LabVIEW®-VI *phy*MOTION™

Virtual Instruments for the *phy*MOTION™ Controller

MANUAL 1301-A001 EN

phytron
Extreme. Precision. Positioning.

LabVIEW® Virtual Instruments

for the *phy*MOTION™

Controller

TRANSLATION OF THE GERMAN ORIGINAL MANUAL

Phytron GmbH

Industriestraße 12

82194 Gröbenzell, Deutschland

Tel.: +49 8142/503-0

Fax: +49 8142/503-190

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. Please send your comments to the following email address: doku@phytron.de

Contents

1 Information 4
2 What is LabVIEW® ? 7
2.1 Requirements 7
2.2 Extent of Supply 7
3 General VI Description 8
4 Description of the *phy*MOTION™ VIs 9
4.1 General 9
4.2 AD MCM.VI 9
4.3 COMM MCM.vi 10
4.4 Counter MCM.vi 10
4.5 DA MCM.vi 11
4.6 Direct Mode MCM.vi 12
4.7 Drive MCM.vi 13
4.8 Encoder MCM.vi 15
4.9 Init MCM.vi 16
4.10 Input MCM.vi 17
4.11 Output MCM.vi 18
4.12 Parameter MCM.vi 19
4.13 Register MCM.vi 20
4.14 Status MCM.vi 21
4.15 Extended Status MCM.vi 22
5 Demo MCM.vi 23
5.1 General Description 23
5.2 Reading the A/D and D/A Inputs 24
5.3 Drive Application for Direct Mode and Motor 25
5.4 Reading and Output of the Internal Distance Counter 27
5.5 Reading and Display of the Encoder Counter 28
5.6 Reading and Setting of the Inputs / Outputs 29
5.7 Reading / Writing the Parameters 30
5.8 Reading and Writing the Registers 31
5.9 Reading the Status 32
6 Parameter 33
7 Copyright and Disclaimer of Warranty 45
8 Index 46

1 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.*

Observe the following safety instructions!

Qualified personnel

WARNING – Serious injury possible!

Serious personal injury or serious damage to the machine and drives could be caused by insufficiently trained personnel!

Without proper training and qualifications damage to devices and injury might result!

- Design, installation and operation of systems may only be performed by qualified and trained personnel.
- These persons should be able to recognize and handle risks emerging from electrical, mechanical or electronic system parts.
- The qualified personnel must know the content of this manual and be able to understand all documents belonging to the product. Safety instructions are to be provided.
- The trained personnel must know all valid standards, regulations and rules for the prevention of accidents, which are necessary for working with the product.

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 What is LabVIEW® ?

This manual describes the LabVIEW graphical programming language from National Instruments. It uses icons to create the application.
LabVIEW stands for „Laboratory Virtual Instrument Engineering Workbench“.
LabVIEW programs are called Virtual Instruments or VIs. These VIs contents the front panel, the user interface and the block diagram, the graphical program code, which is compiled like other high level programming languages.
This manual describes the LabVIEW use for the Phytron controller *phy***MOTION**™.

LabVIEW is a Trade Mark of National Instruments Corporation.

2.1 Requirements

For using the Phytron *phy***MOTION**™ controller VIs it is expected that the user is well-trained in LabVIEW and knows the programming environment. Basic knowledge in programming like data types, loops etc. are required.

The *phy***MOTION**™ VIs are build for LabVIEW 8.0 or higher.

2.2 Extent of Supply

The LabVIEW libraries are saved as Labview_phymotion.zip.

3 General VI Description

LabVIEW programs are called virtual instruments or VIs.
Every VI can be used as a stand-alone program or as a subroutine called sub VI.
The user transfers data among block diagram objects through wires.
The VI starts to run when all input data are available.
If the complete VI is finished, the results are on the outputs. The sequence of the execution is defined by the dependency of the data. There is no predefined sequence (e. g. from right to left).

4 Description of the *phy*MOTION™ VIs

4.1 General

Inputs and outputs which are the same for all libraries:

Name	I/O	Meaning
VISA resource name in	Input	Transfer of the interface parameters
Error in	Input	Input of the error clusters
VISA resource name out	Output	Display of the interface parameters
Error out	Output	Output of the error cluster

Inputs and outputs have the same function. They are only described once.

Cluster is the bundling of different data types in LabVIEW. It can be used as an input or output.

4.2 AD MCM.VI

The A/D value of the *phy*MOTION™ is read.
This VI provides the current A/D converter value as a 14-bit unit to the output while executing.

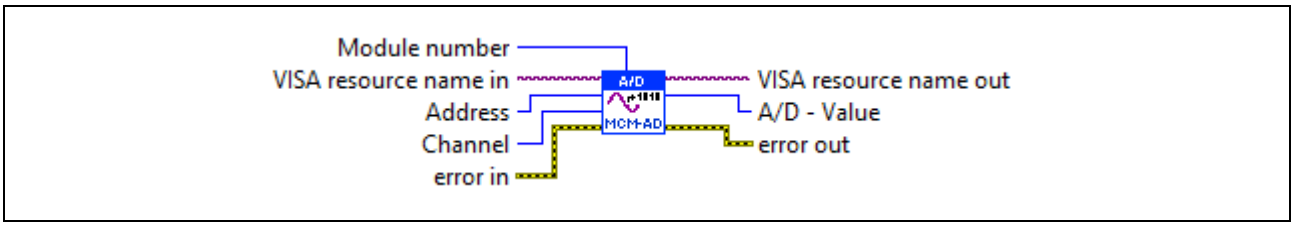


Fig. 1: AD MCM.vi

Name	Meaning	I/O
Address	Configured address at the controller (0 -15; 8-bit unit)	Input
Module number	Number of the addressed board	Input
Channel	Channel of the A/D converter, which should be read (1,2,3 or 4; 8-bit unit)	Input
A/D Value	A/D value in increments (0 – 16384, 14 bit)	Output

4.3 COMM MCM.vi

This VI is internal used by other VIs. It should not be applied for programming user specific applications.

4.4 Counter MCM.vi

This VI reads the selected axis counter.
It displays the counter value of the selected axis by reading the parameter 20 (P20).
You´ll find the parameter description in chapter 6.

