Sample documentation

TERMS OF USE

The use of this sample programs is allowed only under acceptance of following conditions by the user:

The present software which is for guidance only aims at providing customers with sampling information regarding their S7-programs in order to save time. As a result, INSEVIS shall not be held liable for any direct, indirect or consequential damages respect to any claims arising from the content of such software and/or the use made by customers of this sampling information contained herin in connection with their own programs.

Index of content

1 Motivation.....2
2 General principles of the software-design.....2
3 Drive functions (MC-blocks and -types).....3
3.1 Settings for EPOS2.....3
3.2 MC_ReadStatus_E2 (FB).....4
3.3 MC_ReadAxisError_E2 (FB).....5
3.4 MC_ReadActualPosition_E2 (FB).....6
3.5 MC_ReadActualVelocity_E2 (FB).....6
3.6 MC_Reset_E2 (FB).....7
3.7 MC_Power_E2 (FB).....7
3.8 Function blocks in the profile Position Mode.....8
3.8.1 MC_Stop_E2 (FB).....9
3.8.2 MC_MoveAbsolute_E2 (FB).....10
3.8.3 MC_MoveRelative_E2 (FB).....11
3.9 Function blocks in the profile Velocity Mode.....11
3.9.1 MC_MoveVelocity_E2 (FB).....12
3.10 MC_Jog_E2 (FB).....13
3.11 Function blocks in the Homing Mode.....14
3.11.1 MC_Home_E2 (FB).....15
3.12 Special-FB's.....16
3.12.1 E2_Input (FB).....16
3.13 InDataE2Type (UDT).....17
3.14 OutDataE2Type (UDT).....17
3.15 AxisRefE2Type.....17
4 Sample of a MC-block-instance.....18
5 CANopen-configuration with the EPOS-Studio.....19
6 Slave-configuration with the ConfigStage.....21
6.1 Mapping T-PDO1.....21
6.2 Mapping T-PDO2.....22
6.3 Mapping R-PDO1.....22
6.4 Mapping R-PDO2.....22
6.5 Mapping R-PDO3.....22
6.6 Mapping R-PDO4.....22
6.7 Additional SDO-transfers after PDO-mapping.....22
7 S7-Sample-program.....22

1 Motivation

Manufacturer-specific S7-blocks will be offered from different vendors for an easy implementation of their own drive technology into the world of Simatic- and Simatic-compatible PLCs since years. This is often done by a very effective S7-block, adapted to the specials of the vendors drive, containing a monolithic and mostly customized interface and specialized in a certain bus system (normally Profibus DP, also Interbus S and CANopen based on fieldbus-master modules of other manufacturers).

The PLCopen (http://www.plcopen.org) as an international organisation is dedicated to reduce the effords for engineering by using general software interfaces. In the area of drive technology standards were defined, a certification of drives with implemented interfaces is possible. By using bus systems like CANopen with drive interfaces (DS402 drive profile) the effords for the adaption onto a certain bus protocol is unimportant.

In the following the operation on a servo drive Maxon EPOS2 24/5 (http://www.maxonmotor.de) is described. The software was created for INSEVIS-PLC and is based on the PLCopen-standard

On following devices the software test was done:

EPOS2

Testing device	:	EPOS2 24/5
Software-version	:	0x2122
Hardware-version	:	0x6220
EPOS-Studio	:	1.44 revision 1

INSEVIS

Testing device	:	CC300V
operating system	:	2.0.23
S7-Library	:	Insevis_S7-library_from_2_0_22

Actually there will not be supported all possible modes, like e.g. the „Master Encoder Mode“, because the hardware platforms are to assortetly, to work with te same encoder-type. On demand a MC_Gear_E2 could be implemented generally later on.

Company inmotec Automation GmbH (support@inmotec.de) creates and expands drive specific software for INSEVIS-S7-controllers.

2 General principles of the software-design

1. All drive functions (so called Motion-Control-Blocks MC_) will be implemented as single function blocks, e.g. the function block „MC_Power_E2“, a S7-FB, is used to enable the servo drive. Because the motor does need not only to be enabled but also has to do motion functions, more function blocks are neccessary. Of course multiple axes were supported too. To prevent a various number of instances of an function block with separate instance blocks, an instanciation of function blocks in the STAT area of the variables definiton of the „container“-funktion block is recommended.
2. The MC-Blocks use no global resources as M-merker, T-times or Z-counter, but their instanciable IEC-variantes.
3. All drive functions of the INSEVIS-PLC communicate via asynchronous CANopen-PDO's reg. DS301, so that the efford for communication (bus load) is reduced. At the drive profile DS402 will be used operating modes only, what do not require equidistant transfers of demand values. The so called „interpolated mode“ will not be used.
4. The function blocks will be created in origin with SCL (Structured Control Language), an engineering-option to Step7 of Siemens. The use of these function blocks does not need a preinstalled SCL-package on the programming PC of the user.
5. To absorb diversities of the drives and name conflicts of already existing blocks from custom libraries (e.g. at the technology- PLC of Siemens), the MC-Blocks get a postfix like „_C3“ in reference to the regarding drive. There needs to be notified, that the instance name (in the sample „Axis00“) is not touched while swapping drives.
6. Because blocks do not reference each other, block-addresses (absolut numbers) can be adapted to the demands of the users program.

3 Drive functions (MC-blocks and -types)

An operation with encoders is recommended generally, if positioning applications are to do. If the measurement of rotation speed by Hall sensors, no positioning applications can be made. Only an operation by rotation speed is worthwile. It is recommended to drive the motor at least with 1000 r/min.

MC-Block/Symbol	Address	Function
MC_ReadStatus_E2	FB40	Visualization of drive states (currentless, stopping, remaining idle, profil based motion functions active, endlessmove active, synchronized motion functions active, reference move active)
MC_ReadAxisError_E2	FB41	Visualization of the error code of the drive
MC_ReadActualPosition_E2	FB42	Visualization of the actual postion of the motor
MC_ReadActualVelocity_E2	FB43	Visualization of actual velocity of the motor
MC_Reset_E2	FB44	Reset error in the drive
MC_Power_E2	FB45	Enable power stage to the motor or stop/disable fast as possible
MC_Stop_E2	FB46	Stop the motor
MC_MoveAbsolute_E2	FB47	Move to an absolute position
MC_MoveRelative_E2	FB48	Move a relative distance
MC_MoveVelocity_E2	FB50	Endless move (rotation speed assign)
MC_Home_E2	FB52	Execute homing move
MC_Jog_E2	FB53	Jog+/- move, stops on the software-end-delimiters
E2_Input	FB54	Read out inputs
InDataE2Type	UDT100	Data type for input data CANopen, instanciate once per axis
OutDataE2Type	UDT101	Data type for output data CANopen, instanciate once per axis
SWPosE2Type	UDT102	Data type state word CANopen, INTERNAL USE ONLY
CWPosE2Type	UDT103	Data type control word CANopen, , INTERNAL USE ONLY
AxisRefE2Type	UDT104	Data type axis reference, instanciate once per axis

3.1 Settings for EPOS2

Because EPOS2 uses fixed formates for positioning (post-quad-increments), velocities (r/min) and ramps (r/min/s), an implementation is worthwile, what allows an assign of better to understand engineer values.

