

Festo_Positioning_Basic library – Overview FPosB Basicproject and CAN / EtherCAT settings

Commissioning of a selected system with the Festo_Positioning_Basic library is explained step by step, so that it's successfully integrated into the user program.

FPosB

Title Festo_Positioning_Basic library – Overview FPosB Basicproject
Version 1.30
Document no. 100102
Originalen
AuthorFesto

Last saved 02.06.2021

Copyright Notice

This documentation is the intellectual property of Festo SE & Co. KG, which also has the exclusive copyright. Any modification of the content, duplication or reprinting of this documentation as well as distribution to third parties can only be made with the express consent of Festo SE & Co. KG.

Festo SE & Co KG reserves the right to make modifications to this document in whole or in part. All brand and product names are trademarks or registered trademarks of their respective owners.

Legal Notice

Hardware, software, operating systems and drivers may only be used for the applications described and only in conjunction with components recommended by Festo SE & Co. KG.

Festo SE & Co. KG does not accept any liability for damages arising from the use of any incorrect or incomplete information contained in this documentation or any information missing therefrom.

Defects resulting from the improper handling of devices and modules are excluded from the warranty.

The data and information specified in this document should not be used for the implementation of safety functions relating to the protection of personnel and machinery.

No liability is accepted for claims for damages arising from a failure or functional defect. In other respects, the regulations with regard to liability from the terms and conditions of delivery, payment and use of software of Festo SE & Co. KG, which can be found at www.festo.com and can be supplied on request, shall apply.

All data contained in this document do not represent guaranteed specifications, particularly with regard to functionality, condition or quality, in the legal sense.

The information in this document serves only as basic information for the implementation of a specific, hypothetical application and is in no way intended as a substitute for the operating instructions of the respective manufacturers and the design and testing of the respective application by the user.

The operating instructions for Festo products can be found at www.festo.com/sp.

Users of this document (application note) must verify that all functions described here also work correctly in the application. By reading this document and adhering to the specifications contained therein, users are also solely responsible for their own application.

