Subset of the

Technical Specification
PLCopen - Technical Committee 2 – Task Force

Function blocks for motion control

Version 1.1

Appendix A :

Compliance Procedure and Compliance List

July 2005

Appendix A. <u>Compliance Procedure and Compliance List</u>

Listed in this Appendix are the requirements for the compliance statement from the supplier of the Motion Control Function Blocks. The compliance statement consists of two main groups: supported data types (see Appendix A 2 Supported Data types) and supported Function Blocks, in combination with the applicable inputs and outputs (see Appendix A 3 Overview of the Function Blocks and its paragraphs). The supplier is required fill out the tables for the used data types and Function Blocks, according to their product, committing their support to the specification.

By submitting these tables to PLCopen, and after approval by PLCopen, the list will be published on the PLCopen website, www.plcopen.org , as well as a shortform overview, as specified in Appendix A 2 Supported Data types and Appendix A 3 Overview of the Function Blocks.

In addition to this approval, the supplier is granted access and usage rights of the PLCopen Motion Control logo, as described in chapter Appendix A 4 The PLCopen Motion Control Logo and Its Usage.

Data types

The data type REAL listed in the Function Blocks and parameters (e.g. for velocity, acceleration, distance, etc.) may be exchanged to SINT, INT, DINT or LREAL without to be seen as incompliant to this standard, as long as they are consistent for the whole set of Function Blocks and parameters.

Implementation allows the extension of data types as long as the basic data type is kept. For example: WORD may be changed to DWORD, but not to REAL.

Function Blocks and Inputs and Outputs

An implementation which claims compliance with this PLCopen specification shall offer a set of Function Blocks for motion control, meaning one or more Function Blocks, with at least the **basic** input and output variables, marked as "**B**" in the tables. These inputs and outputs have to be supported to be compliant.
For higher-level systems and future extensions any subset of the **extended** input and output variables, marked as "**E**" in the tables can be implemented.
Vendor specific additions are marked with "**V**", and can be listed as such in the supplier documentation.

- **Basic** input/output variables are mandatory	Marked in the tables with the letter "**B**"
- **Extended** input /output variables are optional	Marked in the tables with the letter "**E**"
- **Vendor Specific** additions	Marked in the vendor's compliance documentation with "**V**"

All the vendor specific items will not be listed in the comparison table on the PLCopen website, but in the detailed vendor specific list, which also is published.
All vendor specific in- and outputs of all FBs must be listed in the certification list of the supplier. With this, the certification listing from a supplier describes all the I/Os of the relevant FBs, including vendor-specific extensions, and thus showing the complete FBs as used by the supplier.

Appendix A 1. Statement of Supplier

Supplier name	STÖBER Antriebstechnik GmbH & Co.KG
Supplier address	Kieselbronner Straße 12
City	75177 Pforzheim
Country	Deutschland
Telephone	07231 / 582 – 0
Fax	07231 / 582 – 1000
Email address	Applcations@stoeber.de
Product Name	5th Generation of STÖBER Inverters
Product version	5.2
Release date	05 / 2006

I hereby state that the following tables as filled out and submitted do match our product as well as the accompanying user manual, as stated above.

Name of representation (person): Heiko Berner, Product Management Motion Control

Date of signature (08/09/2006):

Signature:

Appendix A 2.   Supported Data types

Defined datatypes with MC library:	Supported	If not supported, which datatype used
BOOL	Yes	
INT	Yes	
WORD	Yes	
REAL	Yes	
ENUM	Yes	

Table 1: Supported datatypes

Within the specification the following derived datatypes are defined. Which structure is used in this system:

Derived datatypes:	Where used	Supported	Which structure
Axis_Ref	Nearly all FBs	No	Not necessary because of drive based system architecture
MC_Direction (extended)	MC_MoveAbsolute MC_MoveVelocity	Yes Yes	
MC_TP_REF	MC_PositionProfile	No	
MC_TV_REF	MC_VelocityProfile	No	
MC_TA_REF	MC_AccelerationProfile	No	
MC_CAM_REF	MC_CamTableSelect	No	
MC_CAM_ID (extended)	MC_CamTableSelect MC_CamIn	No	
MC_StartMode (extended)	MC_CamIn	Yes	
MC_BufferMode	Buffered FBs	No	