The user has to write following parameters into the axis reference, so that it is possible to program directly in user units, user units/s and user units/s$^{2:}$

```
      L     1.000000e+001                       // e.g. 10 mm feed/turns
      T     "AxisRef".Data.Axis00.fMechanicPitch
      L     4.000000e+003                       // increments Encoder after
                                                // quadruplication,e.g. 4000 at
                                                // a 1000-impuls-Encoder
      T     "AxisRef".Data.Axis00.fEncPQPerMotRev
```

Also the CANopen-Node-ID is to assign, e.g. foracyclic requests (error codes).

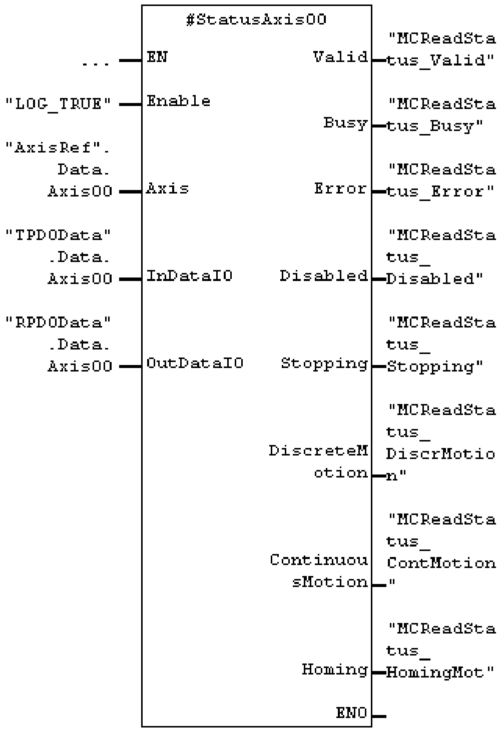
```
      L     4
      T     "AxisRef".Data.Axis00.iNode
```

3.2 MC_ReadStatus_E2 (FB)

The MC_ReadStatus_E2 will be used for visualization (State generation) of different drive states. With these information the PLC-programm can watch all activities of the drive.

ⓘ	This block MUST be implemented into the PLC-program, because beside the state generation the complete axis reference will be processed. That´s why this block size is larger than at the other MC-blocks. Explanations are made in the chapter axis reference. It is only one instance of this FB allowed and reasonable.

Name	Variables area	Type	Function
Enable	IN	Bool	Activate state generation For the processing of the axis reference the enable-input is not important, but anyway the FB MUST be called.
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
InDataIO	IN_OUT	InDataE2Type (UDT)	Reference to IO-data (input-data CANopen)
OutDataIO	IN_OUT	OutDataE2Type (UDT)	Reference to IO-data (output data CANopen)
Valid	OUT	Bool	FB-data are valid (as long Enable = True)
Busy	OUT	Bool	FB-function runs (always False, because created from PDO-data)
Error	OUT	Bool	Axis with error
Disabled	OUT	Bool	Axis is disabled
Stopping	OUT	Bool	Axis is stopping
DiscreteMotion	OUT	Bool	Axis is positioning
ContinuousMotion	OUT	Bool	Axis is positioning endless
Homing	OUT	Bool	Axis executes „homing-move“



In the STAT-area of a Container-FB instanciated MC_ReadStatus_E2 with the instance name StatusAxis00.

3.3 MC_ReadAxisError_E2 (FB)

The MC_ReadAxisError_E2 will be used for visualization of the error code of the axis.
The meaning of the error code is mentioned in the drives help manual.

Name	Variables area	Type	Function
Enable	IN	Bool	Read error code
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Valid	OUT	Bool	FB-data are valid (as long Enable = True)
Busy	OUT	Bool	FB-function (SDO-Transfer) runs (actally)
Error	OUT	Bool	Axis with error
ErrorID	OUT	DWORD	Error code of axis (here 32-bit)

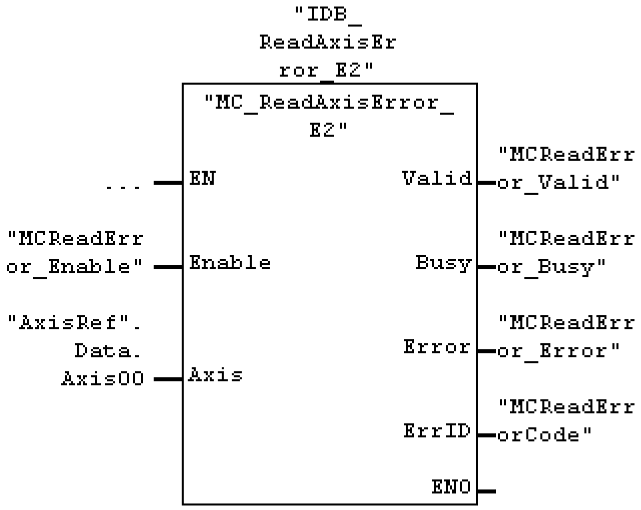
ⓘ

The EPOS2-CANopen-object 0x1003 will be red on subindex 1, what contains the actual error code. The object will be red by CANopen-SDO, when the error bit in the stat word will be set on TRUE (edge) or the enable-input will be set on TRUE (edge).

Occures after a TIME-OUT-time of 150ms no answer from the device or an error will be reported at the SDO-transfer, an error code DW#16#FFFFFFFF (SDO-transfer-error) will be displayed.

The FB in the actual version can not be instanciated in the STAT-area of an instance-DB's of an higher ranked FB's! It must be created a seperate instance for the MC_ReadAxisError_E2.

It is only one instance of this FB allowed and reasonable.

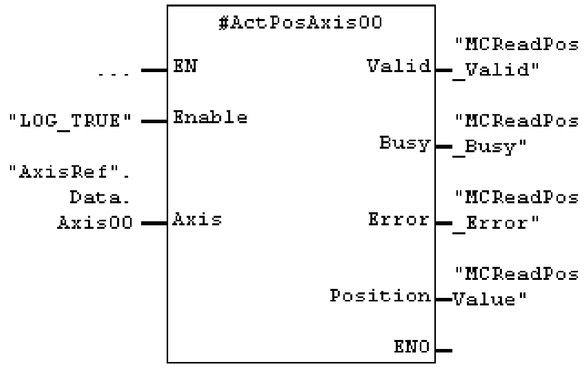


Separate instanciated MC_ReadAxisError_E2 with the instance name ErrorAxis00.

3.4 MC_ReadActualPosition_E2 (FB)

The MC_ReadActualPosition_E2 provides the absolute position of the axis.

