

SOLO

SINGLE-AXIS MOTORIZED MICROMANIPULATOR SYSTEM

WITH USB INTERFACE FOR EXTERNAL CONTROL

OPERATION MANUAL

REV. 1.09B (20220919) (FW V2.55+)



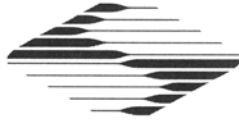
SUTTER INSTRUMENT®

ONE DIGITAL DRIVE
NOVATO, CA 94949

VOICE: 415-883-0128 WEB: WWW.SUTTER.COM
FAX: 415-883-0572 EMAIL: INFO@SUTTER.COM



Copyright © 2022 Sutter Instrument Company. All Rights Reserved.



CE EU Declaration of Conformity

Application of Council Directives:
2014/30/EU (EMC), 2014/35/EU (LVD), and 2015/863/EU (RoHS 3)

Manufacturer's Name:	Sutter Instrument Company	
Manufacturer's Address:	One Digital Drive Novato, CA. 94949 USA Tel: +1 415 883 0128	
Equipment Tested:	SOLO Single-Axis Motorized Micromanipulator System	
Model(s):	SOLO system consisting of SOLO/E (controller & ROE (Rotary Optical Encoder) for user control), with Power adapter (100-240 VAC to 24VDC) (same as, and tested as, a QUAD/E SOLO-25M or SOLO-50M (25- or 50-mm motorized micromanipulator electromechanical, subset of the QUAD/M),	
Conforms to Standards:	EMC Emissions: EN 61326-1:2013, including: EN 55011: 2009 Group 1, Class A; EN 61000-3-2:2015, & EN 61000-3-3:2014 EMC Immunity: EN 61000-4-2:2009, EN 61000-4-3:2011, EN 61000-4-4:2012, EN 61000-4-5:2014, EN 61000-4-6:2014, EN 61000-4-8:2010, & EN 61000-4-11:2004 LVD (Safety): EN 61010-1:2010	
Tested/Verified (as a QUAD system on which the SOLO is based) by:	ITC Engineering Services, Inc 9959 Calaveras Road, PO Box 543 Sunol, CA 94586-0543 USA Tel. +1 925 862 2944 Fax: +1 925 862 9013 Email: itcemc@itcemc.com Web: www.itcemc.com Sutter Instrument	
Test Report(s):	20140120-01R1-Micromanipulator, 20140120-01; SI_EMC_QUAD_20160713; 20200520_REACH and RoHS Compliance Statement	

Sutter Instrument Company hereby declares that the equipment specified above was tested and conforms to the EU Directives and Standards listed above, and further certifies conformation to the requirements of the European Union's Restriction on Hazardous Substances in Electronic Equipment Directive 2015/863 (2011/65/EU Annex II) for RoHS 3.

Project Engineer:

Mark Flaming
President

SUTTER INSTRUMENT®

One Digital Drive, Novato, CA 94949 USA Phone: +1 415 883 0128 Fax: +1 415 883 0572
Email: info@sutter.com Web: <http://www.sutter.com>

DISCLAIMER

The SOLO consists of one electromechanical micromanipulator device and one ROE (Rotary Optical Encoder) with integrated controller. The purpose of the system is for the manipulation at the micro level of micropipettes and probes used in conjunction with a microscope. No other use is recommended.

This instrument is designed for use in a laboratory environment. It is not intended, nor should it be used in human experimentation or applied to humans in any way. This is not a medical device.






Unless otherwise indicated in this manual or by Sutter Instrument Technical Support for reconfiguration, do not open or attempt to repair the instrument.

Do not allow an unauthorized and/or untrained operative to use this device.


Any misuse will be the sole responsibility of the user/owner and Sutter Instrument Company assumes no implied or inferred liability for direct or consequential damages from this instrument if it is operated or used in any way other than for which it is designed.

