



## **SoftMotion: DriveInterface: SercosDrive.lib**

**Document Version 2.18**

**Library Version 1.9.4.0**

## CONTENT

<b>VERSION AND APPLICABILITY INFORMATION</b>	<b>3</b>
<b>1 PARAMETERS IN PLC CONFIG</b>	<b>4</b>
1.1 BusInterface	4
1.2 AxisGroup	4
1.3 supported Drive.wControlType	4
1.4 Additional Structure SERCOS_AXIS_REF	4
1.5 Maximum Number of Axis	5
1.6 Global variables	5
<b>2 FEATURES</b>	<b>6</b>
<b>3 START-UP/CONFIGURATION</b>	<b>7</b>
3.1 AxisGroup.wState	7
3.2 Error sources	7
3.3 Parameter Setting during Startup	8
3.4 Position Scaling (IDN76)	10
3.5 PackProfile conformity check	10
<b>4 SERCOS SPECIFIC FBS FOR ACCESSING DRIVE PARAMETERS</b>	<b>12</b>
4.1 Parameter Identifier (S/P)	12
4.2 Reading/Writing Normal Parameters with a Standard Length	12
4.3 Reading/Writing List Parameters	13
4.4 Reading String Parameters	13
<b>5 SERCOS I/OS</b>	<b>14</b>
<b>6 WAYS OF SYNCHRONIZING PLC AND SERCOS HARDWARE</b>	<b>15</b>
<b>CHANGE HISTORY</b>	<b>16</b>

**Version and Applicability Information**

Hardware interface	SercosBase.lib (interface version 4)
Supported drives	all drives and I/Os supporting Sercos standard
Runtimes	x86 (RTE for Windows NT/2000/XP; CE; Linux)
Components	Sercosdrive.lib SercosBase.lib (interface version 4) or driver with identical interface; SysLibFile.lib; (for drive specific config text file) SysLibCallback.lib
Document Version	2.16 (05.09.2006)

## 1 Parameters in PLC config

### 1.1 BusInterface

wParam1	Not used
wParam2	Not used
dwParam1	Not used
dwParam2	reserved

### 1.2 AxisGroup

wParam1	Baudrate in MBaud
wParam2	brightness of LED (1-63)
wParam3	Not used
wParam4	Not used
dwParam1	Not used
dwParam2	Not used
dwParam3	internal use
dwParam4	internal use
dwParam4	internal use

### 1.3 supported Drive.wControlType

T / - yes	V/V yes	V/P yes	P/P yes	PV/PV yes	V/- yes	CONF yes
-----------	---------	---------	---------	-----------	---------	----------

The cyclically sent/received data can consist of: fSet/ActPosition, fSet/ActVelocity, fSet/ActTorque.

Two additional IDNs can be both sent and received cyclically using awAddRealtimeSend/ReceiveIDN (see below).

### 1.4 Additional Structure SERCOS\_AXIS\_REF

name	Type	
bReceivedRealtimeBit1, bReceivedRealtimeBit2	BOOL	realtime bits of status word from drive
bSendRealtimeBit1, bSendRealtimeBit2	BOOL	realtime bits for control word to drive
wReceivedRealtimeBit1ID, wReceivedRealtimeBit2ID, wSendRealtimeBit1ID, wSendRealtimeBit2ID	WORD	internal use (IDN mapped onto realtime bits (IDN301, 303, 305, 307); checked at startup)
awOperationMode	ARRAY [0..3] OF WORD	content of IDN32, 33, 34, 35
awAddRealtimeSendIDN, awAddRealtimeReceiveIDN	ARRAY [0.. SERC_ADD_CYCLIC] OF	IDN of additional cyclic data must be set during first IEC

