Sample documentation

index of content

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 Parker C3I21T11 (http://www.parker-eme.com) is described. The software was created for INSEVIS-PLC and is based on the PLCopen-standard

On following devices the software test was done:

C3I21T11

Testing device	:	C3I21T11
Software-version	:	2011 R09-11
C3ServoManager	:	V 2.9.2.49 (June 2011)

INSEVIS

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

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_C3“, a S7-FB, is used to enable the motor. 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)

MC-Block/Symbol	Address	Function
MC_ReadStatus_C3	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_C3	FB41	Visualization of the error code of the drive
MC_ReadActualPosition_C3	FB42	Visualization of the actual postion of the motor
MC_ReadActualVelocity_C3	FB43	Visualization of actual velocity of the motor
MC_Reset_C3	FB44	Reset error in the drive
MC_Power_C3	FB45	Enable power stage to the motor or stop/disable fast as possible
MC_Stop_C3	FB46	Stop motor
MC_MoveAbsolute_C3	FB47	Move to an absolute position
MC_MoveRelative_C3	FB48	Move a relative distance
MC_MoveAdditive_C3	FB49	Append relative distance to a move
MC_MoveVelocity_C3	FB50	Endless move
MC_GearIn_C3	FB51	Execute synchronized motion functions (electronic gear), e.g. follow a master drive by encoder-pulses
MC_Home_C3	FB52	Execute homing move
MC_Jog_C3	FB53	Jog+/- move, stops on the software-end-delimiters
C3_Input	FB54	Read out C3-inputs (standard-inputs)
C3_Output	FB55	Write C3- outputs (Standard-outputs)
InDataC3Type	UDT100	Data type for input data CANopen, instanciate once per axis
OutDataC3Type	UDT101	Data type for output data CANopen, instanciate once per axis
SWPosC3Type	UDT102	Data type state word CANopen, INTERNAL USE ONLY
CWPosC3Type	UDT103	Data type control word CANopen, , INTERNAL USE ONLY
AxisRefC3Type	UDT104	Data type axis reference, instanciate once per axis

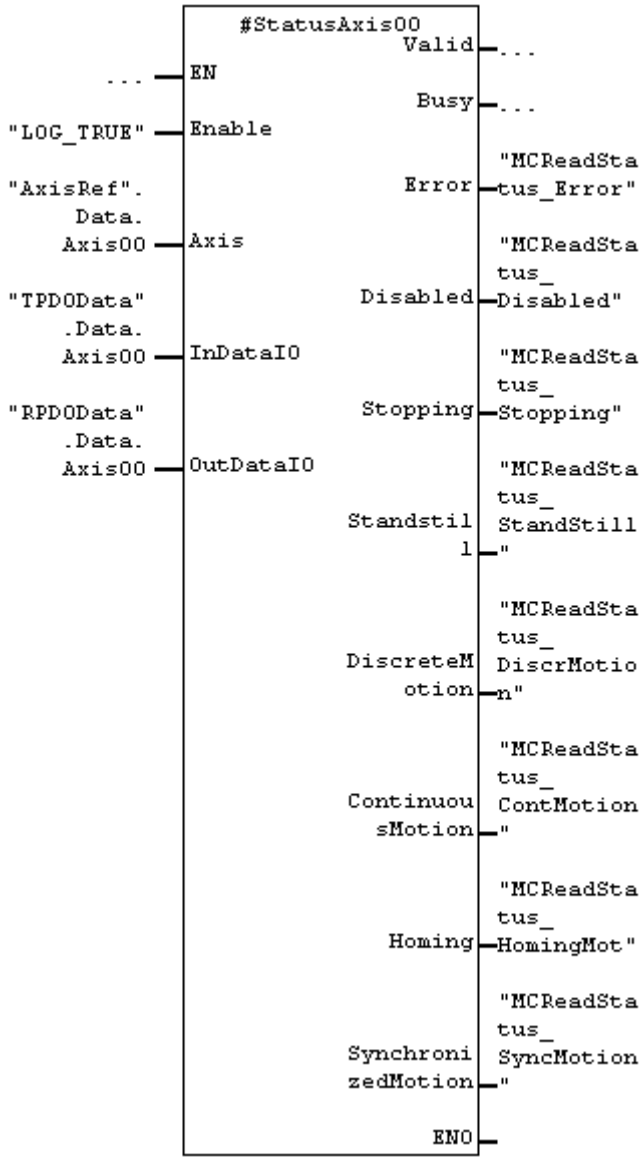
3.1 MC_ReadStatus_C3 (FB)

The MC_ReadStatus_C3 will be used for visualization (State generation) of different drive states. With these information the PLC-programm can watch most 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	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
InDataIO	IN_OUT	InDataC3Type (UDT)	Reference to IO-data (input-data CANopen)
OutDataIO	IN_OUT	OutDataC3Type	Reference to IO-data (output data

Name	Variables area	Type	Function
		(UDT)	CANopen)
Valid	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
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“
SynchronizedMotion	OUT	Bool	Axis is positioning synchronized (e.g. via electronic gear)

