Motorized Microscope Stage 4400LS

Contents

Introduction

The Conix Motorized Stage 4400LS is designed to allow the electronic control of the stage and a Z-focus system.

FEATURES OF THE MOTORIZED STAGE 4400LS:

- Natural feeling joystick control of speed, with speed proportional to joystick deflection. This permits easy selection of speeds from as slow as single-step "jogging" to full turbo traverse speed

- Three user-selectable joystick speed sensitivities

- Cast aluminum controller case shields against RFI radiation from internal microprocessor

- Table-top control unit size is 7"D x 10.5"W x 3"H

- LED display of X, Y, and Z Coordinates

- RS232-C serial communications

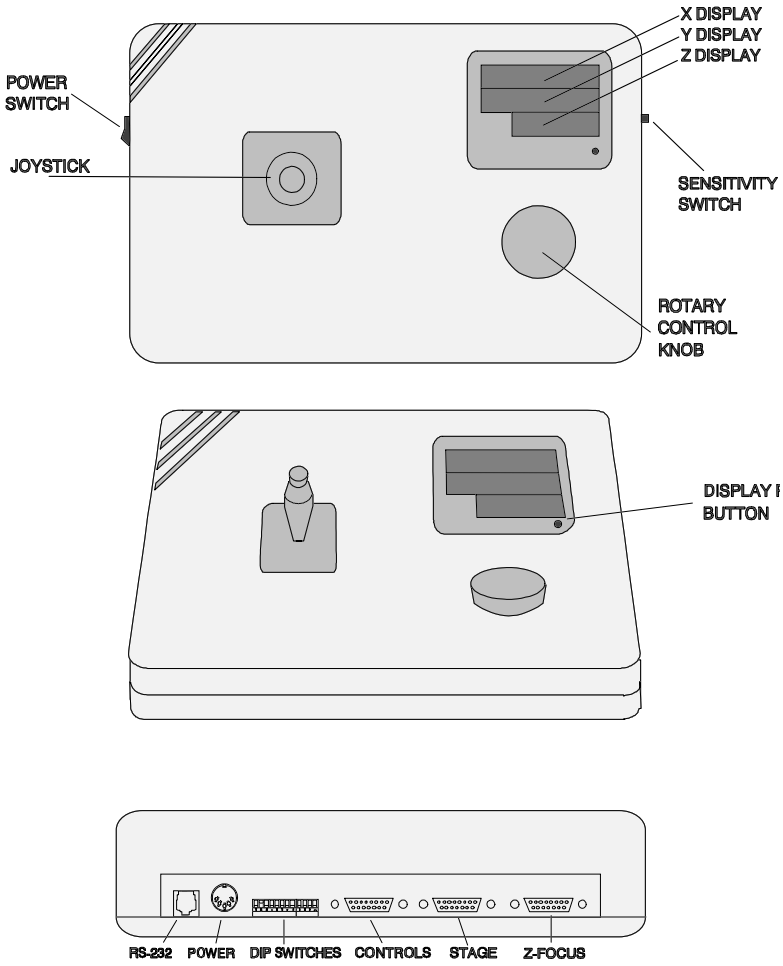
- Maximum speed is in excess of 25 mm/sec

- 3 Axis coordinated motion.

- Remote focus of the Z-Ax is via rotary knob on controller

- Adjustable limit stops.

Control Unit



Installation

Installation of Motorized Stage 4400LS:

Install the stage onto your microscope stand. Hook up the stage cable by plugging it into back of the controller. Plug one end of the Z-Focus cable into the back of the controller and the other end into the Motor Mount Unit.

Check to see if stage hits the neck of the stand when pushed to the extreme rear limit. If it hits, adjust the rear limit stop so the stage engages the stop before hitting the neck. To adjust the limit stop, use the 0.05" allen wrench to loosen the set serew on the stop. Slide the stop to the desired position and retighten the set screw.

