ProfiNet / ProfiBus

Interfaces

TRANSLATION OF THE GERMAN ORIGINAL MANUAL

Version	Modification
10	New Data set for the complete axis status inquiry (chap 8.2.) Lis of Parameters

Phytron GmbH

Industriestraße 12

82194 Gröbenzell, Germany

Tel.: +49(0)8142/503-0

Fax: +49(0)8142/503-190

In this manual you will find the feature descriptions and programming a controller for positioning of a stepper motor.

This manual is supplementary to the "*phy***LOGIC**™-Command References" and "Principles of Positioning".

Every possible care has been taken to ensure the accuracy of this technical manual. All information contained in this manual is correct to the best of our knowledge and belief but cannot be guaranteed. Furthermore we reserve the right to make improvements and enhancements to the manual and / or the devices described herein without prior notification.

We appreciate suggestions and criticisms for further improvement.

Email address: doku@phytron.de

Questions about the use of the product described in the manual that you cannot find answered here, please contact your representative of phytron (http://www.phytron.eu/) in your local agencies.

1 Legal information

This manual:

Read this manual very carefully before mounting, installing and operating the device and if necessary further manuals related to this manual.

- Please pay special attention to instructions that are marked as follows:

	**DANGER – Serious injury!**	*Indicates a high risk of serious injury or death!*
	**DANGER – Serious injury from electric shock!**	*Indicates a high risk of serious injury or death from electric shock!*
	**WARNING – Serious injury possible!**	*Indicates a possible risk of serious injury or death!*
	**WARNING – Serious injury from electric shock!**	*Indicates a possible risk of serious injury or death from electric shock!*
	**CAUTION – Possible injury!**	*Indicates a possible risk of personal injury.*
	**CAUTION – Possible damage!**	*Indicates a possible risk of damage to equipment.*
	**CAUTION – Possible damage due to ESD!**	*Refers to a possible risk of equipment damage from electrostatic discharge.*
	**”Any heading“**	*Refers to an important paragraph in the manual.*

Safety Instructions

CAUTION – Possible damage!

Malfunctions are possible while programming the instruction codes – e.g. sudden running of a connected motor, braking etc.

- Please test the program flow step by step.

CAUTION – Possible damage!

For each application, the functional reliability of software products by external factors such as voltage differences or hardware failure, etc. is affected.

- To prevent damage due to system error, the user should take appropriate safety measures. These include back-up and shut-down mechanisms.

CAUTION – Possible damage!

Each end user system is customised and differs from the testing platform. Therefore the user or application designer is responsible for verifying and validating the suitability of the application.

- The suitability of the device's use must be tested and validated.

CAUTION – Possible damage!

Some modules are set to a default value on delivery. So, e.g., the motor current must be set to the corresponding value (see the motor data from the motor manufacturer). Connected components like motors can be damaged by incorrectly set values.

- Please check before starting, if the parameters are correct.

2 Contents

1 Legal information 3
2 Contents 5
3 Introduction 6
4 Declaration of Incorporation 8
5 Conditions 9
6 Configuration of the *phy*MOTION™ via SIMATIC Manager (Example) 10
7 Download from the ftp Server 11
7.1 Firmware Download 11
7.2 Download of *phy*LOGIC™ Programs (Scripts) 12
8 Interface Assignment 13
8.1 Master Unit (CPU) 13
8.1.1 Data Set (Commands) (8 bytes) 13
8.1.2 Status Code (CPU) 15
8.1.3 Error Code (CPU) 15
8.2 Axes Modules I1AM01 and I4XM01 16
8.2.1 Status Code (Axes) 21
8.2.2 Error Code (Axes) 22
8.3 Digital I/O Modules (DIOM) 23
8.3.1 Status Code (DIOM) 24
8.3.2 Error Code (DIOM) 24
8.4 Analogue Output Module (AIOM or AOM) 25
8.5 Analogue Input Module (AIOM or AIM) 26
8.5.1 Status Code (AIOM) 27
8.5.2 Error Code AIOM 27
9 List of Parameters 28
10 Warranty, Disclaimer and Registered Trademarks 38
10.1 Disclaimer 38
10.2 Warranty 38
10.3 Registered Trademarks 38
11 Index 39

3 Introduction

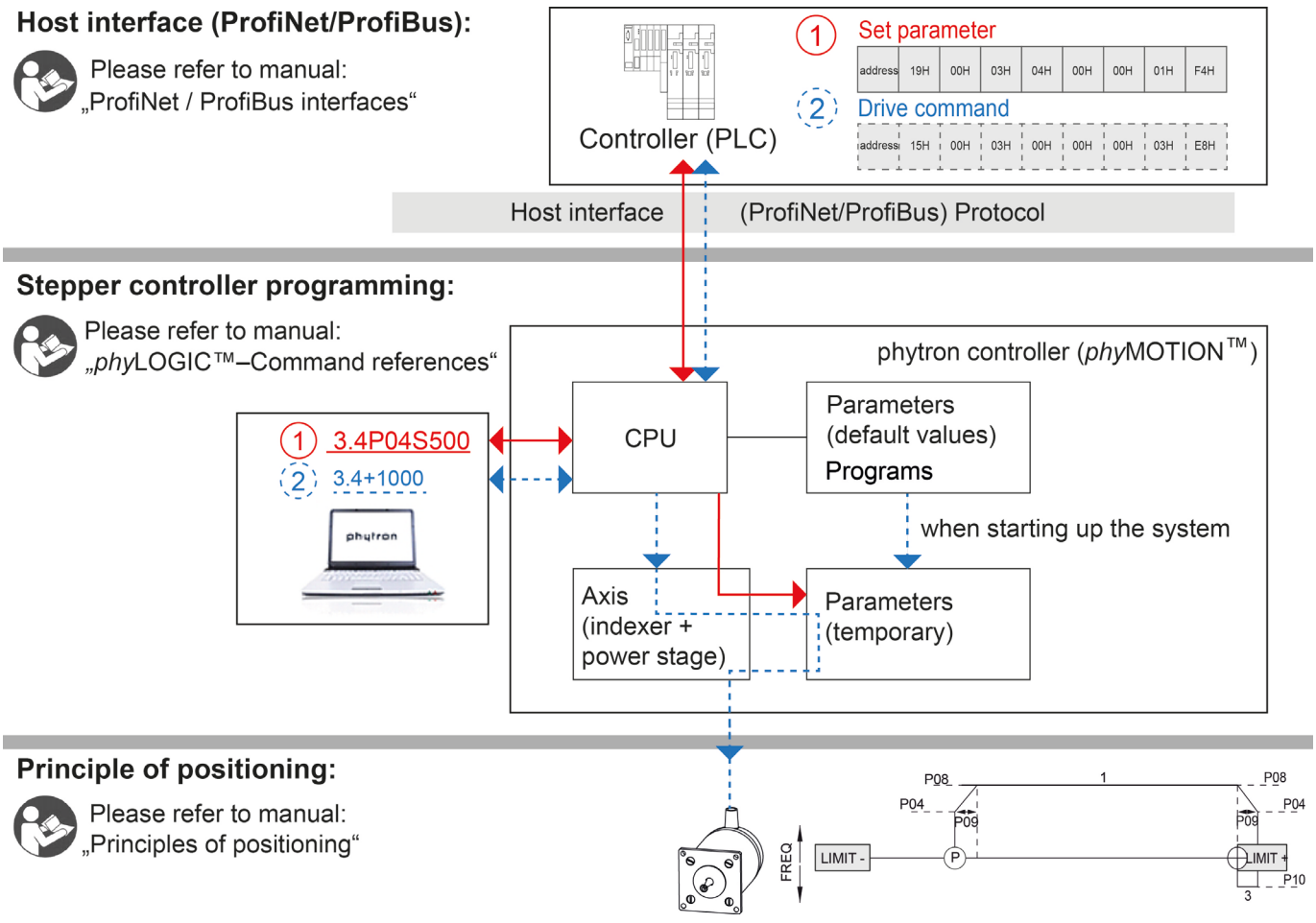
*phy***LOGIC**™ is the programming language to communicate with phytron programmable logic controllers like the MCC-Series or our *phy***MOTION**™.