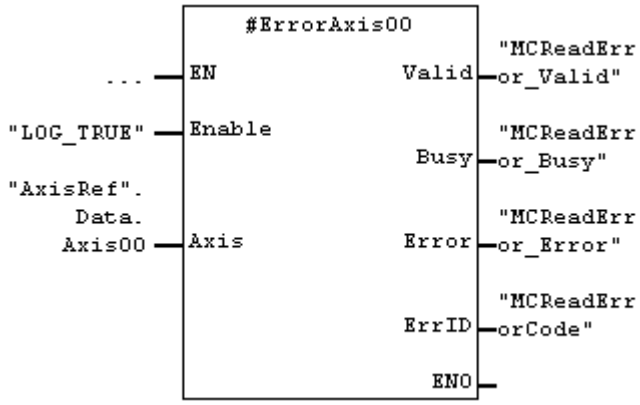


In the STAT-area of a Container-FB instanciated MC_ReadStatus_C3 with the instance name StatusAxis00.

3.2 MC_ReadAxisError_C3 (FB)

The MC_ReadAxisError_C3 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	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Valid	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error
ErrorID	OUT	WORD	Error code of axis (here 16-bit)

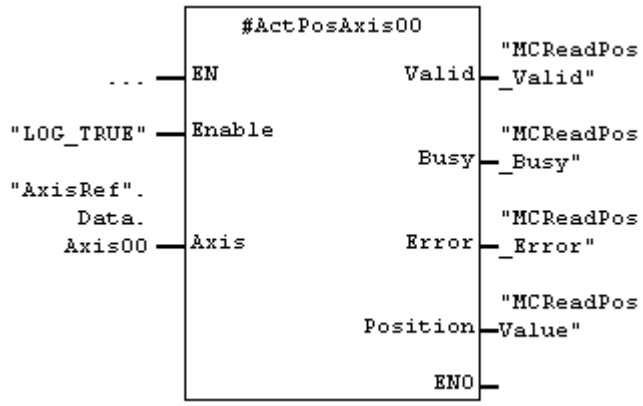


In the STAT-area of a Container-FB instanciated MC_ReadAxisError_C3 with the instance name ErrorAxis00.

3.3 MC_ReadActualPosition_C3 (FB)

The MC_ReadActualPosition_C3 provides the absolute position of the axis. This position can be changed by any kind of positioning, jogging, mechanical movments, homing moves or regulating oscillation.

Name	Variables area	Type	Function
Enable	IN	Bool	Read actual position
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Valid	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error
Position	OUT	REAL	Axis position in user units, e.g. „mm“

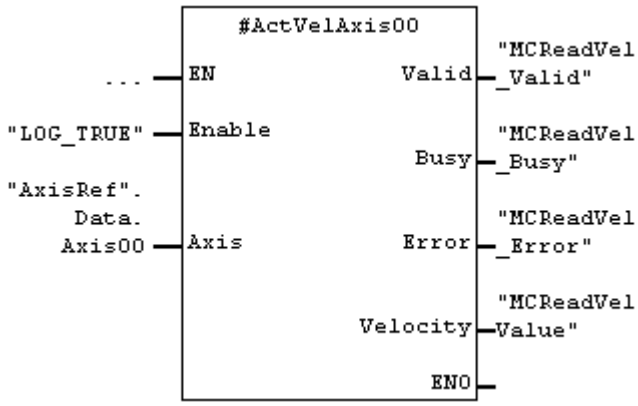


In the STAT-area of a Container-FB instanciated MC_ReadActualPosition_C3 with the instance name ActPosAxis00.

3.4 MC_ReadActualVelocity_C3 (FB)

The MC_ReadActualVelocity_C3 provides the actual velocity of the axis. The velocity can be changed by any kind of positioning, jogging, mechanical movments, homing moves or regulating oscillation.

Name	Variables area	Type	Function
Enable	IN	Bool	Read actual velocity
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Valid	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error
Velocity	OUT	REAL	Axis velocity in user units, e.g. „mm/s“



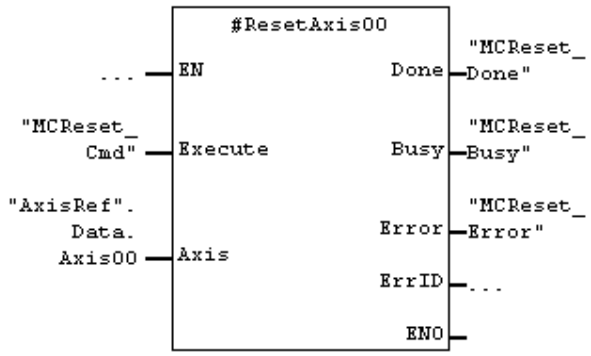
In the STAT-area of a Container-FB instanciated MC_ReadActualVelocity_C3 with the instance name ActVelAxis00.