CAUTION: If a limit stop is moved too close to the outside edge of the stage, the physical limit may be reached before the limit stop is encountered. If that happens, a racheting noise will be heard and an error will occur. Readjust the limit stop in, toward the center of the stage, until limit stop is encountered.

Connect power cables.

Plug power supply cable into the round power connector on back panel of stage controller. Be sure to connect the power supply to the controller prior to plugging it into the wall and prior to turning on the unit. Plug the power supply into a 120VAC 60Hz receptacle.

Installation of the Motor Mount Unit

(1) Remove the fine focus knob that has the looking ring (may be on either side depending on microscope) from the microscope using the shorter of the two allen wrenches included in the parts package. Retain the spring washer on the fine focus shaft.

(2) Attach the pulley to the fine focus shaft with the pulley groove toward the microscope stand. Align pulley set screw to bear on the flat portion of the fine focus shaft. Use the longer of the two allen wrenches supplied in the parts package. Be sure to press the right fine focus knob and pulley towards each other while securing the pulley to compress the spring washer and ensure proper pulley position.

NOTE: Pulley must be installed with pulley groove towards the microscope stand for correct operation of the accessory.

(3) Attach the Motor Unit to the Coarse Tension Adjusting Ring of the coarse focus knob with the two thumb screws.

NOTE: Some models microscope stands have a small access plate near the Coarse Focus Knob, toward the rear of the stand. This access plate and screws must be removed as it will prevent the Motor Unit from mounting flat against the microscope stand.

NOTE: Make sure you plug the power connector into the control unit prior to plugging the supply into a wall receptacle.

Operation

Motorized Operation of the Stage 4400LS.

1) Turn on Motorized Stage Controller by flipping power-on switch on left side of controller.
2) The 3-position Speed Select Switch (located on right side of controller) allows you to change the speed of the stage relative to the deflection of the joystick, slow, medium, fast.
3) Moving the joystick to the right moves the stage rightward, and moving the joystick to the left moves the stage leftward, etc. If the direction of the stage is opposite of what you desire, you can change the dip switch setting (refer to Dip Settings section of this manual) to invert the orientation of either the X-joystick or Y-joystick motion.
4) Pushing the joystick to an extreme position in one direction causes the stage to move quickly, while pushing gently on the joystick causes the stage to move relatively slowly. The speed of the stage is proportional to the deflection of the joystick.
5) Pressing the button on the joystick causes the stage to move many times faster. You can still control the speed by the amount of deflection from the center of the joystick.

NOTE: On power up the computer senses the position of the joystick, this location becomes the 'dead zone'. Do not deflect the joystick during power up.

Computer Control

Communication Specifications

The Stage 4400LS Communications interface is a interface between a host computer and the controller. The communications is established through an RS-232C serial connection. The programming protocol is with text (standard ASCII alpha-numeric characters), along with some special control characters such as carriage returns, spaces and tabs. The controller responds to a set of built-in commands with unique names. The commands can be executed by simply sending the command name with some parameters (if required). the controller will respond in ASCII and may include the result requested.

General Format Of Commands

Each line sent to the Controller should have a command and be terminated with a carriage return character. The first item on the line should be the command. Each line can contain only one command and the Controller's commands are not case sensitive. The allowed commands are listed below. After the command are the parameters, some commands have no parameters. And finally, each command must be terminated with a carriage return character. The carriage return indicates to the Controller the end of a command. The specific items can be separated with white space characters (such as spaces, tabs). The entire command string cannot exceed 40 characters.

(command) [*data*] <cr>

where:

(command)	any valid ASCII command.
[data]	ASCII numeric data (if applicable).

For Example:

Command:	**Where Z<cr>**
	or W Z<cr>
Response:	**:A 1002<cr>**

For Example:

Command:	**Where X<cr>**
	or W X<cr>
Response:	**:A 6000<cr>**

Response

:A <*DATA*><cr>	Everything is ok <*returned data*>
:N <ERROR CODE><cr>	Error.

