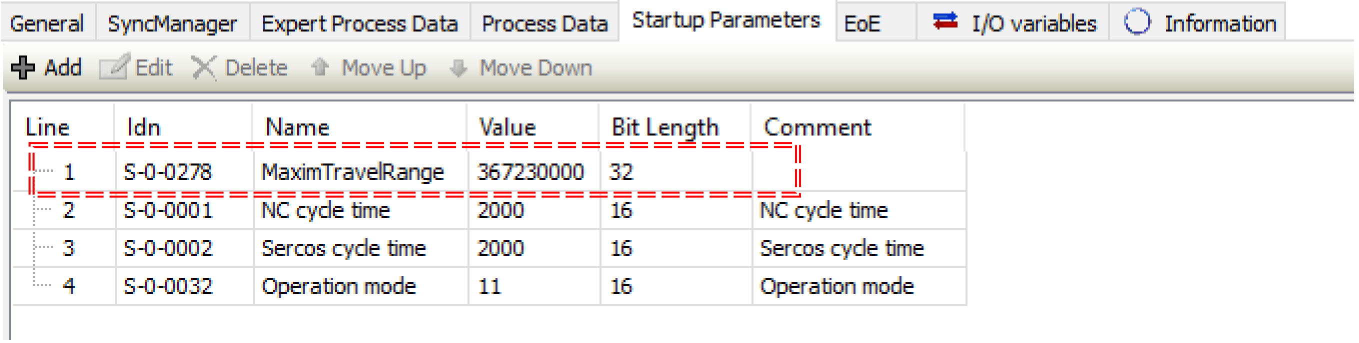
Max. travel range ± 300.0000 mm *S-0-0278*

Modulo value 360.0000 mm

Velocity data: Time unit min

Quitamos tensión

As in the StartUp Parameters the S-0-0278 has value 367230000



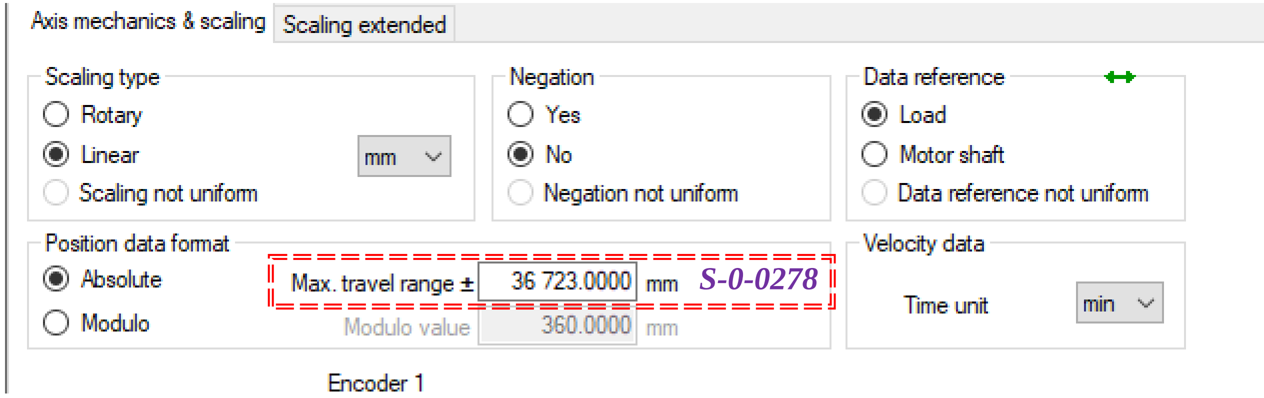
General | SyncManager | Expert Process Data | Process Data | Startup Parameters | EoE | I/O variables | Information

Add | Edit | Delete | Move Up | Move Down

Line	Idn	Name	Value	Bit Length	Comment
1	S-0-0278	MaximTravelRange	367230000	32	
2	S-0-0001	NC cycle time	2000	16	NC cycle time
3	S-0-0002	Sercos cycle time	2000	16	Sercos cycle time
4	S-0-0032	Operation mode	11	16	Operation mode

We start again and visualize the value in Online

When we reconnect we will see that the axis now has a value of 32723.0000 (with four decimal places)