Name	Variables area	Type	Function
Enable	IN	Bool	Read actual position
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Valid	OUT	Bool	FB-data are valid (as long Enable = True)
Busy	OUT	Bool	FB-function runs (always False, because PDO-date)
Error	OUT	Bool	FB-error (always False, because PDO-date)
Position	OUT	REAL	Axis position in user units, e.g. „mm“

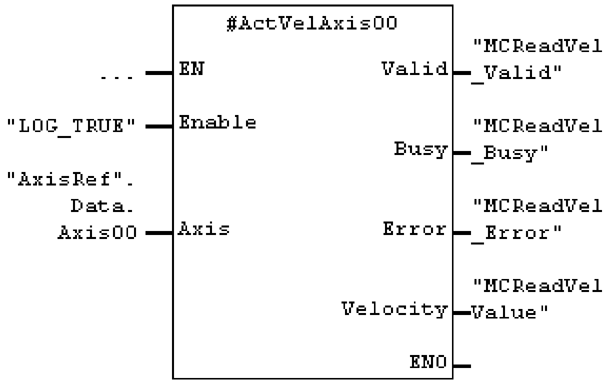


In the STAT-area of a Container-FB instanciated MC_ReadActualPosition_E2 with the instance name ActPosAxis00.

3.5 MC_ReadActualVelocity_E2 (FB)

The MC_ReadActualVelocity_E2 provides the actual velocity of the axis.

Name	Variables area	Type	Function
Enable	IN	Bool	Read actual velocity
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Valid	OUT	Bool	FB-data are valid (as long Enable = True)
Busy	OUT	Bool	FB-function runs (always False, because PDO-date)
Error	OUT	Bool	FB-error (always False, because PDO-date)
Velocity	OUT	REAL	Axis velocity in user units. e.g. „mm/s“



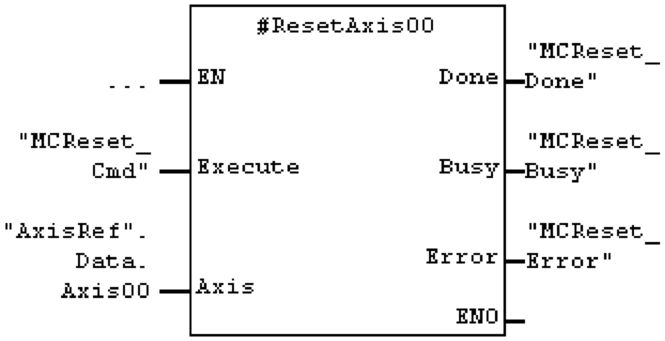
In the STAT-area of a Container-FB instanciated MC_ReadActualVelocity_E2 with the instance name ActVelAxis00.

3.6 MC_Reset_E2 (FB)

With the block MC_Reset_E2 the servo axis error will be reset.

ⓘ	It is only one instance of this FB allowed and reasonable.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge receipes axis
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	FB-function ready and axis without error
Busy	OUT	Bool	FB-function is executed/runs
Error	OUT	Bool	Axis with error



In the STAT-area instanciated MC_Reset_E3 with the instance name ResetAxis00.

3.7 MC_Power_E2 (FB)

With the block MC_Power_E2 the axis will be enabled or disabled

ⓘ	It is only one instance of this FB allowed and reasonable.

Name	Variables area	Type	Function
Enable	IN	Bool	0-1-edge enables power stage axis, 1-0-edge executes an immediate stop with following switching to currentless
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Status	OUT	Bool	1 power stage enabled 0 disabled
Busy	OUT	Bool	Function enable power stage even active
Error	OUT	Bool	Axis with error

3.8 Function blocks in the profile Position Mode

The blocks MC_Stop_E2, MC_MoveAbsolute_E2, MC_MoveRelative_E2 will be processed in the profile „Position Mode“. Beside the assigns on the FB-parameters following CANopen-paramters are to assign via CANopen-SDO or easier via the EPOS-Studio.

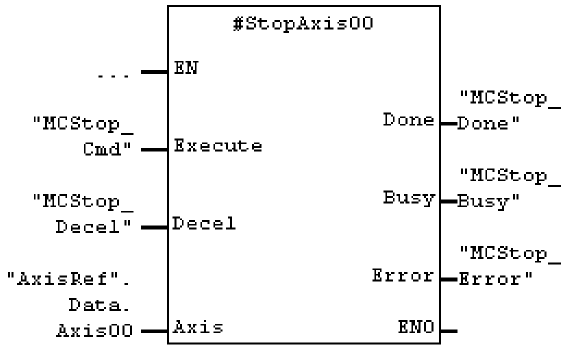
EPOS-Parameter	**Object-index resp. subindex**	**Setting EPOS-Studio resp. SDO-transfer necessary**	**MC_Stop**	**MC_MoveAbsolute MC_MoveRelative**
Position Window (positioning window for DONE-message)	0x6067 / 0x00 [increments]	Yes		
Position Window Time (time in the positioning window for DONE-message)	0x6067 / 0x00 [ms]	Yes		
Software Position Limit	0x607D / 0x01 minimal 0x607D / 0x02 maximal [increments] The software-position-control can be dectivated with -2147483648 resp. +2147483647	Yes		
maximal Profile Velocity	0x607F / 0x00 [r/min]	Yes		
QuickStop Deceleration	0x6085 / 0x00 [r/min/s]		Yes (Decel) [Units/s^2]	
Max Acceleration/ Deceleration	0x60C5 / 0x00 [r/min/s]	Yes		
Target Position (or distance from theactual nominal position)	0x607A / 0x00 [increments]			Yes (position resp. distance) [Units]
Profile Velocity (nominal velocity)	0x6081 / 0x00 [r/min]			Yes (Velocity) [Units/s]
Profile Acceleration (nominal acceleration)	0x6083 / 0x00 [r/min/s]			Yes (Accel) [Units/s^2]
Profile Deceleration (nominal deceleration)	0x6084 / 0x00 [r/min/s]			Yes (Decel) [Units/s^2]
Motion Profile Type	0x6086 / 0x00	Yes 0 = linear ramps 1 = sin^2-ramps		

3.8.1 MC_Stop_E2 (FB)

With the block MC_Stop_E2 the axis will be stopped. Stopping is only possible with a power stage enabled axis.

ⓘ	At an 0-1-edge the axis motion is stopped. Axis moves (new requests) will be blocked at activated Stop-Execute (=1) generally. The QuickStop-Function of the axis will be used! It is only one instance of this FB allowed and reasonable.

