

Description of the Command Sets for

ASI MS-2000

and

LEP MAC-5000

compatible modes

as implemented in the TANGO Controller



Märzhäuser Wetzlar GmbH+Co.KG.
In der Murch 15
35579 Wetzlar
Germany
Tel.: +49/6441/9116-0
www.marzhauser.com

Table of Contents

1. Introduction.....	3
1.1. Basic Rules.....	3
1.2. Reply.....	3
1.4. Negative Reply Formats.....	4
1.5. Error Code Table.....	4
2. Quick Reference.....	5
2.1. Main Operating Commands.....	5
2.2. Customization Commands.....	5
3. Detailed Reference of Commands in alphabetic Order.....	6
3.1 Acceleration Control.....	6
3.2 Backlash Control.....	6
3.3. Centering Stage.....	7
3.4 CDate.....	7
3.5 Error.....	7
3.6 Halt.....	7
3.7 Here.....	8
3.8 Home Motors.....	8
3.9 Joystick Control.....	8
3.10 Move Motor Absolute.....	8
3.11 Move Motor Relative.....	9
3.12. Motor Status Byte Read (hexadecimal).....	9
3.13. Motor Status Byte Read (decimal).....	10
3.14 Reset.....	10
3.2. Remote Key Readings.....	10
3.15 Save.....	10
3.16 Speed.....	11
3.17 Spin.....	11
3.18 Status.....	11
3.19 Units Multiplier.....	12
3.19 Version.....	12
3.20 Wait.....	12
3.21 Where.....	12
3.21 Who.....	12
Document Revision History.....	13

1. Introduction

This manual is intended as technical reference for programmers building Tango software applications based on "LEP MAC-5000" or "ASI MS-2000" compatible commands, excluding the so called "low level" or "binary" part, which is not implemented. The Tango remains in high level mode always.

We decided to document these both command sets in a single manual, since they are very similar. For all other command sets please refer separate available Tango documentation.

The Tango is available with RS232, USB, PCI or PCIe interface. From programmers point of view, all these interfaces behave identical and are transmitting and receiving information to a serial port based on ASCII characters. The only differences are the communication speed and the port number.

The Tango is shipped with a CD containing signed drivers for all actual 32 and 64 bit Microsoft operating systems, like XP, Vista and Windows 7, selecting the correct driver automatically during installation.

Starting from the native Tango command set you may like to change the interpreter mode:

Send the command **!IPRETER 3<CR>** to switch from native to LEP command set

or **!IPRETER 4<CR>** to switch from native to ASI command set (don't forget to send the exclamation mark)

Send **IPRETER 1<CR>** to switch back to native interpreter (without exclamation mark).

The command termination character specified here as <cr> has the value of 13 decimal, 0x0d hexadecimal and is represented in C programming language by "\r" (carriage return).

Programming protocol is with text of ASCII alpha-numeric characters. Also included are some control characters such as carriage return, backspacing, and tab.

There is a set of built-in commands with unique names. These commands can be executed by simply sending the command name with some parameters if required. The reply received will be in a pre set format and may include the result required.

For example: Reading the position of motor X takes the form:

command: **WHERE X<CR>**

ASI reply: **:A 1234<CR><LF>**

LEP reply: **:A 1234<LF>**

indicating a position 123.4 microns aside the origin.

The commands are not case sensitive, e.g. "Where", "WHERE" and "where" are functional identical.

1.1. Basic Rules

Characters sent from a host computer are stored in a buffer and executed after receiving a carriage return. Each line should be started with a command name and terminated with a carriage return character. Each line can only contain one command name, such as CALIB or HALT, but more then one parameter can be appended to a command, by respecting the command format. Each command is executed when a carriage return is received. Commands and parameters should be separated by at least one space or tab character. When a parameter with a number is involved, one or more spaces or tabs are permitted between them.

For example: The following formats are all the same:

where x=123 <cr>

where x =123<cr>

where x = 123 <cr>

The command termination character specified here as <cr> has the value of 13 decimal, 0d hexadecimal and represented in C programming language by "\r" (carriage return).