*phy***LOGIC**™ commands can easily be sent to the controller with phytron's programming software (*phy***LOGIC**™ Toolbox) via USB, embedded into other protocols like Ethernet or into interface protocols like ProfiBus / ProfiNet.

You can parameterise your commands (e.g. a driving command) per axis either just the first time you set up your system, or adjust the parameters temporarily before sending a driving command.

Example: For "relative run" you can set: step resolution (P45), run current (P41), run frequency (P14), start stop frequency (P04), ramp (P15), recovery time (P16), boost (P17), boost current (P42), current hold time (P43), etc.

Use this illustration to find the adequate manual for your programming task:



Each of our programmable controllers comes along with pre set parameters (default values), which are automatically loaded into the temporary memory of each axis while starting the device. These parameters can be changed during your program is executed to optimise your motion tasks at any time.

If you want your controller to wake up with a new set of parameters, you have to explicitly store them in the non volatile storage of the main CPU unit by using a certain command.

4 Declaration of Incorporation

phytron
Beyond Steppers

Declaration of Conformity

according to EC directive 2004/108/EC (EMC-Directive**)**

Name and address of the manufacturer:
Phytron-Elektronik GmbH,
Industriestr. 12
82194 Gröbenzell

We declare that the following product is in conformity with the EC Directives 2004/108/EC relating to EMC.

Product denomination

Part-No.	Title	
10015035	MCM01.1	Main Controller Module
10015036	CANS01.1	CAN Communication Sub Module
10015037	ETHS01.1	Ethernet Communication Sub Module
10015039	PBS01.1	Profibus Communication Sub Module
10015040	PNS01.1	ProfiNet Communication Sub Module
10015041	RSS01.1	RS485/RS232 Communication Sub Module

From serial number 1205xxxxx

Applied harmonized standards

- EN 61000-6-1: 2007-01 Electromagnetic Compatibility (EMC) - Immunity for residential, commercial and light-industrial environmental
- EN 61000-6-2: 2005-08 Electromagnetic compatibility (EMC) - Immunity for industrial environments
- EN 61000-6-3: 2007-01 Electromagnetic compatibility (EMC) - Emission standard for residential, commercial and light-industrial environments
- EN 61000-6-4: 2007-01 Electromagnetic compatibility (EMC) - Emission standard for industrial environments

Comment:
This declaration of conformity is valid only if the device is built in a suitable casing e.g. phyMOTION-6SL-MR-s.

Gröbenzell, 2012-05-10

Johannes Schmid
Technical Director

AP QS-0672-3
CE 7035 Rev. 1

Phytron-Elektronik GmbH
Industriestr. 12 - 82194 Grobenzell
Postfach 1255 - 82180 Grobenzell
T +49-8142-503-0 F +49-8142-503-190
E info@phytron.de W www.phytron.de

Geschäftsführung Birgit Hartmann
Reg.-Gericht München - HRB 44 426
USt.-Ident.-Nr. DE 128 242 222
Steuernummer 117-135-10027

Genossenschaftsbank - Kto. 96610 - BLZ 70169464
IBAN DE67701694640000096610 - BIC GENODEF1M07
Sparkasse Fürstenfeldbruck - Kto. 1801265 - BLZ 70053070
Oberbank München - Kto. 1041021021 - BLZ 70120700
Volksbank Fürstenfeldbruck - Kto. 712531 - BLZ 70163370
Postbank München - Kto. 0286001800 - BLZ 70010080

5 Conditions

- You have an S7 station, consisting of a power supply module and a CPU and configured.
- The PG is connected to the PROFINET I/O.
- The *phy***MOTION**™ controller is connected to the superior main station via ProfiNet interface.

6 Configuration of the *phy*MOTION™ via SIMATIC Manager (Example)

- Install the GSD data file which comes with your delivery on your PC.
- Start the SIMATIC manager and open the project that you have created.
- Add the *phy*MOTION™ rack from the hardware catalogue (HW Config) with drag & drop.
- Drag the individual modules according to the module assembly in the housing (from left to right) from the hardware catalogue *phy*MOTION™.
- **Important:** The POWM01 and POWM02 modules are ignored because they are not addressed!