Name	Variables area	Type	Function
Execute	IN	Bool	1 Stops axis 0 Moves enabled
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Decel	IN	Dint	Stop ramp [Units/s^2]
Done	OUT	Bool	Axis stopped
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error

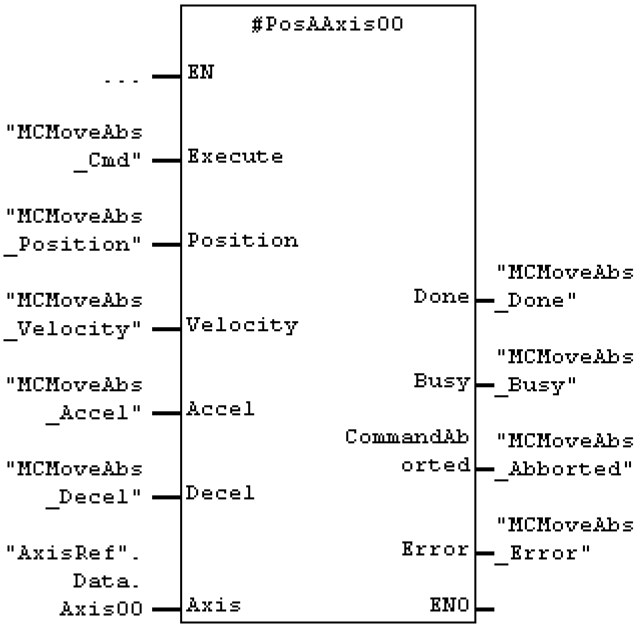


In the STAT-area instanciated MC_Stop_E2 with the instance name StopAxis00.

3.8.2 MC_MoveAbsolute_E2 (FB)

The block MC_MoveAbsolute_E2 will be used for absolute positioning. Reference point of the absolute position is defined by homing reference travel or by determined absolute point of origin (mathematical zero-point).

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge starts the move
Position	IN	Real	Absolute position in user units, e.g. „mm“
Velocity	IN	Real	Positioning velocity in user units, e.g. „mm/s“
Accel	IN	Dint	Acceleration in user units, e.g. „mm/s^2“
Decel	IN	Dint	Deceleration in user units, e.g. „mm/s^2“
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Axis has reached target position
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error



In the STAT-area instanciated MC_MoveAbsolute_E2 with the instance name PosAAxis00.

3.8.3 MC_MoveRelative_E2 (FB)

The block MC_MoveRelative_E2 will be used for relative positioning (for a distance). Reference point for the distance is the actual target position. This kind of positioning is referred as chain positioning.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge startes the move
Distance	IN	Real	Distance in user units e.g. „mm“
Velocity	IN	Real	Positioning velocity in user units e.g. „mm/s“
Accel	IN	Dint	Acceleration in user units, e.g. „mm/s^2“
Decel	IN	Dint	Deceleration in user units, e.g. „mm/s^2“
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Axis has reached target postion
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error

3.9 Function blocks in the profile Velocity Mode

The blocks MC_Stop_E2, MC_MoveVelocity_E2 and MC_Jog_E2 will be processed in the profile „Velocity Mode“. Beside the assigns on the FB-parameters following CANopen-parameters are to assign via CANopen-SDO or easier via the EPOS-Studio.

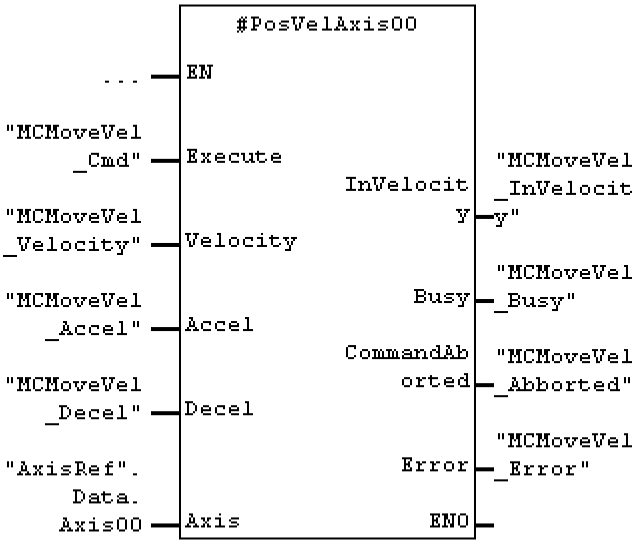
EPOS-Parameter	Object-index resp. subindex	Setting EPOS-Studio resp. SDO-transfer necessary	MC_Stop	MC_MoveVelocity MC_Jog
Position Window (positioning window for DONE-message)	0x606D / 0x00 [r/min]	Yes		
Position Window Time (time in the positioning window for DONE-message)	0x606E / 0x00 [ms]	Yes		
Maximal Profile Velocity	0x607F / 0x00 [r/min]	Yes		
QuickStop Deceleration	0x6085 / 0x00 [r/min/s]		Yes (Decel) [Units/s^2]	
Max Acceleration / Deceleration	0x60C5 / 0x00 [r/min/s]	Yes		
Target Velocity ((nominal velocity)	0x60FF / 0x00 [r/min]			Yes (Velocity) [Units/s]
Profile Acceleration (nominal acceleration)	0x6083 / 0x00 [r/min/s]			Yes (Accel) [Units/s^2]
Profile Deceleration (nominal deceleration)	0x6084 / 0x00 [r/min/s]			Yes (Decel) [Units/s^2]
Motion Profile Type	0x6086 / 0x00	Yes 0 = linear ramps 1 = sin^2-ramps		

3.9.1 MC_MoveVelocity_E2 (FB)

The block MC_MoveVelocity_E2 will be used for endless moves.

ⓘ	The move MUST be stopped with the block MC_Stop_E2. Will be assigned the velocity 0.0 units/s on the block MC_MoveVelocity, the drives moves with the profile- (nominal-) velocity 0.0 units/s. It means, the assigned profile-(nominate) values are still active. If the velocity shell be changed „flying“, the velocity can be changed by an additional instance of MC_MoveVelocity or the Execute of the „one“ instance can be re-triggered (another 0-1-edge).

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge startes the move
Velocity	IN	Real	Positioning velocity in user units e.g. „mm/s“
Accel	IN	Dint	Acceleration in user units, e.g. „mm/s^2“
Decel	IN	Dint	Deceleration in user units, e.g. „mm/s^2“
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
InVelocity	OUT	Bool	Axis has reached profile (target-) postion (distance driven)
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error



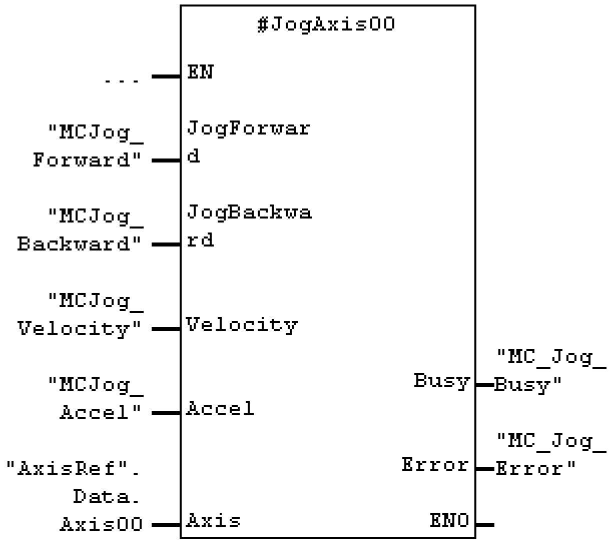
In the STAT-area instanciated MC_MoveVelocity_E2 with the instance name PosVelAxis00.