Axis mechanics & scaling | Scaling extended

Scaling type: ○ Rotary ● Linear (mm) ○ Scaling not uniform

Negation: ○ Yes ● No ○ Negation not uniform

Data reference: ● Load ○ Motor shaft ○ Data reference not uniform

Position data format: ● Absolute ○ Modulo

Max. travel range ± 36 723.0000 mm *S-0-0278*

Modulo value 360.0000 mm

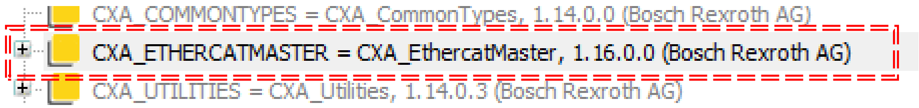
Velocity data: Time unit min

Encoder 1

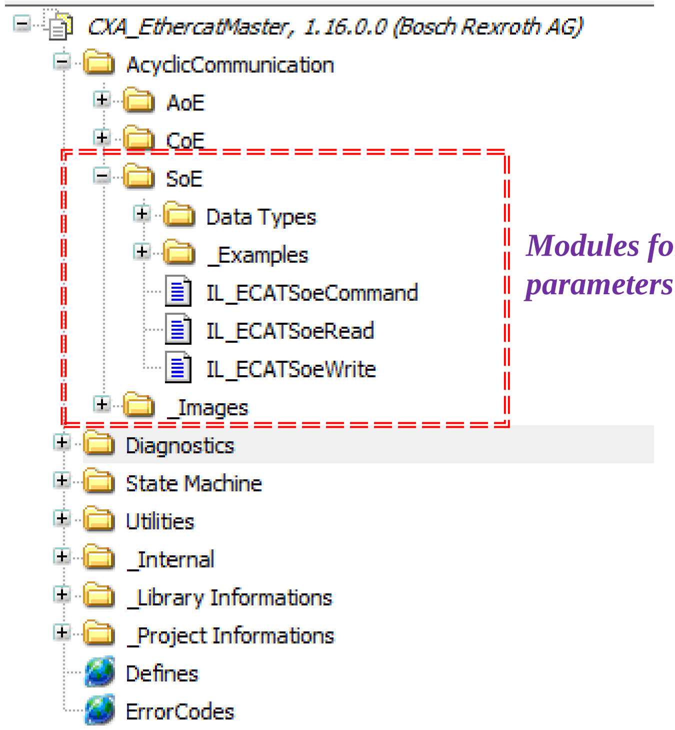
Modules for reading / writing EtherCat parameters

ctrlX - *Modules for reading / writing EtherCat parameters*

- *The system also has, within the "CXA_ETHERCATMASTER" library, the option for reading and writing access to parameters in the EtherCat network*

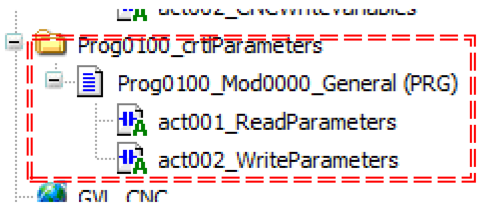


CXA_COMMONTYPES = CXA_CommonTypes, 1.14.0.0 (Bosch Rexroth AG)
CXA_ETHERCATMASTER = CXA_EthercatMaster, 1.16.0.0 (Bosch Rexroth AG)
CXA_UTILITIES = CXA_Utilities, 1.14.0.3 (Bosch Rexroth AG)



```
CXA_EthercatMaster, 1.16.0.0 (Bosch Rexroth AG)
  AcyclicCommunication
    AoE
    CoE
    SoE
      Data Types
      _Examples
      IL_ECATSoeCommand
      IL_ECATSoeRead
      IL_ECATSoeWrite
    _Images
  Diagnostics
  State Machine
  Utilities
  _Internal
  _Library Informations
  _Project Informations
  Defines
  ErrorCodes
```

Modules for reading / writing EtherCat parameters using SoE protocol

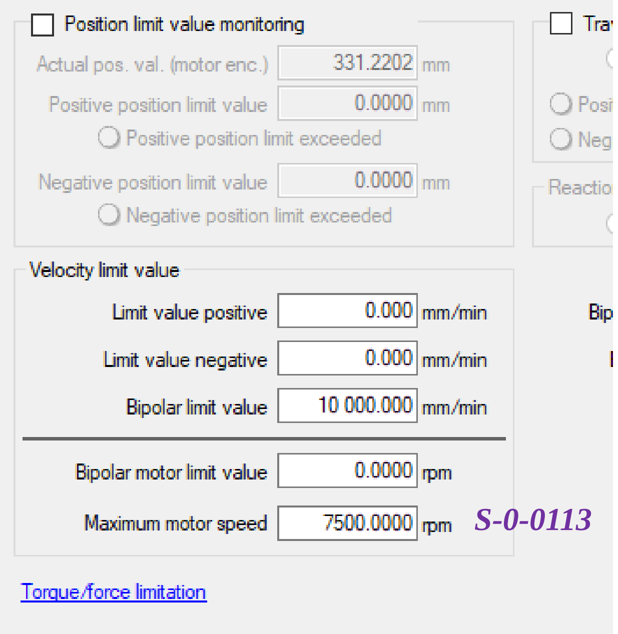


```
act002_CNCWriteVariables
Prog0100_ctrlParameters
  Prog0100_Mod0000_General (PRG)
    act001_ReadParameters
    act002_WriteParameters
GVL_CNC
```