3.5 MC_Reset_C3 (FB)

With the block MC_Reset_C3 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	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Data are valid
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error
ErrorID	OUT	WORD	Error code of axis (here 16-bit)



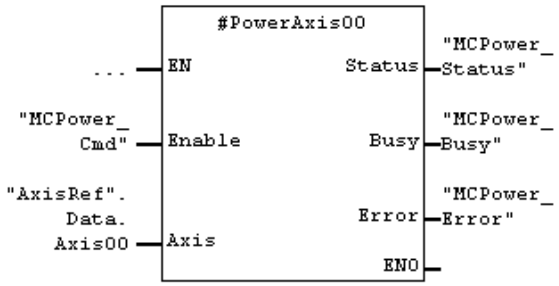
In the STAT-area instanciated MC_Reset_C3 with the instance name ResetAxis00.

3.6 MC_Power_C3 (FB)

With the block MC_Power_C3 the axis will be enabled or disabled.

ⓘ	It is only one instance of this FB allowed and reasonable. The immediate stop ramp and the immediate stop jerk will be configuredwerden by the C3-ServoManager.

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	AxisRefC3Type (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



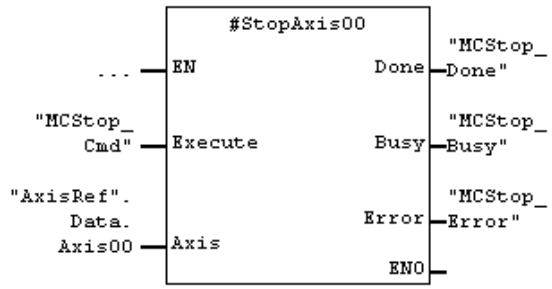
In the STAT-area instanciated MC_Power_C3 with the instance name PowerAxis00.

3.7 MC_Stop_C3 (FB)

With the block MC_Stop_C3 the axis will be stopped. Stopping is only possible with a power stage enabled axis.

ⓘ	It is only one instance of this FB allowed and reasonable. The stop ramp and the stop jerk will be configured with the C3 ServoManager. At an 0-1-edge the axis motion is stopped. Axis moves (new requests) will be blocked at activated Stop-Execute (=1) generally.

Name	Variables area	Type	Function
Execute	IN	Bool	1 Stoppes axis 0 Move enable of axis
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Axis stopped
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error

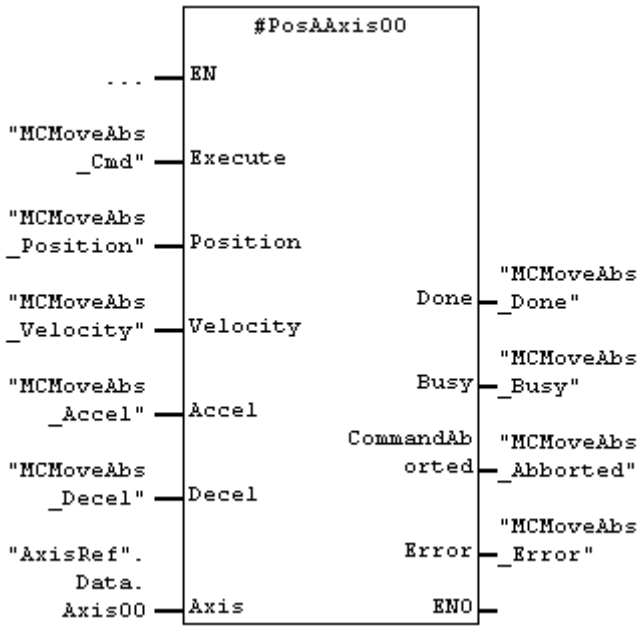


In the STAT-area instanciated MC_Stop_C3 with the instance name StopAxis00.

3.8 MC_MoveAbsolute_C3 (FB)

The block MC_MoveAbsolute_C3 will be used for absolute positioning. Reference point of the absolute position (the mathematical zero-pointis is defined by teaching (absolute encoder) or by homing reference travel.

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	AxisRefC3Type (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

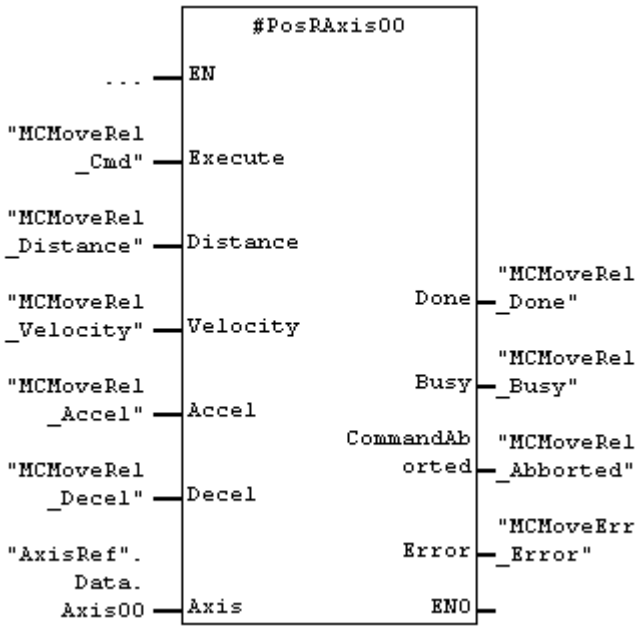


In the STAT-area instanciated MC_MoveAbsolute_C3 with the instance name PosAAxis00.