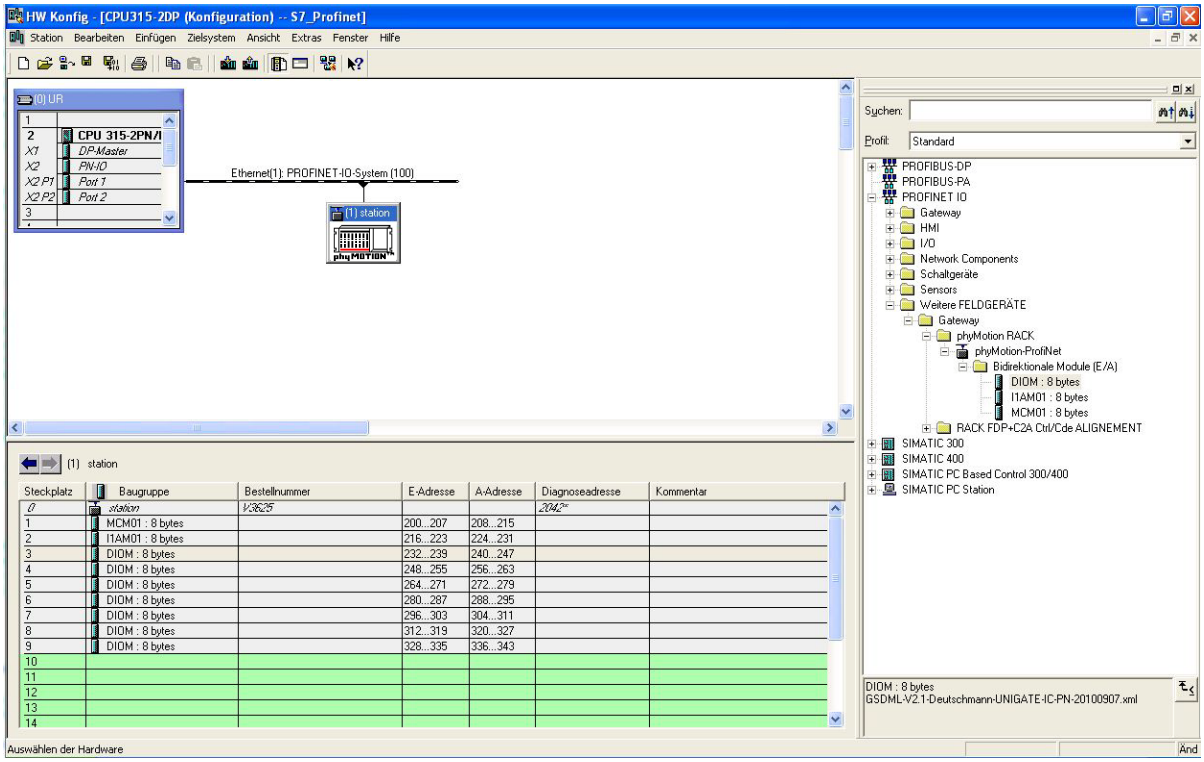


Fig.1 Configuration of the *phy*MOTION™ with HW Config

Save and compile the hardware configuration with s*tation* > *save and compile*.

7 Download from the ftp Server

The phytron PNS01 sub module contains an internal ftp server. This ftp server can be used for updating the firmware of every active *phy***MOTION**™ module like MCM01, MCM02, I1AM01, DIOM01, etc.

You can also transmit *phy***LOGIC**™ programs (scripts) via the ftp server to the *phy***MOTION**™.

7.1 Firmware Download

The firmware of the modules is developed and delivered by phytron.

To update the firmware of the individual modules proceed as follows:

1. Open the Windows Explorer and enter in the command line:
   //ftp <IP-address of the *phy***MOTION**™>
   The IP address is assigned by the PLC after the initialisation.
2. Enter in the ftp client window:
   User name: Phytron
   Password: Phytron
   Confirm with the button LOGIN.
3. A window with a list of files stored on the ftp server is opened, e.g.

   MCM0x.phyMotion: Firmware file for the master (MCM0x)

   I1AM01.phyMotion: Firmware file for the 1 axis drive (I1AM01)

   DIOM01.phyMotion: Firmware file for the digital I/O module (DIOM01)

4. To a file, it has to be highlighted. Then press the DELETE button.
5. Open the folder containing the new firmware in a second window.

   Save the new firmware files by drag and drop on the ftp server.

   IMPORTANT: The firmware files must be deleted before a restoring, because they can't be overwritten on the ftp server!

   If a new firmware is loaded to the ftp server, the "new" firmware is always transmitted to all modules of the same type.
6. First, after a restart or reset the firmware version no. on the ftp server is compared with the version no of the MCM module.
   If they differ, the firmware of the MCM module is automatically updated with the firmware of the ftp server.
7. If they don't differ, no firmware will be updated. The files remain even after successful ftp download on the ftp server.

Leave the current files on the ftp server, so a module recently inserted into the *phy***MOTION**TM is automatically updated to the latest firmware status, which is stored on the ftp server.

IMPORTANT: It is only checked whether the update version of the ftp server differs from the version of the module. Therefore, you can set the firmware back to an older version by replacing the file on the ftp server with the older firmware file.

7.2 Download of *phy*LOGIC™ Programs (Scripts)

- Write a script in *phy***LOGIC**TM, then save it with the name <no>.mpr. There are possible more than 1 to 255 as <nr> program name.
- Open the Windows Explorer and enter in the command line:
  //ftp <IP address of the *phy***MOTION**TM>
  The IP address is assigned by the PLC after the initialisation.
- Enter in the ftp client window:
  User name: Phytron
  Password: Phytron
  Confirm with the button LOGIN.
- Open the folder in a second window where the sequential programs for the *phy***MOTION**TM are stored, e.g.:

  1.mpr sequential program 1 for stepper motor controller

  2.mpr sequential program 2 for stepper motor controller

  255.mpr sequential program 255 for stepper motor controller

- Save the required sequential programs by drag and drop on the ftp server
- After restart or reset all „mpr“-files are transmitted from the ftp server to the MCM module and then are automatically deleted on the ftp server.