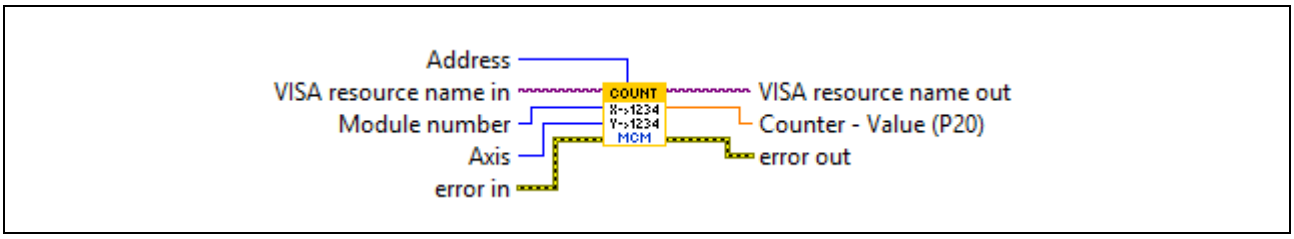


Fig. 2: Counter MCM.vi

Name	Meaning	I/O
Address	Configured address at the controller (0 -15; 8-bit unit)	Input
Axis	Axis, the counter should be read (1,2,3 or 4; 8-bit unit)	Input
Module number	Number of the addressed board	Input
Counter Value	Counter value of the axis (Double)	Output

4.5 DA MCM.vi

The D/A value of the *phy***MOTION**™ is read.
This functional module reads the values adjusted in the cluster and acknowledges with transmission OK.

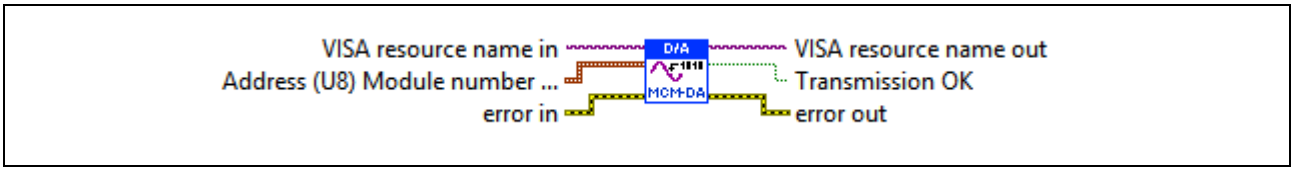


Fig. 3: DA MCM.vi

Name	Meaning	I/O
Cluster: Address Module number Channel Value	consists of the following file types: • Controller address (0 -15; 8-bit unit) • Number of the addressed board • D/A converter channel, which should be read (1,2, 3 or 4; 8-bit unit) • Increments (0….65535, 16 bit) are read	Input
Transmission OK	True, if the controller acknowledged the command (ACK) False, if the command was invalid (NAK)	Output

4.6 Direct Mode MCM.vi

An instruction is transmitted to the controller.
The VI transmits the string at the input Send String to the controller and picks the answer from the controller.
For detailed description of the controller parameters see chap.6.

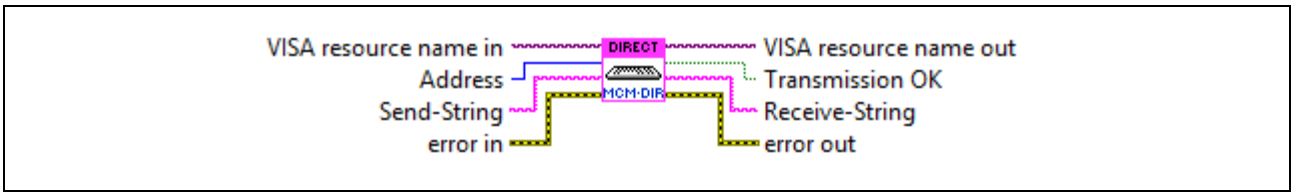


Fig. 4: Direct mode MCM.vi

Name	Meaning	I/O
Address	Configured address at the controller (0 -15; 8-bit unit)	Input
Send String	Command, which is transmitted to the controller (e. g. 1.1+1000 corresponds to drive 1000 steps)	Input
Transmission OK	True, if the controller acknowledged the command (ACK) False, if the command was invalid (NAK)	Output
Receive String	Response String of the controller (without control character and ACK) It´s empty in commands without response	Output

4.7 Drive MCM.vi

This VI sends drive instructions to the *phy***MOTION**™.
This functional module reads the value adjusted in the cluster and generates a drive instruction for the *phy***MOTION**™ .

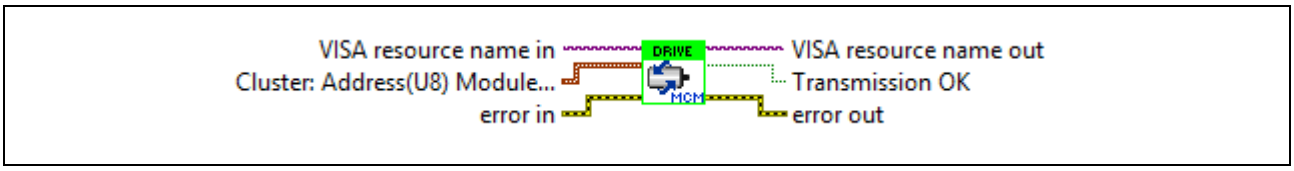


Fig. 5: Drive MCM.vi

Name	Meaning	I/O
Cluster: Axis Address Module number Position Mode (ENUM)	consists of the following file types: • Axis (8-bit unit): axis, where the drive instruction is written (1 to 18) • Address (8-bit unit): Controller address (0-15) • Number of the addressed board • Position Mode (ENUM): the following adjustments are available[1)]: ○ Relative, generates and transmits a relative drive instruction ○ Absolute, generates and transmits an absolute drive instruction ○ Initialisation Plus, generates and transmits an initialisation in the positive direction ○ Initialisation Minus, generates and transmits an initialisation in the negative direction ○ Initialisation Center Plus, generates and transmits an initialisation to the Center Switch in the positive direction ○ Initialisation Center Minus, generates and transmits an initialisation to the Center Switch in the negative direction ○ Free Run Plus, starts a free run in the positive direction ○ Free Run Minus, starts a free run in the negative direction	Input

Distance	• Distance (DLB): this input is used as a distance for relative and absolute drive instructions	
Transmission OK	True: the controller accepts the instruction False: invalid command	Output

[1)]More information:

Further Manual

An overview of axis commands and associated parameters, as well as schematic representations of the driving parameters can be found in the following manual:

„Principles of Positioning for Stepper Motor Controllers“

4.8 Encoder MCM.vi

The encoder counter reads the selected axis.
Parameter 22 (P22) is read out for the respective axis.
You´ll find the description of the parameters in chapter 6 parameters.