SAFETY WARNINGS AND PRECAUTIONS

Electrical

- Operate the SOLO using 110 – 240 VAC., 50-60 Hz line voltage. This instrument is designed for use in a laboratory environment that has low electrical noise and mechanical vibration. Surge suppression is always recommended
-  NOTE: There are no user-replaceable fuses in the SOLO system.
-   The SOLO system's power supply consists of an external AC to DC switching power adapter. If the external power adapter is damaged due to a mains over- or under-voltage, it must be replaced.
-   GROUNDING/EARTHING: Proper grounding protects the ROE/controller electronics, reduces/eliminates electromagnetic interference, and improves the safety of the system operator. The ROE/controller provides a socket (labeled GROUND) that accepts a banana plug attached to a suitably gauged insulated wire, the other end of which (alligator clip) connects to a solid, proper ground.

Avoiding Electrical Shock and Fire-related Injury

- Always use the grounded power cord provided to connect the system's power adapter to a grounded/earthed mains outlet (3-prong). This is required to protect you from injury if an electrical hazard occurs.
- Do not disassemble the system. Refer servicing to qualified personnel.
-  To prevent fire or shock hazard do not expose the unit to rain or moisture.

Electromagnetic Interference

To comply with FDA and CE/EU electromagnetic immunity and interference standards; and to reduce the electromagnetic coupling between this and other equipment in your lab always use the type and length of interconnect cables provided for interconnecting the electromechanical devices and ROE/controller (refer to Technical Specifications for more details).

Operational

Failure to comply with any of the following precautions may damage this device.

- This instrument is designed for operation in a laboratory environment (Pollution Degree I) that is free from mechanical vibrations, electrical noise, and transients.
- **⚠ DO NOT CONNECT OR DISCONNECT THE CABLES BETWEEN THE CONTROLLER AND THE MECHANICAL UNITS WHILE POWER IS ON.** Please allow at least 20 seconds after turning the unit off before disconnecting the mechanical units. Failure to do so may result in damage to the electronics.
- Operate this instrument only according to the instructions included in this manual.
- Do not operate if there is any obvious damage to any part of the instrument.
- **⚠ Do not operate this instrument near flammable materials.** The use of any hazardous materials with this instrument is not recommended and, if undertaken, is done so at the users' own risk.
- **⚠ Do not operate if there is any obvious damage to any part of the instrument.** Do not attempt to operate the instrument with the SOLO/M electromechanical manipulator shipping tape in place or severe motor damage may result. When transporting the mechanical manipulator, be sure to reinstall the shipping tape (using masking tape or equivalent only) to the original locations. Failure to do this may result in damage to the motors.
- **⚠ Never touch any part of the micromanipulator electromechanical device while it is in operation and moving.** Doing so can result in physical injury (e.g., fingers can be caught and pinched between the moving parts of the micromanipulator).
- **⚠ If the SOLO system is used in a microinjection environment, please observe the following.** As with most micromanipulation devices, sharp micropipettes can fly out of their holder unexpectedly. Always take precautions to prevent this from happening. Never loosen the micropipette holder chuck when the tubing is pressurized, and never point micropipette holders at yourself or others. Always wear safety glasses when using sharp glass micropipettes with pressure tubing.
- **⚠ Take care to ensure no cables pass close to the SOLO/M electromechanical micromanipulator within the spherical movement limits of all its axes combined.**

Other

- Retain the original packaging for future transport of the instrument.
- Sutter Instrument reserves the right to change specifications without prior notice.
- Use of this instrument is for research purposes only.

Handling Micropipettes

⚠ Failure to comply with any of the following precautions may result in injury to the users of this device as well as those working in the general area near the device.

- The micropipettes used with this instrument are very sharp and relatively fragile. Avoid contact with micropipette tips to prevent accidentally impaling oneself.
- Always dispose of micropipettes by placing them into a well-marked, spill-proof "sharps" container.