3.9 MC_MoveRelative_C3 (FB)

The block MC_MoveRelative_C3 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 starts 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/s²“
Decel	IN	Dint	Deceleration in user units, e.g. „mm/s²“
Axis	IN_OUT	AxisRefC3Type (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

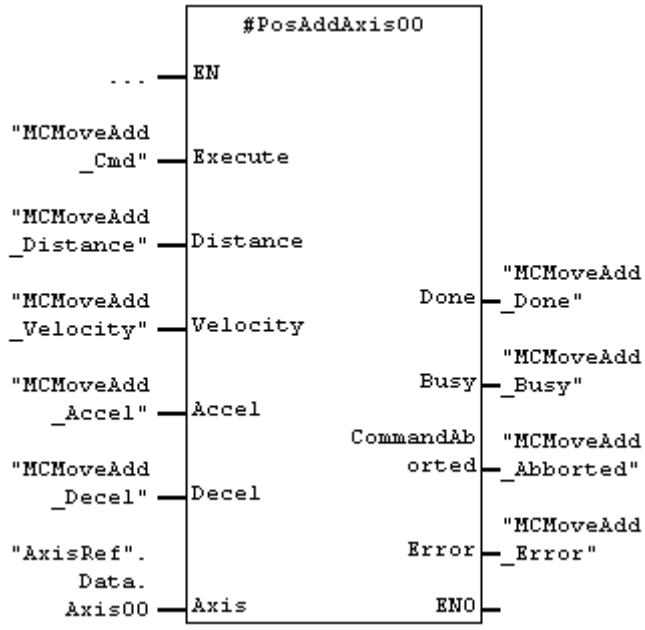


In the STAT-area instanciated MC_MoveRelative_C3 with the instance name PosRAxis00.

3.10 MC_MoveAdditive_C3 (FB)

The bock MC_MoveAdditive_C3 will be used for additive positioning (for a distance). In difference to the relative positioning the distance will be added here onto the actual target.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge starts 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/s²“
Decel	IN	Dint	Deceleration in user units, e.g. „mm/s²“
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Axis has reached 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

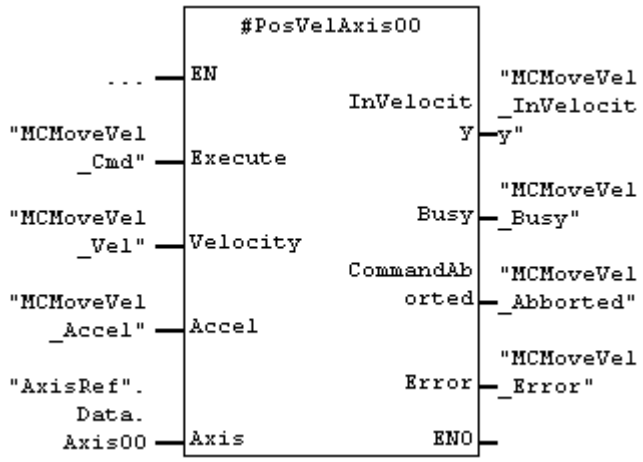


In the STAT-area instanciated MC_MoveAdditive_C3 with the instance name PosAddAxis00.

3.11 MC_MoveVelocity_C3 (FB)

The block MC_MoveVelocity_C3 will be used for endless moves, whereat the position controller stays active, so that during position. The move MUST be stopped witht MC_Stop_C3.

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 and deceleration in user units, e.g. „mm/s²“
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
InVelocity	OUT	Bool	Axis has reached target velocity
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_C3 with the instance name PosVelAxis00.

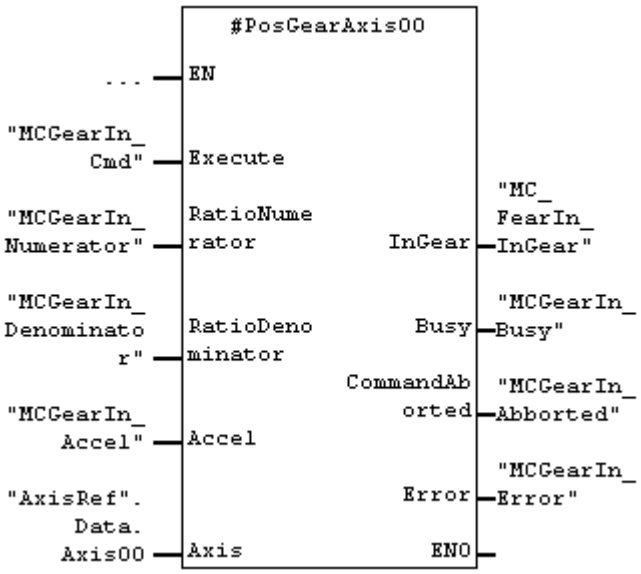
3.12 MC_GearIn_C3 (FB)

The block MC_GearIn_C3 will be used, if the axis should follow a master drive (e.g. another axis or an encoder). In that case for a positioning a time based move profile will not be calculated, but a under limitation on a maximale acceleration resp. deceleration a guide value, what can be a RS422-encoder signal or an analog value or a motion-bus-setpoint (using C3 the so called HEDA-Bus). Type and resolution of the master setpoint values will be configured by the C3-ServoManager.