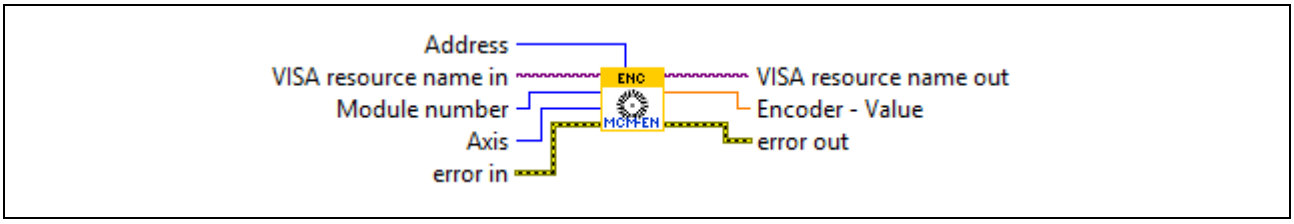


Fig. 6: Encoder MCM.vi

Name	Meaning	I/O
Address	Configured address at the controller (0 -15; 8-bit unit)	Input
Module number	Number of the addressed board	Input
Axis	Axis, whose counter is to be read (1 to 18; 8-bit unit)	Input
Encoder Value	Read encoder value of the axis (double)	Output

4.9 Init MCM.vi

This VI displays the initiator status.
The *phy***MOTION**™ initiator status is imported and displayed as Boolean Cluster.

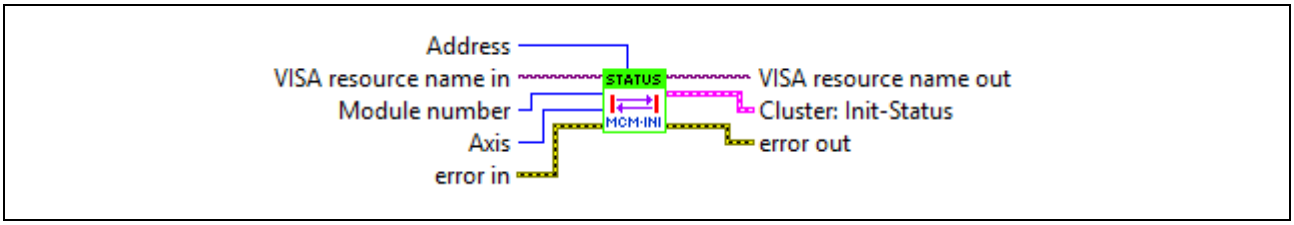


Fig. 7: Init MCM.vi

Name	Meaning	I/O
Address	Configured address at the controller (0 -15; 8-bit unit)	Input
Module number	Number of the addressed board	Input
Axis	The count of the axis is to be read (1 to 18, 8-bit unit)	Input
Cluster: Initiator Status	The Initiator Status consists of five elements (BOOL) • Axis +, activated = TRUE, free = FALSE • Axis –, activated = TRUE, free = FALSE • Axis Center, activated = TRUE, free = FALSE • Axis SW+, activated = TRUE, free = FALSE • Axis SW–, activated = TRUE, free = FALSE	Output

4.10 Input MCM.vi

Reads the *phy***MOTION**™ input status.
The status of the *phy***MOTION**™ is displayed as a Boolean Cluster.

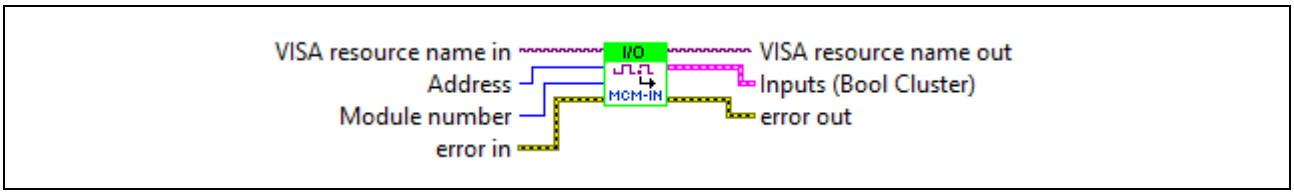


Fig. 8: Input MCM.vi

Name	Meaning	I/O
Address	Configured address at the controller (0 -15 8-bit unit)	Input
Module number	Number of the addressed board	Input
Cluster: Inputs	consists of eight elements (Boolean) TRUE = Input High FALSE = Input Low	Output

4.11 Output MCM.vi

This VI sets the outputs at the *phy***MOTION**™.
Executing this routine the status at the input is set as output status.

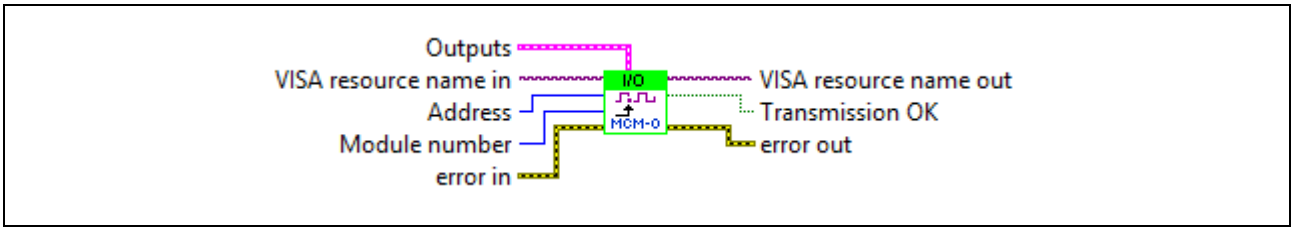


Fig. 9: Output MCM.vi

Name	Meaning	I/O
Address	Configured address at the controller (0 -15; 8-bit unit)	Input
Module number	Number of the addressed board	Input
Cluster: Outputs	consists of eight elements (Boolean) TRUE = Output High FALSE = Output Low	Input
Transmission OK	True: the controller accepts the instruction False: invalid command	Output

4.12 Parameter MCM.vi

This VI reads or sets the *phy***MOTION**™ parameters.
The *phy***MOTION**™ reads or transmits the parameter which is adjusted in the Parameter Number.