TABLE OF CONTENTS

DISCLAIMER	3
SAFETY WARNINGS AND PRECAUTIONS	3
Electrical	3
Avoiding Electrical Shock and Fire-related Injury.....	3
Electromagnetic Interference	3
Operational	3
Other.....	4
Handling Micropipettes.....	4
1. INTRODUCTION	9
1.1 Structure of the SOLO Documentation Package.....	9
1.2 Components of the SOLO System	9
1.3 Overview	10
1.3.1 Features.....	10
1.3.2 Description.....	10
2. INSTALLATION	11
2.1 Mounting Instructions	11
2.1.1 Mounting the SOLO/M to the Stand or Platform	11
2.2 Headstage Mounting	12
2.3 Other Accessories.....	12
2.4 Electrical Connections and Initial Operating Instructions.....	12
2.5 ROE/Controller Rear Panel Controls and Configuration.....	13
2.5.1 Power Switch	13
2.5.2 Configuration Switches	14
3. OPERATIONS	15
3.1 Main Controls and Indicators on the ROE/Controller	15
3.2 Display.....	15
3.2.1 Initial Startup	15
3.3 Control Operations	16
3.3.1 Maximum Positive Position Values:	16
3.3.2 Setting Position for HOME or WORK.....	16
3.3.3 Moving to the Home Position	16
3.3.4 Moving to the Work Position	17
3.3.5 Setting Absolute/Relative Coordinates Mode.....	17
3.3.6 Mode Indications.....	17
3.3.7 Speed Control and ROE Knob Movements (SPEED).....	18
3.3.8 Pausing Home Movements	18
3.3.9 Pausing Work Movements	18
3.3.10 Pulse Mode.....	18
4. EXTERNAL CONTROL	19
4.1 Virtual COM Port (VCP) Serial Port Settings.....	19
4.2 Protocol and Handshaking	19
4.3 Command Sequence Formatting	19
4.4 Axis Position Command Parameters.....	20

4.5 Microsteps and Microns (Micrometers).....	20
4.6 Ranges and Bounds:	21
4.7 Travel Speed	21
4.1 Commands	21
4.1.1 Get Current Position ('c' or 'C') Command	21
4.1.2 Move to Controller-Defined HOME Position ('h') Command.....	22
4.1.3 Move to Controller-Defined WORK Position ('w') Command	22
4.1.4 Move to Specified "Home" Position ('H') Command	22
4.1.5 Move to Specified "Work" Position ('W') Command.....	23
4.1.6 Move to Specified Position ('x' or 'X') Command	23
4.1.7 Set Velocity for all Externally Controlled Movements ('v') Command (Firmware Ver. 2.55+)	24
4.2 Notes.....	24
5. MAINTENANCE.....	28
APPENDIX A. LIMITED WARRANTY	29
APPENDIX B. ACCESSORIES	30
APPENDIX C. TECHNICAL SPECIFICATIONS	31
C.1. SOLO/M.....	31
C.2. SOLO ROE/Controller	31
APPENDIX D. QUICK REFERENCE	33
Manual Operation.....	33
Configuration	33
External Control	33
INDEX.....	39

TABLE OF FIGURES

Figure 1-1. The SOLO system.....	9
Figure 2-1. Side view of SOLO/M showing mounting adapter plate and lock screws.....	11
Figure 2-2. Mounting the SOLO/M on the Adapter Plate	12
Figure 2-3. Rear of SOLO ROE/Controller cabinet	13
Figure 2-4. Configuration switches on rear of SOLO ROE/Controller unit (switch positions shown are factory defaults).....	14
Figure 3-1. Startup screen	15
Figure 3-2. Factory default startup calibrated position	15
Figure 3-3. Maximum positive values.....	16
Figure 3-4. Moving to Home position (screen is amber while moving)	16
Figure 3-5. Factory default Home position.....	16
Figure 3-6. Example Home position defined and saved	16

Figure 3-7. Example Work position.....	17
Figure 3-8. Relative mode.....	17

TABLE OF TABLES

Table 2-1. Configuration DIP switch settings.....	14
Table 3-1. Maximum positive position value of each axis.....	16
Table 3-2. Screen colors and modes.....	17
Table 4-1. USB-VCP interface serial port settings.....	19
Table 4-2. Microns/microsteps conversion.....	21
Table 4-3. Ranges and bounds.....	21
Table 4-4. Travel speeds.....	21
Table 4-5. Get Current Position and Angle ('c' or 'C') command.....	22
Table 4-6. Move to controller-defined HOME position ('h') command.....	22
Table 4-7. Move to controller -defined WORK position ('w') command.....	22
Table 4-8. Move to specified "Home" position ('H') command.....	23
Table 4-9. Move to specified "Work" position ('W') command.....	23
Table 4-10. Move to specified X-axis position ('x' or 'X') command.....	23
Table 4-11. Set velocity command ('v') command.....	24
Table C-1. SOLO cables and receptacles/connectors.....	32
Table D-1. Configuration Switches 1 – 4.....	33
Table D-2. USB-VCP interface serial port settings.....	33
Table D-3. Microns Microns/microsteps conversion.....	34
Table D-4. Ranges and bounds.....	35
Table D-5. Travel speeds.....	35
Table D-6. SOLO external control commands.....	35

(This page intentionally left blank.)