	WORD	cycle
abyAddRealtimeSendLength, abyAddRealtimeReceiveLength	ARRAY [0.. SERC_ADD_CYCLIC] OF BYTE	length of additional cyclic data in bytes  must be set during first IEC cycle
adwAddRealtimeSendData, adwAddRealtimeReceiveData	ARRAY [0.. SERC_ADD_CYCLIC] OF DWORD	additional cyclic data
abyActiveTriggers	ARRAY[0..1] OF BYTE	internal use
strConfigFromFile	STRING	full path and name of config text file (see below)  must be set during first IEC cycle
ascl, ascsl	ARRAY[0..4] OF Sercos_Cyclic_Entry	internal use (cyclic data)
asrl, assl	ARRAY[0..SERC_MAX_ENTRIE S] OF Sercos_Entry	internal use (config lists at startup)
sc, scr, srds, sw, sr, srl, swl	SercosBase-FBs	internal use
byOperatingMode	BYTE	operation Mode
blsIO	BOOL	FALSE (default) for drives; TRUE for I/Os

## 1.5 Maximum Number of Axis

The maximum number of axis (SERC\_MAX\_DRIVES) is defined in SercosBase.lib and not limited by this library.

The minimum cycletime for n axis is depending on the amount of cyclic data and the baudrate.

## 1.6 Global variables

In this lib the following global variables are defined, which shouldn't been touched by the user, but give the driver of the next lower level (e.g. SercosBase.lib) the possibility to configure the functionality programmed in Sercosdrive.lib:

g_iInterruptStartingState	State from which the interface starts producing interrupts
g_bInterfaceDoesTiming	if set, no SlotCalculation and reading/writing of timing parameters is done by Sercosdrive.lib
g_bInterfaceDoesCommandsForPhaseSwitch	If set, commands 127 and 128 are not executed by Sercosdrive.lib
g_bTurnDWORDS	If set, lower and upper word of (read/to be written) DWORDS are turned
g_bAlwaysUseConfigCyclicTeleType	If set, no preferred cyclic telegram type (IDN 15) is used, but always the configurable one (7)
g_byWaitDuringStartupInPhase	If set <>255, the start-up procedure is stopped in the specified phase and not continued until this value is changed.
g_byCurrentPhase	monitors the current Sercos phase (255: unknown)

## 2 Features

- **RegulatorOn, DriveStart**
- Detecting and acknowledging **errors**
- **reading/writing** SoftMotion parameters
- **reading/writing** drive parameters (see 4)
- any **gearing factors** (dwRatioTechUnitsDenom/iRatioTechUnitsNum)
- **linear/rotary axes**: the rotary drive only works with incrementally weighted positions (ID76 = 0)
- **controlling modes**: position (ID 32), velocity (ID 33), torque (ID 34)
- drive internal **homing** (configure the homing method with Sercos-ID 147 and 52 etc.)
- **limit switches** should be connected to the drive
- **configuration from file** (see below)
- **configuration from dialog settings in CoDeSys PLC configuration**
- **latching**:

Trigger. iTriggerNumber	Sercos Trigger	capture occured?	capture position
1	Sercos Trigger1, positive latched	409	130
2	Sercos Trigger1, negative latched	410	131
3	Sercos Trigger2, positive latched	411	132
4	Sercos Trigger2, negative latched	412	133

Make sure, that ID170 is set to 3 (e.g. at startup). When starting capturing, ID405 or ID406 is enabled automatically.

For fast latching (cyclic data transfer; no service channel) configure RealtimeBits to transmit ID 405/406 and receive ID 409-412 and add ID130-133 to cyclic data.

example for fast latching (in config file):

```
301, 0, 2, 405, 2; (Probel Enable 405 -> Realtime Control Bit1)
305, 0, 2, 409, 2; (Probel pos. latched 409 -> Realtime Status Bit1)
169, 0, 2, 2#0001, 2; (Enable Probel for rising edge)
170, 0, 2, 2#11, 4; (Enable Probing)
```

Set in the PLC configuration dialog or during the first IEC cycle :

```
Drive_MS.abvAddRealtimeReceiveLength[0] := 4;
Drive_MS.awAddRealtimeReceiveIDN[0] := 130;
```

- **2 real time inputs** (bReceivedRealtimeBit1, bReceivedRealtimeBit2)
- **2 real time outputs** (bSendRealtimeBit1, bSendRealtimeBit2)
- switching operation modes to torque, velocity or position controlled with **SMC\_SetControlMode**