8 Interface Assignment

*phy***LOGIC**™ commands usually contain the module address (e.g. when you are using *phy***LOGIC**™ Toolbox).
As you can see in the screenshot of the HW Config (Fig.1) every *phy***MOTION**™ module card already is addressed in the Profinet system. Once addressed commands can be sent to every single module card. In the following section the instruction set per module type is given referring to the *phy***LOGIC**™ command reference manual

8.1 Master Unit (CPU)

8.1.1 Data Set (Commands) (8 bytes)

	Function of commands (8 bytes)	**Byte**								**According to *phy*LOGIC™ command (refer to the command reference manual)**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**command**	**chap.**
Send	Read CPU status	00_H	00_H	00_H	00_H	00_H	00_H	00_H	00_H	ST	6.17
	Status and Error Reset	00_H	01_H	00_H	00_H	00_H	00_H	00_H	00_H	STC	6.17
	Controller Reset	01_H	00_H	00_H	00_H	00_H	00_H	00_H	00_H	CR	6.3
	Save controller's parameters	02_H	01_H	00_H	00_H	00_H	00_H	00_H	00_H	SA	6.14
	Terminates the save of the controller's parameters[1)]	02_H	00_H	00_H	00_H	00_H	00_H	00_H	00_H	–	
	Start program script	03_H	01_H	00_H	00_H	00_H	00_H	00_H	XX_H[2)]	QPname A	6.14
	Stop program script	03_H	00_H	00_H	00_H	00_H	00_H	00_H	00_H	QPE	6.14
	Emergency stop of all axes and outputs are set to zero	04_H	00_H	00_H	00_H	00_H	00_H	00_H	00_H	–	
	Read register										
	signed long	40_H	10_H	Register no		00_H	00_H	00_H	00_H	RnnR	6.15
	float	40_H	40_H	Register no		00_H	00_H	00_H	00_H	RnnR	6.15

	Function of commands (8 bytes)	Byte								According to *phy*LOGIC™ command (refer to the command reference manual)	
		1	2	3	4	5	6	7	8	command	chap.
	Write register										
	signed long	41_H	10_H	Register no		Register value				Rnn Svalue	6.15
	float	41_H	40_H	Register no		Register value				RnnS value	6.15

1) The command 0201_H sets the flag ‚Save parameters'. This flag must be reset before resaving the parameters. The command 0200_H resets this flag.

2) [2)] XX_H has to be replaced by the script number which should be started. It must be registered in hexadecimal code. E.g. code $0A_H$ starts script no. 10.mpr (see chap. 6.3).

Function		Byte							
		1	2	3	4	5	6	7	8
Response	Controller Reset status	00_H	Error code	Status		00_H	00_H	00_H	00_H
	Read master status	01_H	Error code	Status		00_H	00_H	00_H	00_H
	Save controller's parameters	02_H	Error code	Status		00_H	00_H	00_H	00_H
	Start / stop program	03_H	Error code	Status		00_H	00_H	00_H	00_H
	Emergency stop program	04_H	Error code	Status		00_H	00_H	00_H	00_H
	Read register								
	signed long	40_H	10_H	Register no		Register value			
	float	40_H	40_H	Register no		Register value			
	Write register								
	signed long	41_H	10_H	Register no		Register value			
	float	41_H	40_H	Register no		Register value			

8.1.2 Status Code (CPU)

Status (2 bytes)	Meaning
0001_H	Command error
0002_H	Range value error
0004_H	Checksum error (CRC)
0008_H	ADDR error (card could not be addressed)
0010_H	Timeout error on bus
0020_H	Bad value error
0040_H	Interface error (frame)
0080_H	Software error
0100_H	Internal program is running
0200_H	Forced switch over to remote via PC
1000_H	Programming error: internal program
2000_H	Flag ‚Parameters are changed‘; this flag is reset by the command $0201_{H.}$
4000_H	Input inquiry active (wait for input status)
8000_H	Remote/Local switch on Remote

8.1.3 Error Code (CPU)

Error (1 byte)	Meaning
00_H	OK
$0m_H$	Error in module no. m m: module number 1 to 14_H (module 1 to 20 in one sub rack)

8.2 Axes Modules I1AM01 and I4XM01

Important: Byte ‘3’ (axis no.) differs for the two axis modules:

I1AM01: axis no. = 00_H
I4XM01: axis no. = 00_H to 03_H

	Function	Byte 1	2	3	4	5	6	7	8	Acc. to *phy*LOGIC™ command (refer to the command reference manual) command	chap.
Send	Axis status										
	Read	10_H	00_H	Axis no.	00_H	00_H	00_H	00_H	00_H	SEm.a	6.17
	Error reset	10_H	01_H	Axis no.	00_H	00_H	00_H	00_H	00_H	n/a	
	Read the complete status	10_H	02_H	Axis no.	00_H	00_H	00_H	00_H	00_H	SEm.a	6.17
	Axis instruction Reset	11_H	00_H	Axis no.	00_H	00_H	00_H	00_H	00_H	–	
	Axis stop										
	normal	12_H	00_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aS	6.25
	with emergency stop ramp	12_H	01_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aSN	6.25
	Reference run										
	– direction	13_H	00_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR–	6.25
	+ direction	13_H	01_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR+	6.25
	CENTER switch via – direction	13_H	02_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR–C	6.25
	CENTER switch via + direction	13_H	03_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR+C	6.25
	– direction with encoder track zero	13_H	04_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR–^I	6.25
	+ direction with encoder track zero	13_H	05_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR+^I	6.25

	Function	Byte 1	2	3	4	5	6	7	8	Acc. to *phy*LOGIC™ command (refer to the command reference manual) command	chap.
	CENTER switch via – direction with encoder track zero	13_H	06_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR–C^I	6.25
Send	CENTER switch via + direction with encoder track zero	13_H	07_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR+C^I	6.25
	only pulse zero in – direction	13_H	08_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR–I	6.25
	only pulse zero in + direction	13_H	09_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aR+I	6.25
	Centre to initiator OFF	13_H	$0A_H$	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aRC-	6.25
	Centre to initiator ON	13_H	$0B_H$	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aRC+	6.25
	Centre to initiator OFF with encoder track zero	13_H	$0C_H$	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aRC-^I	6.25
	Centre to initiator ON with encoder track zero	13_H	$0D_H$	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aRC+^I	6.25
	Centre only in + direction	13_H	$0E_H$	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aRCW	6.25
	Centre only in – direction	13_H	$0F_H$	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aRCCW	6.25
	Free running – direction	14_H	00_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aL– (m.aLr)	6.25
	Free running + direction	14_H	01_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aL+ (m.aLr)	6.25