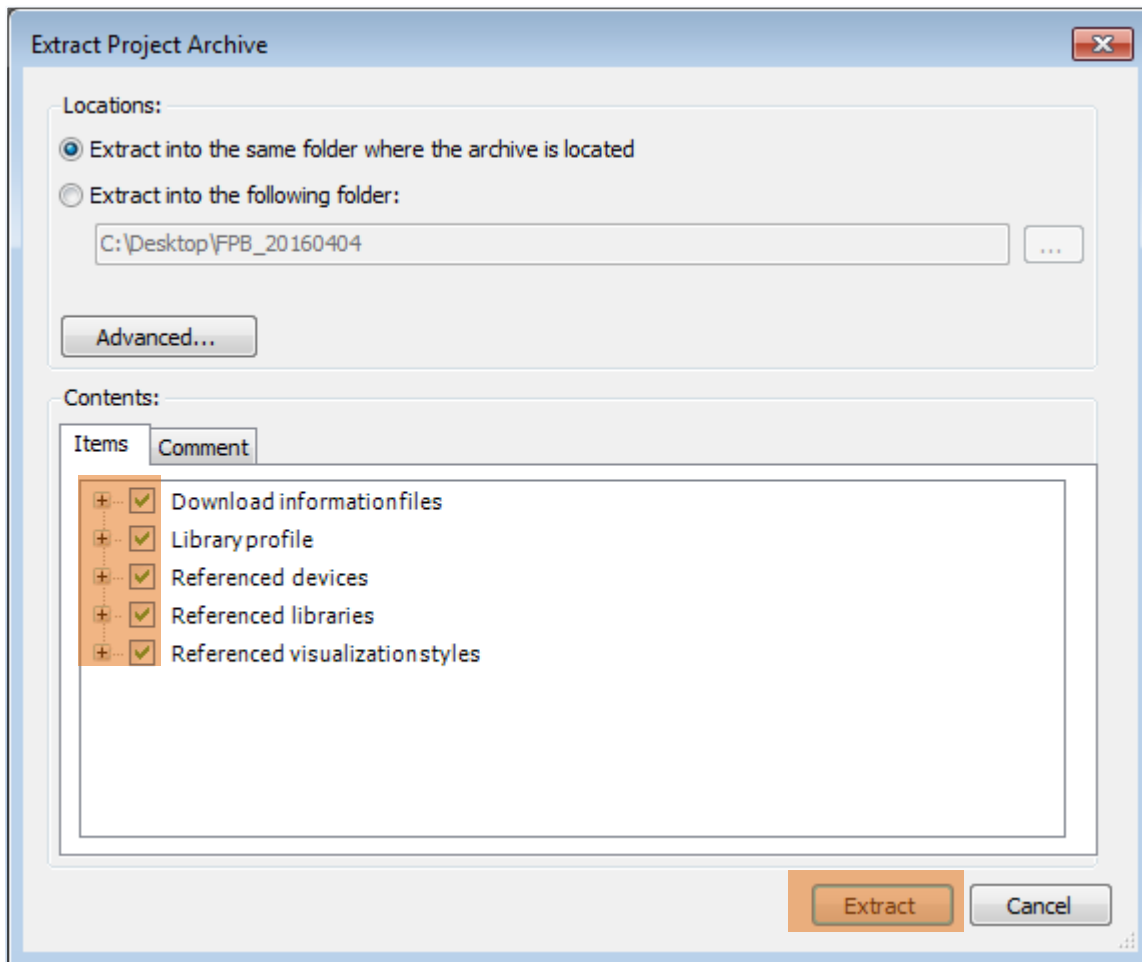
Table of contents

1	Downloading and installing FPOSB.lib	5
2	Basic project overview	8
2.1	Devices	8
2.1.1	CNC settings und 00_CNCs	8
2.1.2	Task configuration	9
2.1.3	Visualisation	12
2.1.4	CAN bus	13
2.1.5	EtherCAT	15
2.1.6	Soft motion general axis pool	16
2.2	POUs	16
2.2.1	00_GVLs.....	17
2.2.2	00_Main_Motion.....	18

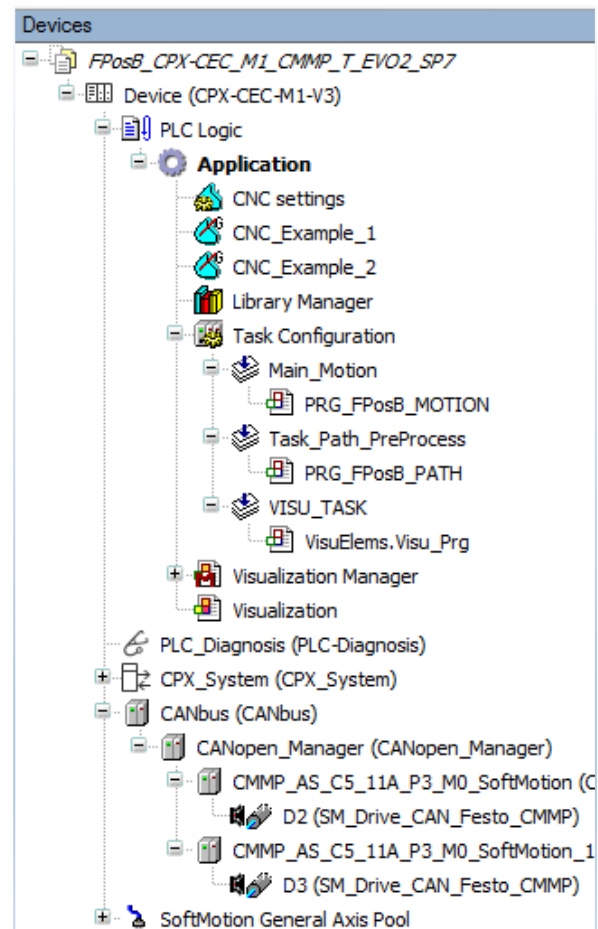
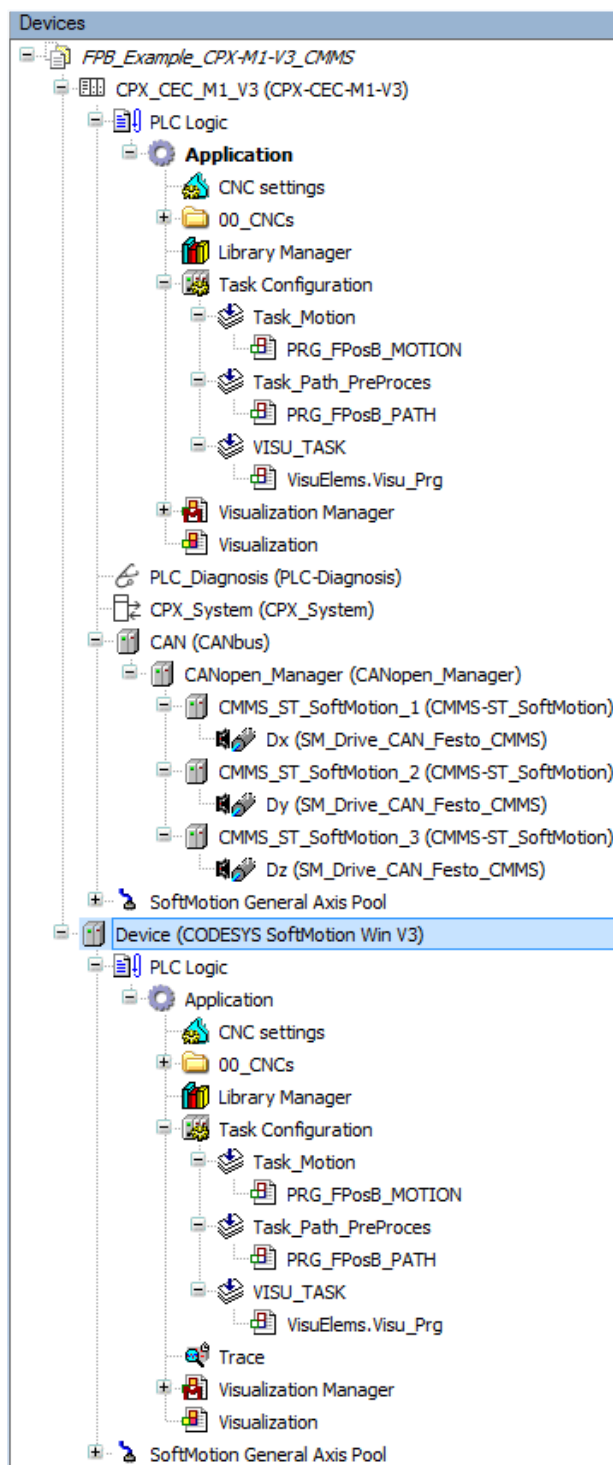
1 Downloading and installing FPOSB.lib