1.2. Reply

A reply is sent back from controller to the host upon reception of a command. Every reply starts with an ASCII colon character (":") and terminates by a character or sequence. The delimiter of this sequence depends on the selected mode.

In interpreter mode 3 (LEP) this is a single character having a value of 10 decimal, 0A hexadecimal and represented as "\n" in C programming language.

In interpreter mode 4 (ASI) mode the delimiter is the consecutive sequence of the two hex values 0D 0A representing "\r\n" in C language.

Reception of a reply means controller is ready to accept the next command from the host computer. The replies can be divided to two groups regarded as positive and negative. A positive reply is sent back if there are no errors encountered within the command structure. A positive reply character is an ASCII ("A") following the reply start character. Internally every command received is assigned a reference number which is sent to host with the positive reply. A

negative reply is sent back if there are errors within the command structure. A negative reply character is an ASCII ("N") following the reply start character. Sending a command that does not exist, not respecting the command formats or trying to execute commands with no corresponding modules installed, are some of the reasons that will cause negative replies.

1.3. Positive Reply Formats

Positive reply without any other parameters.

Move X

:A

Positive reply with a value requested.

Where X Y

:A -2000 1000

In this case inquired X position is -2000 and Y is 1000.

A positive reply may have error code replacing the value requested.

Where X Y

:A -2000 N-2

Inquired X position is -2000, but Y-axis is not installed.

1.4. Negative Reply Formats

An error code is added to the negative replies:

Xyxtter

:N -1

The command Xyxtter is unknown to the controller.

Move X

:N -2

The axis X is not installed.

1.5. Error Code Table

When a command is received that the controller cannot interpret, for one reason or another, an error message is returned in the following format:

:N <error code>

The error codes are as follows:

- 1 Unknown Command
- 2 Illegal Axis Parameter
- 3 Missing Parameter or not enough parameters (e.g. move r=)
- 4 Parameter out of range
- 5 Operation failed
- 6 Undefined Error (command is incorrect, but for none of the above reasons)
- 21 Process aborted by the HALT command

2. Quick Reference

The following section describes the command set of the Tango when switched to LEP or ASI mode. If you don't need to know everything in detail, just use the quick reference below to get started.

2.1. Main Operating Commands

 or <bs> - Abort current command and flush input buffer

LEP Command	ASI Command	ASI Shortcut	Description
	CDATE	CD	Returns Date/Time current firmware was compiled
halt	HALT	\	Halts all serial commands being executed
here	HERE	H	Writes a position to an axis position buffer
home	HOME	!	Tells stage to go to physical limit switches
move	MOVE	M	Writes a position to an axis target buffer
movrel	MOVREL	R	Writes a relative position to target buffer
rdstat	RDSTAT	RS	Returns a Status Information byte for an axis in decimal ASCII format
remres	RESET	~	Resets the controller
speed	SPEED	S	Sets the maximum velocity/speed of axis
spin	SPIN	@	Causes axis to spin motor at given DAC rate
status	STATUS	/	Returns B-Busy, N-Not Busy
where	WHERE	W	Returns current position

2.2. Customization Commands

These commands support setup parameters. In most cases, these commands would be used only once after the unit is powered up.

LEP Command	ASI Command	ASI Shortcut	Description
accel	ACCEL	AC	Changes/Displays ramp time in milliseconds
	BACKLASH	B	Changes axis backlash correction motion constant
	ERROR	E	Changes/ Displays max position error allowable before the controller will start re-correcting position.
joystick	JOYSTICK		Enables/Disables/Assigns manual control input for an axis
	PCROS	PC	Changes/Displays position error at which controller considers a move to be complete
	SAVESET	SS	Saves current set-up parameters to non volatile memory.

3. Detailed Reference of Commands in alphabetic Order

This chapter explains all implemented commands in detail.

3.1 Acceleration Control

Description for LEP mode:

Command: ACCEL

Format: Accel motor-id [motor-id] [...]
or
Accel motor-id=value [motor-id=value][...]

Function: This command is similar to Speed command. Acceleration number can have values from 1 to 255, which controls the acceleration curve of the motors. This ramp value is inversely proportional to the ramping time. The smaller the ACCEL value, the shorter the ramp time.