	Function	Byte 1	2	3	4	5	6	7	8	Acc. to *phy*LOGIC™ command (refer to the command reference manual) command	chap.
	Relative Positioning with parameter P14	15_H	10_H	Axis no.	00_H	distance signed long				m.a*rvalue*	6.25
	P14	15_H	40_H	Axis no.	00_H	distance float				m.a*rvalue*	6.25
	P4	15_H	11_H	Axis no.	00_H	distance signed long				m.a*rvalue*	6.25
	P4	15_H	41_H	Axis no.	00_H	distance float				m.a*rvalue*	6.25
Send	Absolute Positioning with parameter P14	16_H	10_H	Axis no.	00_H	position signed long				m.aArvalue	6.25
	P14	16_H	40_H	Axis no.	00_H	position float				m.aArvalue	6.25
	P4	16_H	11_H	Axis no.	00_H	position signed long				m.aArvalue	6.25
	P4	16_H	41_H	Axis no.	00_H	position float				m.aArvalue	6.25
	Deactivate axis	17_H	00_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aMD	6.25
	Activate axis	17_H	01_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aMA	6.25
	Reset power stage	18_H	00_H	Axis no.	00_H	00_H	00_H	00_H	00_H	m.aC	6.25

Function	Byte 1	2	3	4	5	6	7	8	Acc. to *phy*LOGIC™ command (refer to the command reference manual) command	chap.
Write parameters										
signed long	19_H	10_H	Axis no.	Parameter no.	Parameter value				m.aPmm Svalue	6.25
float	19_H	40_H	Axis no.	Parameter no.	Parameter value				m.aPmm Svalue	6.25
Read parameters										
signed long	$1A_H$	10_H	Axis no.	Parameter no.	00_H	00_H	00_H	00_H	m.aPmmR	6.25
float	$1A_H$	40_H	Axis no.	Parameter no.	00_H	00_H	00_H	00_H	m.aPmmR	6.25

	Function	Byte 1	2	3	4	5	6	7	8
Response	Read axis status	10_H	Error code	Status		Current position			
	Read the complete axis status	10_H	Error code	Status		Status 32 bit			
	Axis instruction Reset	11_H	Error code	Status		Current position			
	Axis stop	12_H	Error code	Status		Current position			
	Reference running	13_H	Error code	Status		Current position			
	Free running	14_H	Error code	Status		Current position			
	Relative Positioning	15_H	Error code	Status		Current position			
	Absolute Positioning	16_H	Error code	Status		Current position			
	Deactivate axis/ Activate axis	17_H	Error code	Status		Current position			
	Reset power stage	18_H	Error code	Status		Current position			
	Write parameters	19_H	Data type	Axis no.	Parameter no.	Parameter value			
	Read parameters	$1A_H$	Data type	Axis no.	Parameter no.	Parameter value			

8.2.1 Status Code (Axes)

Status (2 bytes)	Meaning
0001_H	Busy (motor is running)
0002_H	Not now (command cannot be executed because the motor is running)
0004_H	Wait for SYNC (only ProfiNet)
0008_H	Reference run successful
0010_H	Limit switch '+' is active
0020_H	Limit switch '–' is active
0040_H	Limit switch 'center' is active
0080_H	Software switch '+' responded
0100_H	Software switch '–' responded
0200_H	Power stage is busy
0400_H	Axis is in the ramp
0800_H	Internal error
1000_H	Limit switch error while positioning
2000_H	Power stage error (short circuit, under voltage)
4000_H	SFI error
8000_H	Encoder error
10000_H	Axis is running
20000_H	Axis is in recovery time (s. parameter P13 or P16)
40000_H	Axis is in stop current delay time (parameter P43)
80000_H	Axis is in position
100000_H	Axis APS is ready
200000_H	Axis is positioning mode

Status (2 bytes)	Meaning
400000_H	Axis is in free running mode
800000_H	Axis multi F run
1000000_H	Axis SYNC allowed

8.2.2 Error Code (Axes)

Error (1 byte)	Meaning
00_H	OK
01_H	Data error
02_H	Motor temperature warning
04_H	Motor shut-off temperature is reached
08_H	Power stage temperature > 85°C
10_H	Error limit switch
20_H	Error power stage
40_H	SFI step failure
80_H	Error encoder

8.3 Digital I/O Modules (DIOM)

Function		Byte								Acc. to *phy*LOGIC™ command (refer to the command reference manual)	
		1	2	3	4	5	6	7	8	command	chap.
Send	Read I/O status	20_H	00_H	00_H	00_H	00_H	00_H	00_H	00_H	**AZ**n.a **AG**1R **EZ**n.a **EG**1R	6.3 6.3 6.5 6.5
	Reset I/O Status	20_H	01_H	00_H	00_H	00_H	00_H	00_H	00_H	**SI0**m**C**	6.20
	Read input status and set output status	21_H	00_H	00_H	00_H	00_H	00_H	Output		-	-

Function		Byte							
		1	2	3	4	5	6	7	8
Response	Read I/O status	20_H	Error code	Sta tus	00_H	Input		Output	
	Read input status and set output status	21_H	Error code	Sta tus	00_H	Input		Output	

8.3.1 Status Code (DIOM)

Status (1 byte)	Meaning
01_H	Error
10_H	Interrupt 1
20_H	Interrupt 2

8.3.2 Error Code (DIOM)

Error (1 byte)	Meaning
00_H	OK
01_H	unknown Instruction
02_H	incorrect instruction
04_H	incorrect data
10_H	Short circuit at output
20_H	24 V electrically isolated is missed (no 24 V applied)

8.4 Analogue Output Module (AIOM or AOM)

Function		Byte								Acc. to *phy*LOGIC™ command (refer to the command reference manual)	
		1	2	3	4	5	6	7	8	Command	chap.
Send	Read analogue output channel	50_H	00_H	00_H	channel no	00_H	00_H	00_H	00_H	**AZ**n.a **AG**1R **EZ**n.a **EG**1R	6.2 6.2 6.5 6.5
	Analogue I/O error Reset	50_H	01_H	00_H	00_H	00_H	00_H	00_H	00_H	**SI0**m**C**	6.15
	Set output channel	51_H	00_H	00_H	channel no.	00_H	00_H	Analogue value		-	-
	Read output channel configuration	52_H	00_H	00_H	channel no	00_H	00_H	00_H	00_H		
	Write output channel configuration	53_H	00_H	00_H	channel no	00_H	00_H	00_H	Function		

