CARE AND MAINTENANCE

Cleaning the painted or plastic surfaces.

Avoid the use of any organic solvents (such as thinner, alcohol, ether, etc.) to clean the painted or plastic surfaces of the accessory. Instead, use a mild solution of soap and water or a neutral detergent.

Never attempt to dismantle.

Never attempt to dismantle the instrument, thereby avoiding the possibility of impaired operational efficiency or accuracy. Contact an authorized Conix distributor for service and repair.

When not in use.

When not in use, turn off power to the accessory with the power switch on the Controller Module. When unit is not in use for an extended period, unplug transformer from its 120VAC 60Hz outlet.

Handle with care.

Handle all equipment with care. Install equipment in an environment with limited exposure to direct sun light, dust, high temperatures, humidity, and vibration. Avoid any machanical or electrical shocks.

NOTE: We reserve the right to make alterations in design or function. For this reason, specifications or illustrations in this manual may not conform with models in current production.

Well Plate Positioner (Model #200)

Contents

Introduction 2

Control Unit 3

Installation 4

Installation of Motorized Stage 4

Connection of power cables 4

Operation 5

Motorized Operation of the Well Plate Positioner 5

Computer control 6

Communication Specifications 6

General Format Of Commands 6

ASCII Commands 9

Introduction

The Conix Well Plate Positioner is designed to allow the electronic control of the stage (holding multiple well plates) and a Z system.

FEATURES OF THE WELL PLATE POSITIONER:

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

- RS232-C serial communications

- Maximum speed is in excess of 25 mm/sec

- 3 Axis coordinated motion.

- 25 micron acuracy with submicron resolution.

-Adjustable limit stops.

-Allows access to entire area of two well plates.

-Mounting block for OEM supplied dispensing unit.

Installation

Plug power supply and RS-232 cables into connectors on back panel of Well Plate Positioner.

Computer Control

Communication Specifications

The Well Plate Positioner Communications interface is a interface between a host computer and the controller. The communications is established through an RS-232C serial connection set at a baud rate of 9600. The format is 8-Bits, 1-Stop Bit, and No Parity. 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 unit 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 unit 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 Well Plate Positioner 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 Well Plate Positioner 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 Well Plate Positioner 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 unit's 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 Positioner'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

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).
Default unit setting is in millimeters.

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

ASCII Commands

To set the internal transformation matrix.

format: **Matrix A11 A12 A21 A22<cr>**

There exists an internal transformation matrix. It is a 4x4 matrix. When a command is given to the controller it is first transformed from the users coordiante system to the internal coordinates using the internal matrix.
X = A11*X' + A12*Y'
Y = A21*X' + A22*Y'

Some basic rules about the internal matrix.
The internal matrix must be unitary. The system uses the fact that for a real matrix, the inverse of a unitary matrix is the transpose.
$I = A*A^t$
Each entry is normalized to 16384. For example, the identity matrix (as initilized by the system) is:
A11=16384, A12=0, A21=0 A22=16384
The range for each entry is {-32768, 32767}
The display system will display a decimal in the X & Y left-most digit when the matrix is not the Identity matrix.

NOTE: The rotate command changes this internal matrix.

Response:

The system will respond with the current value for the internal matrix. If you don't pass any arguments the system will return the current matrix without changing it.

:A 16384 0 0 16384

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

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

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 ':' is returned immediately, then 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>**

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 units currently selected for the motion of the Well Plate Positioner. The units can be changed to millimeters, inches, or steps. Default unit setting is in millimeters.

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.

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: **:Well Plate Positioner<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.

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

VERSION J.3.4