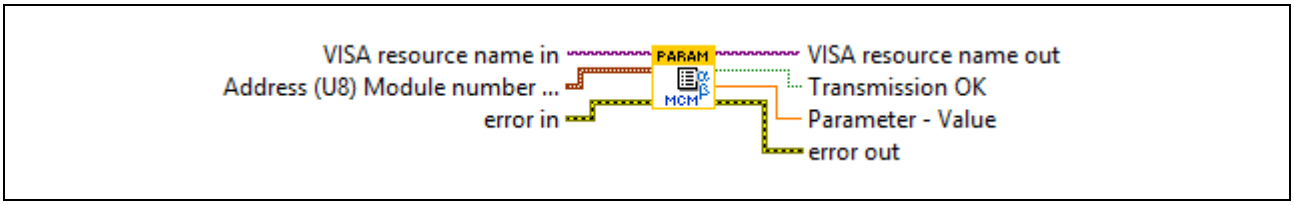


Fig. 10: Parameter MCM.vi

Name	Meaning	I/O
Cluster: Address Module number Axis ParameterNumber Parameter Value Read / Write	consists of the following data types: • 8-bit unit, controller address (0-15) • Number of the addressed board • 8-bit unit: axis, the parameter is read/written (1 to 18) • 8-bit unit: parameter number, to be read or written • Parameter Value (Double): Parameter value to be written. Only with the choice ‚write'! • Read / Write (ENUM): Contains the entry Read and Write. Read: Parameter is read and the Parameter Value is displayed at the output. Write: Parameter is written with the value from the input Parameter Value.	Input
Transmission OK	True: the controller accepts the instruction False: invalid command (Boolean)	Output
Parameter Value	The Read function displays the parameter value of the selected parameter (Double).	Output

4.13 Register MCM.vi

The register of the *phy***MOTION**™ register is read or set..
The register set in Register Number is read by the *phy***MOTION**™ or transmitted to the *phy***MOTION**™.

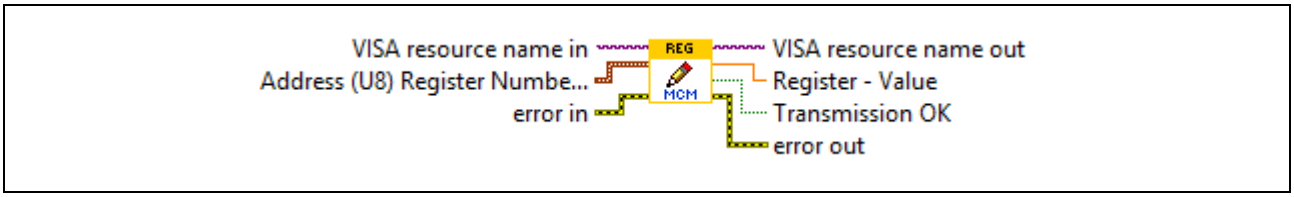


Fig. 11: Register MCM.vi

Name	Meaning	I/O
Cluster: Address Register Number Register Value Read / Write	It has the following data types: • Address (8-bit unit): controller address (0-15) • Register Number (16-bit unit): register number, to be read or written. • Register Value (Double): register value to be written. Only with the choice ‚write'! • Read / Write (ENUM): contains the entry Read and Write. Read: Parameter is read and the Register Value is displayed at the output. Write: Parameter is written with the value from the input Register Value.	Input
Transmission OK	True: the controller accepts the instruction False: invalid command (Boolean)	Output
Register Value	The Read function displays the Register Value of the selected parameter (Double).	Output

4.14 Status MCM.vi

The *phy***MOTION**™ status is read and the result is displayed as Boolean Cluster.

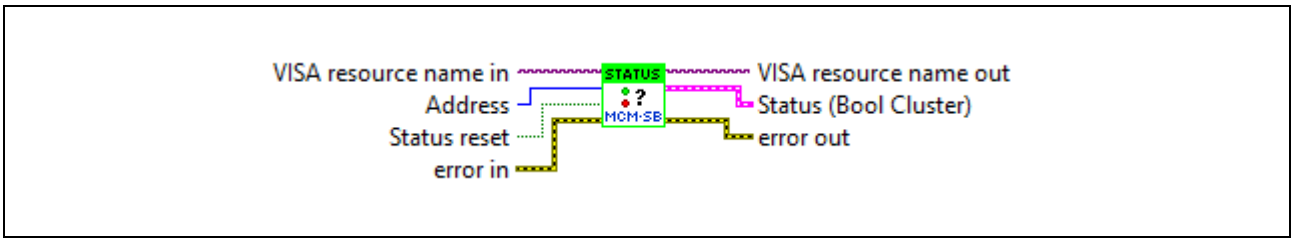


Fig. 12: Status MCM.vi

Name	Meaning	I/O
Address	Configured address at the controller (0 -15; 8-bit unit)	Input
Module number	Number of the addressed board	Input
Status Reset	Resets any pending status (error)	Input
Cluster: Status	Displays the *phy***MOTION**™ status as Boolean cluster. The status is read binary. You´ll find further information in the *phy***LOGIC**™ Programming Manual for *phy***MOTION**™ available under the instruction ST.	Output

4.15 Extended Status MCM.vi

The extended status of the *phy***MOTION**™ is read and displayed as Boolean Cluster.

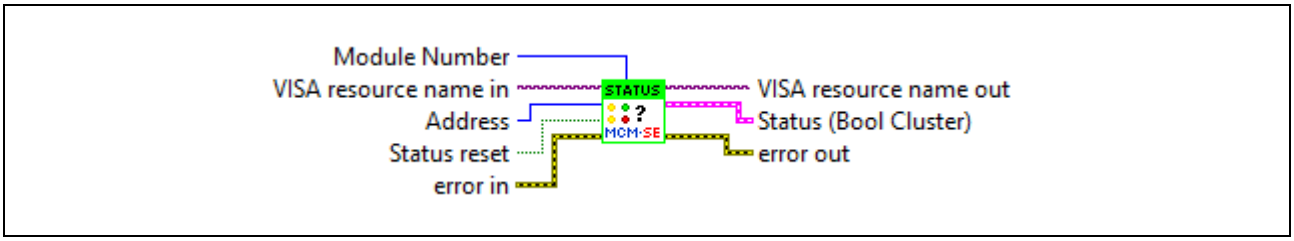


Fig. 13: Extended Status MCM.vi

Name	Meaning	I/O
Address	Configured address at the controller (0 -15; 8-bit unit)	Input
Status Reset	Resets any pending status (error)	Input
Cluster: Status	Displays the *phy***MOTION**™ status as Boolean cluster. The status is read binary. You´ll find further information in the *phy***LOGIC**™ Programming Manual for *phy***MOTION**™ available under the instruction 1.1SE.	Output