1. INTRODUCTION

1.1 Structure of the SOLO Documentation Package

The SOLO 1-Axis Micromanipulator System is comprised of a ROE/controller, a power adapter, and a SOLO-25 or SOLO-50 stepper-motor-based electromechanical micromanipulator. This manual consists of four parts: This chapter, Introduction, which provides an overview and general description of the SOLO system; Chapter 2, Installation, which describes how to install, set up, and configure all components of the system; Chapter 3, Operations, which describes how to operate the SOLO; Chapter 4, Maintenance, describes how to perform routine and other maintenance of the SOLO; and Chapter 5, Reconfiguration, describes the reconfiguration possibilities of the SOLO system.



Figure 1-1. The SOLO system

1.2 Components of the SOLO System

Carefully remove all components from the shipping container. In addition to this manual, the following should be included:

- SOLO ROE Rotary Optical Encoder input device with built-in controller and external power adapter.
- SOLO-25 or SOLO-50 electromechanical micromanipulator
- 9-pin DSUB cable (connects the ROE/controller to the SOLO/M electromechanical micromanipulator).
- Power adapter
- Power adapter AC mains cable appropriate for your location
- Ground/Earth cable
- USB Cable

IMPORTANT

Once the SOLO system has been unpacked, remove the shipping tape from the various locations on the SOLO/M electromechanical micromanipulator. The shipping tape must be removed before operating the SOLO system. If you need to transport the SOLO/M in the

future, reapply 2 to 3-inch pieces of masking tape to the same locations. Once the tape has been removed, handle the SOLO/M with care. The mechanisms can be damaged if any of the axes are inadvertently moved without the tape in place.

1.3 Overview

1.3.1 Features

- One axis (X) with 25 or 50 mm of travel.
- Sub-micron 100nm resolution
- Digital display indicates coordinates in relative or absolute
- User-friendly, fanless compact controller with ROE preserves bench space
- Push button control of multiple functions – work, home, pulse, and relative
- Robotic home- and work-position moves for easy automated pipette exchange

1.3.2 Description

The **SOLO**, the newest Sutter Instrument motorized manipulator, is easy to use and has one axis. The X axis provides 25 or 50 mm range of motion. The ROE controller has a digital display and keys for Home, Work, Pulse, and Relative. The compact, intuitive controller takes up minimal bench space, is fan-free, and easy to use.

The **SOLO's** ROE provides fine control of electrode position and the rate of rotation of ROE dials for each axis determines the speed of travel. The finest step size is less than 100nm. Four conveniently located buttons on the ROE provide control of all the basic functions you will need in normal operation (Work, Home, Relative, and Pulse).

Press and hold **WORK** (for 3 seconds) to quickly store a work position, tap **HOME** to move all axes to an initial location that is useful for changing electrodes, or press and hold the **HOME** button (for 3 seconds) to memorize a new **HOME** position.

Pressing and holding the **RELATIVE** button for three seconds at any location causes the display coordinates to all zeroes. When activating relative mode, the display turns blue.

To return to viewing the absolute coordinates, tap the **RELATIVE** button to toggle back. Finally, tapping the **PULSE** button causes a 3 μ m advance in the diagonal. This rapid burst of forward motion can assist in sharp electrode cell penetration.

All the electronics, except for a small power supply, are housed within the **SOLO** ROE and no separate controller or computer is required.

In keeping with the compact and simple design, the **SOLO** has no need of computer connectors. If the user desires a computer connection, a USB port is provided on the rear of the **SOLO** ROE.

2. INSTALLATION

When installing the SOLO system for the first time, it is recommended that the components of the system be installed in the following order: SOLO/M electromechanical micromanipulator first, followed by the SOLO/E ROE/Controller.

2.1 Mounting Instructions

The following sections describe how to mount the SOLO/M manipulator to a stand using the mounting adapter plate, how to adjust the pipette angle and how to mount different headstages.