3.10 MC_Jog_E2 (FB)

The block MC_Jog_E2 will be used to move the axis „manually“ (also named „inching“). Both directions are possible. The FB wors like the MC_MoveVelocity, but ist can be moved directional.

ⓘ	If the Jog-process will be finished (JogForward and JogBackward both FALSE), than internal will be triggered QuickStop, where in difference wobei to the „normal“ stopping via MC_Stop_E2 the Accel-ramp from FB-input is used, so can be stopped more „soft“ Note, that the Accel-value is valid for acceleration as well as for deceleration.

Name	Variables area	Type	Function
JogForward	IN	Bool	0-1-edge startes jogging clockwise, 1-0-edge stops the axis
JogBackward	IN	Bool	0-1-edge startes jogging counter-clockwise, 1-0-edge stops the axis
Velocity	IN	Real	Manual velocity in user units, e.g. „mm/s“
Accel	IN	Dint	Acceleration / deceleration in user units, e.g. „mm/s2“
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Busy	OUT	Bool	FB-Function is executed/runs
Error	OUT	Bool	Axis with error



In the STAT-area instanciated MC_Jog_E2 with the instance name JogAxis00.

3.11 Function blocks in the Homing Mode

The blocks MC_Home_E2 will be executed in the homing mode. Beside the assign on the FB-parameters following CANopen-paramters are to assign via CANopen-SDO or easier via the EPOS-Studio.

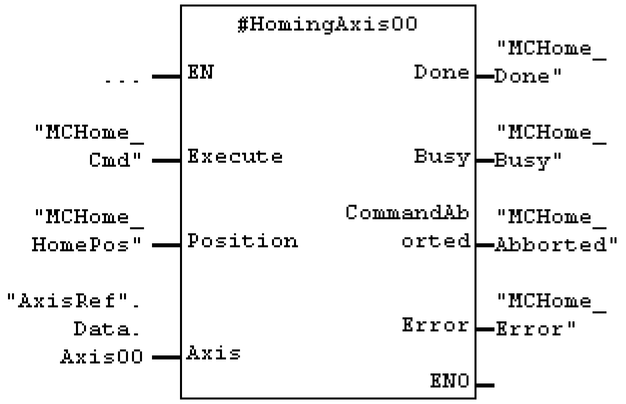
EPOS-Parameter	Object-index resp. subindex	Settings EPOS-Studio resp. SDO-transfer neccessary	MC_Home
Homing Method (Method of referenciation by position switches, mechanical arresters, set position,...)	0x6098 / 0x00	Yes	
Homing Speeds (search velocity at the referencation)	0x6099 / 0x00 [U/min]	Yes	
Homing Acceleration (ramps at the referencation)	0x609A / 0x00 [U/min/s]	Yes	
Home Offset (after the referenc-ation, this course will be driven too, than the value of the parameter Home position will be set as new actual position.	0x607C / 0x00 [Inkremente]	Yes	
Current Threshold for homing modes -1 to -4 (If it should be driven against an arrester while referencing, current limit)	0x2080 [mA]	Yes	
Home position (Set position after sucessfully Homing and driving the course in the Home Offset)	0x2081 / 0x00 [increments]		Yes (Set position after finished homing-procedure) [Units]
Motion Profile Type	0x6086 / 0x00	Yes 0 = linear ramps 1 = sin^2-ramps	

3.11.1 MC_Home_E2 (FB)

Ther block MC_Home_C3 will be used to define the mathematical reference- (zero-) point of the axis. The reference kinds (Modi) are to be find in the EPOS2-manual. The modi „35“ and „-3“ were tested in the labaratories.

ⓘ	It is only one instance of this FB allowed and useful.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge startes the reference move
Position	IN	REAL	Set position (new target position) in user units, eg.g. „mm“, the value of the parameter 0x2081/0x00 will be described
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Reference move (incl. Driving the distance in the parameter 0x607C/0x00 Home Offset) finished, Set position was set, the mathematical zero point and if so, the Software-End-Limits are valid
Busy	OUT	Bool	Function is executed/runs
CommandAbborted	OUT	Bool	Command was cancelled by new positioning, jogging disabled motor, stopping, etc.
Error	OUT	Bool	Axis with error or referencation (Homing) finished with error



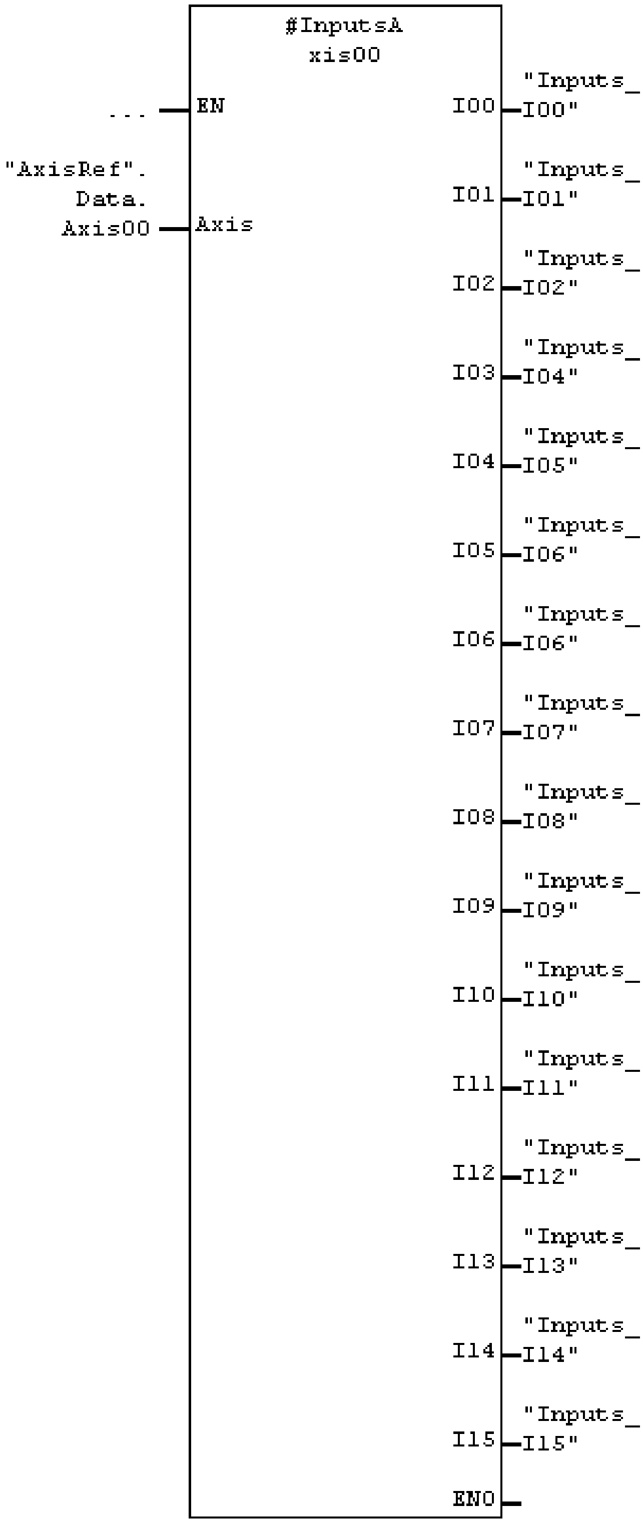
In the STAT-area instanciated MC_Home_E2 with the instance name HomingAxis00.