Codesys V3.5 SP7 patch 2 is a prerequisite for use of the library. The library is integrated into the project archive of the basic project. Codesys V3.5 SP7 patch 2 and the basic project can both be found in the support portal.

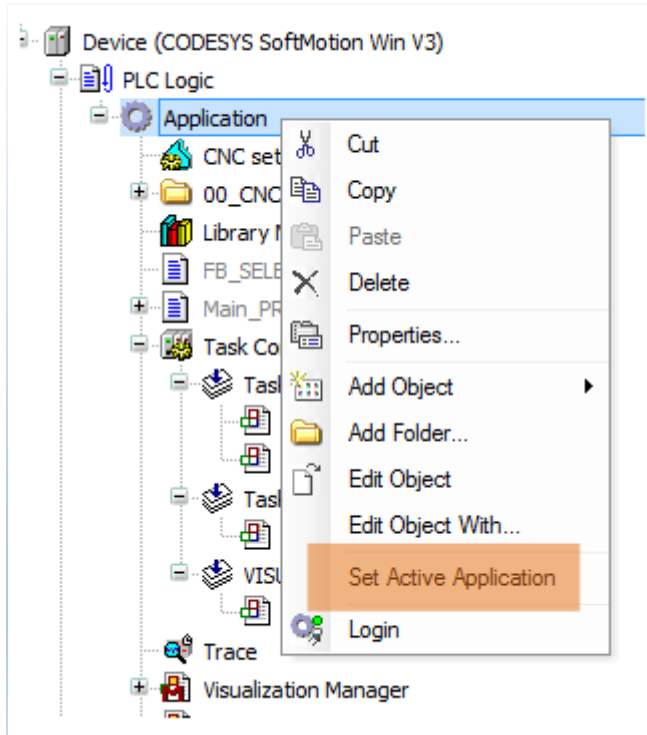
1. First of all, Codesys version V3.5 must be installed.
Then extract the basic project by double clicking the project archive (select all files).



- As soon as all of the data have been extracted, a window similar to the following 2 screenshot should appear at the left: (depends which Example Project are used)



How to set active an application, if you have more than one applications in one project:

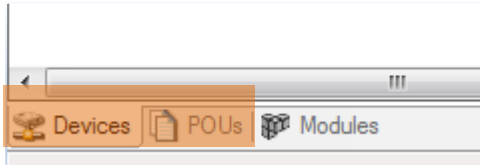


2 Basic project overview

The structure of the project is explained in this section in order to provide an overview.