2.1.1 Mounting the SOLO/M to the Stand or Platform

The SOLO/M attaches to the mounting adapter plate using four M3X0.5 hex head locking screws.

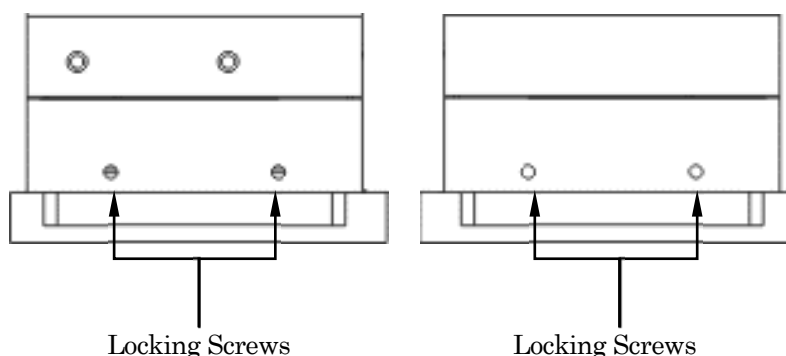


Figure 2-1. Side view of SOLO/M showing mounting adapter plate and lock screws.

The SOLO/M is shipped with the adapter plate in place. It is attached using four tapered pegs, along with four locking screws.

To remove it, first loosen the four hex screws that secure the manipulator to the pegs in the adapter plate. The rear pair is in a similar location in the back of the manipulator. Once the locking screws are sufficiently loosened, lift the SOLO/M upwards from the adapter plate.

Before attaching the adapter plate to the SOLO/M, you need to decide where to position the manipulator on your stand/platform. The stand can be any flat surface carrying $\frac{1}{4}$ -20, 10-32, or M6 holes on one-inch centers (such as a Sutter Instrument MT-series stand or MD series platform).

Examine the space of the platform onto which installation is to take place. Attach the control cable to SOLO/M and move the entire unit around on the platform until the precise desired position is determined. A small bag containing the necessary hardware to attach the SOLO to the stand is included.

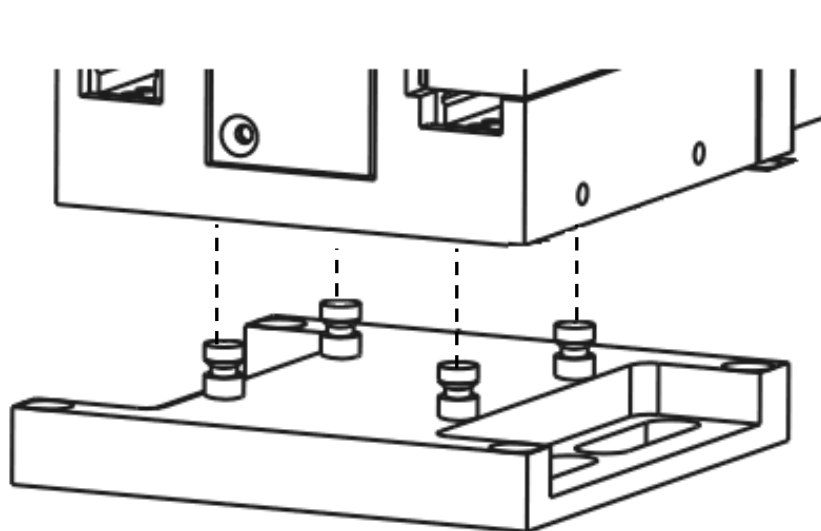


Figure 2-2. Mounting the SOLO/M on the Adapter Plate

Once the plate is mounted, align the pegs on top of the plate with the holes in the manipulator, push the X-axis firmly onto the plate, and re-tighten the locking hex set screws.

2.2 Headstage Mounting

Sutter IPA headstage, Axon headstages 203B or CV-7, and the Heka EPC-10 headstage have an integral dovetail that fits directly into the rotary dovetail slide bracket on the SOLO/M. The dovetail slide bracket on the SOLO/M also supports older Axon and Heka headstages when using the 4" dovetail extension.