Reply: If there are no errors, a positive reply is sent.

Example: ACCEL R=100 T=60 Z=10
:A
ACCEL R T Z
:A 100 60 10

Description for ASI mode:

Command: ACCEL

Shortcut: AC

Format: ACCEL [X=time] [Y=time] [Z=time]

Function: This command sets the amount of time in milliseconds that it takes an axis motor speed to go from the start velocity to the maximum velocity and then back down again at the end of the move. At a minimum, this acceleration / deceleration time must be greater than t_{step} (the amount of time it takes for the controller to go through one loop of its main execution code. Use the INFO command to determine the t_{step}).

Example: AC X=50 Y=50 Z=50
:A
ACCEL X? Y? Z?
:X=50 Y=50 Z=50 A

The command in this example will make the controller take 50 milliseconds to accelerate the motors on each axis during a move command. When the controller gets within 50 milliseconds of finishing the move, it will begin to decelerate the motors back down to the start velocity where the pulses take over to bring the axes within the pulse crossover position error.

3.2 Backlash Control

Command: BACKLASH

Shortcut: B

Format: BACKLASH [X= distance] [Y= distance] [Z= distance]

Function: This command sets (or displays) the amount of distance in millimeters to travel to absorb the backlash in the axis' gearing. This backlash value works with an anti backlash routine built into the controller. The routine ensures that the controller always approaches the final target from the same direction. A value of zero (0) disables the anti-backlash algorithm for that axis.

Example: B X=.05 Y=.05 Z=0
:A

The command in this example will make the controller move the X and Y axes to a location 50 microns away from the final target before moving to the final target, while the anti-backlash algorithm for the Z axis is disabled.

3.3. Centering Stage

Command: CENTER

Format: Center motor-id=speed [motor-id=speed] [...]

Function: The syntax of this command is similar to the SPIN command described previously with an extra function. It will be used to center the motor in the middle of the axis or at the center pulse location if one is provided. To use this command the motor should be on an axis with two switches located at the ends of the axis. If center pulse is not provided motor is located in the center of the two end limits. If center pulse is provided the motor is located at the center pulse location.

Reply: If there are no errors, a positive reply is sent back. This reply does not signal the end of centering process. The status of motor can be determined by reading the status byte. with RDSTAT.

Example: center x=10000
:A

3.4 CDate

Command: CDATE

Shortcut: CD

Format: CDATE

Function: This command returns the date and time the current firmware was compiled.

Example: CD
Jun 23 2009:15:36:21

This example shows that the firmware running was compiled on June 23th year 2009 at 3:36:21 PM.

3.5 Error

Command: ERROR

Shortcut: E

Format: ERROR [X= position] [Y= position] [Z= position]

Function: This command sets the Drift Error setting. This setting controls the crossover position error (in millimeters) between the target and position at which the Tango controller considers an axis to be too far out of position. When this limit is reached, the controller will re-attempt to move the axis back within the Finish Error (PC) limit. The current value for this setting can be viewed using the INFO command or by placing a ? after the axis name. Entries of zero value, e.g., ERROR X=0<CR>, are ignored.

Examples: E X=.0004
:A

Input values equal to or less than zero are acknowledged by ":A ", but ignored. The command in this example would cause the controller to consider a difference between the target and the current position greater than 400nm to be too large. If this large of an error were detected, the controller would re-engage the move algorithm to place the position error back inside of the Finish Error (PC) limit.

3.6 Halt

Command: HALT

Shortcut: \ (the backslash character)

Format: HALT

Function: This command will stop all active motors.

Reply: If there are no errors, a positive reply of ":A" will be returned. If the "HALT" command is given while a commanded move is *in motion*, the controller will reply with the :N-2I error.

Example: HALT
:A

3.7 Here

Command: HERE
 Shortcut: H
 Format: HERE axis=*position* [axis=*position*] [axis=*position*]
 Function: Assign the specified number to the axis's current position buffer. The unit of measurement is in tenths of microns. This defines the current position to be a specific distance from the origin (0), i.e., the origin may change.
 Reply: If there are no errors, the positive reply “:A” will be sent back from the controller.
 Example: H X=1234 Y=4321 Z
 :A

The X position will change to 123.4 microns from the origin, Y will change to 432.1 microns, and the Z will be zeroed.