The ratio of numerator/ denominator will be transferred to the servo drive. It is necessary to pay attention that as minimum gear reduction 0.001 can be transferred, also the ratio is not better adjustable than with 0.001.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge startes the move
RatioNumerator	IN	Real	Numerator gear factor
RatioDenominator	IN	Dint	Denominator gear factor
Accel	IN	Dint	Maximal acceleration and deceleration in user units e.g. „mm/s²“
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
InVelocity	OUT	Bool	Axis has reached target velocity

Name	Variables area	Type	Function
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



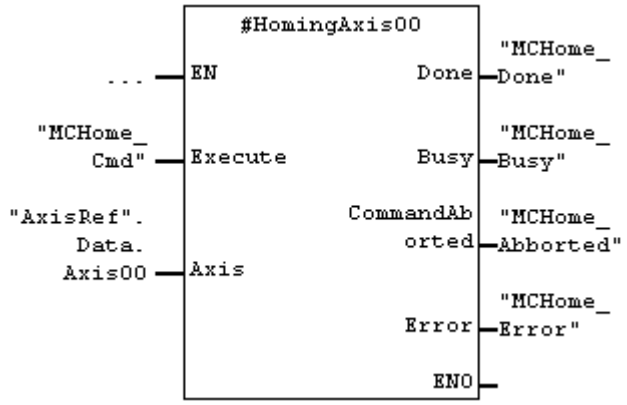
In the STAT-area instanciated MC_GearIn_C3 with the instance name PosGearAxis00.

3.13 MC_Home_C3 (FB)

the C3-manual. If an absolute encoder or an absolute encoder simulation will be used, the mode must be set to 0 after teaching (no referencing necessary). The profile values (velocity, acceleration, jerk) are part of the C3-configuration.

ⓘ	It is only one instance of this FB allowed and reasonable. (1) Set up the homing-offset in the configuration. After the referencing the axis will be reported as actual position with a value of -1.0 * homing-offset. (2) It is also possible to set up in the C3-ServoManager, if later on during the homing move the mathematical zero point shell be driven too.

Name	Variables area	Type	Function
Execute	IN	Bool	0-1-edge startes the reference move
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Done	OUT	Bool	Reference move finished, the set position was set, the mathematic zero point and if necessary 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



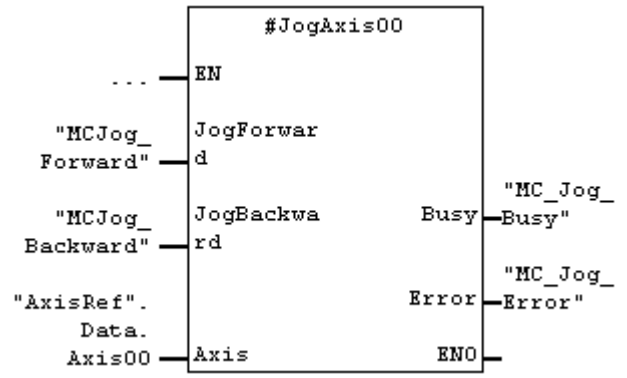
In the STAT-area instanciated MC_Home_C3 with the instance name HomingAxis00.

3.14 MC_Jog_C3 (FB)

The block MC_Jog_C3 will be used to move the axis „manually“ (also named „inching“). Both directions are possible, the profile values (acceleration, deceleration, jerk) will be defined by the C3-configuration. If there are desinded software-end-limits, the axis stopps on the defined software-end-limits.

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

Name	Variables area	Type	Function
JogForward	IN	Bool	0-1-edge starts jogging clockwise, 1-0-edge stops the axis
JogBackward	IN	Bool	0-1-edge starts jogging counterclockwise, 1-0-edge stops the axis
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
Busy	OUT	Bool	Function is executed/runs
Error	OUT	Bool	Axis with error

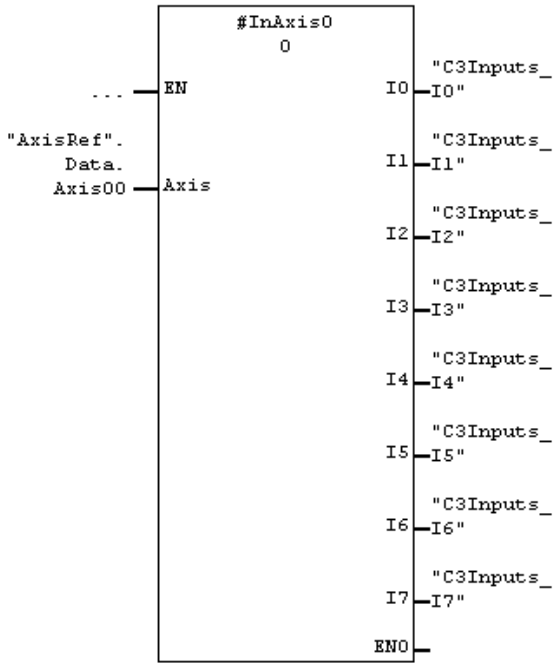


In the STAT-area instanciated MC_Jog_C3 with the instance name JogAxis00.