### 3 Start-up/Configuration

#### 3.1 AxisGroup.wState

The variable wState of an AxisGroup represents its internal state. In case of this SercosDriver there is the following correlation with the Sercos phase:

wState	phase	action
0	?	program not started yet
1	?	start SERCOS base driver
2	?	wait for interrupt
10	?	switch phase to 0
11	0	switch to phase 0 successful
20	0	switch phase to 1
21	1	switch to phase 1 successful
30	1	switch to phase 2
31	2	switch to phase 2 successful; read basic IDs for each drive
32	2	slot calculation for each drive
33	2	clear configuration list for each drive
34	2	read configuration file of each drive
35	2	prepare and write configuration list (phase 2) to each drive
38	2	set command 127 and check list IDN 21
39	2	wait for end of command 127
40	2	switch to phase 3
50	3	switch to phase 3 successful; write configuration list (phase 3) to each drive
60-65	3	check configuration (command 128, IDN 22)
70	3	switch to phase 4
80	4	switch to phase 4 successful; write configuration list (phase 4) to each drive
81	4	read IDNs and calculate scaling factors
99	4	switch to normal operation
100	4	normal operation
200-210	4	prepare for resetting AxisGroup
>1000	?	error during startup in state wState-1000

#### 3.2 Error sources

If the AxisGroup does not come to state 100, please first check the global variable in SM\_DriveBasic.lib g\_strBootupError. Additionally check wState of the AxisGroup:

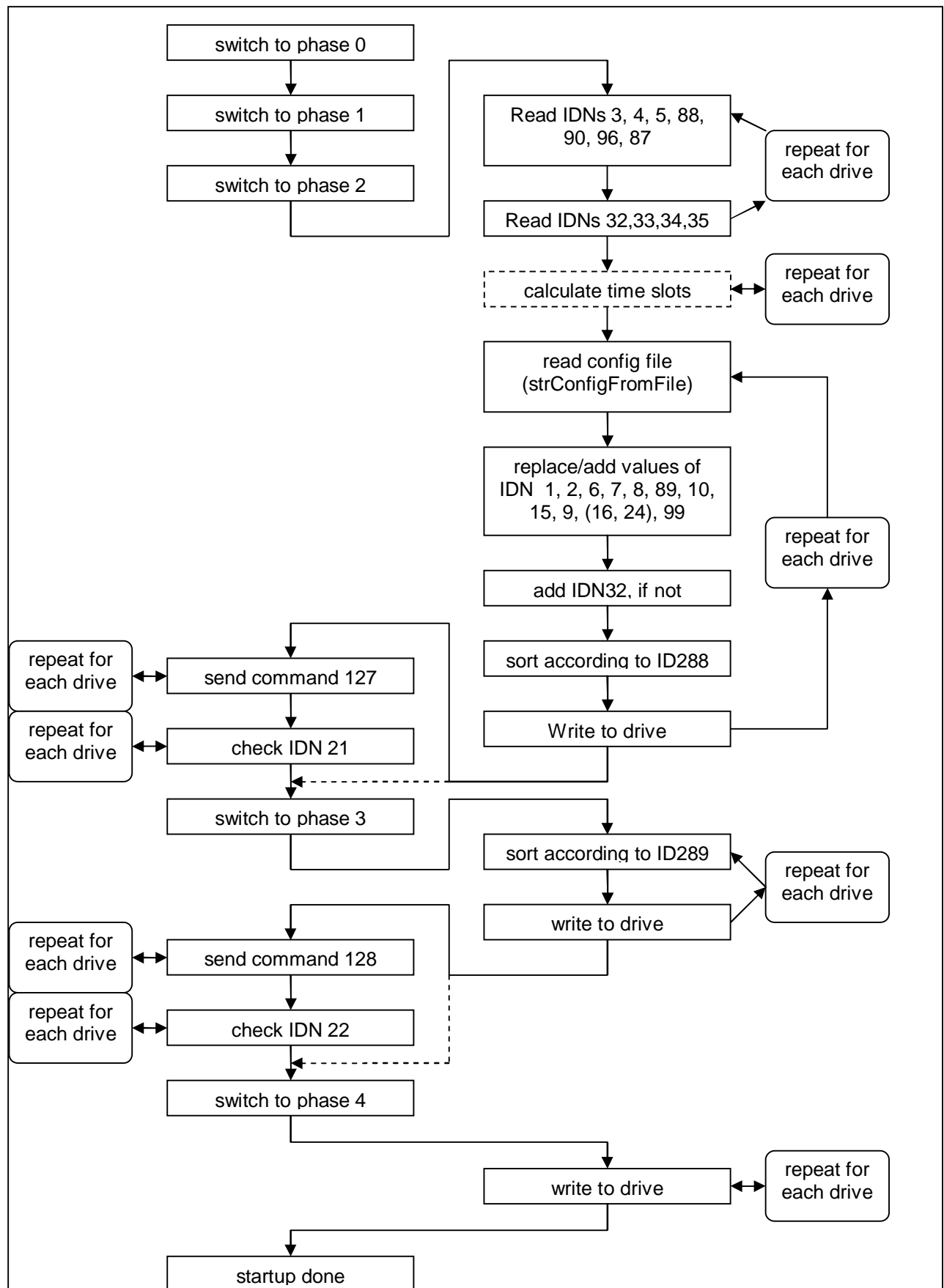
wState	Errors
0	program not started yet; task config not correct; general problem with the PLC
2	no interrupt received; problem with the hardware driver; task config not correct
1000	a) Versions of hardware driver and SercosBase.lib don't fit. Contact your system manufacturer;