3.8 Home Motors

Command: HOME
 Shortcut: ! (the exclamation mark character)
 Format: HOME axis [*axis*] [*axis*]
 Function: Move specified axis motors toward their *HOME* position. The default location for the *HOME* position (1000 mm) is far past the positive limit of the stage travel. If a hardware or firmware limit switch is encountered, the motor will stop.
 Reply: If there are no errors, an “:A” is returned.
 Example: ! X Y Z
 :A

The X and Y-axis motors will start moving towards the *HOME* position. A HALT command can stop the motors.

Note: The stage will be positioned at the limit switches or at the previously defined *HOME* position at the completion of this command. See SETHOME.

3.9 Joystick Control

Command: JOYSTICK
 Shortcut: J
 Format: JOYSTICK [X±] [Y±] [Z±] or JOYSTICK [X=*dev*] [Y=*dev*] [Z=*dev*]
 Function: This command enables (+) or disables (–) the input from the default manual control device for the axis (joystick or knob).
 Reply: If there are no errors, the positive reply “:A” will be returned from the controller.
 Example: J X+ Y+ Z-
 :A

The above command enables the default X and Y joystick control and disables the Z control knob.

3.10 Move Motor Absolute

Command: MOVE
 Shortcut: M
 Format: MOVE axis= *position* [axis= *position*] [axis= *position*]
 Function: Move one or more axis motors to an absolute position. The unit of measurement is in tenths of microns. If no position is specified, a position of 0 (the origin) is assumed.
 Reply: A positive reply of “:A” is sent back when the command is received correctly. Reception of the reply does not mean the end of execution, and the command STATUS can be used to determine if the move has been completed.
 Examples: M X=1234 Y=4321 Z
 :A

The controller will move the X-axis to position 123.4 microns from the origin using the maximum set speed (see SPEED). Simultaneously, it will move the Y-axis to position 432.1 microns, and the Z-axis to the zero (0) position. During this movement, the Joystick and Encoder inputs will be locked-out and cannot alter the target positions entered. The motors will stop when they have reached their target or when their limit switch is encountered. To stop the motors during a serial MOVE command, use the HALT (\) command.

3.11 Move Motor Relative

Command: MOVREL
 Shortcut: R
 Format: MOVREL axis=*distance* [axis=*distance*] [axis=*distance*]
 Function: Move one or more axis motor a distance relative from its current position. This command is very similar to the MOVE command. The unit of measurement is also in tenths of microns.
 Reply: A positive reply of “:A” is sent back when the command is received correctly. Reception of the reply does not mean the end of execution, and the command STATUS can be used to determine if the move has been completed.
 Examples: R X=1234 Y=-321 Z
 :A

The controller will move the X-axis an additional 123.4 microns in the positive direction at the maximum set speed (see SPEED). Simultaneously, the Y-axis will move 32.1 microns in the negative direction, while the Z-axis will not move at all.

During this movement, the Joystick and Encoder input will be locked-out and cannot alter the target positions entered. The motors will stop when they have reached their target, or if their limit switch is encountered. To stop the motors during a serial MOVREL command, use the HALT (\) command.

3.12. Motor Status Byte Read (hexadecimal)

Command: RDSBYTE
 Shortcut: RB
 Format: RDSBYTE axis [*axis*] [*axis*]
 Function: Requests the MS-2000 to respond with the Status Byte. The number is one byte, which can be broken down into 8 bits that represent the following internal flags:
 Bit 0: 0 = No commanded move is in progress. 1 = A commanded move is in progress. This bit is synonymous with the STATUS command. If the bit is set, then STATUS returns 'B', otherwise STATUS returns 'N'.
 Bit 1: 0 = The axis is disabled. It can be reenabled by one of the following: High Level command MC <axis>+, cycling the clutch switch for the Z-axis, Low Level StartMotor command (hex 47), or a system reset. This feature is available in versions 6.2c and later; 1 = The axis is enabled.
 Bit 2: 0 = Motor is inactive (off), 1 = Motor is active (on).
 Bit 3: 0 = Joystick/Knob disabled, 1 = Joystick/Knob enabled
 Bit 4: 0 = Motor not ramping, 1 = Motor ramping
 Bit 5: 0 = Ramping up, 1 = Ramping down
 Bit 6: Upper limit switch: 0 = open, 1 = closed
 Bit 7: Lower limit switch: 0 = open, 1 = closed
 Reply: : <byte as hexadecimal>
 Examples: RB X
 :<0x8A>
 RB X Y
 :<0x8A><0x02>