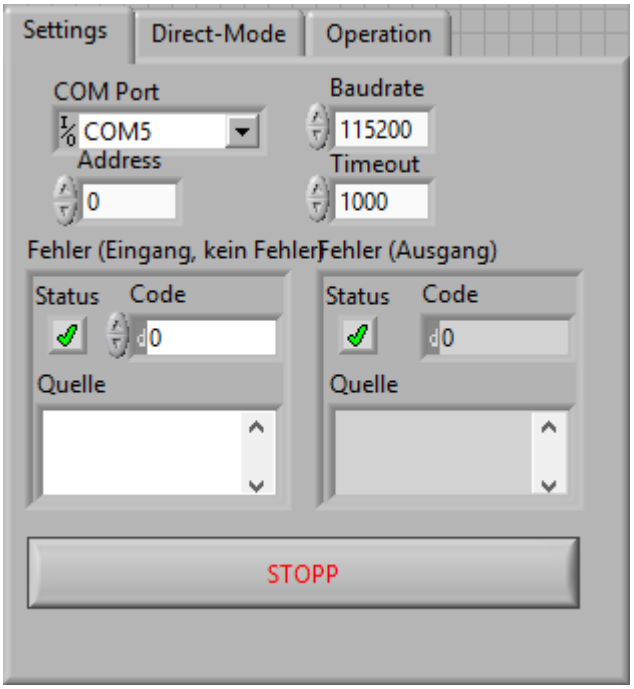
5 Demo MCM.vi

5.1 General Description

The VI demos use the VIs from the phymotion.llb. These are demos of register cards with different tabs.
The first register card (Settings) is equal in all demos and only described once.

Settings register card:

Here are the general settings for the interface:

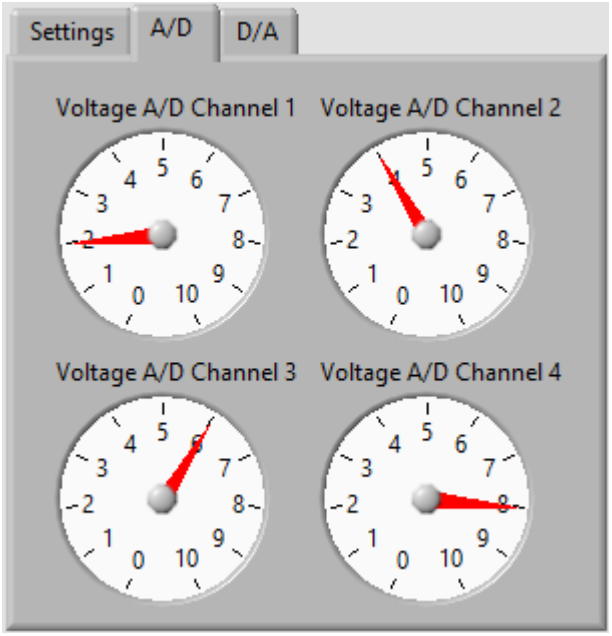


COM Port	Configures the interface used. In this case the serial interface COM5.
Address	The Address adjusted at the controller
Baud Rate	Configuration of the controller´s baud rate, e. g. 115 200.
Timeout	The time waiting for an answer. If there is no answer in the denoted time, VISA-VI displays an error.
Error	This output displays an error message, which occurs during communication.
Stop	Stops the program.

5.2 DEMO A/D-D/A MCM Reading the A/D and D/A Inputs

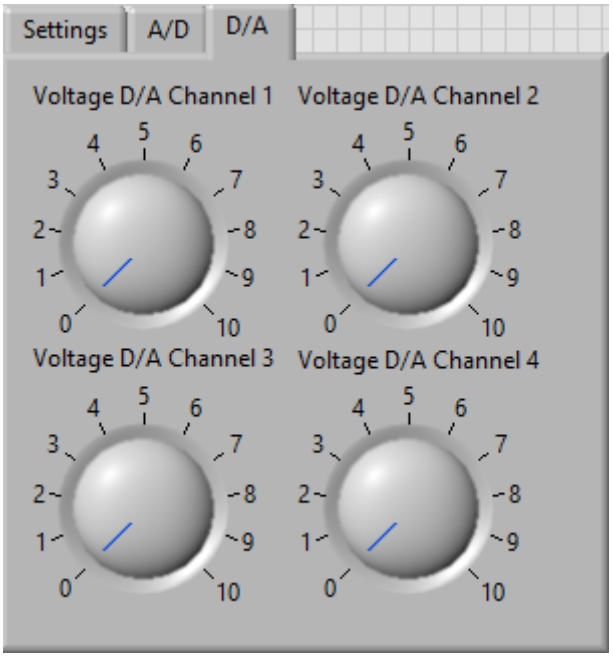
This demo has one register card with three tabs: Settings, A/D and D/A.
(For the description of Settings register card please see above.)

A/D register card:



This register card shows the present voltage at channel 1 to 4 of the A/D inputs of the *phy***MOTION**™ These are both graphically displayed.

D/A register card:

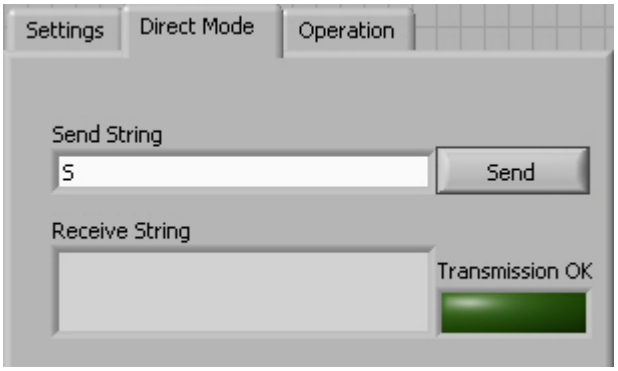


Here, the voltage value for the analogue output is set at the channel 1 to 4 of the *phy***MOTION**™.