Program structure used in the example

ctrlX - *Modules for reading / writing EtherCat parameters*

- ***Example of reading the maximum motor speed parameter***



☐ Position limit value monitoring

Actual pos. val. (motor enc.) 331.2202 mm

Positive position limit value 0.0000 mm

○ Positive position limit exceeded

Negative position limit value 0.0000 mm

○ Negative position limit exceeded

Velocity limit value

Limit value positive 0.000 mm/min

Limit value negative 0.000 mm/min

Bipolar limit value 10 000.000 mm/min

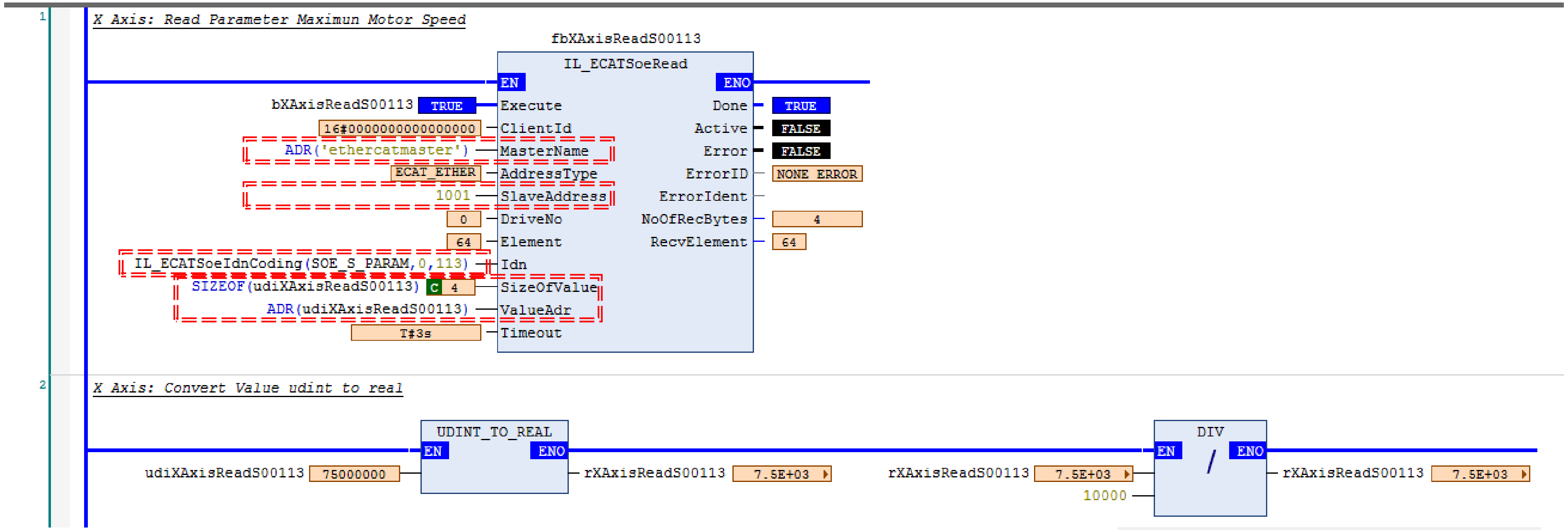
Bipolar motor limit value 0.0000 rpm

Maximum motor speed 7500.0000 rpm

S-0-0113

Torque/force limitation

The reading module should only contain the master 'ethercatmaster' as a pointer, the address of the ethercat slave number, the parameter indexer and obviously the parameter on which the value will be read



X Axis: Read Parameter Maximun Motor Speed

fbXAxisReadS00113 : IL_ECATSoeRead

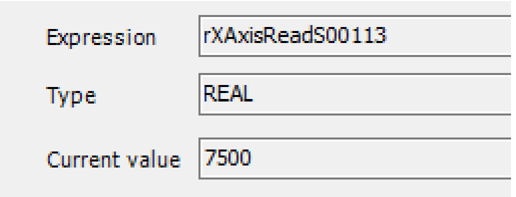
- EN / ENO
- bXAxisReadS00113 TRUE — Execute
- 16#0000000000000000 — ClientId
- ADR('ethercatmaster') — MasterName
- ECAT_ETHER — AddressType
- 1001 — SlaveAddress
- 0 — DriveNo
- 64 — Element
- IL_ECATSoeIdnCoding(SOE_S_PARAM,0,113) — Idn
- SIZEOF(udiXAxisReadS00113) C 4 — SizeOfValue
- ADR(udiXAxisReadS00113) — ValueAdr
- T#3s — Timeout
- Done — TRUE
- Active — FALSE
- Error — FALSE
- ErrorID — NONE_ERROR
- ErrorIdent
- NoOfRecBytes — 4
- RecvElement — 64

X Axis: Convert Value udint to real

- UDINT_TO_REAL: udiXAxisReadS00113 75000000 → rXAxisReadS00113 7.5E+03
- DIV: rXAxisReadS00113 7.5E+03 / 10000 → rXAxisReadS00113 7.5E+03

```
udiXAxisReadS00113: UDINT;
rXAxisReadS00113: REAL;
```

