



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**

DISCLAIMER OF WARRANTIES

THIS DOCUMENT IS PROVIDED ON AN “AS IS” BASIS AND MAY BE SUBJECT TO FUTURE ADDITIONS, MODIFICATIONS, OR CORRECTIONS. PLCOPEN HEREBY DISCLAIMS ALL WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, FOR THIS DOCUMENT. IN NO EVENT WILL PLCOPEN BE RESPONSIBLE FOR ANY LOSS OR DAMAGE ARISING OUT OF OR RESULTING FROM ANY DEFECT, ERROR OR OMISSION IN THIS DOCUMENT OR FROM ANYONE’S USE OF OR RELIANCE ON THIS DOCUMENT.

## Appendix A. Compliance Procedure and Compliance List

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](http://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.

- <b>Basic</b> input/output variables are mandatory	Marked in the tables with the letter “ <b>B</b> ”
- <b>Extended</b> input /output variables are optional	Marked in the tables with the letter “ <b>E</b> ”
- <b>Vendor Specific</b> additions	Marked in the vendor’s compliance documentation with “ <b>V</b> ”

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	OMRON Europe B.V.
Supplier address	C/ Olesa 14-16 08027
City	Barcelona
Country	Spain
Telephone	(+34) 93.214.06.00
Fax	(+34) 93.214.06.31
Email address	<a href="mailto:Josep.lario@eu.omron.com">Josep.lario@eu.omron.com</a>
Product Name	<ul style="list-style-type: none"><li>• <b>CJ1W-NC881</b></li><li>• <b>CJ1W-NC481</b></li><li>• <b>CJ1W-NC281</b></li><li>• <b>CJ1W-NCF81</b></li><li>• <b>CJ1W-NC882</b></li><li>• <b>CJ1W-NC482</b></li></ul>
Product version	1.0
Release date	-

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): Josep Lario

Date of signature (dd/mm/yyyy): 07/04/2010



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	NO	DINT
ENUM	NO	

**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	Axis Ref is defined by the UnitNumber and the Axis Number input parameters in the FB.
MC_Direction (extended)	MC_MoveAbsolute MC_MoveVelocity	NO	
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	NO	
MC_BufferMode	Buffered FBs	NO	

**Table 2: Supported derived datatypes**

**Appendix A 3. Overview of the Function Blocks**

<b>Single Axis Function Blocks</b>	<b>Supported Yes / No</b>	<b>Comments (&lt;= 48 char.)</b>
MC_MoveAbsolute	Yes	
MC_MoveRelative	Yes	
MC_MoveAdditive	No	
MC_MoveSuperimposed	No	
MC_MoveVelocity	Yes	
MC_Home	Yes	
MC_Stop	Yes	
MC_Power	Yes	
MC_ReadStatus	Yes	
MC_ReadAxisError	No*	Vendor specific FB is used: Read Error_NCx8x203_ReadError
MC_Reset	Yes	
MC_ReadParameter	Yes	
MC_ReadBoolParameter	No*	Use MC_ReadParameter or direct memory addressing.
MC_WriteParameter	Yes	
MC_WriteBoolParameter	No*	Use MC_WriteParameter or direct memory addressing.
MC_ReadActualPosition	Yes	
MC_PositionProfile	No	
MC_VelocityProfile	No	
MC_AccelerationProfile	No	
<b>Multi-Axis Function Blocks</b>	<b>Supported Yes / No</b>	<b>Comments (&lt;= 48 char.)</b>
MC_CamTableSelect	No	
MC_CamIn	No	
MC_CamOut	No	
MC_GearIn	No	
MC_GearOut	No	
MC_Phasing	No	

**Table 3: Short overview of the Function Blocks**

### Appendix A 3.1 MoveAbsolute

If Supported	MC_MoveAbsolute	Sup. Y/N	Comments
<b>VAR_IN_OUT</b>			
B	Axis	Yes	
<b>VAR_INPUT</b>			
B	Execute	Yes	
B	Position	Yes	
E	Velocity	Yes	
E	Acceleration	Yes	
E	Deceleration	Yes	
E	Jerk	No	
E	Direction	No	Use Vendor specific FB for absolute positioning with modulo axis.
E	BufferMode	No	
<b>VAR_OUTPUT</b>			
B	Done	Yes	
E	Busy	Yes	
E	Active	No	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

### Appendix A 3.2 MoveRelative

If Supported	MC_MoveRelative	Sup. Y/N	Comments
<b>VAR_IN_OUT</b>			
B	Axis	Yes	
<b>VAR_INPUT</b>			
B	Execute	Yes	
B	Distance	Yes	
E	Velocity	Yes	
E	Acceleration	Yes	
E	Deceleration	Yes	
E	Jerk	No	
E	BufferMode	No	
<b>VAR_OUTPUT</b>			
B	Done	Yes	
E	Busy	Yes	
E	Active	No	
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		
VAR_INPUT			
B	Execute		
B	Distance		
E	Velocity		
E	Acceleration		
E	Deceleration		
E	Jerk		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