	b) at least one drive switched back from phase 4 unexpectedly: possibly a ring-brake occurred
1010	problem in optical wires; wrong baudrate; wrong brightness of LED
1020	DriveID incorrect; one of the drives is missing / defect

### 3.3 Parameter Setting during Startup

The startup of the drive is done as shown in the figure below:





This start-up procedure is executed until it is completely finished (AxisGroup.wState=100). To stop the start-up in a certain phase, one can set **g\_byWaitDuringStartupInPhase** to the phase number. The start-up process will stop as soon as it has reached the specified phase and wait until the global variable is set to another value. It is not intended to switch back to a lower phase during operation.

This can be done by setting the phase **g\_byWaitDuringStartupInPhase** and using the FB **SMC\_ResetAxisGroup**.

You can configure the Sercos Master to send additional parameters during startup by using the dialogs in the PLC configuration inside CoDeSys. Alternatively you can configure your drive from a ASCII text file. Simply set the full path and name of this file in the Variable **<DriveName>\_MS.strConfigFromFile** during the first PLC cycles.

Be aware that if you use this method, all parameters from the dialog are removed before reading the file.

The text file must obey the following rules:

- one line represents one ID (or one list entry for list IDs); comment or empty lines are allowed
- a comment is marked by brackets (nested comments are not supported!)
- one line consists of 5 attributes: wIndex (ID), wListIndex (0 for normal, 1..n for list entries, 16#FFFF for commands), wSize (in bytes), dwValue (value), byPhase (2, 3, 4 or 255)
- all attributes are separated by a comma; a entry is terminated by a semicolon; spaces and tabs are allowed

Example:

```
(Konfigurationsfile für Kollmorgen Seidel ServoStar 600)
(S-Parameter,list position,length,      value,      phase)
76,          0,          2,          10,          2;      (Wichtung Position)
44,          0,          2,          2,           2;      (Wichtung Geschw.)
160,         0,          2,          0,           2;      (Wichtung Beschl.)
169,         0,          2,          2#0101,      2;      (Messtaster-
Steuerparameter)
```

A byPhase attribute of 255 signals that this entry mustn't be sent.

If you want to configure **list parameters**, you must specify the complete list, starting with index 1. It is not allowed, to specify only a single list parameter.

### 3.4 Position Scaling (IDN76)

Some drives don't support the wrap around in position values at  $2^{32}-1$  to 0. This is not a problem when running a linear axis (with limited position values).