Function		Byte							
		1	2	3	4	5	6	7	8
Response	Read analogue output channel	50_H	Error code	Status	channel no	00_H	00_H	Analogue value	
	Set output channel	51_H	Error code	Status	channel no	00_H	00_H	Analogue value	
	Read output channel configuration	52_H	Error code	Status	channel no	00_H	00_H	00_H	Function
	Write output channel configuration	53_H	Error code	Status	channel no.	00_H	00_H	00_H	Function

8.5 Analogue Input Module (AIOM or AIM)

Function		Byte								Acc. to *phy*LOGIC™ command (refer to the command reference manual)	
		1	2	3	4	5	6	7	8	Command	chap.
Send	Read analogue input channel	60_H	00_H	00_H	Channel no	00_H	00_H	00_H	00_H	**AZ**n.a **AG**1R **EZ**n.a **EG**1R	6.2 6.2 6.5 6.5
	Analogue I/O error Reset	60_H	01_H	00_H	00_H	00_H	00_H	00_H	00_H	**SI0**m**C**	6.15
	Read input channel configuration	62_H	00_H	00_H	Channel no	00_H	00_H	00_H	00_H		
	Write input channel configuration	63_H	00_H	00_H	Channel no	00_H	00_H	00_H	Function		

Function		Byte							
		1	2	3	4	5	6	7	8
Response	Read analogue input channel	60_H	Error code	I/O status	Channel no	00_H	00_H	Analogue value	
	Read input channel configuration	62_H	Error code	I/O status	Channel no	00_H	00_H	00_H	Function
	Write input channel configuration	63_H	Error code	I/O status	Channel no	00_H	00_H	00_H	Function

8.5.1 Status Code (AIOM)

Status (1 byte)	Meaning
01_H	Error
10_H	Interrupt 1
20_H	Interrupt 2

8.5.2 Error Code AIOM

Error (1 byte)	Meaning
00_H	Module OK
01_H	unknown Instruction
02_H	incorrect instruction
04_H	incorrect data
10_H	DAC error
20_H	24 V electrically isolated is missed (no 24 V applied)

9 List of Parameters

No.	Meaning	Default
P01	Type of movement (free run, relative / absolute, reference run) 0 = Rotational movement (ignoring limit switches) 1 = Hardware limit switches are monitored for XY tables or other linear systems, 2 limit switches: Mechanical zero and limit direction – Limit direction + 2 = Software limit switches are monitored 3 = Hardware and software limit switches are monitored	0
P02	Measuring units of movement: only used for displaying 1 = step 2 = mm 3 = inch 4 = degree	1
P03	Conversion factor for the thread 1 step corresponds to ... If P03 = 1 (steps) the conversion factor is 1. Computing the conversion factor: $Conversion\,factor = \frac{Thread}{Number\,of\,steps\,per\,revolution}$ Example: 4 mm thread pitch 200-step motor = 400 steps/rev. in the half step mode $Conversion\,factor = \frac{4}{400} = 0.01$	1
P04	Start/stop frequency The start/stop frequency is the maximum frequency to start or stop the motor without ramp. At higher frequencies, step losses or motor stop would be the result of a start or stop without ramp. The start/stop frequency depends on various factors: type of motor, load, mechanical system, power stage. The frequency is programmed in Hz.	400

No.	Meaning	Default
P05 **P06**	not used	
P07	Emergency stop ramp Input for I1AM0x: in 4000 Hz/s steps I4XM01: in 1 Hz/s steps	100 000
P08	f_{max} MØP (mechanical zero point) Run frequency during initializing (referencing) Enter in Hz (integer value) I1AM0x: 40 000 maximum I4XM01: 4 000 000 maximum	4000
P09	Ramp MØP Ramp during initializing, associated to parameter P08 Input for I1AM0x: in 4000 Hz/s steps I4XM01: in 1 Hz/s steps	4000
P10	f_{min} MØP Run frequency for leaving the limit switch range Enter in Hz	400
P11	MØP offset for limit switch direction + (away from “LIMIT+” switch, towards “LIMIT–” switch) Distance between reference point MØP and limit switch activation Unit: is defined in parameter P02 P11>=0	0
P12	MØP offset for limit switch direction – (away from “LIMIT–” switch, towards “LIMIT+” switch) Distance between reference point MØP and limit switch activation Unit: is defined in parameter P02 P12>=0	0

No.	Meaning	Default
P13	Recovery time MØP Time lapse during initialization Enter in msec	20
P14	f_{max} Run frequency during program operation Enter in Hz (integer value) I1AM0x: 40 000 maximum I4XM01: 4 000 000 maximum	4000
P15	Ramp for run frequency (P14) Input for I1AM0x: in 4000 Hz/s steps I4XM01: in 1 Hz/s steps	4000
P16	Recovery time position Time lapse after positioning Input in msec	20
P17	Boost (current is defined in P42) 0 = off 1 = on during motor run 2 = on during acceleration and deceleration ramp <u>Remarks:</u> The boost current is set in parameter P42 for internal power stages. You can select with parameter P17 in which situation the controller switches to boost current. P17 = 1 means, the boost current always is switched on during motor run. During motor standstill the controller switches to stop current.	0
P18	Internally used for linear interpolation	
P19	Encoder deviation MØP counter	
P20	Mechanical zero counter This counter contains the number of steps referred to the mechanical zero (MØP). If the axis reaches the MØP, P20 will be set to zero.	0

No.	Meaning	Default
P21	Absolute counter Encoder, multi turn and also for single turn. The value of P22 is extended to P21 by software. The encoder counters have a fixed resolution, e.g. 10 bit (for single-turn encoders: the resolution is bits per turn), then the read value repeats. A saw tooth profile of the the numerical values is produced during a continuous motor running. This course is "straightened" by software. P20 and P21 will be scaled to the same value per revolution by P3 and P39 and are therefore directly comparable, see P36.	0
P22	Encoder counter Indicates the true absolute encoder position. Is only set for A/B encoders to zero (after reset), the absolute encoder remains the value.	0
P23	Software Limit Switch (Axial limitation pos. direction +) If the distance is reached, the run in + direction is aborted. 0 = no limitation	0
P24	Software Limit Switch (Axial limitation neg. direction –) If the distance is reached, the run in – direction is aborted. 0 = no limitation	0
P25	Compensation for play Indicates the distance, the target position in the selected direction is passed over and afterwards is started in reverse direction. 0 = no compensation for play	0

<table>
<tr><th>No.</th><th>Meaning</th><th>Default</th></tr>
<tr><td>P26</td><td>The data transfer rate is set by P26 (ONLY for SSI encoder), by which the encoder is read. The transfer rate is dependent on the length of the cable by which the encoder is connected to the device. The shorter the cable, the encoder can more quickly be read.