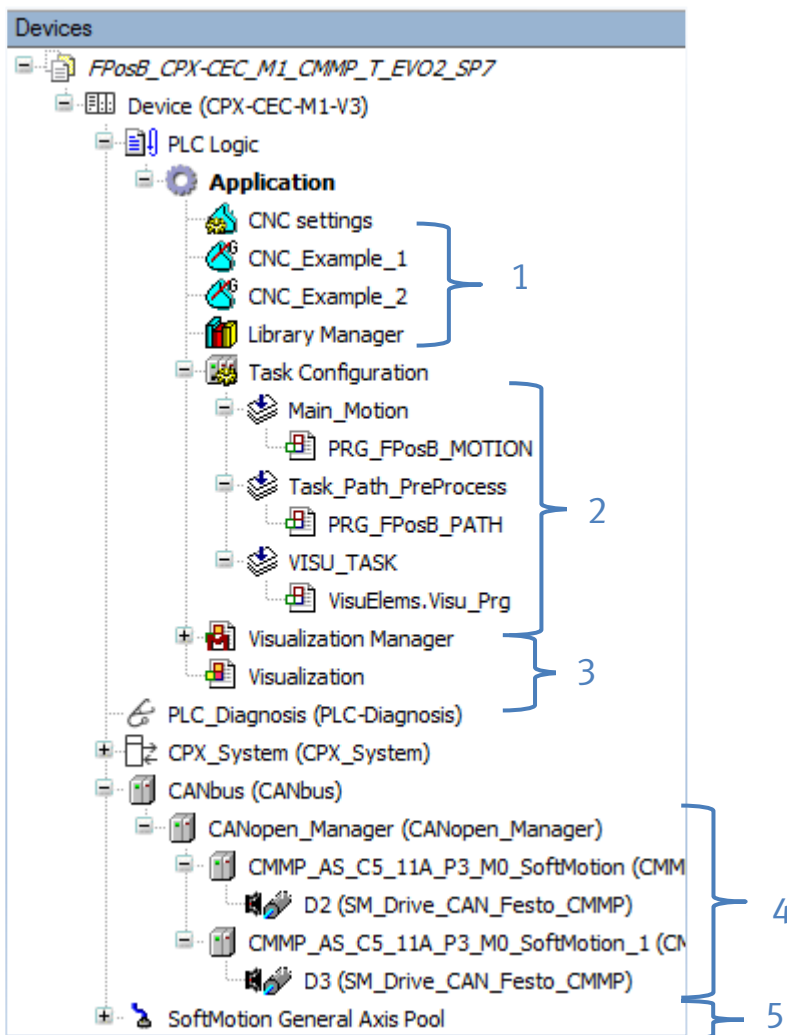
The user is responsible for any changes made to the project. The basic project provides a basis with which a system can be quickly commissioned.

The most important functions of the basic project are included in the “Devices” and “POUs” tabs.



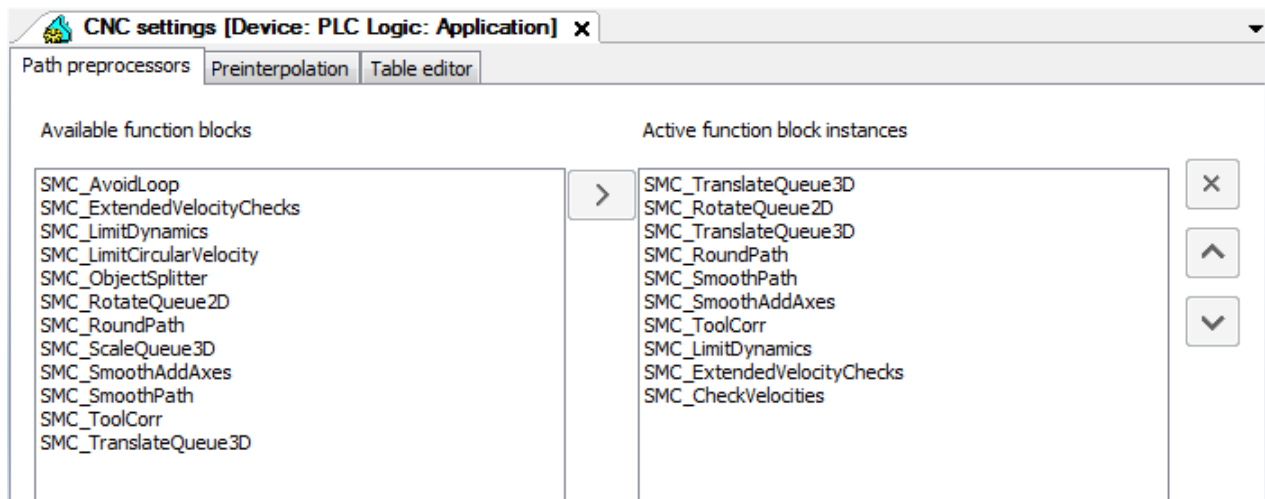
2.1 Devices

The “Devices” section in the sample project can be roughly subdivided into 5 groups:

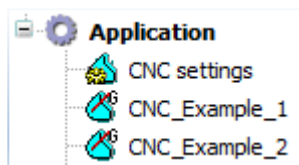


2.1.1 CNC settings und 00_CNCs

Additional active function blocks can be selected by the user in the CNC settings (path pre-processors tab). 5 blocks are already active in the sample project.

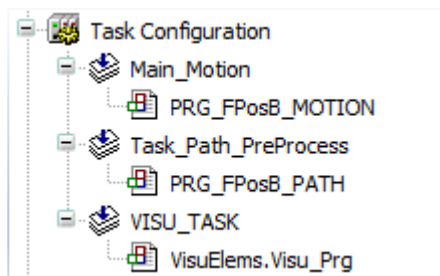


Sample CNCs can be found in the Application Tree. These can be used to test the library's CNC mode. The user can save his own CNCs here later.



2.1.2 Task configuration

A total of three tasks are set up in the task configuration: Task_Motion, Task_Path_PreProcess und VISU_TASK.



Task_Motion

PRG_FPOSB_Motion is invoked by this task, which is responsible for configuration and input/output process data.

This task has highest priority within the application and is invoked cyclically every 8 ms.

Configuration

Priority (0..31): 2

Type
Cyclic Interval (e.g. t#200ms): t#8ms

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

+ Add Call ✕ Remove Call ✎ Change Call ⬆ Move Up ⬇ Move Down ➡ Open POU

POU	Comment
PRG_FPosB_MOTION	

Note 1: The sync signal in the CANopen_Manager is also set to 8 ms (see section [4.1.4](#)).

Note 2: Task_Motion must also be assigned to the controller's bus cycle task.

PLC settings Edit Licenses...

PLC shell

Users and Groups

Bus cycle options
Bus cycle task: Task_Motion

Additional settings

Task_Path_PreProcess

The task invokes PRG_FPOSB_PATH, which is responsible for preparing the path. The priority is lower and the interval is greater than for Task_Motion.

Configuration

Priority (0..31): 5

Type
Cyclic Interval (e.g. t#200ms): t#16ms

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

+ Add Call ✕ Remove Call ✎ Change Call ⬆ Move Up ⬇ Move Down ➡ Open POU

POU	Comment
PRG_FPosB_PATH	

VISU_TASK

This task processes the actual visualisation. Since this is the direct interface to the user, neither high priority nor a short cycle time are required. There's no need to adjust these data because they're set automatically when a WebVisu is added.

Configuration

Priority (0..31): 31

Type

Cyclic

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

Watchdog

☐ Enable

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

Sensitivity:

+ Add Call

✕ Remove Call

✎ Change Call

↑ Move Up

↓ Move Down

→ Open POU

POU	Comment
VisuElems.Visu_Prg	

2.1.3 Visualisation

The visualisation can be evoked here directly at the Codesys level. The visualisation is prepared to the extent that it can be used to operate all essential functions.

This visualisation can also be invoked via a web browser.

The WebVisu is invoked via the following address:

`http:// <IP_address of web server>:8080/webvisu.htm`

Further details are included in application note 04 visualization and in OnlineHelp of Codesys (keyword: visualization).

Festo_Positioning_Basic_3

Operation Mode

No Mode

Configuration Mode

Homing Mode

Jogging Mode

Stepping Mode

PTP Mode

Internal CNC Mode

External CNC Mode

Enable System

Reset System

Ack Error

Select Coordinate System

MCS

UCS

WPCS

Override 50

Please change the operation mode on the left side

☐ Config Done
☐ Is Portal
☐ ReadyTo Enable
☐ Enabled
☐ Homed
☐ Resetting
☐ Busy
☐ Step Mode Active
☐ Step Done
☐ IsMoving
☐ Halt Active
☐ Interrupt Active
☐ Aborted
☐ Motion Complete
☐ Path Corr Active
☐ M_Active
☐ Error
☐ IPO Done