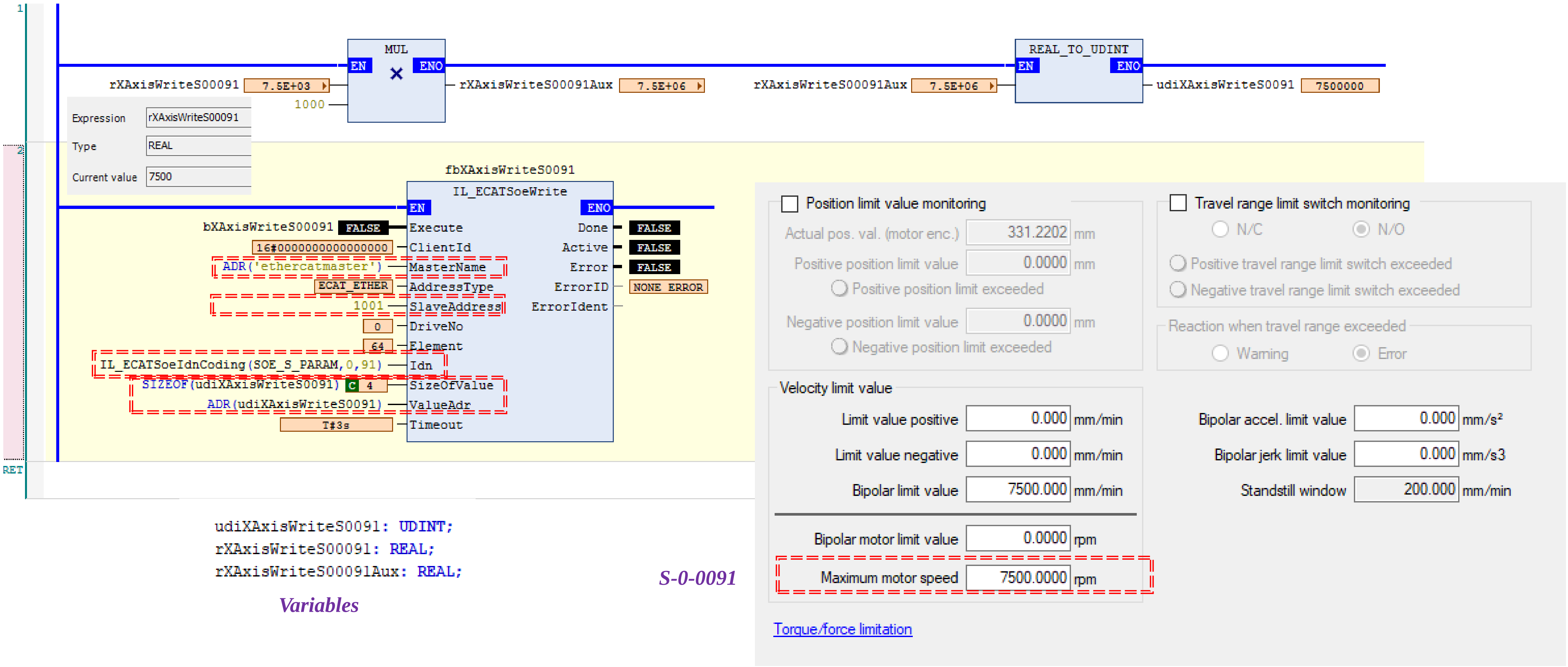
Variables



Expression	rXAxisReadS00113
Type	REAL
Current value	7500

ctrlX - *Modules for reading / writing EtherCat parameters*

- *Example of writing the maximum speed bipolar parameter*



```
1
MUL
EN            ENO
rXAxisWriteS00091 7.5E+03   ×   rXAxisWriteS00091Aux 7.5E+06
1000

REAL_TO_UDINT
EN            ENO
rXAxisWriteS00091Aux 7.5E+06    udiXAxisWriteS0091 7500000

Expression      rXAxisWriteS00091
Type            REAL
Current value   7500

2
fbXAxisWriteS0091
IL_ECATSoeWrite
EN                                            ENO
bXAxisWriteS00091 FALSE  Execute        Done        FALSE
16#0000000000000000      ClientId       Active      FALSE
ADR('ethercatmaster')    MasterName     Error       FALSE
ECAT_ETHER               AddressType    ErrorID     NONE ERROR
1001                     SlaveAddress   ErrorIdent
0                        DriveNo
64                       Element
IL_ECATSoeIdnCoding(SOE_S_PARAM,0,91)   Idn
SIZEOF(udiXAxisWriteS0091) C 4          SizeOfValue
ADR(udiXAxisWriteS0091)                 ValueAdr
T#3s                                    Timeout
RET
```

```
udiXAxisWriteS0091: UDINT;
rXAxisWriteS00091: REAL;
rXAxisWriteS00091Aux: REAL;
```

Variables

S-0-0091

☐ Position limit value monitoring

Actual pos. val. (motor enc.)	331.2202	mm
Positive position limit value	0.0000	mm
○ Positive position limit exceeded		
Negative position limit value	0.0000	mm
○ Negative position limit exceeded		

☐ Travel range limit switch monitoring

○ N/C ● N/O

○ Positive travel range limit switch exceeded

○ Negative travel range limit switch exceeded

Reaction when travel range exceeded

○ Warning ● Error

Velocity limit value

Limit value positive	0.000	mm/min
Limit value negative	0.000	mm/min
Bipolar limit value	7500.000	mm/min
Bipolar motor limit value	0.0000	rpm
Maximum motor speed	7500.0000	rpm

Bipolar accel. limit value	0.000	mm/s²
Bipolar jerk limit value	0.000	mm/s3
Standstill window	200.000	mm/min

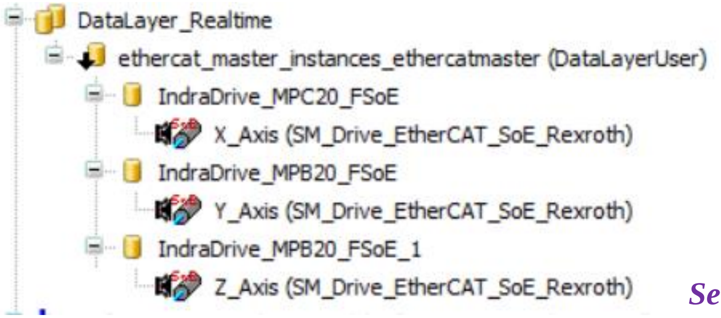
Torque/force limitation

Task assignment

ctrlX - *Task assignment*

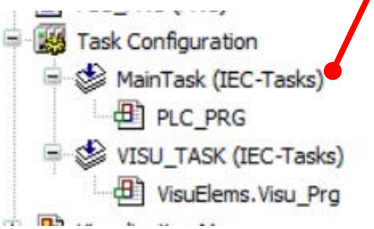
- ***For the example of the application, two tasks have been used, each one with a different behavior and also in the case of the MainTask depending on whether we are in simulation mode or in simulation mode.***