3.12 Special-FB's

3.12.1 E2_Input (FB)

The block E2_Input is a special version for the EPOS2-axis (Note the different versions of the EPOS2-family.).

Name	Variables area	Type	Function
Axis	IN_OUT	AxisRefE2Type (UDT)	Axis refererence (axis pointer)
I00 bis I15	OUT	Bool	Inpots 0 to 15 as state



In the STAT-area instanciated E2_Input with the instance name InAxis00.

3.13 InDataE2Type (UDT)

This data type is to instanciate while using in a data block with a name, e.g.. Axis00. Exactly 1 instance per axis will be needed. The instance data correlate to the T-PDO-data of the EPOS2-axis.

ⓘ	The best solution is to set up all instances (of the different axis´) of InDataE2Type in a separate DB (e.g. „TPDODATA“).

```
wStatusWord      :   WORD;       // TPDO1, async, 0x6041 + 0x00, Status word
wDigInWord       :   WORD;       // TPDO1, async, 0x2071 + 0x01, Input state
byActModeOfOp    :   BYTE;       // TPDO1, async, 0x6061 + 0x00, Actual mode of operation
diActPosition    :   DINT;       // TPDO2, async, 0x6064 + 0x00, Actual position [pq Enc.]
diActVelocity    :   DINT;       // TPDO2, async, 0x606C + 0x00, Actual velocity [rev/min]
```

3.14 OutDataE2Type (UDT)

This data type is to instanciate while using in a data block with a name, e.g.. Axis00. Exactly 1 instance per axis will be needed. The instance data correlate to the R-PDO-data of the EPOS2-axis.

ⓘ	The best solution is to set up all instances (of the different axis´) of OutDataE2Type in a separate DB (e.g. „RPDODATA“).

```
wControlWord         :   WORD;       // RPDO1, async, 0x6040 + 0x00, Control word
diTarget             :   DINT;       // RPDO1, async, 0x607a + 0x00, Target [pq Enc.]
byModeOfOp           :   BYTE;       // RPDO1, async, 0x6060 + 0x00, Mode of operation
diProfVelocity       :   DINT;       // RPDO2, async, 0x6081 + 0x00, Profile velocity [rev/min]
diHomePosition       :   DINT;       // RPDO2, async, 0x2081 + 0x00, Home position [pq Enc.]
diProfAccel          :   DINT;       // RPDO3, async, 0x6083 + 0x00, Prof. acceleration [rev/min/s]
diProfDecel          :   DINT;       // RPDO3, async, 0x6084 + 0x00, Prof. deceleration [rev/min/s]
diQuickStopDecel     :   DINT;       // RPDO4, async, 0x6085 + 0x00, Prof. deceleration [rev/min/s]
diSpeed              :   DINT;       // RPDO4, async, 0x60FF + 0x00, Prof. deceleration [rev/min]
```

3.15 AxisRefE2Type

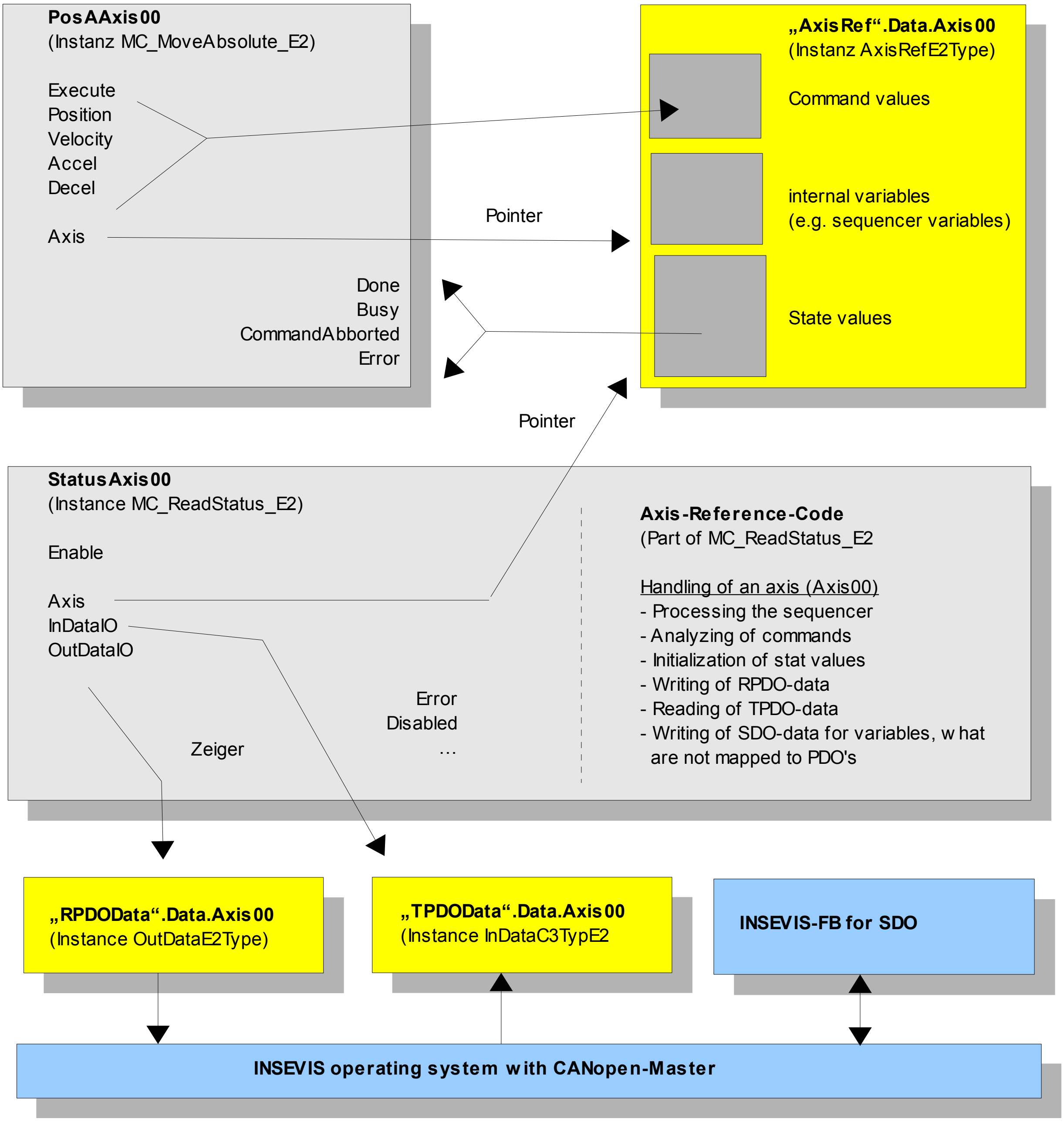
This data type will be used internally as axis working data of the axis. Because the instanciated variables were handled over as IN_OUT, the efford for copying is low.