The X-axis example value of 0x8A means the following:

B7: 1 - X Axis is at its lower limit
 B6: 0 - X Axis upper limit switch open
 B5: 0 - Ramping down, if ramping
 B4: 0 - Motor not ramping
 B3: 1 - Joystick/Knob is enabled
 B2: 0 - Motor power is off.
 B1: 1 - X Axis is enabled
 B0: 0 - No commanded move is in progress

Note: Motor power can be on while a commanded move is not in progress and the stage appears not to be moving. This happens when the motor is either making a final adjustment to a commanded move or when it is applying a force to maintain the stage position.

3.13. Motor Status Byte Read (decimal)

Command: RDSTAT
 Shortcut: RS
 Format: RDSTAT *axis* [*axis*] [*axis*]
 Function: Same as RDSBYTE, except the data is returned in ASCII decimal format.
 Example: RS X
 :A 138

3.14 Reset

Command: RESET (or REMRES)
 Shortcut: ~ (this is the tilde character)
 Format: RESET (or REMRES)
 Function: This command causes the controller to do a software reset. A software reset reinitializes all variables back to their pre-assigned values.
 Reply: If there are no errors, a positive reply of “:A”, followed by the startup sequence.
 Example: ~
 :A

3.2. Remote Key Readings

Command: REMKEY
 Format: REMKEY
 Function: There are up to 4 switches on the Tango joystick input device. This command is used to read status of these switches. The command will always return a value in a one-character string form. The values range from 0 to 4. If there is no switch closure is detected previous to this command the string character "0" is returned. Otherwise string characters from "1" to "4" is returned, representing the pressed switch number.
 Reply: A positive reply is sent back when command is received correctly. Reception of the reply means the end of execution. Return value is included in the reply.
 Example: REMKEY
 :A 0 will mean no switch pressed since last inquiry.
 :A 2 switch number 2 was pressed since last inquiry.

3.15 Save

Command: SAVESET or SAVE
 Shortcut: SS
 Format: SAVESET Z - saves settings to flash memory
 SAVESET Y - restores previously saved settings after a SAVESET X
 SAVESET X - will reload factory defaults upon next power-up
 Function: SAVESET allows the user to save current parameters settings to Flash memory.
 Reply: Upon the start of execution of this command, the controller will reply with a “:”.
 When the execution is complete, an “A” will follow the colon.
 Note 1: During the time interval between the “:” and the “A”, no serial or manual moves should be given.
 Note 2: In Versions 6.1u and later (see VERSION command), limit settings (see SETLOW and SETUP) are saved if and only if the SAVEPOS command is issued *after* the command SAVESET Z.
 Example: SS Z Saves current settings to flash memory.
 :A

3.16 Speed

Command: SPEED
 Shortcut: S
 Format: SPEED [X=*maximum_speed*] [Y=*maximum_speed*] [Z=*maximum_speed*]
 SPEED X? [Y?] [Z?]
 Function: Sets the maximum speed at which the stage will move. Speed is set in millimeters per second. Maximum speed is = 7.5 mm/s for standard 6.5 mm pitch leadscrews.
 Reply: If there are no errors, a positive reply of “:A” is sent back.
 Example: S X=1.23 Y=3.21 Z=0.2
 :A

In the example, the X-axis maximum speed is set to 1.23 mm/s, the Y-axis is set to 3.21 mm/s, and Z-axis is set to 0.2 mm/s.