Every command returns a response: The response is in the form of a colon followed by a status character (either an A or N). The colon is sent by the Controller as soon as the command is received. The status character is not sent until the function has completed (i.e. after the motor has moved/stopped). Do not send another command until the last function has been completed and returned a response. If for some unknown reason the Controller does not respond with a colon, then the command was not received properly (due to communications problems) and the command must be resent. In this case, the Controllers internal buffer must be emptied by sending an ESC character (ASCII 27). This is necessary since your last command may have been partially received and may still reside in the controller's internal buffer. It is not a bad idea to send an ESC character before every command, but it is not necessary.

Examples:

command:	**M Z=1001<cr>**	(move to location 1001)
response:	**:A <cr>**	(everything is okay)
command:	**W Z <cr>**	(where is z-axis?)
response:	**:A 1001 <cr>**	(z-axis position is 1001)

command:	**AQRST<cr>**	(an illegal command)
response:	**:N -1 <cr>**	(error code -1)

PRESENTLY ASSIGNED ERROR CODES

-1	unknown command

ASCII Commands

Halt Motor: (Special Interface requirements)

format (ASCII Only):

HALT

The ASCII version of this command behaves differently then the hex code version. The ASCII version like all other ASCII commands is only interpreted after the previous command is completed. This makes the ASCII form of the command less useful then the hex code version. It still may be used.

Response (ASCII Only):

A positive response is sent back immediately after the command is completed.

:A

Hex code: 0x7D (HEX Only)

The hex code version of this command is interpreted differently then standard commands. The moment the processor receives the hex code it stops the motors. DO NOT SEND a line terminator, it is then interpreted as an empty string, which results in an ':N -1 Unknown Command' ERROR. This command also flushes the internal receive buffer.

There is no response from this command itself, and if a previously entered command has been halted the normal response from that command will be returned.

ASCII Commands

Set Current Location:

format: **HERE X=? Y=? Z=?<cr>**
or **H X=? Y=? Z=?<cr>**
HERE X Y Z

This command will change the internal (to the controller) location of the X-axis, Y-axis, and Z-axis, respectively. This will effectively adjust the location of the origin.

Response:

A positive response is sent back immediately after the command is received.
:A<CR>

Example:

HERE X=1000 Y=1500 Z=2000 <cr>

The current locations of the X-axis, Y-axis, and Z-axis become the 1000 position, the 1500 position, and the 2000 position, respectively. the actual location depends on the setting of units (which Units the system is currently using).

This command can also take the form of simply adjusting any one or two of the axes.

Example:

HERE X=1000 Y=1500<cr>

The current locations of the X-axis and Y-axis become the 1000 position, and the 1500 position, respectively.

ASCII Commands

Move to limit switches:

format: **HOME <cr>**

This command will move the stage to the upper limit switches. (This command only works with stages that have limit switches). Then the system position is set to Zero. This command moves quickly to the limits, hits them, backs off slightly and reapproaches them at a slower velocity. the slower velocity helps ensure the position is as acurate as possible.

Response:

A positive response is sent back immediately after the command is complete.
:A<CR>

Example:

HOME <cr> The stage moves to the limit switches.

ASCII Commands

Inbit:

format: **INBIT1<cr>**, **INBIT2<cr>**, **INBIT3<cr>**

These commands will retrieve the state of the INBITS. Each INBIT (port Controls pin 13,6,14) has a 4.7K pull-up resistor and is active low. The disconnected state is inactive (+5VDC). The inputs are standard +5VDC TTL levels. DO NOT EXCEED +5.5VDC.

Response: A positive response is sent back immediately after the command is complete with the current state.
:A ON/OFF<CR>

Example: **INBITS1 <cr>** Get the current state of INBIT1

ASCII Commands

Enable/Disable the joystick:

format: **JOYSTICK ENABLE/DISABLE<cr>**

This command will enable or disable the joystick from operation. if the joystick is disabled, any movement of the joystick will be ignored by the system.

Response:

A positive response is sent back immediately after the command is received.