With the axes in EtherCat and therefore with the "simulation" mode deactivated

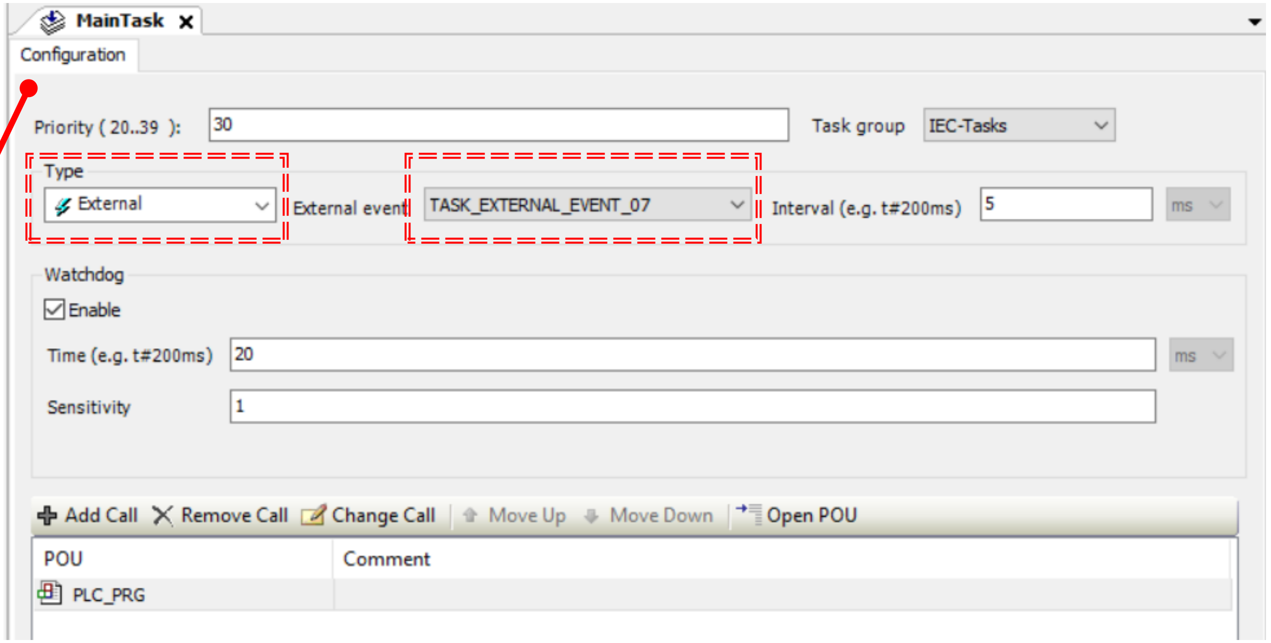


DataLayer_Realtime
ethercat_master_instances_ethercatmaster (DataLayerUser)
IndraDrive_MPC20_FSoE
X_Axis (SM_Drive_EtherCAT_SoE_Rexroth)
IndraDrive_MPB20_FSoE
Y_Axis (SM_Drive_EtherCAT_SoE_Rexroth)
IndraDrive_MPB20_FSoE_1
Z_Axis (SM_Drive_EtherCAT_SoE_Rexroth)

Selection of external event and Task_External_Event_07 (EtherCat Master)



Task Configuration
MainTask (IEC-Tasks)
PLC_PRG
VISU_TASK (IEC-Tasks)
VisuElems.Visu_Prg



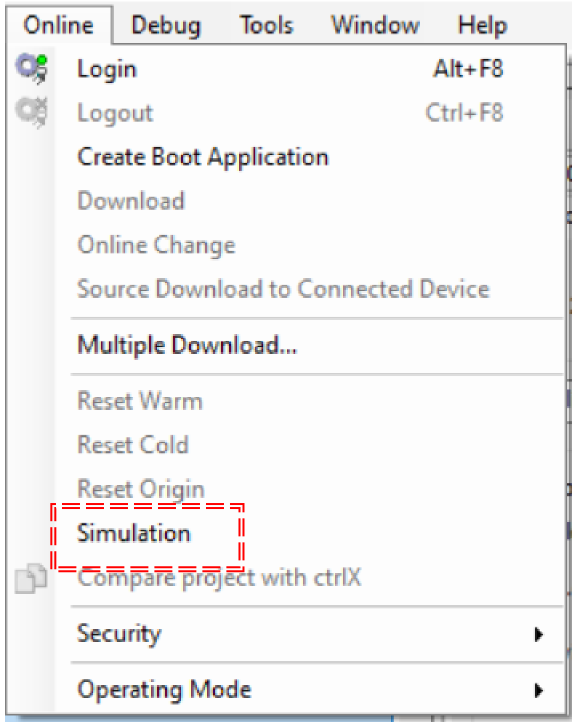
Priority (20..39): 30
Task group: IEC-Tasks

Type: External
External event: TASK_EXTERNAL_EVENT_07
Interval (e.g. t#200ms): 5 ms