Data transfer rate 1 to 10 (= 100 to 1000 kHz)
1 = 100 kHz
2 = 200 kHz
3 = 300 kHz
4 = 400 kHz
5 = 500 kHz
6 = 600 kHz
7 = 700 kHz
8 = 800 kHz
9 = 900 kHz
10 = 1000 kHz</td><td>1</td></tr>
<tr><td>P27</td><td>Limit switch type
NCC: normally closed contact
NOC: normally open contact

<table><tr><th></th><th>LIMIT–</th><th>Center/Ref</th><th>LIMIT+</th></tr><tr><td>0</td><td>NCC</td><td>NCC</td><td>NCC</td></tr><tr><td>1</td><td>NCC</td><td>NCC</td><td>NOC</td></tr><tr><td>2</td><td>NOC</td><td>NCC</td><td>NCC</td></tr><tr><td>3</td><td>NOC</td><td>NCC</td><td>NOC</td></tr><tr><td>4</td><td>NCC</td><td>NOC</td><td>NCC</td></tr><tr><td>5</td><td>NCC</td><td>NOC</td><td>NOC</td></tr><tr><td>6</td><td>NOC</td><td>NOC</td><td>NCC</td></tr><tr><td>7</td><td>NOC</td><td>NOC</td><td>NOC</td></tr></table></td><td>0</td></tr>
<tr><td>P28</td><td>Axis options
0 = Power stage is deactivated after power on
1 = Power stage is activated after power on</td><td>0</td></tr>
<tr><td colspan="3">P29 not used</td></tr>
</table>

<table>
<tr><th>No.</th><th>Meaning</th><th>Default</th></tr>
<tr><td>P30</td><td>For I4XM01 only!
Frequency band setting
0 = manual
1 = automatic
<u>Remark</u>:
It is recommended to work with the automatic setting mode.
For each run frequency (P14) and ramp (P15) the controller automatically selects suitable settings.</td><td>1</td></tr>
<tr><td>P31</td><td>For I4XM01 only!
Frequency and ramp predivider (only if P30 = 0, manual)
This parameter changes the predivider which supplies the hardware (frequency generated) with a clock of 20 MHz derived.</td><td>3</td></tr>
<tr><td></td><td>
<table>
<tr><th>P31</th><th>Run frequency</th><th>resolution</th><th>predivider</th></tr>
<tr><td>0</td><td>1 Hz … 8 kHz</td><td>⅛ Hz</td><td>2440</td></tr>
<tr><td>1</td><td>1 Hz … 16 kHz</td><td>¼ Hz</td><td>1220</td></tr>
<tr><td>2</td><td>1 Hz … 32 kHz</td><td>½ Hz</td><td>609</td></tr>
<tr><td>3</td><td>1 Hz … 65 kHz</td><td>1 Hz</td><td>304</td></tr>
<tr><td>4</td><td>2 Hz … 130 kHz</td><td>2 Hz</td><td>152</td></tr>
<tr><td>5</td><td>4 Hz … 260 kHz</td><td>4 Hz</td><td>75</td></tr>
<tr><td>6</td><td>8 Hz … 520 kHz</td><td>8 Hz</td><td>37</td></tr>
<tr><td>7</td><td>16 Hz … 1 MHz</td><td>16 Hz</td><td>18</td></tr>
<tr><td>8</td><td>32 Hz … 2 MHz</td><td>32 Hz</td><td>9</td></tr>
<tr><td>9</td><td>64 Hz … 4 MHz</td><td>64 Hz</td><td>4</td></tr>
</table>
The parameter can be used for individual settings when automatic frequency band setting for the specific application is not appropriate.
</td><td></td></tr>
<tr><td>P32</td><td>TBD</td><td>1</td></tr>
</table>

No.	Meaning	Default
P33	TBD	1
P34	Encoder type 0 = no encoder 1 = incremental 5.0 V 2 = incremental 5.5 V 3 = serial interface SSI binary Code 5.0 V 4 = serial interface SSI binary Code 5.5 V 5 = serial interface SSI Gray Code 5.0 V 6 = serial interface SSI Gray Code 5.5 V 7 = EnDat 5 V 8 = EnDat 5.5 V 9 = resolver 10 = LVDT 4-wire 11 = LVDT 5/6-wire	0
P35	Encoder resolution for SSI and EnDat encoder Enter max. encoder resolution in Bit (max. 48 Bit) Special feature EnDat: if the parameter is set to zero, the controller uses the resolution which is read from the connected instrument.	10
P36	Encoder function This parameter specifies the use of P21 as a pure counter or whether its value is continuously compared with the value of the P20 counter, if the counter values vary too much, the motion is aborted with an error message. 0 = counter 1 = counter+SFI	0
P37	Encoder tolerance for SFI Enter tolerance value for SFI evaluation Input: tolerance value for SFI-evaluation in the selected resolution (P3 * P20). If P21 is used for step failure indication the scale of the counter P20 * P3 must be equal to the scale of the counter P21 * P39 and P21 must be set to zero after initialization of the scaling (or can be set to the same value as P20). e.g. scaling to 360°/rev.: Motor 200 steps per revolution, 1/20 step, →P3 = 360 / 200 / 20 = 0.09, encoder 10 bit / rev. → P39 = 360 / 2^{10} = 0.3515625	0