5.3 Drive Application for Direct Mode and Motor

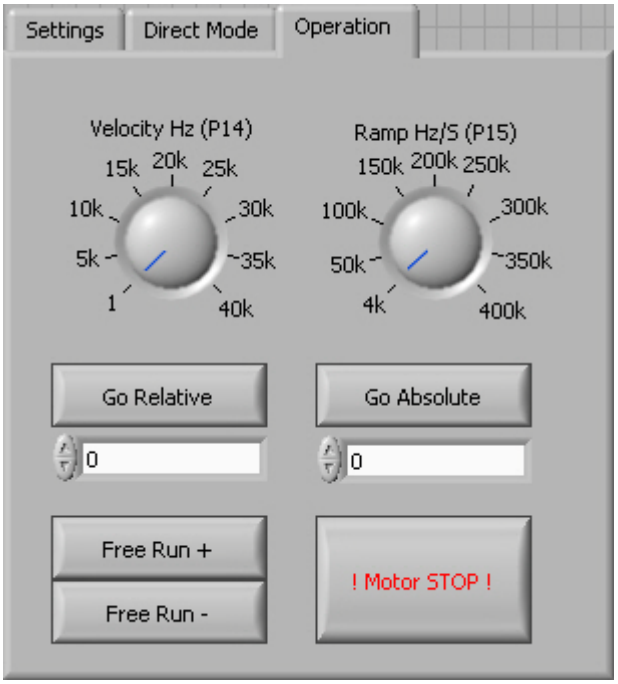
This demo is a small application to demonstrate
- **Direct Mode MCM.vi**
- **Parameter MCM.vi** and
- **Drive MCM.vi** files.

Direct Mode register card:



Send String	Entry of the transmitted command
Send	Transfer of the command
Receive String	Display of the *phy***MOTION**™ answer
Transmission OK	Command recognized: LED on (ACK) command not recognized: LED off (NAK)

Operation register card:

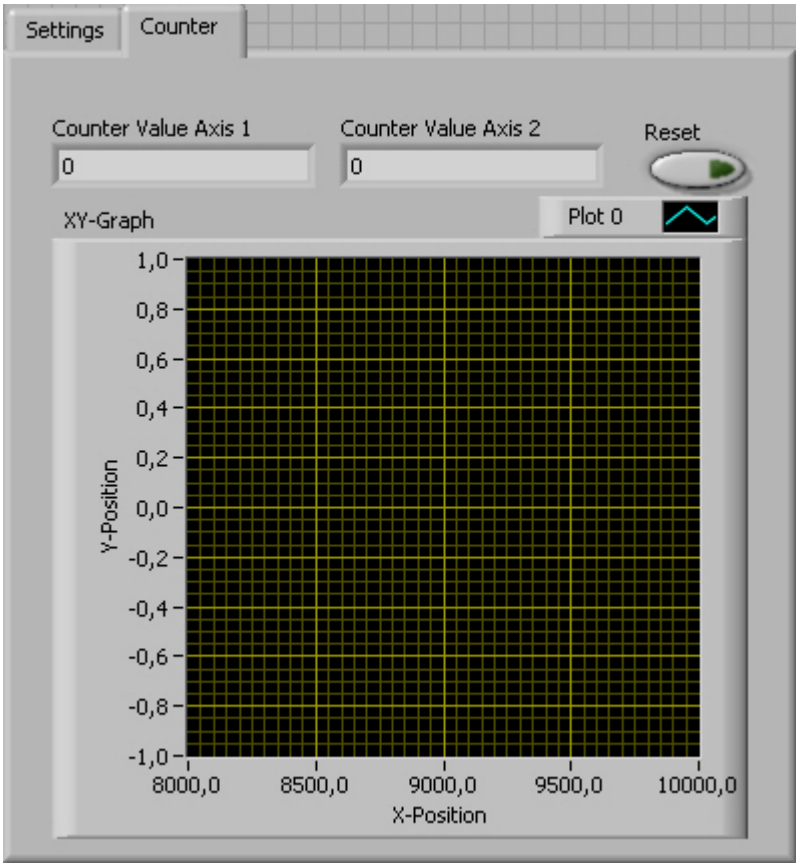


Velocity Hz (P14)	Sets the driving speed of the *phy***MOTION**™. Free Run + or – enables to change the velocity also during the run.
Ramp Hz/S (P15)	Sets the acceleration- and deceleration of the ramp. The value can only be adopted when the motor is at a standstill.
Go Relative	Moves from the actual position by the entered value (box beneath).
Go Absolute	The entered value (box beneath) is referred to the zero point.
Free Run +	Starts a free run in positive direction
Free Run -	Starts a free run in negative direction
! Motor STOP !	Cancels each running positioning and stops the motor.

5.4 DEMO COUNT MCM Reading and Output of the Internal Distance Counter

This demo reads the internal *phy***MOTION**™ distance counter (P20) and displays both as a counter value as a diagram.

Counter register card:

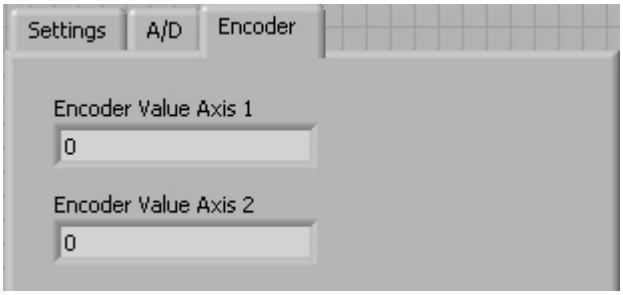


Counter Value Axis 1	Counter value display of axis 1.1
Counter Value Axis 2	Counter value display of axis 2.1
Reset	Deletes the graph
XY Graph	Diagram of the varying counter value at axis X and Y and in the system of coordinates.

5.5 Reading and Display of the Encoder Counter

This register card imports the *phy***MOTION**™ Encoder Counter (P22) and the counter value is displayed in the text box.

Encoder register card:

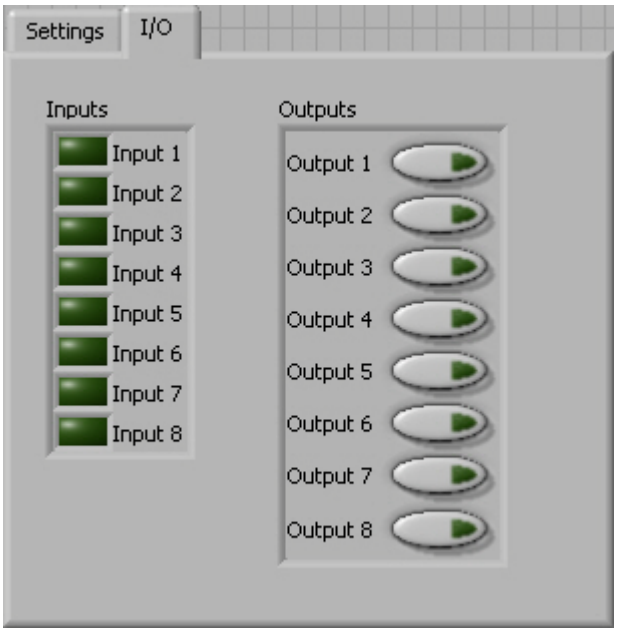


Encoder Value Axis1	The encoder counter value of the axis 1.1
Encoder Value Axis2	The encoder counter value of the axis 2.1

5.6 DEMO I/O MCM Reading and Setting of the Inputs / Outputs

This demo reads and displays the *phy***MOTION**™ inputs and activates the outputs.