Table 2: Supported derived datatypes

Appendix A 3. Overview of the Function Blocks

Single Axis Function Blocks	Supported Yes / No	Comments (<= 48 char.)
MC_MoveAbsolute	Yes	
MC_MoveRelative	Yes	
MC_MoveAdditive	Yes	
MC_MoveSuperimposed	Yes	
MC_MoveVelocity	Yes	
MC_Home	Yes	
MC_Stop	Yes	
MC_Power	No	Available as system function
MC_ReadStatus	No	Available as system function
MC_ReadAxisError	No	Available as system function
MC_Reset	Yes	
MC_ReadParameter	No	Available as system function
MC_ReadBoolParameter	No	Available as system function
MC_WriteParameter	No	Available as system function
MC_WriteBoolParameter	No	Available as system function
MC_ReadActualPosition	No	Available as system function
MC_PositionProfile	No	
MC_VelocityProfile	No	
MC_AccelerationProfile	No	
Multi-Axis Function Blocks	**Supported Yes / No**	**Comments (<= 48 char.)**
MC_CamTableSelect	No	Combined with MC_CamIn
MC_CamIn	Yes	
MC_CamOut	Yes	
MC_GearIn	Yes	
MC_GearOut	Yes	
MC_Phasing	No	

Table 3: Short overview of the Function Blocks

Appendix A 3.1 MoveAbsolute

If Supported	MC_MoveAbsolute	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
B	Position	Yes	
E	Velocity	Yes	
E	Acceleration	Yes	
E	Deceleration	Yes	
E	Jerk	Yes	Reserved for future implementation
E	Direction	Yes	
E	BufferMode	No	
V	Use Brake	Yes	
V	Step ID	Yes	
V	Execute Timestamp	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	
E	Active	Yes	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

Appendix A 3.2 MoveRelative

If Supported	MC_MoveRelative	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
B	Distance	Yes	
E	Velocity	Yes	
E	Acceleration	Yes	
E	Deceleration	Yes	
E	Jerk	Yes	Reserved for future implementation
E	BufferMode	No	
V	Use Brake	Yes	
V	Step ID	Yes	
V	Execute Timestamp	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	
E	Active	Yes	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

Appendix A 3.3 MoveAdditive

If Supported	MC_MoveAdditive	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
B	Distance	Yes	
E	Velocity	Yes	
E	Acceleration	Yes	
E	Deceleration	Yes	
E	Jerk	Yes	Reserved for future implementation
E	BufferMode	No	
V	Use Brake	Yes	
V	Step ID	Yes	
V	Execute Timestamp	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	
E	Active	Yes	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

Appendix A 3.4 MoveSuperimposed

If Supported	MC_MoveSuperimposed	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
B	Distance	Yes	
E	VelocityDiff	Yes	
E	Acceleration	Yes	
E	Deceleration	Yes	
E	Jerk	Yes	Reserved for future implementation
V	Use Brake	Yes	
V	Step ID	Yes	
V	Execute Timestamp	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	
E	Active	Yes	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

Appendix A 3.5 MoveVelocity

If Supported	MC_MoveVelocity	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
E	Velocity	Yes	
E	Acceleration	Yes	
E	Deceleration	Yes	
E	Jerk	Yes	Reserved for future implementation
E	Direction	Yes	
E	BufferMode	No	
V	Use Brake	Yes	
V	Step ID	Yes	
VAR_OUTPUT			
B	InVelocity	Yes	
E	Busy	No	
E	Active	Yes	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

Appendix A 3.6 Home

If Supported	MC_Home	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
B	Position	Yes	
E	HomingMode	No	Available as system function
E	BufferMode	No	
V	Use Brake	Yes	
V	Step ID	Yes	
V	Execute Timestamp	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	
E	Active	Yes	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

Appendix A 3.7 Stop

If Supported	**MC_Stop**	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
E	Deceleration	Yes	
E	Jerk	Yes	Reserved for future implementation
E	BufferMode	No	
V	Use Brake	Yes	
V	Step ID	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	
E	Active	Yes	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

Appendix A 3.8 Power
Not supported, available as system function

Appendix A 3.9 ReadStatus
Not supported, available as system function

Appendix A 3.10 ReadAxisError
Not supported, available as system function

Appendix A 3.11 Reset

If Supported	**MC_Reset**	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
V	Step ID	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	
B	Error	Yes	
B	ErrorID	Yes	
V	Active	Yes	

Appendix A 3.12 ReadParameter
Not supported

Appendix A 3.13 ReadBoolParameter

If Supported	MC_ReadBoolParameter	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Drive based architecture
VAR_INPUT			
B	Valid	No	
B	ParameterNumber	No	
VAR_OUTPUT			
B	Done	No	
E	Busy	No	
B	Error	No	
E	ErrorID	No	
B	Value	No	Available as system function