No.	Meaning	Default
P38	Encoder preferential direction of rotation 0 = + (positive) 1 = – (negative)	0
P39	Encoder conversion factor 1 increment corresponds to ... Computing the conversion factor: $Conversion\,factor = \frac{Thread}{Encoder\,steps\,per\,revolution}$	1
P40	Stop current in 0.01 $A_{r.m.s.}$ steps depending on the power stage I1AM01: 0 to 250 (0 to 2.5 $A_{r.m.s.}$) I1AM02: 0 to 350 (0 to 3.5 $A_{r.m.s.}$) ZMX^{+}: 0 to 630 (0 to 6.3 $A_{r.m.s.}$) MCD^{+}: 0 to 63 (0 to 6.3 $A_{r.m.s.}$) APS: 0 to 350 (0 to 3.5 $A_{r.m.s.}$) MSX52: 0 to 280 (0 to 2.8 $A_{r.m.s.}$) MSX102: 0 to 560 (0 to 5.6 $A_{r.m.s.}$) MSX152: 0 to 840 (0 to 8.4 $A_{r.m.s.}$)	2
P41	Run current in 0.01 $A_{r.m.s.}$ steps I1AM01: 0 to 250 (0 to 2.5 $A_{r.m.s.}$) I1AM02: 0 to 350 (0 to 3.5 $A_{r.m.s.}$) ZMX^{+}: 0 to 630 (0 to 6.3 $A_{r.m.s.}$) MCD^{+}: 0 to 63 (0 to 6.3 $A_{r.m.s.}$) APS: 0 to 350 (0 to 3.5 $A_{r.m.s.}$) MSX52: 0 to 280 (0 to 2.8 $A_{r.m.s.}$) MSX102: 0 to 560 (0 to 5.6 $A_{r.m.s.}$) MSX152: 0 to 840 (0 to 8.4 $A_{r.m.s.}$)	6
P42	Boost current in 0.01 $A_{r.m.s.}$ steps I1AM01: 0 to 250 (0 to 2.5 $A_{r.m.s.}$) I1AM02: 0 to 350 (0 to 3.5 $A_{r.m.s.}$) ZMX^{+}: 0 to 630 (0 to 6.3 $A_{r.m.s.}$) MCD^{+}: 0 to 63 (0 to 6.3 $A_{r.m.s.}$) APS: 0 to 350 (0 to 3.5 $A_{r.m.s.}$) MSX52: 0 to 280 (0 to 2.8 $A_{r.m.s.}$) MSX102: 0 to 560 (0 to 5.6 $A_{r.m.s.}$) MSX152: 0 to 840 (0 to 8.4 $A_{r.m.s.}$)	10

No.	Meaning	Default
P43	Current hold time in msec	20
P44	For I4XM01 only! Origin of the Control pulses for the axis 0 = 1:1 (Input=Output) 1 = from X 2 = from Y 3 = from Z 4 = from U 5 = from external	0
P45	Step resolution 1 to 512 0 = 1/1 step 7 = 1/16 step 1 = 1/2 step 8 = 1/20 step 2 = 1/2.5 step 9 = 1/32 step 3 = 1/4 step 10 = 1/64 step 4 = 1/5 step 11 = 1/128 step 5 = 1/8 step 12 = 1/256 step 6 = 1/10 step 13 = 1/512 step (e.g. APS01) Important: for I1AM: step resolution from 1/1 to 1/128 step	3
P46	not used	
P47	not used	
P48	not used	
P49	Power stage temperature in 1/10 °C	(read only)
P50	Divider for Control pulses only for I4XM01 Control pulses $_{Output}$=1/(n+1) * Control pulses $_{Input}$ 0 : 1/(0+1)=1 1: 1/(1+1)= 1/2 2: 1/(2+1) =1/3 3: 1/(3+1)=1/4 4: 1/(4+1)=1/5 5: 1/(5+1)=1/6	n=0
P51	Pulse width: (n+1)*100 ns only for I4XM01 n: 0….255 e.g. n=19: (19+1)*100 ns=2000 ns= 2µs -> F_{max}=1/(2*2 µs)=250 kHz	n=19

No.	Meaning	Default
P52	Internally used for trigger position.	
P53	Power stage monitoring 0 = off 1 = on	1
P54	Motor temperature in 1/10 °C -999999: Temperature module not existent -9999: negative overflow or temperature lower -220 °C at PT100 9999: positive overflow or temperature higher +390 °C at PT100	-999999 (read only)
P55	Motor temperature **warning** in 1/10 °C If the motor warmed up to a defined temperature value, a warning occurs. We recommend to operate the motor until it is cooled again.	0
P56	Motor temperature **shut-off** in 1/10 °C If the motor warmed up to a defined temperature value, the controller switches off and the power stage must be reset.	0
P57	Resolver voltage n=3…10 (Volt)	3
P58	Resolver ratio (ratio of primary to secondary winding) 0=1/8 1=1/4 2=1/2 3=1 4=2	2

10 Warranty, Disclaimer and Registered Trademarks

10.1 Disclaimer

Phytron GmbH has verified the contents of the manual to match with the hardware and software. However, errors and omissions are exempt and Phytron GmbH assumes no responsibility for complete compliance. The information contained in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

10.2 Warranty

The *phy***MOTION**™ modules are subject to **legal warranty**. Phytron will repair or exchange devices which show a failure due to defects in material or caused by the production process. This warranty does not include damage caused by the customer, for example, not intended use, unauthorised modifications, incorrect handling or wiring.

10.3 Registered Trademarks

In this manual several trademarks are used which are no longer explicitly marked as trademarks within the text. The lack of these signs may not be used to draw the conclusion that these products are free of rights of third parties. Some product names used herein are for instance.

- *phy***MOTION**™ is a trademark of the Phytron GmbH.
- *phy***LOGIC**™ is a trademark of the Phytron GmbH.
- Microsoft is a registered trade mark and WINDOWS® is a trade mark of the Microsoft Corporation in the USA and other countries.

11 Index

A

Axes module 16, 18
Axis 21, 22

C

Compensation for play 31
Configuration 10
Copyright 2
Counter 31
CPU 13, 15

D

Download 11

F

Firmware 11
ftp client 11
ftp server 11, 12

G

GSD 10

H

HW Config 10, 13

I

I/O module 23, 24

M

MCM 13

P

Parameter 6
*phy***LOGIC**™ 6
ProfiNet interface 9

S

S7 station 9
Safety instructions 4
Scripts 11
Start-/Stop frequency 28

W

Warranty 38