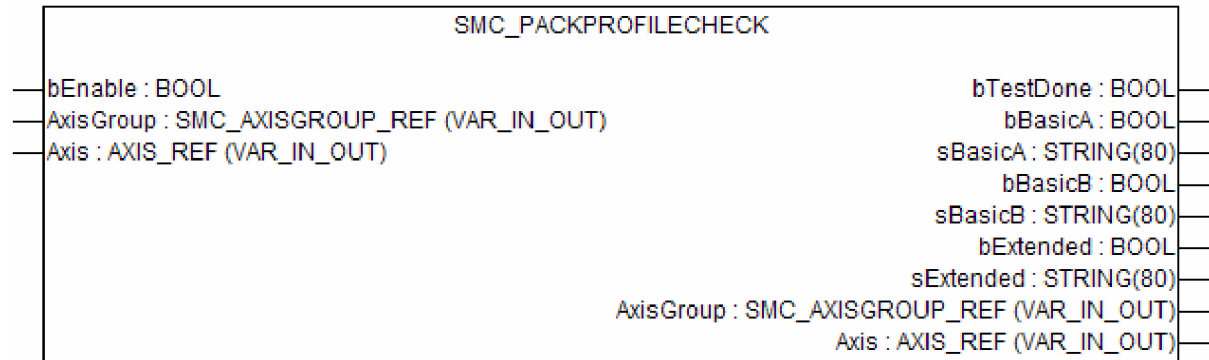
Using an endless (rotary/modulo) axis, this will cause a problem (typically a position error fault at that point). In this case use a rotary position scaling (bit2-0 = 2) with modulo functionality (bit7 = TRUE) and define a modulo value for the drive (IDN103), with maximum possible size (or at least many times over one turn around (IDN79)). Moreover, you should ensure that the increments per SoftMotion-turn-around  $(dwOneTurn = fPositionPeriod * dwRatioTechUnitsDenom / iRatioTechUnitsNum)$  is either identical with IDN103 or at least two (or more) times less than IDN103.

### 3.5 PackProfile conformity check

In addition to the offline mechanism to check, whether the configured parameters are according to an PackProfile standard (Version 1.2), there is a check for conformity done online.

This is especially necessary, if the configuration from file is used, as it cannot be checked offline, what is stored in these files.

The results of the check can be gathered with this function block:



**bEnable** : Enable the function block

**AxisGroup, Axis**: Axis group to be checked.

**bTestDone**: TRUE: start-up is done and below outputs are definite, FALSE: start-up is not completed. Therefore the results in the below outputs are not yet definite.

**bBasicA, bBasicB, bExtended**: TRUE: start-up complies to PackProfile BasicA/BasicB/Extended

**sBasicA, sBasicB, sExtended**: reasons, why start-up is not compliant to PackProfile BasicA/BasicB/Extended

## 4 Sercos Specific FBs for Accessing Drive Parameters

You can read or write all drive IDNs with the FBs of this chapter. This includes normal parameters (2Byte/4Byte) as well as lists and strings and beside the normal data value can access also other data block elements (like minimum/maximum value, name etc.).

The inputs Enable and Axis and the outputs Done, Error and ErrorID of those FBs are similar to MC\_Read/WriteParameter.

Use these FBs with care, as they may (if permanently enabled) block the standard DriveInterface acyclic communication channel!

The global variable g\_strSercosError is a string and includes a detailed error message, when reading/writing a parameter (list) via the service channel failed.

### 4.1 Parameter Identifier (S/P)

The standard (S) parameters lie between 1 and 32768 (Bit15 not set). Product specific (P) parameters are offsetted with 32768 (Bit 15 set).

The parameter sets (digit after S or P) determine Bit14-12.

The data block number (last (main) number) is stored in Bit11-0

example: S-0-0100 = wIDN 100 ;      S-2-0001 = wIDN 8193 ;      P-0-1000 = wIDN 33768

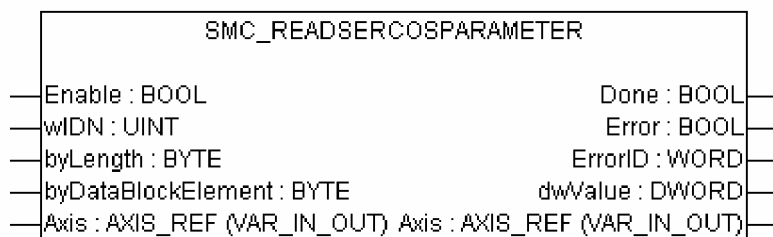
### 4.2 Reading/Writing Normal Parameters with a Standard Length