3.17 Spin

Command: SPIN
 Shortcut: @
 Format: SPIN X=*rate* [Y=*rate*] [Z=*rate*]
 Function: Tells controller to ‘spin’ the motor of specified axis at a rate expressed as its DAC value, a bit value from 0 to 128.
 Reply: If there are no errors, a positive reply of “:A” is sent back.
 Example: @ X=100 Y=-100 Z
 :A

This example shows a command that will instruct the X-axis turn at a motor rate of 100 DAC bits in one direction, the Y-axis at the same rate but in the other direction, and stop any rotation or motion of the Z-axis.

NOTE: To stop rotation, give a value of zero, or just the type the axis letter without an assignment as shown in the example above, or use the HALT (\) command.

NOTE: The HALT command will not return an *:N-2I* when stopping a SPIN command.

3.18 Status

Command: STATUS
 Shortcut: /
 Format: STATUS
 Function: Inquires regarding the motor status of all axes. Queries the controller whether or not any of the motors are still busy moving following a serial command. Using the shortcut / is the preferred method for rapid polling of the controller for a busy state. The / is handled quickly in the command parser.
 Reply: The positive reply can come in two forms:
N - there are no motors running from a serial command
B - there is a motor running from a serial command
 Example: MOVE X=12345
 :A
 STATUS
 B
 /
 N

In this example, the command MOVE started the X-axis moving towards the position 1.2345 millimeters from the origin. The first STATUS command returned a “*B*” showing that the motor is still busy moving towards the target. The second time, the STATUS command returned an “*N*” signifying that the MOVE command is finished and there is no longer any motor movement.

3.19 Units Multiplier

Command: UM (Units Multiplier)
 Shortcut: UM
 Format: UM [X=n] [Y=n] [Z=n]
 Function: Specifies the multiplier for most serial commands such as MOVE and WHERE.
 Default values are 10000 (/mm), setting the default input scaling to 0.1? m/count.
 The UM value always returns to the default upon power up and cannot be saved permanently using the Save Settings command.
 Reply: If there are no errors, a positive reply of “:A” is returned.

3.19 Version

Command: VERSION
 Shortcut: V
 Format: VERSION
 Function: Requests controller to report which firmware version it is currently using.
 Reply: If there are no errors, a positive reply of “:A” will be returned, followed by the version number.
 Example: V
 :A Version: USB-8.6a

3.20 Wait

Command: WAIT
 Shortcut: WT
 Format: WAIT [X=msecs] [Y=msecs] [Z=msecs]
 Function: Sets the length of time *msec*, in milliseconds, the controller will pause at the end of a move. The Busy status is not cleared during this Pause state. During the Pause state, the servo loop remains actively attempting to position the axis on target.
 Example: WT X=20
 :A
 Sets the wait time for the X-axis to 20 ms.

3.21 Where

Command: WHERE
 Shortcut: W
 Format: WHERE *axis* [*axis*] [*axis*]
 Function: Returns the current position of the motor for the axis specified.
 Reply: If there are no errors, a positive reply of “:A” will be followed by the current position, in tenths of microns.
 Example: W X Y Z
 :A 1234.5 432.1 0 (in ASI mode)
 :A 1234 432 0 (in LEP mode)

In this example, X is 123.45 microns from the origin, Y is 43.21 microns from the origin, and Z is sitting on the origin.
 Notes: No matter which order the X, Y, and Z's are specified in the WHERE command, the reply will always be in the order X, Y, Z. The reporting precision of the WHERE command can be changed with the Setup Control Commands (below). Default includes a single fractional digit, which represents 10 nanometer precision. If fractional decimals cannot be handled by the user's software, use the appropriate Setup Control Command (below) so only integer data is returned (100 nanometer precision).

3.21 Who

Command: WHO
 Shortcut: N
 Format: WHO
 Function: Inquires the controller to reply with its name. Allows computer software to automatically determine what stage instrument is attached at the end of the serial line.
 Reply: If there are no errors, the Tango will reply with a positive response of “:A”, followed by its name.
 Example: N

:A ASI-MS2000-XYBR-Zs-USB

Document Revision History

No.	Revision	Date	Changes	Remarks
01		7th Sept 2009		birthday
	A			
	B			
	C			