Drive present

Drive homed

	MCS	UCS	WPCS
X	0.000	0.000	0.000
Y	0.000	0.000	0.000
Z	0.000	0.000	0.000
V	0.000	0.000	0.000
W	0.000	0.000	0.000
CS Quality	1.000	1.000	
Act. Velocity		0.000	
Act. Override		50	

CNC Prg Name:
No_CNC

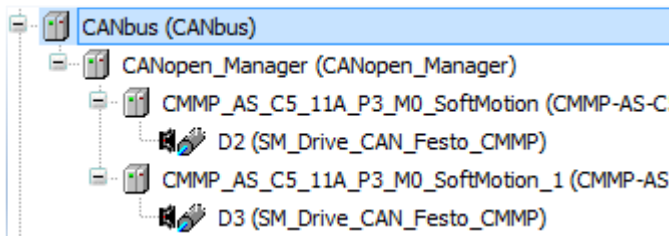
OP-Mode

Diagnosis

2.1.4 CAN bus

In the case of CAN bus, all settings such as baud rate and node ID are set by means of the software. These settings must coincide with those of the controller.

Bit rate / baud rate (CAN bus)



General

CANbus I/O Mapping

Status

Information

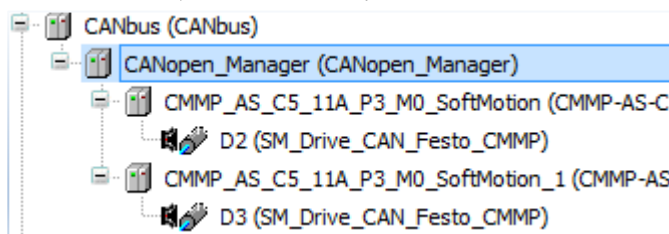
General

Network: 0

Baudrate (bit/s): 1000000

CAN

Activating the sync signal (CANopen_manager)



General

CANopen I/O Mapping

Status

Information

General

Node ID: 126

Check and fix configuration...

CANopen

☒ Autostart CANopenManager ☒ Polling of optional slaves

☒ Start Slaves NMT Error Behaviour: Restart Slave

☐ NMT Start All (if possible)

Guarding

Sync

☒ Enable Sync Producing

COB-ID (Hex): 16# 80

Cycle Period (µs): 8000

Window Length (µs): 0

☐ Enable Sync Consuming

TIME

CAN bus participants

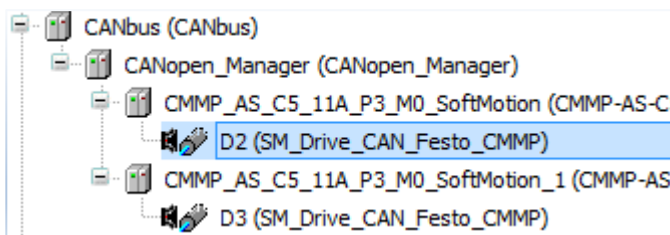
Since three controllers are used in our example, three CAN participants are added – each as a CMMS-ST soft motion axis.

A name is assigned to each axis so that the participants are unique.

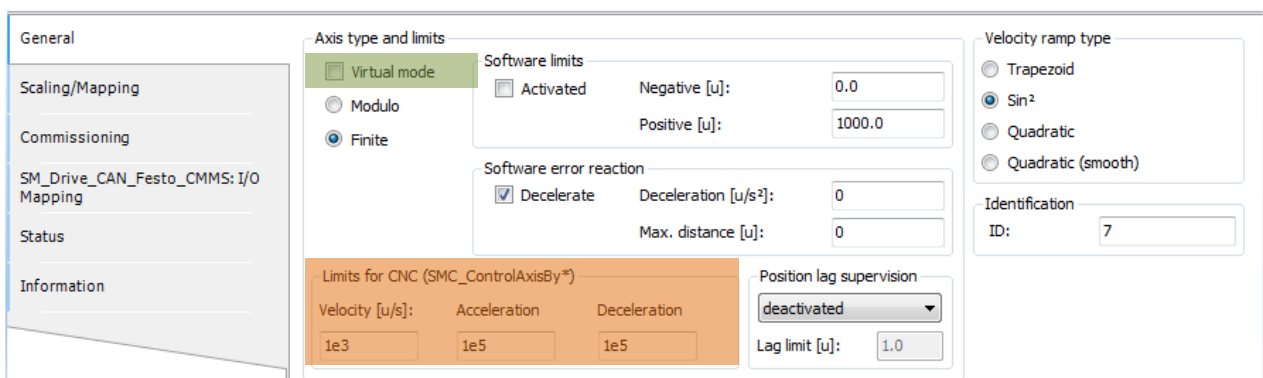
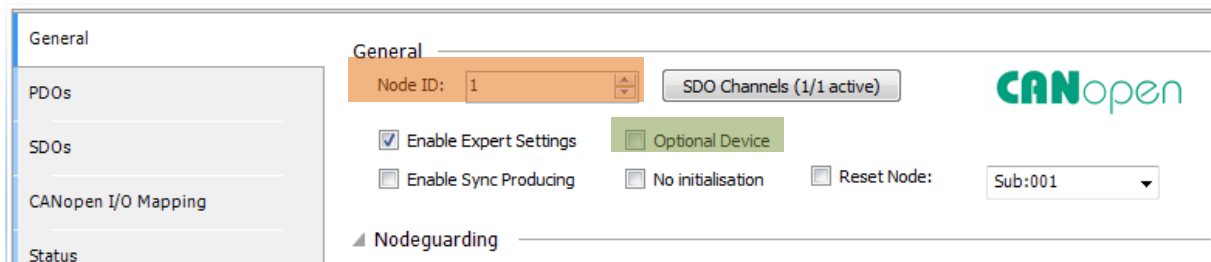
The following table applies in this respect:

Node ID	2D gantry / XY gantry	Linear gantry	In the sample project (T-Gantry)
1	Motor 1 (2D gantry) / X	X	D1
2 (1 for M1 T-Gantry Example)	Motor 2 (2D gantry) / Y	Motor 1 (linear gantry)	D2 (Motor1)
3 (2 for M2 T-Gantry Example)	Z	Motor 2 (linear gantry)	D3 (Motor2)
4	A / U	A / U	D4
5	B / V	B / V	D5
6	C / W	C / W	D6

Table 1: Labelling



The CAN settings for the added axes are suitable to the greatest extent and don't have to be changed as a rule. Only the node ID and the CNC limits have to be adjusted.



The dynamic limit values for the individual axes depend on the limit of the axis (technical data) and the application.

If scaling/mapping is accordingly adapted, these values are in mm/s and mm/s².

If any axes are physically unavailable, they can also be simulated. The corresponding controller is optionally set to this end and the virtual mode is activated (see fields highlighted in green).

Scaling/mapping

Due to the fact that the soft motion modules work with soft motion units, these must be defined.

It's advisable to adjust these settings such that one soft motion unit is equal to one millimetre at the kinematic system.

For the T-Gantry in the Example Project without gearbox, this results in the following setting for motor 1 and motor 2 controllers, because the feed constant is equal to 100 mm per revolution:

These settings can be also changed within the library itself. Please refer to chapter 1.1.2 GVL_Config in the document 03 Configuration H-T-XY-Gantry and coordinate systems.

2.1.5 EtherCAT

CMMT servo drives do not support CANbus. These servo drives are able to communicate via EtherCAT, Profinet or Ethernet/IP. In this example we focus on EtherCAT because the drives are connected with a PLC (CPX-E-CEC-M1) via EtherCAT.

Cycle time & sync offset

For most softmotion applications a cycle time of 8ms is sufficient. The sync offset should be set to 50% to avoid any issues with different sign and interpretations of it.

EtherCAT master setting

EtherCAT task setting

EtherCAT devices

Devices need to be added accordingly to their physical connection. For further details please check EtherCAT specifications.

The assignment of devices to the specific axes is done later in the configuration.

The dynamic limit values for the individual axes depend on the limit of the axis (technical data) and the application.

Based on the settings in Festo Automation Suite these values are usually in mm/s and mm/s²

Scaling/Mapping

You do not have to change any settings here as the needed scaling is done in Festo Automation Suite for each servo drive.

Keep these values

FPosB provides also a possibility to adapt the feed constant within the library (Please refer to chapter 1.1.2 GVL_Config in the document 03 Configuration H-T-XY-Gantry and coordinate systems). Using CMMT drives it is not recommended to change it within Codesys because you also have to take care about the settings in Festo Automation Suite. So every needed change should be also done in Festo Automation Suite to avoid any problems.