I/O register card:

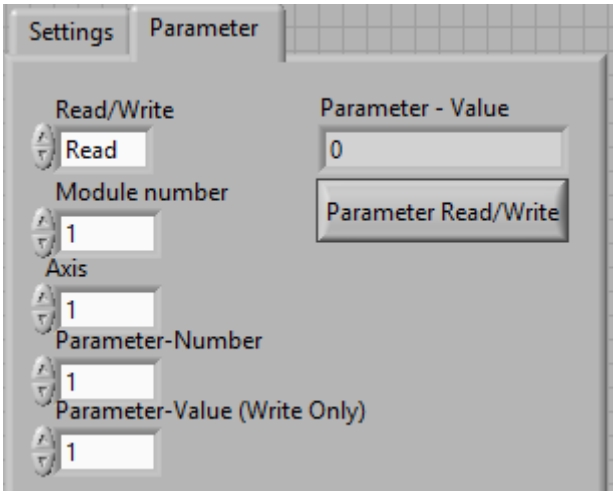


Inputs	Display of the MCM input status: LED on = Status High LED off = Status Low
Outputs	The respective MCM outputs can be switched.

5.7 DEMO PARAM MCM Reading / Writing the Parameters

This demo reads and writes the *phy***MOTION**™ parameters.

Parameter register card:

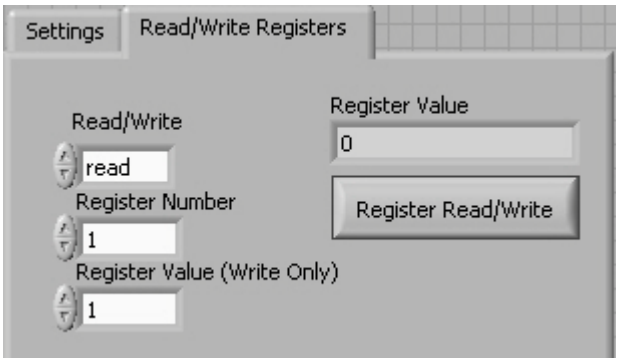


Read/Write	Reading or writing the parameters
Module number	Number of the addressed board
Axis	Axis, whose parameters are changed
Parameter Number	Parameter number, which is modified
Parameter Value (Write Only)	Value to which the parameter is changed (write only)
Parameter Value	Read parameter of the controller (read only)

5.8 DEMO REG MOM Reading and Writing the Registers

This demo reads and writes the *phy***MOTION**™ Register.

Read/Write Registers Register card:

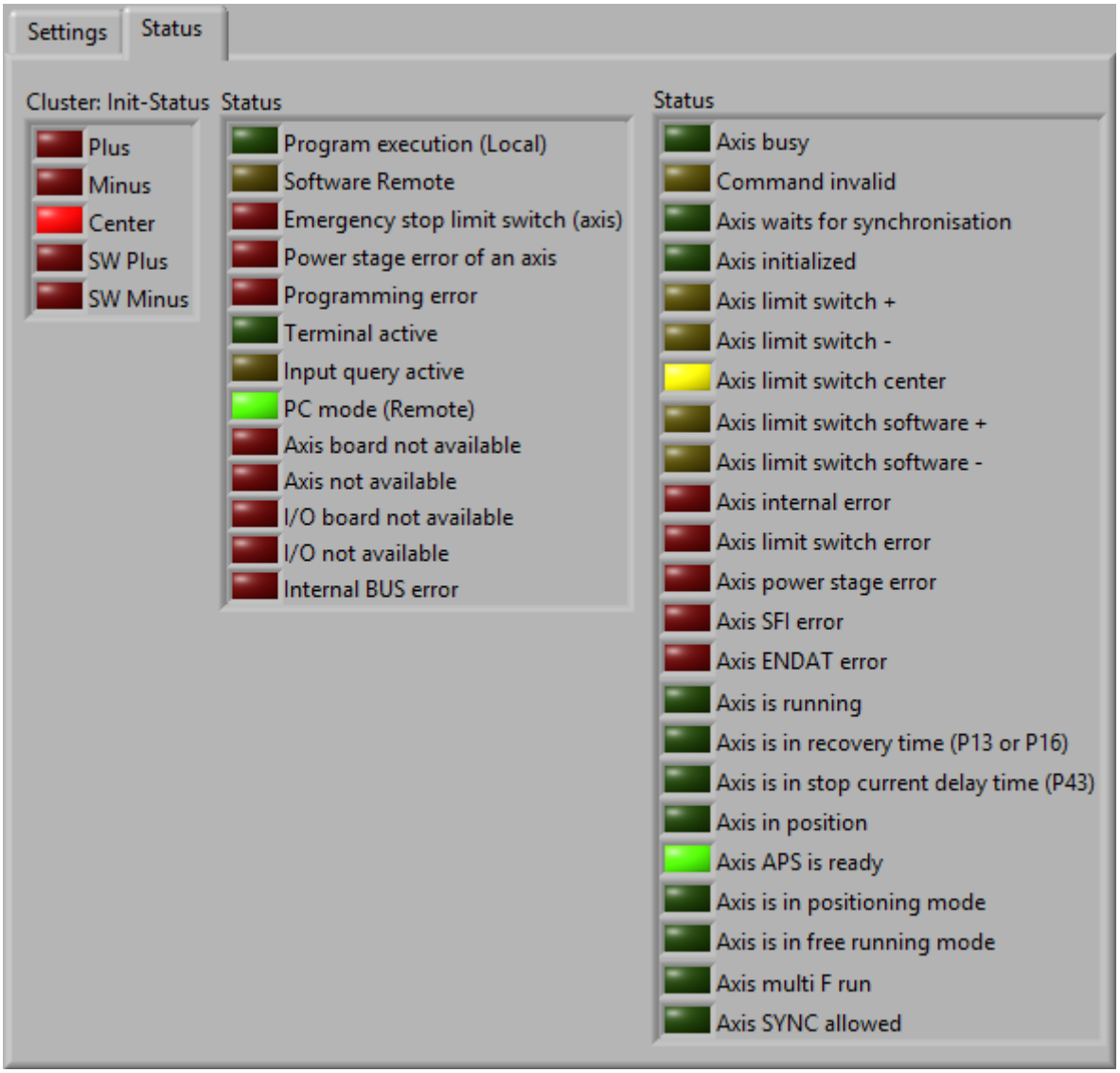


Read/Write	Reads or writes the registers
Register Number	Register number which is read or written
Register Value (Write Only)	Value that is written in the register. (write only)
Register Value	Value that is written out of the register. (read only)

5.9 Reading the Status

This demo reads and displays the initiator status and the general and extended *phy***MOTION**™ status.