The block MC_ReadStatus_E2 uses the variables, defined by the axis reference for processing the sequencers.

ⓘ	The best solution is to set up all instances (of the different axis´) of AxisRefE2Type in a separate DB (e.g. „AXISREF“)

4 Sample of a MC-block-instance

Following figure shows the use and the data flow of a MC-block for one axis with the name Axis00.

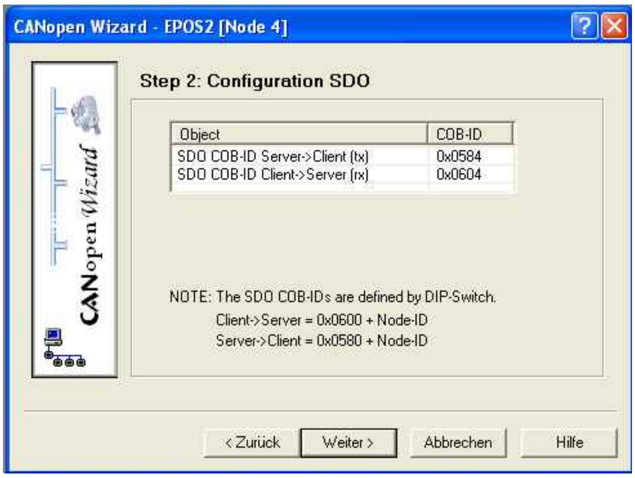


5 CANopen-configuration with the EPOS-Studio

Gererally there can not payed attention for the common configuration of a EPOS2-servo drive. Important for the CANopen-part is only, that actual configuration should be tested via the CANopen-Wizard (after configuration with the ConfigStage and start of the INSEVIS-PLC), because the setting of the CANopen-Slaves are carried out by the INSEVIS-PLC thereselfe.

The CANopen-Node-ID is to be set on the referring jumper (CAN-ID) too. The CAN-Bitrate (0x2001) is default assigned to the value „9“ (automatic detection). It is recommended to keep this value unchanged for an easier parameterization.

ⓘ	Use the „Object Dictionary“, to test or to change actual values.

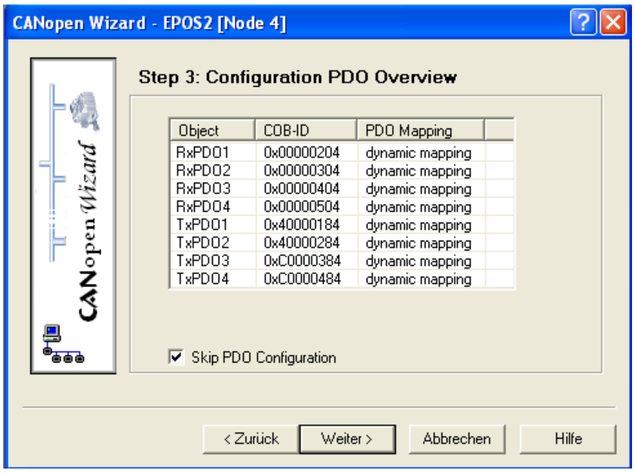


Step 2: Configuration SDO

Object	COB-ID
SDO COB-ID Server->Client (tx)	0x0584
SDO COB-ID Client->Server (rx)	0x0604

NOTE: The SDO COB-IDs are defined by DIP-Switch.
Client->Server = 0x0600 + Node-ID
Server->Client = 0x0580 + Node-ID

Check of the COB-ID's for the SDO's, (normally no changes needed).

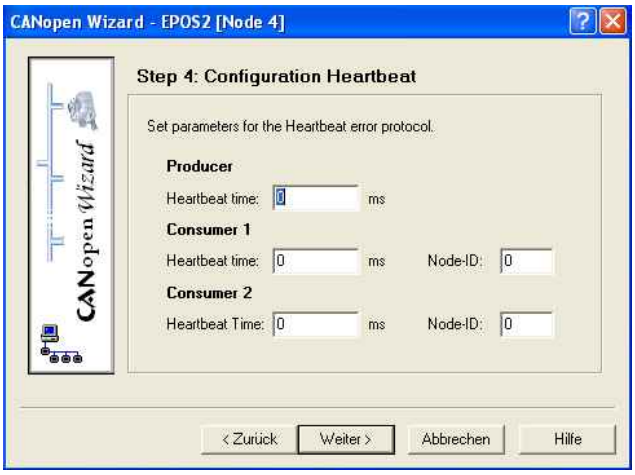


Step 3: Configuration PDO Overview

Object	COB-ID	PDO Mapping
RxPDO1	0x00000204	dynamic mapping
RxPDO2	0x00000304	dynamic mapping
RxPDO3	0x00000404	dynamic mapping
RxPDO4	0x00000504	dynamic mapping
TxPDO1	0x40000184	dynamic mapping
TxPDO2	0x40000284	dynamic mapping
TxPDO3	0xC0000384	dynamic mapping
TxPDO4	0xC0000484	dynamic mapping

☑ Skip PDO Configuration

The PDO-configuration can be skipped completey.



Step 4: Configuration Heartbeat

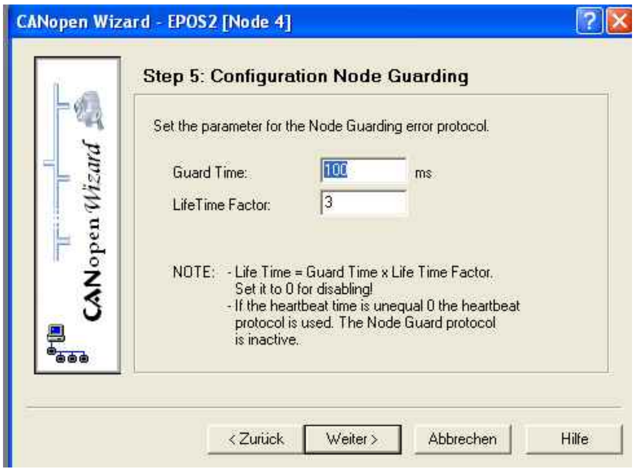
Set parameters for the Heartbeat error protocol.