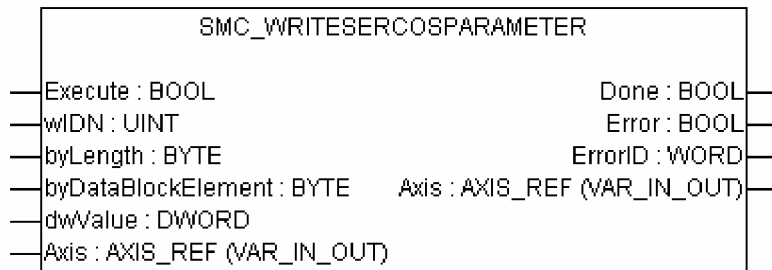
For reading normal parameters you have to specify the IDN, the length of this parameter (in Bytes; typically 2 or 4), and finally the data block element you want to see.

The data block element is specified according to Sercos specification:

byDataBlockElement	data block element
0 or 7	data value (default)
1	data state
2	name
3	attribute
4	unit
5	minimum
6	maximum

Be aware that not all data block elements consist of normal parameters (e.g. string) and therefore must be handled with the string or list reading FBs.

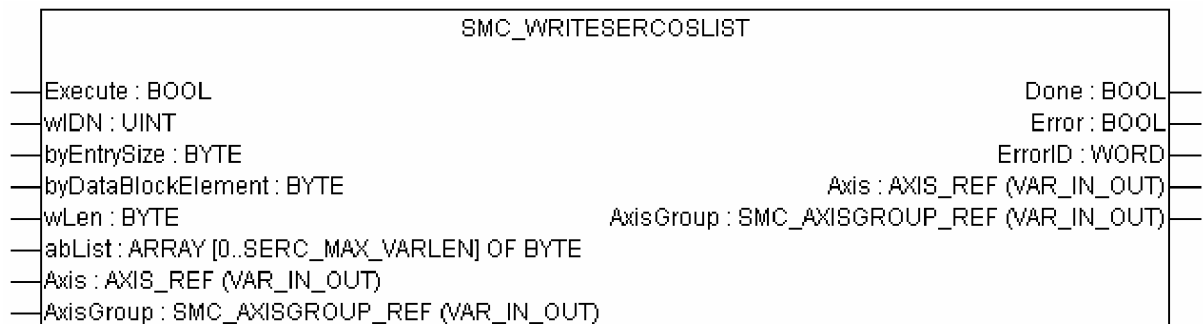
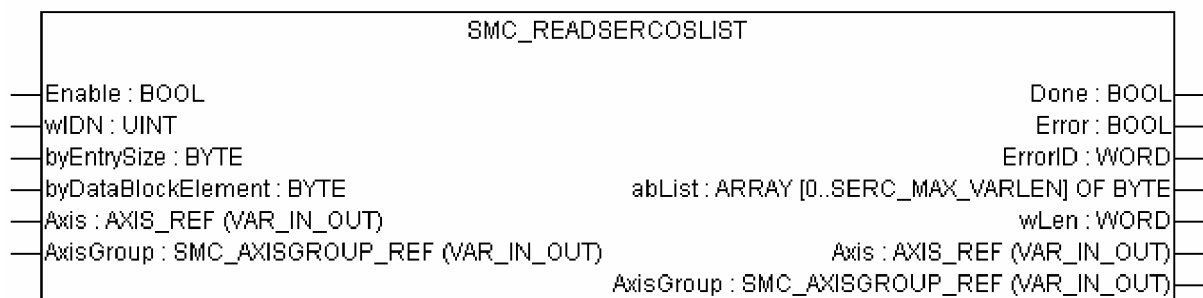