Watchdog
☑ Enable
Time (e.g. t#200ms): 20 ms
Sensitivity: 1

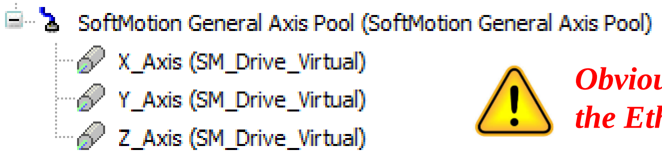
POU	Comment
PLC_PRG	

Simulation Off



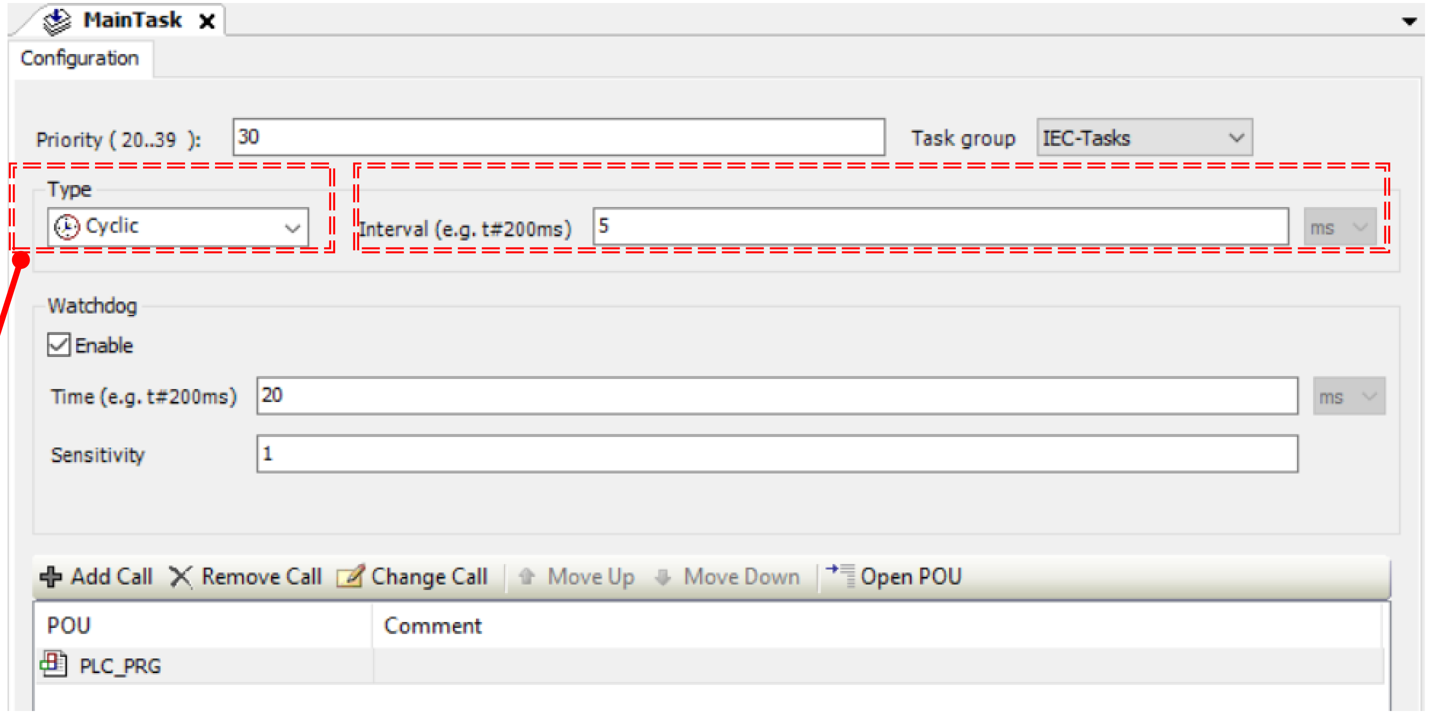
ctrlX - *Task assignment*

If we are going to use the simulation mode we will have to modify the data of the Main Task so that it can activate the modules that are executed from it



SoftMotion General Axis Pool (SoftMotion General Axis Pool)
- X_Axis (SM_Drive_Virtual)
- Y_Axis (SM_Drive_Virtual)
- Z_Axis (SM_Drive_Virtual)

Obviously, the virtual axes also work with the EtherCat part activated.