:A ENABLE<CR> The current status of the joystick is ENABLE.

ASCII Commands

Change the Joystick Position:

format: **JSPosition X, Y** **X, Y in the range {-127..0..127}**

With this command, you can quary the current position of the joystick or you can manually set a new position. to set a new position, you must first disable the joystick. This will disable the computer from updating the internal registor for the joystick position and allow an external computer to set them manually. This command allows an external computer to simulate the function of the internal joystick. While the computer is moving under joystick control, an external computer may quary the current location of the xyz system. (the more you quary the less responsive the joystick becomes.)

Response:

The values returned are the current location of the joystick.

:A 0 10<CR> The current position (x,y) of the joystick.

Special Hex code: 0xCF 0xXX 0xYY

This command has a special hex code version. you must send all three bytes uniterrupted. The form of the value is 7 LSB {0-0x7F} hold the position and the MSB the sign bit. (this is not normal for negative values -1 = 0x81).

ASCII Commands

Change the Joystick sensitivity:

format: **JSScale XX** **XX in the range of {1..255}**

This command will change the speed/sensitivity of the joystick . The default is 1. The smaller the number the faster the stage moves. This allows more control of the sensitivity than just the side slide switch. The side slide switch still will function properly, but the fastest motion will be smaller as this setting increases. Each position of the side slide switch is approximately a factor of two different.

Response:

A positive response is sent back immediately after the command is received.

:A 1<CR> The current setting of the joystick sensitivity.

ASCII Commands

Set Min Speed:

format: **MINSPEED<cr>**
This command sets the start up speed for movement of the stage. The operator can choose a value from 50 to 60,000, where a larger number signifies a slower MINSPEED.

Response: A positive response is sent back when the command is complete with the current setting.
:A XXX<CR>

Example: **MINSPEED 1000<cr>** This will set the MINSPEED to :A 1000
This command can also be used to simply view the current MINSPEED setting.

Example: **MINSPEED<cr>**

Response: **:A 1000<cr>**

ASCII Commands

Move Absolute:

format: **MOVE X=? Y=? Z=?<cr>**
or **M X=? Y=? Z=?<cr>**
MOVE X Y Z

This command will move the X-axis, Y-axis, and Z-axis to the respective locations in the current units. The current units may be steps, millimeters, or inches.

Response: A positive response is sent back when the command is complete.
:A<CR>

Example: **MOVE X=1000 Y=1500 Z=2000<cr>**
This will move the x-axis, y-axis, and z-axis to +1000, +1500, and +2000 steps from the origin, respectively. The order of the X=? Y=? Z=? is irrelevant. For example an alternate command would be MOVE Y=1500 Z=2000 X=1000.

This command can also take the form of simply moving any one or two of the axes.

Example: **MOVE Z=1000 <cr>**

This will move the Z-axis to +1000 steps from the origin.

Special Hex code:

Each time you send one of these Hex Codes, the X, Y or Z axis moves a little bit.

Step size:	1	2	4
X -Axis (-):	0xD0	0xD1	0xD2
X-Axis (+):	0xD3	0xD4	0xD5
Y-Axis (-):	0xD6	0xD7	0xD8
Y-Axis (+):	0xD9	0xDA	0xDB
Z-Axis (+):	0xDC	0xDD	0xDE
Z-Axis (-):	0xDF	0xE0	0xE1

ASCII Commands

OutBit:

format: **OUTBIT1<cr> or OUTBIT2<cr>**

This command will set or retrieve the state of the output bits (controls port pins 3 & 11). These Bits are active low.

Response:

A positive response is sent back when the command is complete with the current state.
:A ON/OFF<CR>

Example: **OUTBIT1 ON <cr>**

This command will set OUTBIT1 active (low)

Response: **:A ON<cr>** The current OUTBIT1 state is ON

ASCII Commands

Rampslope:

format: **RAMPSLOPE<cr>** Range (1-255)

This command will set the rate at which the velocity changes. the larger the number, the slower the change in velocity.