### 4.3 Reading/Writing List Parameters

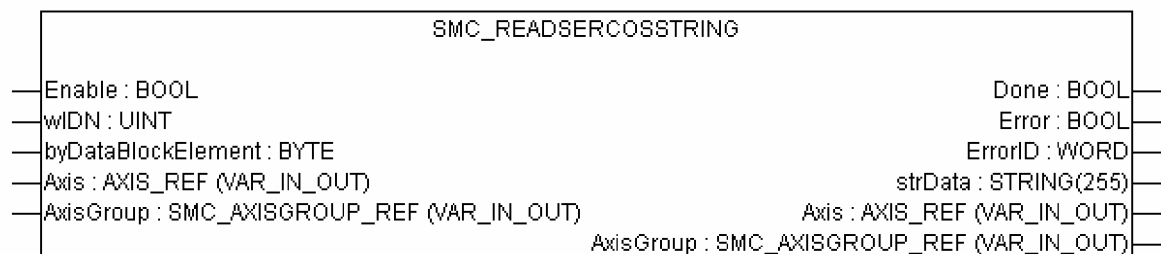
In opposite to the above FBs those for reading/writing lists need beside a reference to the axis also the AxisGroup. Moreover, the entry size (size of one list element in bytes) must be specified.

Be aware that the number of elements of a list to be read or written are limited to 4000.



### 4.4 Reading String Parameters

This FB reads a string parameter and works similar to SMC\_ReadSercosList. The string size is limited to 255 characters.



## 5 Sercos I/Os

Sercos I/Os can be controlled with this driver by setting the variable bIsIO during the first IEC cycle. The I/O data IDNs should be transferred as cyclic data; therefore configure it in the arrays awAddRealtimeSendIDN, abyAddRealtimeSendLength, adwAddRealtimeSendData and awAddRealtimeReceiveIDN, abyAddRealtimeReceiveLength, adwAddRealtimeReceiveData.

Example:

```
*****)
(* this must be set once during the first cycle *)
IF NOT bInit THEN
    IO_MS.bIsIO := TRUE;
    IO_MS.abyAddRealtimeSendLength[0] := 2;
    IO_MS.awAddRealtimeSendIDN[0] := 16#2000;
    IO_MS.abyAddRealtimeReceiveLength[0] := 2;
    IO_MS.awAddRealtimeReceiveIDN[0] := 16#2000 + 10;
    bInit := FALSE;
END_IF
(*****)

(* use a counter for test *)
byOut := byOut+1;
(* write output data in the real-time data structure *)
IO_MS.adwAddRealtimeSendData[0] := byOut;

(* read input data from real-time data structure *)
byIn := DWORD_TO_BYTE(IO_MS.adwAddRealtimeReceiveData[0]);
```

## **6 Ways of Synchronizing PLC and Sercos Hardware**

(A) If your PLC supports an externally triggered Sercos interrupt task:

- configure your Task to run externally event-triggered with the Sercos event
- don't forget to set the cycle time in your AxisGroup

(B) Else (e.g. SP RTE):

- configure your Task to run event-triggered by <AxisGroup>.bSync (CoDeSys SP RTE > V2.3.3.9) or g\_bSercosInterrupt (former RTE versions)
- don't forget to set the cycle time in your AxisGroup
- if you run very fast cycles (1ms), decrease also the cycle time of the scheduler (SP RTE: Config->Timebase)

**Change History**

<b>Version</b>	<b>Description</b>	<b>Editor</b>	<b>Date</b>
---	Issued	HP	06.10.03
---	Last update, library version 2.40	HP	20.05.05
2.00	Adaptation to new template + Start of document versioning, Release	MN	30.05.05
2.01	updated for library version 2.56	HP	06.07.05
2.10	updated for library version 1.9.0.0	HP	29.07.05
2.11	updated for library version 1.9.0.2	HP	04.01.06
2.12	updated for library version 1.9.1.0	HP	13.01.06
2.13	updated for library version 1.9.1.1	HP	09.03.06
2.14	updated for library version 1.9.1.2	HP	30.05.06
2.15	updated for library version 1.9.1.3	HP	28.07.06
2.16	updated for library version 1.9.2.0	HP	05.09.06
2.17	updated for library version 1.9.3.0	HP	16.04.07
2.17+	Spelling mistake	HP	20.06.07
2.18	updated for library version 1.9.4.0	HP	04.12.07