3.15 MC3_Input (FB)

The block C3_Input is a special version for the C3-axis, which provides the „lower“ C3-inputs 0 to 7 as state. Input 5 and 6 are reserved for position switches, so that the polarity can be reverse there by correlative configuration. Input I7 is reserved for the reference/homing switch.

Name	Variables area	Type	Function
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis refererence (axis pointer)
I0 bis I7	OUT	Bool	Inputs 0 to 7 as state

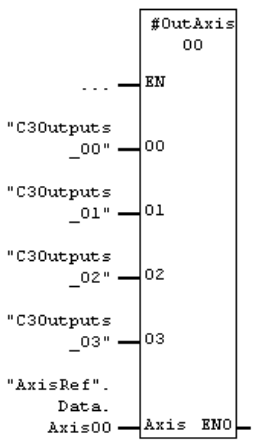


In the STAT-area instanciated C3_Input with the instance name InAxis00.

3.16 C3_Output (FB)

The block C3_Output is a special version for the C3-axis, which provides the „lower“ C3-outputs0 to 3 to set/reset.

Name	Variables area	Type	Function
Axis	IN_OUT	AxisRefC3Type (UDT)	Axis reference (axis pointer)
O0 bis O3	IN	Bool	Outputs 0 to 3 to set/reset



In the STAT-area instanciated C3_Output with the instance name OutAxis00.

3.17 InDataC3Type (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 C3-axis.

ⓘ	The best solution is to set up all instances (of the different axis´) of InDataC3Type in a separate DB (e.g. „TPDODATA“).

```
wStatusWord    :   WORD;      // TPDO1, async, 0x6041 + 0x00, Status word
iActModeOfOp   :   INT;       // TPDO1, async, 0x6061 + 0x00, Actual mode of operation
wDigInWord     :   WORD;      // TPDO1, async, 0x6100 + 0x01, Digital inputs (Standard)
wErrCode       :   WORD;      // TPDO1, async, 0x603f + 0x00, Error code
diActPosition  :   DINT;      // TPDO2, async, 0x6064 + 0x00, Actual position [units * 1000]
diActVelocity  :   DINT;      // TPDO2, async, 0x606C + 0x00, Actual velocity [units/s * 1000]
```

3.18 OutDataC3Type (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 C3-axis.

ⓘ	The best solution is to set up all instances (of the different axis´) of OutDataC3Type in a separate DB (e.g. „RPDODATA“).

```
wControlWord   :   WORD;      // RPDO1, async, 0x6040 + 0x00, Control word
iModeOfOp      :   INT;       // RPDO1, async, 0x6060 + 0x00, Mode of operation
diTarget       :   DINT;      // RPDO1, async, 0x607a + 0x00, Target [units * 1000]
diProfVelocity :   DINT;      // RPDO2, async, 0x6081 + 0x00, Profile velocity [units/s * 1000]
wDigOutWord    :   WORD;      // RPDO2, async, 0x6300 + 0x01, Digital outputs (Standard)
diProfAccel    :   DINT;      // RPDO3, async, 0x6083 + 0x00, Prof. acceleration [units/s2]
diProfDecel    :   DINT;      // RPDO3, async, 0x6084 + 0x00, Prof. deceleration [units/s2]
```