Name	B/E	R/W	Sup. Y/N	Comments
CommandedPosition	B	R	Yes	available as system function
SWLimitPos	E	R/W	Yes	available as system function
SWLimitNeg	E	R/W	Yes	available as system function
EnableLimitPos	E	R/W	Yes	available as system function
EnableLimitNeg	E	R/W	Yes	available as system function
EnablePosLagMonitoring	E	R/W	Yes	available as system function
MaxPositionLag	E	R/W	Yes	available as system function
MaxVelocitySystem	E	R	Yes	available as system function
MaxVelocityAppl	B	R/W	Yes	available as system function
ActualVelocity	B	R	Yes	available as system function
CommandedVelocity	B	R	Yes	available as system function
MaxAccelerationSystem	E	R	Yes	available as system function
MaxAccelerationAppl	E	R/W	Yes	available as system function
MaxDecelerationSystem	E	R	Yes	available as system function
MaxDecelerationAppl	E	R/W	Yes	available as system function
MaxJerk	E	R/W	No	available as system function

Table 4: Parameters for ReadParameter and WriteParameter

Appendix A 3.14 WriteParameter
Not supported, available as system function

Appendix A 3.15 WriteBoolParameter
Not supported, available as system function

Appendix A 3.16 ReadActualPosition
Not supported, available as system function

Appendix A 3.17 PositionProfile
Not supported, available as system function

Appendix A 3.18 VelocityProfile
Not supported, available as system function

Appendix A 3.19 AccelerationProfile
Not supported, available as system function

Appendix A 3.20 CamTableSelect

If Supported	MC_CamTableSelect	Sup. Y/N	Comments
VAR_IN_OUT			
B	Master	No	Drive based architecture
B	Slave	No	Drive based architecture
B	CamTable	Yes	
VAR_INPUT			
B	Execute	No	
E	Periodic	Yes	Combined with MC_CamIn
E	MasterAbsolute	Yes	Combined with MC_CamIn
E	SlaveAbsolute	Yes	Combined with MC_CamIn
V	Step ID	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	
B	Error	Yes	
E	ErrorID	Yes	
E	CamTableID	No	

Appendix A 3.21 CamIn

If Supported	MC_CamIn	Sup. Y/N	Comments
VAR_IN_OUT			
B	Master		Drive based architecture
B	Slave		Drive based architecture
VAR_INPUT			
B	Execute	Yes	
E	MasterOffset	Yes	
E	SlaveOffset	Yes	
E	MasterScaling	Yes	
E	SlaveScaling	Yes	
E	StartMode	Yes	
E	CamTableID	Yes	
E	BufferMode	No	
V	Step ID	Yes	
V	Execute Timestamp	Yes	
VAR_OUTPUT			
B	InSync	Yes	
E	Busy	No	
E	Active	Yes	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	
E	EndOfProfile	No	available as system function

Appendix A 3.22 CamOut

If Supported	MC_CamOut	Sup. Y/N	Comments
VAR_IN_OUT			
B	Slave	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
V	Execute Timestamp	Yes	
V	Step ID	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	

B	Error	Yes	
E	ErrorID	Yes	

Appendix A 3.23 GearIn

If Supported	MC_GearIn	Sup. Y/N	Comments
VAR_IN_OUT			
B	Master	No	Drive based architecture
B	Slave	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
B	RatioNumerator	No	System parameter, real time usability
B	RatioDenominator	No	System parameter, real time usability
E	Acceleration	Yes	
E	Deceleration	Yes	
E	Jerk	Yes	Reserved for future implementation
E	BufferMode	No	
V	Execute Timestamp	Yes	
V	Step ID	Yes	
VAR_OUTPUT			
B	InGear	Yes	
E	Busy	No	
E	Active	Yes	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

Appendix A 3.24 GearOut

If Supported	MC_GearOut	Sup. Y/N	Comments
VAR_IN_OUT			
B	Slave	No	Drive based architecture
VAR_INPUT			
B	Execute	Yes	
V	Step ID	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	No	
B	Error	Yes	
E	ErrorID	Yes	
V	Active	Yes	

Appendix A 4.   The PLCopen Motion Control Logo and Its Usage

For quick identification of compliant products, PLCopen has developed a logo for the motion control Function Blocks:

Figure 1: The PLCopen Motion Control Logo

This motion control logo is owned and trademarked by PLCopen.

In order to use this logo free-of-charge, the relevant company has to fulfill all the following requirements:
1. the company has to be a voting member of PLCopen;
2. the company has to comply with the existing specification, as specified by the PLCopen Task Force Motion Control, and as published by PLCopen, and of which this statement is a part;
3. this compliance application is provided in written form by the company to PLCopen, clearly stating the applicable software package and the supporting elements of all the specified tables, as specified in the document itself;
4. in case of non-fulfillment, which has to be decided by PLCopen, the company will receive a written statement concerning this from PLCopen. The company will have a one month period to either adopt their software package in such a way that it complies, represented by the issuing of a new compliance statement, or remove all reference to the specification, including the use of the logo, from all their specification, be it technical or promotional material;
5. the logo has to be used as is - meaning the full logo. It may be altered in size providing the original scale and color setting is kept.
6. the logo has to be used in the context of Motion Control.