Producer
Heartbeat time: 0 ms

Consumer 1
Heartbeat time: 0 ms Node-ID: 0

Consumer 2
Heartbeat Time: 0 ms Node-ID: 0

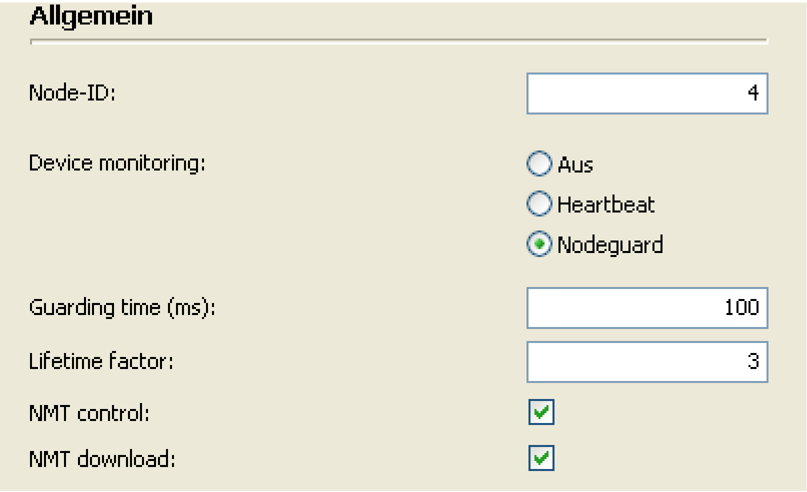
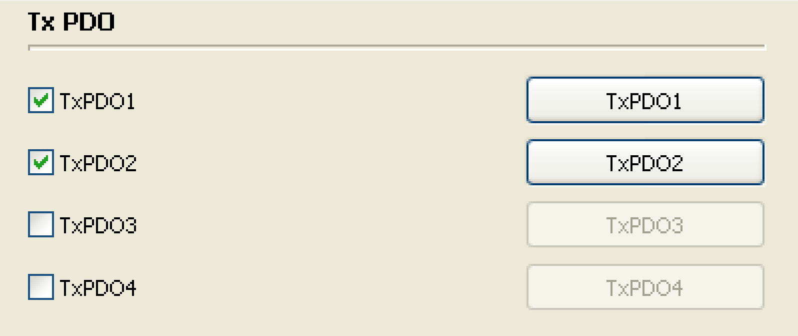
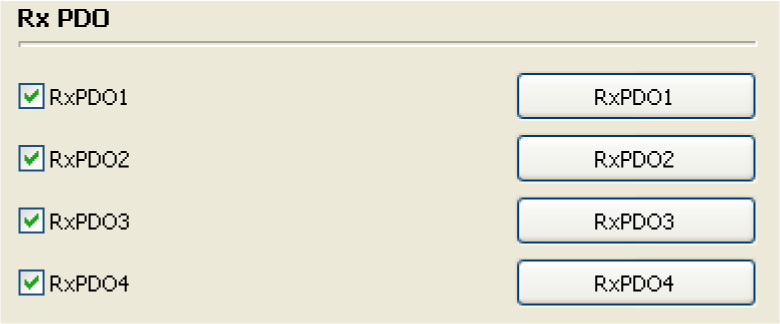
A heartbeat-monitoring is not used.

 CANopen Wizard - EPOS2 [Node 4] Step 5: Configuration Node Guarding Set the parameter for the Node Guarding error protocol. Guard Time: 100 ms LifeTime Factor: 3 NOTE: - Life Time = Guard Time x Life Time Factor. Set it to 0 for disabling! - If the heartbeat time is unequal 0 the heartbeat protocol is used. The Node Guard protocol is inactive. CANopen Wizard < Zurück Weiter > Abbrechen Hilfe	Nodeguarding will be configurated and activated by the INSEVIS-PLC.

6 Slave-configuration with the ConfigStage

With the ConfigStage-software will amongst others configurated the CANopen-Master and each CANopen-Slave. Also the connection from PLC-data (e.g. DataBlock and offset in the DataBlock) to the CANopen-data (R-PDO's, T-PDO's) will bedefined.

The axis can be taken over as type into the library of the ConfigStage!

 Allgemein Node-ID: 4 Device monitoring: ○ Aus ○ Heartbeat ● Nodeguard Guarding time (ms): 100 Lifetime factor: 3 NMT control: ☑ NMT download: ☑	Definition of Node-ID and guardings EPOS2 supports node guarding CANopen-settings (like COB-ID's) to load to EPOS2
 Tx PDO ☑ TxPDO1 [TxPDO1] ☑ TxPDO2 [TxPDO2] □ TxPDO3 [TxPDO3] □ TxPDO4 [TxPDO4]	TPDO (EPOS2→ CANopen-Master) 2 T-PDO's are necessary for the receive direction. activate the download of the communication parameter and mapping Response characteristics TPDO1 Typ: 254, no blocking time Response characteristics TPDO2 Typ: 254, define a blocking time of e.g. 100ms !
 Rx PDO ☑ RxPDO1 [RxPDO1] ☑ RxPDO2 [RxPDO2] ☑ RxPDO3 [RxPDO3] ☑ RxPDO4 [RxPDO4]	RPDO (CANopen-Master → EPOS2) 4 R-PDO's are necessary for the send direction. activate the download of the communication parameter and mapping Response characteristics RPDO1 Typ: 254, no blocking time Response characteristics RPDO2 Typ: 254, no blocking time Response characteristics RPDO3 Typ: 254, no blocking time Response characteristics RPDO3 Typ: 254, no blocking time
 SDO [SDOs]	additional configuration via SDO No additional SDOs are neccessary. (If you want, you can transfere here CANopen-data, what otherwise must be transferred by the EPOS-Studio.)

6.1 Mapping T-PDO1

Offset in data area (e.g. data block) of an instance from type „InDataE2Type“: **0** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6041	0x00	16 Bit/Word	State word DS402
2	0x2071	0x01	16 Bit /Word	Input word of the digital inputs
3	0x6061	0x00	16 Bit/Word	Actual operation mode DS402

6.2 Mapping T-PDO2

Offset in data area (e.g. data block) of an instance from type „InDataE2Type“: **6** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6064	0x00	32 Bit/DWord	Actual position [Post quadrature encoder]
2	0x606C	0x00	32 Bit/DWord	Actual velocity [U/min]

6.3 Mapping R-PDO1

Offset in data area (e.g. data block) of an instance from type „OutDataE2Type“: **0** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6040	0x00	16 Bit/Word	Control word DS402
2	0x607A	0x00	32Bit/DWord	Target [Post quadrature encoder]
3	0x6060	0x00	16 Bit/Word	Operation mode DS402

6.4 Mapping R-PDO2

Offset in data area (e.g. data block) of an instance from type „OutDataE2Type“: **8** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6081	0x00	32 Bit/DWord	Profile velocity [r/min]
2	0x2081	0x00	32 Bit/DWord	Home-position [Post quadrature encoder]

6.5 Mapping R-PDO3

Offset in data area (e.g. data block) of an instance from type „OutDataE2Type“: **16** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6083	0	32 Bit/DWord	Profile acceleration [r/min/s]
2	0x6084	0	32 Bit/DWord	Profile deceleration [r/min/s]

6.6 Mapping R-PDO4

Offset in data area (e.g. data block) of an instance from type „OutDataE2Type“: **24** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6085	0	32 Bit/DWord	Quick-stop-ramp [r/min/s]
2	0x60FF	0	32 Bit/DWord	Velocity in the profile Velocity Mode [r/min]

6.7 Additional SDO-transfers after PDO-mapping

Number	Index	Subindex	Size	Value	Explanation

7 S7-Sample-program

The sample project consists of an S7-program, what demonstrates the application of the MC-blocks.