MainTask

Configuration

Priority (20..39): 30

Task group: IEC-Tasks

Type: Cyclic

Interval (e.g. t#200ms): 5 ms

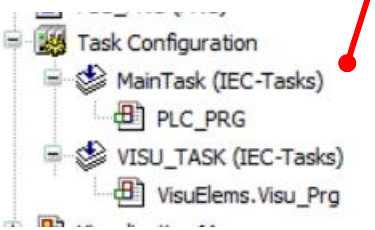
Watchdog

☑ Enable

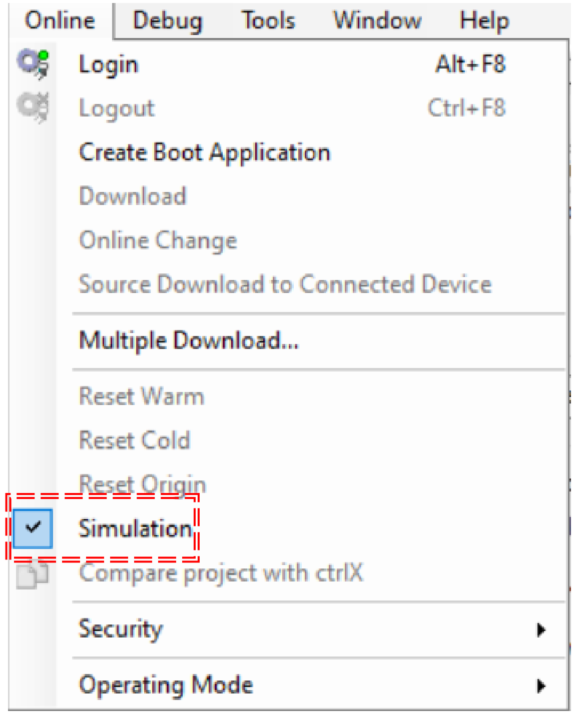
Time (e.g. t#200ms): 20 ms

Sensitivity: 1

POU	Comment
PLC_PRG	



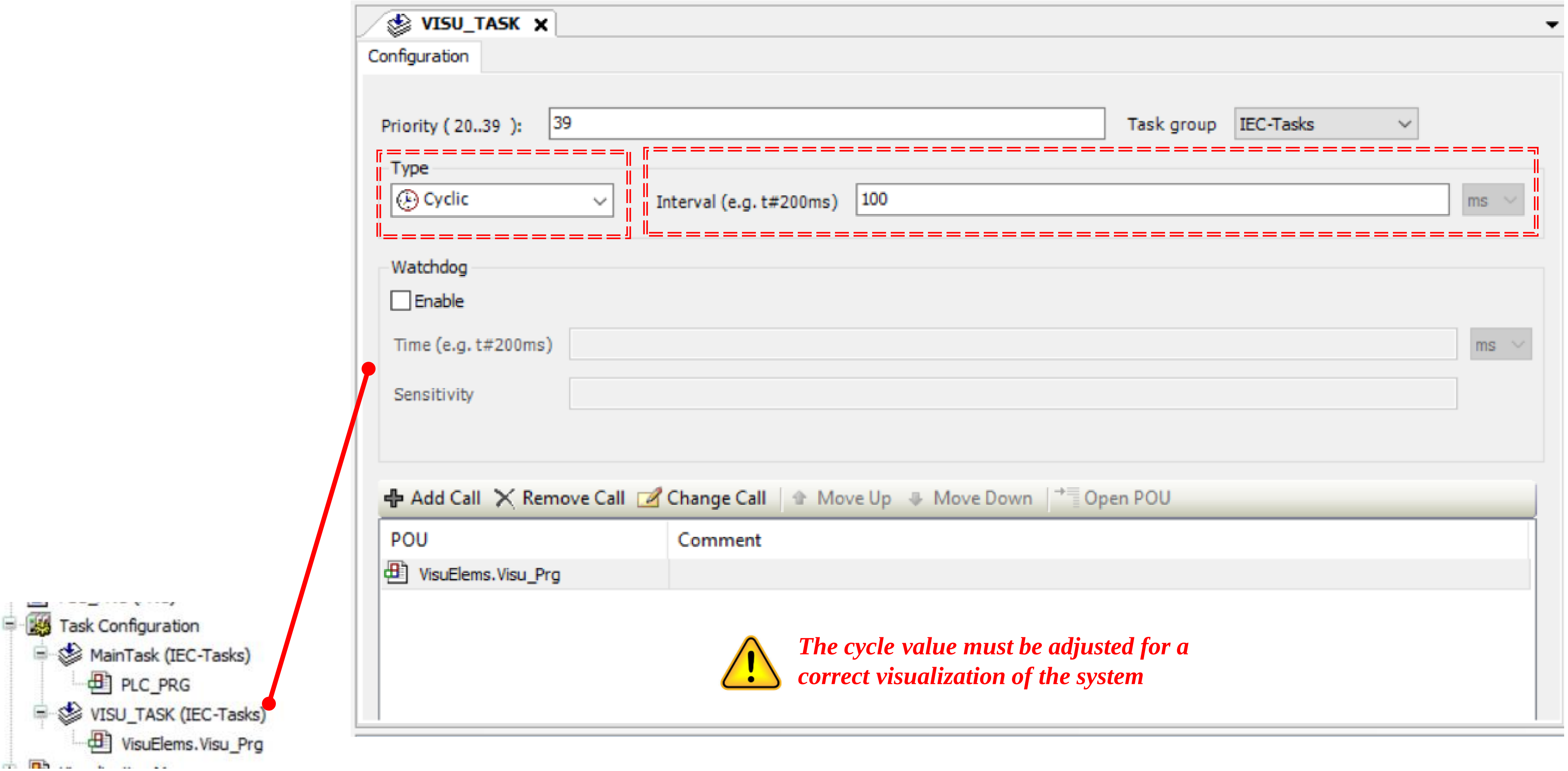
Task Configuration
- MainTask (IEC-Tasks)
  - PLC_PRG
- VISU_TASK (IEC-Tasks)
  - VisuElems.Visu_Prg



Simulation On

ctrlX - *Task assignment*

The visualization task is used in both cases cyclically and the "call" time can be modified depending on the speed of execution that we want to visualize.



VISU_TASK

Configuration

Priority (20..39): 39

Task group: IEC-Tasks

Type: Cyclic

Interval (e.g. t#200ms): 100 ms

Watchdog

□ Enable

Time (e.g. t#200ms): ms

Sensitivity:

POU	Comment
VisuElems.Visu_Prg	

- Task Configuration
  - MainTask (IEC-Tasks)
    - PLC_PRG
  - VISU_TASK (IEC-Tasks)
    - VisuElems.Visu_Prg

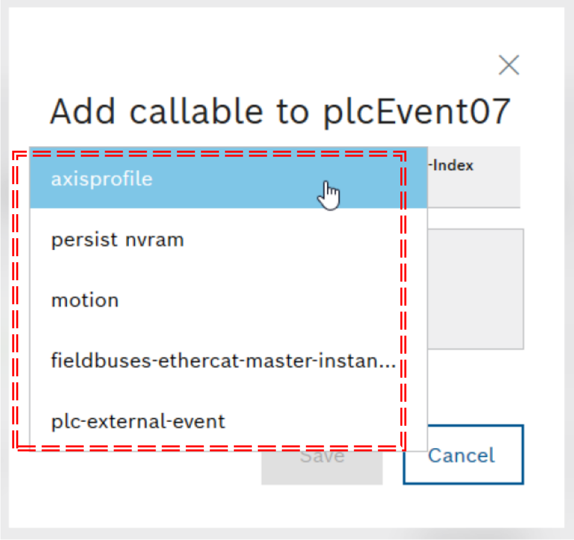
The cycle value must be adjusted for a correct visualization of the system