3.19 AxisRefC3Type

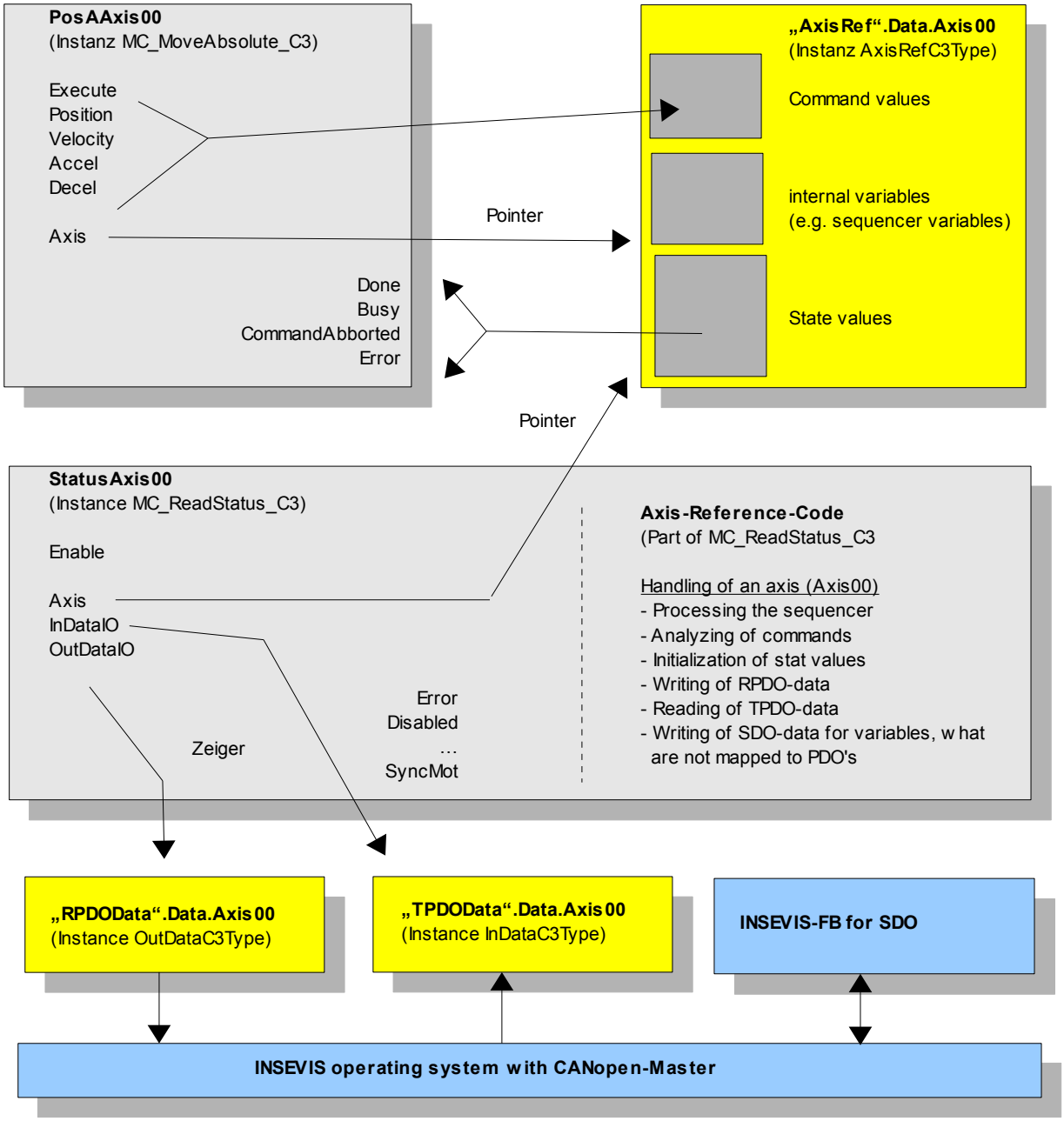
This data type will be used internally as axis working data of the axis. Because the instanciated variables were handled over as IN_OUT, the efford for copying is low.

The block MC_ReadStatus_C3 uses these variables, definded by the axis reference, for processing the sequencers.

ⓘ	The best solution is to set up all instances (of the different axis´) of AxisRefC2Type in a separate DB (e.g. „AXISREF“)

4 Sample of a MC-block-instance

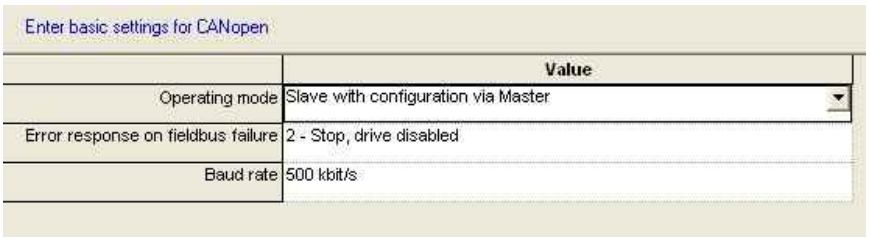
Following figure shows the use and the data flow of a MC-block for one axis with the name Axis00.



5 CANopen-configuration with the C3-ServoManager

5.1 Configure communication

Gererally there can not payed attention for the common configuration of a C3-servo drive. Important-for the CANopen-part is only, that C3I21 can be configured with up to 4 PDO's in each direction and that it supports SDO's. Following settings have to be done by the C3-Wizard for CAN.



operation mode
Slave with configuration via master
→ The PDO's will be configured and allocated from CANopen-Master

Error reaction while bus failure
→ Here was choosen an alternative, where the servo drive stopps at bus failure and switches currentless.

Baud rate
Adjustable in steps from 20 kBit/s up to 1 MBits/s

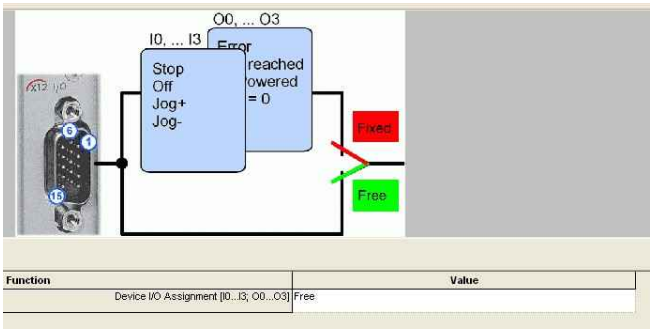
On the C3-device the Node-ID must be set up, optional the baud rate can be set, what is only useful, if the device would be configured completely via CAN. The efford for this is really high (see the manufacturers instruction for more).

Set up the Node-ID
→ DIP-switch 1..7: binary setting (OFF/ON) of the Node-ID from 1 bis 127, DIP-switch 8: OFF

5.2 Configure C3-device

The device is to be configured referring to the requrements of your hardware (device, motor) and application. Parameters, like e.g. the jog-profile are to provide, because not every parameter can be provided via the MC-blocks.