Response:

A positive response is sent back when the command is complete.
:A<CR>

Example: **RAMPSLOPE 100 <cr>**

This command will set the current RAMPSLOPE to 100.

Response: **:A 100<cr>** The current RAMPSLOPE is 100

ASCII Commands

Move Relative:

format: **RELMOVE X=? Y=? Z=?<cr>**
or **RM X=? Y=? Z=?<cr>**
RELMOVE X Y Z

This command will move the X-axis, Y-axis, and Z-axis a relative amount of ?,?,? from the current location in number of units.

Response:

A positive response is sent back when the command is complete.
:A<CR>

This command can also be used to relatively move any one or two of the axes.

Example:

RELMOVE Z=1000 <cr>

This command will move the focus (Z-axis) 1000 units from the current location.

ASCII Commands

Reset the system:

format: **RESET<cr>**
hex code: 0x7f

This command will reset the system, as if the power had been turned off. When the hex code is used this command does an automatic power on reset regardless if a command is being executed. No response is given if hex code is used.

Response: A positive response is sent back prior to the command being completed; The command responds prior to reset.
:A<CR>

Example: **RESET<cr>**

ASCII Commands

Rotate the transformation matrix.

format: **ROTATE Angle<cr>**

Refernce the Matrix Command. This command will change the internal matrix so as to rotate the X & Y axis.The result of this command is that the joystick and computer commands are not longer parallel to the stage axis. The Angle must be an integer. Range {0, 359}

Response:

:A CurrentAngle<CR>

Example: **Rotate 45**

Response: **:A 45<cr>**

ASCII Commands

ScanH (Please reference **SETSCAN**):

format: **SCANH Rows Columns<cr>**

The purpose of this command is to scan the stage in the horizontal direction. Prior to this command, the stage must be moved to the reference point. As defined by:

Ref X = Edge of first column - **Margin**
Ref Y = Center of first row.

The scan starts from the current location and is defined by motion in the X axis
Xmotion = **Columns***XPitch + 2***Margin**
During the Motion, The output bit 1 (**ExtOutBit1**, Controls pin 3) becomes active when the stage is in the center of each Column +- the **Window**.
At the end of the X Motion the Y axis moves **YPitch** amount. Output bit 2 (**ExtOutBit2**, Controls pin 11) becomes active during the Y motion. This is repeated for each **Row.**

Response: A ':' is returned immediately, then a positive response is sent back when the command is complete.
:A<CR>

ASCII Commands

ScanV (Please reference **SETSCAN**):

format: **SCANV Rows Columns<cr>**

The purpose of this command is to scan the stage in the Vertical direction. Prior to this command, the stage must be moved to the reference point. As defined by:

Ref X = Center of first Column.
Ref Y = Edge of first row - **Margin**

The scan starts from the current location and is defined by motion in the Y axis
Ymotion = **Rows***YPitch + 2*Margin
During the Motion, The output bit 1 (**ExtOutBit1**, Controls pin 3) becomes active when the stage is in the center of each Row +- the **Window**.
At the end of the Y Motion the X axis moves **XPitch** amount. Output bit 2 (**ExtOutBit2**, Controls pin 11) becomes active during the X motion. This is repeated for each **Column.**

Response: A ':' is returned immediately, then a positive response is sent back when the command is complete.
:A<CR>

ASCII Commands

SetScan (Please refernece ScanH & ScanV)

format: **SETSCAN XPitch YPitch Window Margin<cr>**

This command sets up the parameters for the ScanH and ScanV commands.Please refer to each of these commands to determine how the parameters are used.

Response: A positive response is sent back immediately after the command is received with the current values
:A 1000 1000 100 1000<CR> (defaults)

ASCII Commands

Speed XY axis only:

format: **SPEED<cr>**

This command will tell the operator the current value of the maximum speed of movement for the HOME and MOVE commands. The range of speed is 1 to 65535, with a larger number representing a slower speed.