### Appendix A 3.4 MoveSuperimposed

If Supported	MC_MoveSuperimposed	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
B	Distance		
E	VelocityDiff		
E	Acceleration		
E	Deceleration		
E	Jerk		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

### Appendix A 3.5 MoveVelocity

If Supported	MC_MoveVelocity	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	Yes	
VAR_INPUT			
B	Execute	Yes	
E	Velocity	Yes	
E	Acceleration	Yes	
E	Deceleration	Yes	
E	Jerk	No	
E	Direction	No	This FB supports both positive and negative Velocity values
E	BufferMode	No	
VAR_OUTPUT			
B	InVelocity	Yes	
E	Busy	Yes	
E	Active	No	
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	Yes	
VAR_INPUT			
B	Execute	Yes	
B	Position	Yes	
E	HomingMode	No*	This is a property of the axis, and is configured by direct memory addressing or Software Tool.
E	BufferMode	No	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	Yes	
E	Active	No	
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	Yes	
VAR_INPUT			
B	Execute	Yes	
E	Deceleration	No	
E	Jerk	No	
E	BufferMode	No	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	Yes	
E	Active	No	
E	CommandAborted	Yes	
B	Error	Yes	
E	ErrorID	Yes	

### Appendix A 3.8 Power

If Supported	MC_Power	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	Yes	
VAR_INPUT			
B	Enable	Yes	
E	Enable_Positive	No	Enable is always both Positive and Negative
E	Enable_Negative	No	Enable is always both Positive and Negative
E	BufferMode	No	
VAR_OUTPUT			
B	Status	Yes	
E	Busy	No	
E	Active	No	
B	Error	Yes	
E	ErrorID	Yes	

### Appendix A 3.9 ReadStatus

If Supported	MC_ReadStatus	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	Yes	
VAR_INPUT			
B	Enable	Yes	
VAR_OUTPUT			
B	Valid	Yes	
E	Busy	No	
B	Error	Yes	
E	ErrorID	Yes	
B	Disabled	Yes	
B	Errorstop	Yes	
B	Stopping	Yes	
B	StandStill	Yes	
B	DiscreteMotion	Yes	
B	ContinuousMotion	Yes	
E	SynchronizedMotion	No	
E	Homing	Yes	
E	ConstantVelocity	Yes	
E	Accelerating	Yes	
E	Decelerating	Yes	

### Appendix A 3.10 ReadAxisError

If Supported	MC_ReadAxisError	Sup. Y/N	Comments
			<b>(Vendor Specific FB ReadError_NCx8x203_ReadError is used) This FB covers MC_ReadAxisError functionality</b>
VAR_IN_OUT			
B	Axis	Yes	
VAR_INPUT			
	Enable	Yes	
VAR_OUTPUT			
B	Valid	Yes	
E	Busy	No	
B	Error	Yes	
B	ErrorID	Yes	

### Appendix A 3.11 Reset

If Supported	MC_Reset	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	Complete Unit errors (Including all axis in the unit) are reset.
VAR_INPUT			
B	Execute	Yes	
VAR_OUTPUT			
B	Done	Yes	
E	Busy	Yes	
B	Error	Yes	
B	ErrorID	Yes	

### Appendix A 3.12 ReadParameter

If Supported	MC_ReadParameter	Sup. Y/N	Comments
<b>VAR_IN_OUT</b>			
B	Axis	No*	This function block is able to read different data, Axis parameters are a particular case. Additional input parameters are used to select Axis Parameter source.
<b>VAR_INPUT</b>			
B	Enable	No*	Execute is used instead.
B	ParameterNumber	No*	This FB is able to read several parameters in one shot. Therefore vendor specific input are used.
<b>VAR_OUTPUT</b>			
B	Valid	No*	Execute is used, in consequence "Done" output is used.
E	Busy	Yes	
B	Error	Yes	
E	ErrorID	Yes	
B	Value	No*	This FB is able to read several parameters in one shot. Therefore vendor specific outputs are used.