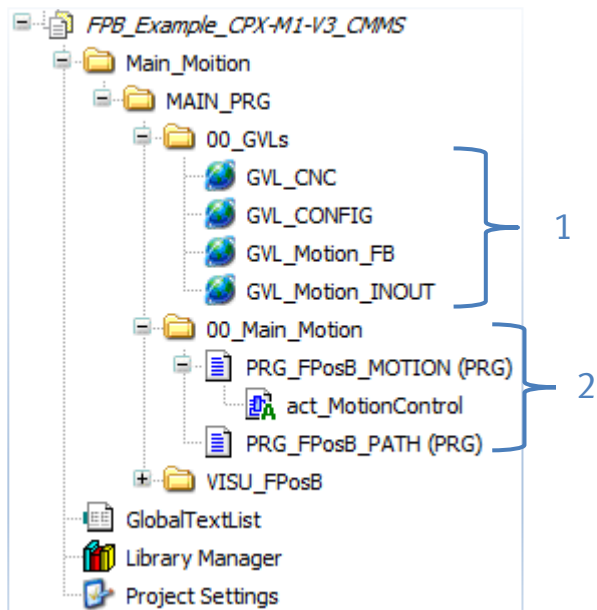
2.1.6 Soft motion general axis pool

If no kinematic system is available or if components will only be simulated, e.g. for test purposes, virtual axes can be added.



2.2 POUs

Roughly speaking there are two important groups for the user where the POUs are concerned: 00_GVLs and 00_Main_Motion.



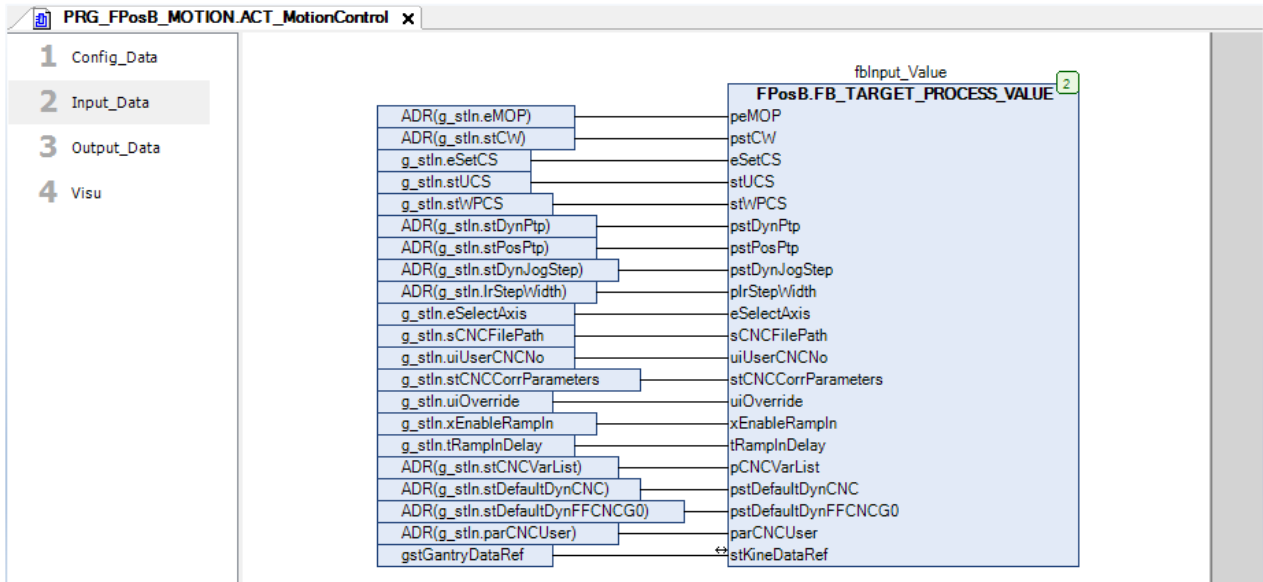
2.2.1 00_GVLs

Basic settings for CNC (GVL_CNC) and the system (GVL_Motion_INOUT) are entered, the system is configured (GVL_CONFIG) and function blocks are instantiated (GVL_Motion_FB) in the four global variables lists. Further information can be found in section in application note 03 configuration H-/T-/XY-gantry.

2.2.2 00_Main_Motion

This part is responsible for the system's actual motion generation. The PRG_FPOSB_MOTION program invokes the act_MotionControl action, in which the configuration, input and output data and the visualisation are invoked.

The previously defined variables are assigned to the individual modules. Using the input data as an example, this results in the following:



The PRG_FPOSB_PATH program simply invokes path preparation with the corresponding DataRef structure (see GVL_CONFIG).

```

1  PROGRAM PRG_FPosB_PATH
2  VAR
3      fbFHP : FPosB.FB_PATH_PREPROCESS;
4  END_VAR
5
1  fbFHP(stKineDataRef:= gstGantryDataRef);

```