Note following setting for the in-/outputs on the C3:



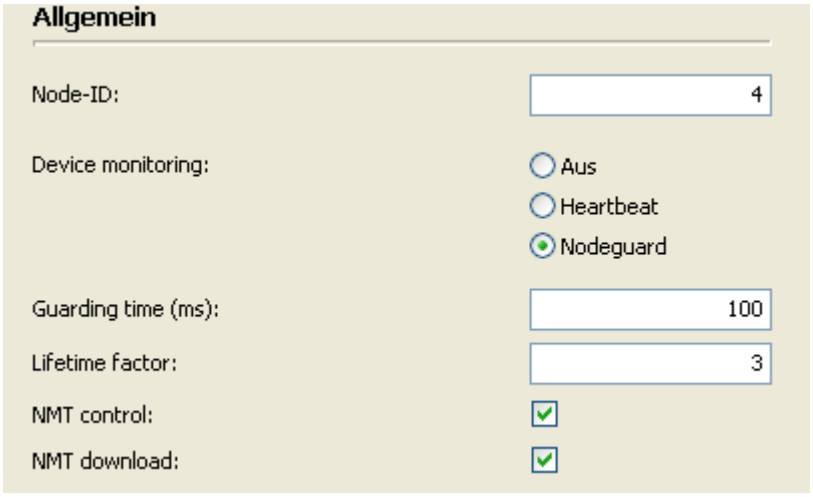
The 24V-inputs I0...I7 can be used with this setting on the C3 „free“ and can be used as additional resources for the PLC. If position switches will be used I5 resp. I6 are designed for this purpose. If a switch is used on a reference move, I7 must be used for it.

The 24V-outputs O0...O3 are available as additional resources for the PLC. Note the boundary conditions like (current capability, switching of inductive loads, and so on).

6 Slave-Configuration with 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.

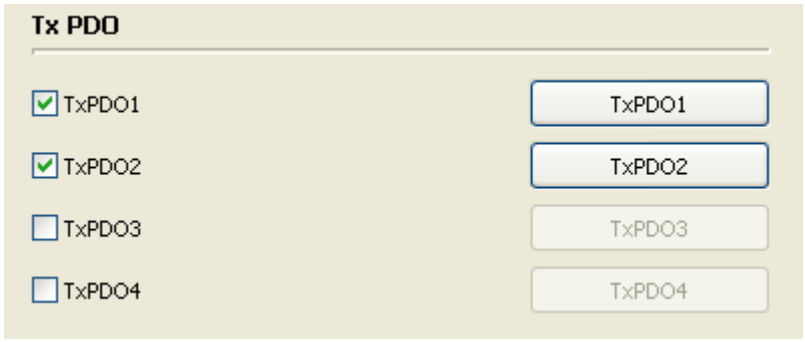
The axis can be taken over as type into the library of the ConfigStage!



Definition of Node-ID and Guardings

C3 supports Nodeguarding

CANopen-settings (like COB-ID's) to load to C3



TPDO (C3 → 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 !

RPDO (CANopen-Master → C3)

3 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

additional configuration via SDO

After the download of the mapping parameters (initiated by PLC-firmware) further settings on the C3, what can not be made by the C3-ServoManager-configuration, will be transfererd by SDO

6.1 Mapping T-PDO1

Offset in data area (e.g. data block) of an instance from type „InDataC3Type“: **0** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6041	0	16 Bit/Word	State word DS402
2	0x6061	0	16 Bit/Word	Actual operation mode DS402
3	0x6100	1	16Bit/Word	Basic inputs C3
4	0x603F	0	16Bit/Word	Error code C3

6.2 Mapping T-PDO2

Offset in data area (e.g. data block) of an instance from type „InDataC3Type“: **8** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6064	0	32 Bit/DWord	Actual position {units * 1000]
2	0x606C	0	32 Bit/DWord	Actual velocity [units/s * 1000]

6.3 Mapping R-PDO1

Offset in data area (e.g. data block) of an instance from type „OutDataC3Type“: **0** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6040	0	16 Bit/Word	Control word DS402
2	0x6060	0	16 Bit/Word	Operation mode DS402
3	0x607A	0	32Bit/DWord	Desired value 1 (variable), [e.g. units * 1000]

6.4 Mapping R-PDO2

Offset in data area (e.g. data block) of an instance from type „OutDataC3Type“: **8** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6081	0	32 Bit/DWord	Profile velocity [units/s * 1000]
2	0x6300	1	16Bit/Word	Basic outputs C3

6.5 Mapping R-PDO3

Offset in data area (e.g. data block) of an instance from type „OutDataC3Type“: **12** (Byte-Offset)

Number	Index	Subindex	Size	Explanation
1	0x6083	0	32 Bit/DWord	Profile acceleration [units/s^2 * 1000]
2	0x6084	0	32 Bit/DWord	Profile deceleration [units/s^2 * 1000]

6.6 Additional SDO-transfers after PDO-mapping

Number	Index	Subindex	Size	Value	Explanation
1	0x605A	0	16 Bit/Word	6	„Quick stop mode“ set up so, that a stopp causes a reject of the position

7 S7-Sample-program

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