### Appendix A 3.13 ReadBoolParameter

If Supported	MC_ReadBoolParameter	Sup. Y/N	Comments
<b>VAR_IN_OUT</b>			
B	Axis		
<b>VAR_INPUT</b>			
B	Valid		
B	ParameterNumber		
<b>VAR_OUTPUT</b>			
B	Done		
E	Busy		
B	Error		
E	ErrorID		
B	Value		

Name	B/E	R/W	Sup. Y/N	Comments
CommandedPosition	B	R		
SWLimitPos	E	R/W		
SWLimitNeg	E	R/W		
EnableLimitPos	E	R/W		
EnableLimitNeg	E	R/W		
EnablePosLagMonitoring	E	R/W		
MaxPositionLag	E	R/W		
MaxVelocitySystem	E	R		
MaxVelocityAppl	B	R/W		
ActualVelocity	B	R		
CommandedVelocity	B	R		
MaxAccelerationSystem	E	R		
MaxAccelerationAppl	E	R/W		
MaxDecelerationSystem	E	R		
MaxDecelerationAppl	E	R/W		
MaxJerk	E	R/W		

**Table 4: Parameters for ReadParameter and WriteParameter**

### Appendix A 3.14 WriteParameter

If Supported	MC_WriteParameter	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis	No	This function block is able to write different data, Axis parameters are a particular case. Additional input parameters are used to select Axis Parameter target
VAR_INPUT			
B	Execute	Yes	
B	ParameterNumber	No*	This FB is able to write several parameters in one shot. Therefore vendor specific input are used.
B	Value	No*	This FB is able to write several parameters in one shot. Therefore vendor specific input are used.
VAR_OUTPUT			
B	Done	Yes	
E	Busy	Yes	
E	Buffered	No	
B	Error	Yes	
E	ErrorID	Yes	

### Appendix A 3.15 WriteBoolParameter

If Supported	MC_WriteBoolParameter	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
B	ParameterNumber		
B	Value		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Buffered		
B	Error		
E	ErrorID		

### Appendix A 3.16 ReadActualPosition

If Supported	MC_ReadActualPosition	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Enable		
VAR_OUTPUT			
B	Valid		
E	Busy		
B	Error		
E	ErrorID		
B	Position		

### Appendix A 3.17 PositionProfile

If Supported	MC_PositionProfile	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis		
B	TimePosition		
VAR_INPUT			
B	Execute		
B	TimeScale		
E	PositionScale		
E	Offset		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

### Appendix A 3.18 VelocityProfile

If Supported	MC_VelocityProfile	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis		
B	TimeVelocity		
VAR_INPUT			
B	Execute		
B	TimeScale		
E	VelocityScale		
E	Offset		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

### Appendix A 3.19 AccelerationProfile

If Supported	MC_AccelerationProfile	Sup. Y/N	Comments
VAR_IN_OUT			
B	Axis		
B	TimeAcceleration		
VAR_INPUT			
B	Execute		
B	TimeScale		
E	AccelerationScale		
E	Offset		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

### Appendix A 3.20 CamTableSelect

If Supported	MC_CamTableSelect	Sup. Y/N	Comments
VAR_IN_OUT			
B	Master		
B	Slave		
B	CamTable		
VAR_INPUT			
B	Execute		
E	Periodic		
E	MasterAbsolute		
E	SlaveAbsolute		
VAR_OUTPUT			
B	Done		
E	Busy		
B	Error		
E	ErrorID		
E	CamTableID		

### Appendix A 3.21 CamIn

If Supported	MC_CamIn	Sup. Y/N	Comments
VAR_IN_OUT			
B	Master		
B	Slave		
VAR_INPUT			
B	Execute		
E	MasterOffset		
E	SlaveOffset		
E	MasterScaling		
E	SlaveScaling		
E	StartMode		
E	CamTableID		
E	BufferMode		
VAR_OUTPUT			
B	InSync		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		
E	EndOfProfile		

### Appendix A 3.22 CamOut

If Supported	MC_CamOut	Sup. Y/N	Comments
VAR_IN_OUT			
B	Slave		
VAR_INPUT			
B	Execute		
VAR_OUTPUT			
B	Done		
E	Busy		
B	Error		
E	ErrorID		

### Appendix A 3.23 GearIn

If Supported	MC_GearIn	Sup. Y/N	Comments
VAR_IN_OUT			
B	Master		
B	Slave		
VAR_INPUT			
B	Execute		
B	RatioNumerator		
B	RatioDenominator		
E	Acceleration		
E	Deceleration		
E	Jerk		
E	BufferMode		
VAR_OUTPUT			
B	InGear		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

### Appendix A 3.24 GearOut

If Supported	MC_GearOut	Sup. Y/N	Comments
VAR_IN_OUT			
B	Slave		
VAR_INPUT			
B	Execute		
VAR_OUTPUT			
B	Done		
E	Busy		
B	Error		
E	ErrorID		

### Appendix A 3.25 Phasing

If Supported	MC_Phasing	Sup. Y/N	Comments
VAR_IN_OUT			
B	Master		
B	Slave		
VAR_INPUT			
B	Execute		
B	PhaseShift		
E	Velocity		
E	Acceleration		
E	Deceleration		
E	Jerk		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		



## 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.