Status register card:



Initiator Status	Initiator status of the controller. The LED is on, when the initiator is activated.
Status	General status of the controller. The LEDs display the status. The ST command is used. Extended status: 1.1SE

The LED colors are described in the *phy***LOGIC**™ ToolBox communication software for PC Manual.

6 Parameter

For operating a stepper motor controller several presetting as speed, acceleration ramps or waiting time are required. These presetting are called **Parameters**.

Default parameters are stored which can be used in several applications at delivery. You can read and edit these parameters with LabVIEW Parameter-VI or in *phy***LOGIC**™ ToolBox.

Several counters are also contained in the list of parameters, which will be continuously actualized by the program. The counters can be read and some of them can be edited, too.

- For each axis separate parameters have to be set. Insert a module and axis number to mark the axis in front of the parameter number.

  Example: m.aP15 is the acceleration ramp value for axis m.a.

- Parameter values can be entered or read.
- P49 can only be read.
- P19 to P22 are counters. They will be actualized by the program during axis movement.
- P27 to P54 are special parameters for the *phy***MOTION**™.
- Current values (P40 to P42) and P45 only apply to the INTERNAL power stages or power stages, which are connected via a bus:

*phy***MOTION**™	**Supply**	**Power stage module(s)**	**P45**
	EXTERNAL	INAM, I1AM01, I1AM02,…	as described in the parameter list
		EXAM	without function; external power stage is set via DIP or rotary switch
	INTERNAL	integrated (MSX+, ZMX+,…)	invalid; Distribution of step resolution (0 to 15) according to the power stage table (see power stage manual)

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
P05 **P06**	not used	

No.	Meaning	Default
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
P13	Recovery time MØP Time lapse during initialization Enter in msec	20

No.	Meaning	Default
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 Remarks: 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!

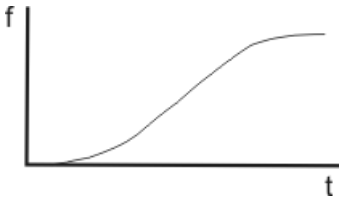
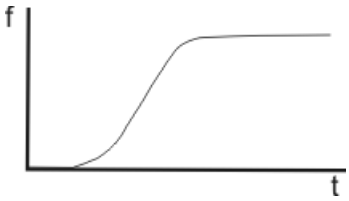
Frequency band setting

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>
</table>

No.	Meaning	Default
P32	Positioning ramp shape 0 = s-shape 1 = linear ramp Remark: The s-shape ramp can be modified with P33 parameter.	1
P33	Arc value setting for s-shape ramp Values: OMC: 1 to 8191 TMC: 1 to 32767  P33: low value  P33: high value	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

No.	Meaning	Default
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
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

No.	Meaning	Default
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
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

No.	Meaning	Default
P45	Step resolution 1 to 512 0 = 1/1 step 1 = 1/2 step 2 = 1/2.5 step 3 = 1/4 step 4 = 1/5 step 5 = 1/8 step 6 = 1/10 step 7 = 1/16 step 8 = 1/20 step 9 = 1/32 step 10 = 1/64 step 11 = 1/128 step 12 = 1/256 step 13 = 1/512 step (e.g. APS01) Important: for I1AM: step resolution from 1/1 to 1/128 step P45 only applies to the INTERNAL power stages or power stages, which are connected via a bus (see chap. 9).	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
P52	Internally used for trigger position.	
P53	Power stage monitoring 0 = off 1 = on	1

No.	Meaning	Default
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

7 Copyright and Disclaimer of Warranty

The communication software Lab VIEW and any documentation delivered with it are protected by copyright law. The manual must not be copied, reproduced, put into machine readable form, neither complete nor in parts, without the prior written permission of National Instruments Corporation. It is permitted to create backups of the Phytron Freeware VIs for personal use. However, the program must not be modified or sold.

Limitation of warranty

The Lab VIEW-VI software and its documentation have been made with great carefulness and have been reproduced under effective controlling measures. Nevertheless there might be mistakes.

We refer to our terms of delivery and payment, in particular to item VII liability and item IX software utilization.

You'll agree to our delivery and payment conditions, if you instal, copy or use otherwise the software. If you disagree to these conditions you aren't authorized to install or use the software.

The Phytron specific VIs and the associated manual were made with great care and reproduced with the involvement of effective control measures. Each CD-ROM is checked before delivery with a well-known scanner program for viruses of all kinds. Nevertheless errors can exist.

Phytron does not warrant that the software is free of errors.
Phytron is not liable for any damages arising by using this software. The user or application designer is ultimately responsible for the suitability of the software.

By installing, copying or otherwise using the software product you agree with our terms of delivery and payment. If you don´t agree to these terms and conditions of purchase, you are not justified to install or use the software product.

Trademarks

In this manual several trademarks are used which are no longer explicitly marked as trademarks within the text. The lack of this 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:

- LabVIEW is a registered trademark of National Instruments Corporation.
- *phy***MOTION**™ is a registered trade mark of the Phytron GmbH.
- *phy***LOGIC**™ is a registered trade mark of the Phytron GmbH.

8 Index

A

A/D converter 9
A/D Inputs 24

B

Boolean Cluster 16, 17

C

Cluster 9
Copyright 45
Counter 10, 27

D

D/A converter 11
Demo 23
Direct mode 12, 25
Drive instruction 13

E

Encoder 15
Encoder Counter 24
Error cluster 9

F

Freeware 45

I

I/O 29
Init 16
Initiator Status 16
Input 17
Inputs / Outputs 29
Interface parameter 9

L

LabVIEW 7

O

Operation 26

P

Parameter 19, 30
Parameter Number 19
*phy***LOGIC** 21, 22

R

Read / Write register 31
Register 20
Register card 23
Register Number 20

S

Settings 23
Status 21, 22, 32

T

Trademarks 45

V

VI 7
Virtual instruments 7

W

Warranty 45

Phytron GmbH
Industriestraße 12 – 82194 Gröbenzell
T+49-8142-503-0 F +49-8142-503-190
www.phytron.eu