Response:

A positive response is sent back immediately after the command is received.
:A<CR>

Example: **SPEED<cr>**

Response: **:A 100<cr>** The maximum speed is set at 100.

ASCII Commands

Change Units:

format: **UNITS ??<cr>**

This command will change the current units that are displayed on the controller. The units can be changed to millimeters, inches, or steps.

Response: A positive response is sent back immediately after the command is received.
:A<CR>

Example:	**UNITS MM<cr>**	The units are changed to millimeters.
Example:	**UNITS STEPS<cr>**	The units are changed to steps.
Example:	**UNITS INCH<cr>**	The units are changed to inches.

NOTE: All of the commands return and accept responses in current units. The dip switch settings determine the initial power up units, but may be overridden by this command.

ASCII Commands

Get Version:

format: **VERSION<cr>**
Hex Code: 0x7c

This command returns the current version code of the firmware.

Response:

A positive response is sent back when the command is complete.
:A version j.x.x<CR>

ASCII Commands

Get Current Location(s):

format: **WHERE X Y Z <cr>**
or **W X Y Z <cr>**

This command will query the controller for the current location of the axes.

Response: A positive response is sent back immediately after the command is received.

:A ????<CR>	The current location in number of units.

Example:

WHERE X Y Z <cr>	The current location is sent back from the controller.

Response:

:A 500 4000 300<CR>	The current location in number of units.

This command can also be used to query the controller for the location of any one or two of the axes.

Example:	**WHERE Y <cr>**	The current location is sent back from the controller.
Response:	**:A 4000<CR>**	The current location of the y-axis in number of units.

ASCII Commands

Get Current Accessory:

format: **WHO<cr>**

This command will query the controller for the current accessory being used. In this case it will be the Stage 4400 system

Response:

A positive response is sent back immediately after the command is received.
:A<CR>

Example:

WHO<cr>

Response: **:Stage 4400LS System<cr>**

ASCII Commands

Set Zero Of Origin:

format: **ZERO <cr>**

This command will set the origin to the current location. This results in the current location being the new ZERO (origin).

Response:

A positive response is sent back immediately after the command is received.
:A<CR>

Example: **ZERO <cr>** The current location becomes the ZERO position.

Dip Switches

The default setting for dip switches 1 thru 12 are as follows:
Up, Up, Down, Down, Down, Down , Down, Down, Up, Up, Up, Up.

Dip Switch definitions:

Switch 1 - RS-232C 0
Switch 2 - RS-232C 1
Switch 3 - N/A Leave Down
Switch 4 - Right/Left hand Operations (z-knob orientation)
Switch 5 - Z Doubler (Up=200um/rev down=100um/rev)
Switch 6 - Z Orientation (Up=Right Mount, Down=Left Mount)
Switch 7 - Steps/ Units
Switch 8 - Units Millimeter/Inch
Switch 9(1) - N/A Leave Up
Switch 10(2) - N/A Leave Up
Switch 11(3) - Y-Axis Orientation
Switch 12(4) - X-Axis Orientation

Baud Rates: 300, 1200, 2400, 9600

S1	S2	Baud Rate
Up	Up	9600
Down	Up	2400
Up	Down	1200
Down	Down	300

Right/Left Handed Operations:

This selects the direction of rotation of the motor with respect to operator motion of the Rotary Control Knob.

S3	ORIENTATION
Down	Right
Up	Left

NOTE: Switches are only interrogated at power up, so before making adjustments turn the unit off.

For warranty repair return the product to the warranty department of Conix Research Inc. at the following location:

Conix Research Inc.
857 28TH
Springfield, OR 97477
(541) 747-8512

You should provide a written description of the problem with the unit. Consumer must prepay all postage, shipping, insurance, and delivery costs associated with the return of the product.

For more information refer to the Conix Research Inc. Limited Warranty Card provided with this product.

CONIX RESEARCH Inc
857 28TH SPRINGFIELD, OR 97477
PHONE (541) 747-8512 FAX (541) 747-8528