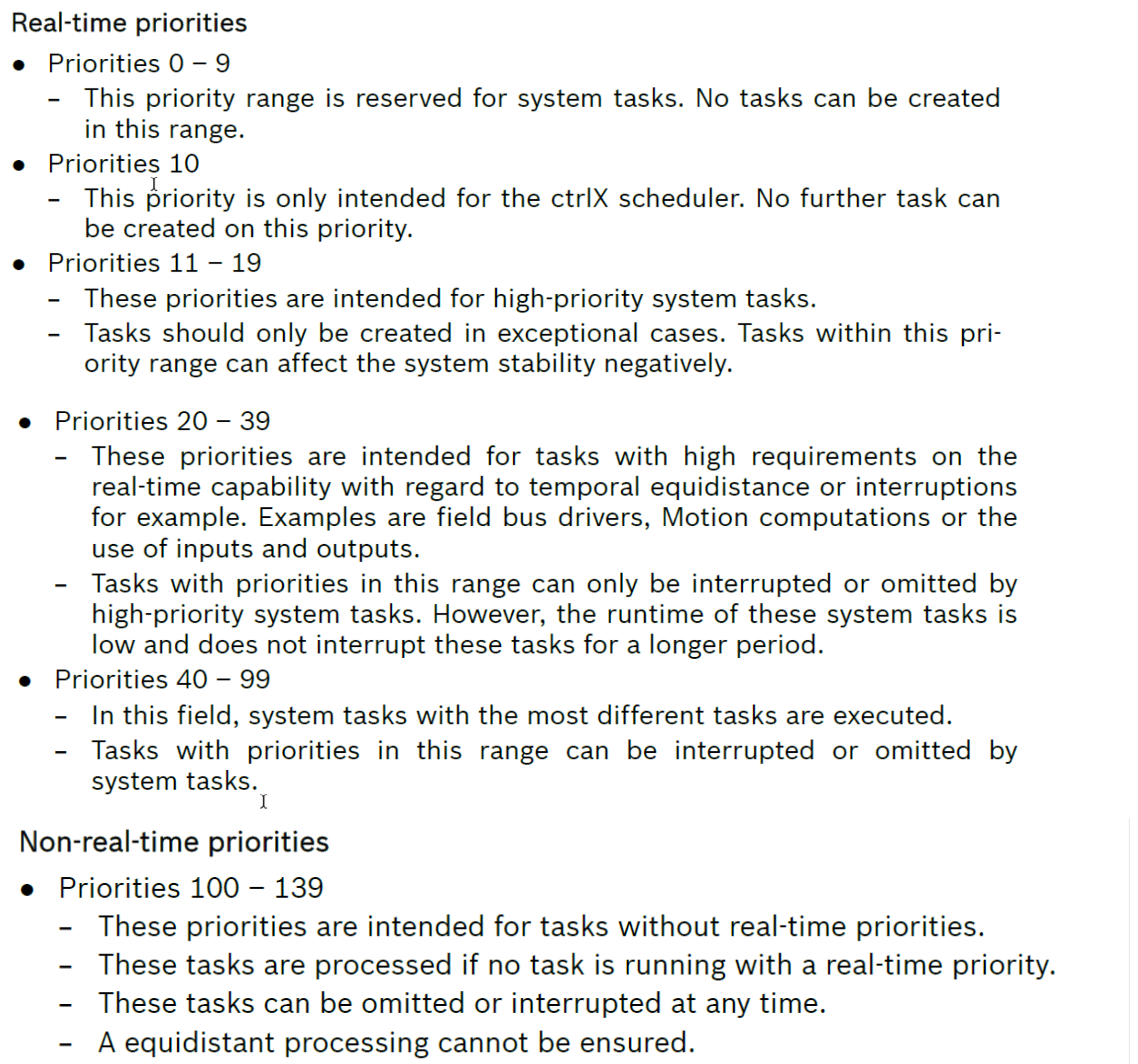
ctrlX - *Task assignment*

Details of the priorities and the operation of the team's tasks are available in the ctrlX Core manual.

Table 1: Task priorities

Priority	Note
0	Highest real-time priority
10	ctrlX scheduler
11-19	High-priority system tasks
20-39	ctrlXAutomation task, PLC tasks...
...	
99	Lowest real-time priority
100	Highest non-real-time priority
...	
139	Lowest non-real-time priority





Real-time priorities

- Priorities 0 – 9
  - This priority range is reserved for system tasks. No tasks can be created in this range.
- Priorities 10
  - This priority is only intended for the ctrlX scheduler. No further task can be created on this priority.
- Priorities 11 – 19
  - These priorities are intended for high-priority system tasks.
  - Tasks should only be created in exceptional cases. Tasks within this priority range can affect the system stability negatively.
- Priorities 20 – 39
  - These priorities are intended for tasks with high requirements on the real-time capability with regard to temporal equidistance or interruptions for example. Examples are field bus drivers, Motion computations or the use of inputs and outputs.
  - Tasks with priorities in this range can only be interrupted or omitted by high-priority system tasks. However, the runtime of these system tasks is low and does not interrupt these tasks for a longer period.
- Priorities 40 – 99
  - In this field, system tasks with the most different tasks are executed.
  - Tasks with priorities in this range can be interrupted or omitted by system tasks.

Non-real-time priorities

- Priorities 100 – 139
  - These priorities are intended for tasks without real-time priorities.
  - These tasks are processed if no task is running with a real-time priority.
  - These tasks can be omitted or interrupted at any time.
  - A equidistant processing cannot be ensured.

Thanks for your attention