Rod-mounted headstages and micro tools are accommodated using a rod clamp that fits into the dovetail (not shown). All the headstage adapters and mounting hardware are included with the manipulator and are shipped in a zip lock plastic bag.

2.3 Other Accessories

One or more accessories may have been ordered and received for mounting the SOLO/M and/or modifying the headstage mount to the manipulator (i.e., rotating base, microscope stage mount, gantry, dovetail extension, etc.). Setup of these accessories is normally covered in documentation accompanying the accessory.

2.4 Electrical Connections and Initial Operating Instructions

Initially, you may want to simply connect the SOLO/M micromanipulator and the ROE/Controller together and try some gross movements in order to get a feel for the controls and how to make simple movements. It is perfectly acceptable to set the manipulators in the middle of a bench top, make all electrical connections and then observe each unit's movement by eye.

⚠ CAUTION: Unless the SOLO/M micromanipulator electromechanical baseplate is firmly bolted down to a breadboard or solidly to a firm surface, the SOLO/M is likely to tip over when fully extending its axis, especially if it is loaded with a headstage that extends beyond the SOLO/M's current center of gravity.

Upon deciding to directly install the SOLO system in your rig, it is useful to follow the initial setup procedure to learn how to move the units to allow easy access to the mounting screws.

1. With the power switch on the back of the ROE in the OFF (0) position, connect the power adapter's 24VDC cable to the POWER receptacle.

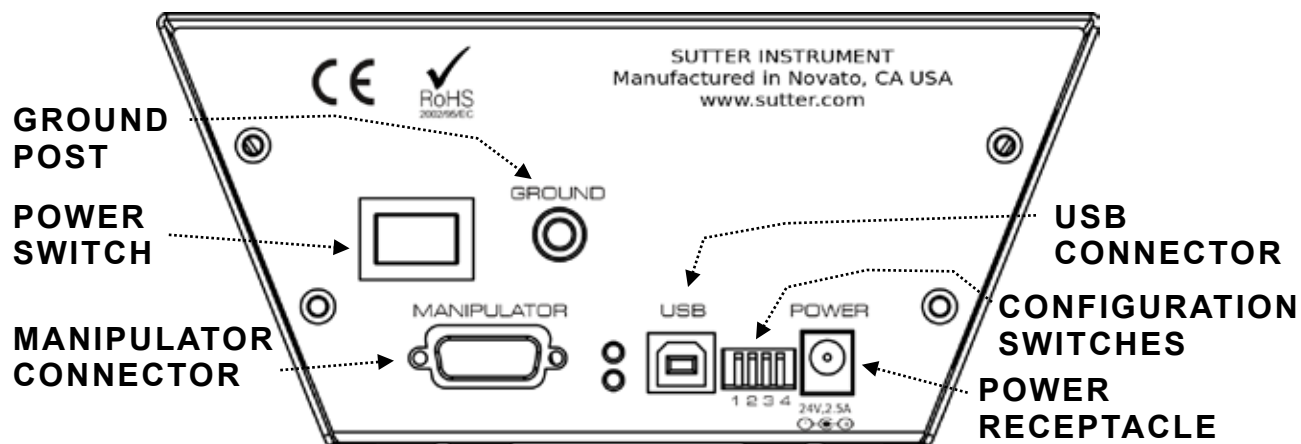


Figure 2-3. Rear of SOLO ROE/Controller cabinet

2. With the power OFF (rear panel switch in the “0” position), connect a well-grounded/earthed wire to the GROUND post/banana plug socket.
3. With the power OFF, connect the male end of the DB-9 cable to the MANIPULATOR connector on the ROE, the other end of which is connected to the SOLO/M micromanipulator electromechanical. (See **cautionary note below.**)
4. Verify that the four switches on the rear of the ROE are set as desired.
5. Power up the system by moving the power switch on the rear of the ROE to the “1” position.

*  **CAUTION: NEVER CONNECT OR DISCONNECT THE ROE/CONTROLLER FROM THE SOLO/M WHILE THE POWER IS ON!**

2.5 ROE/Controller Rear Panel Controls and Configuration

2.5.1 Power Switch

The power switch for the SOLO system is located on the rear panel of the ROE/controller. At power up, the microprocessor in the ROE/controller scans the attached equipment and configures the system accordingly.

2.5.2 Configuration Switches

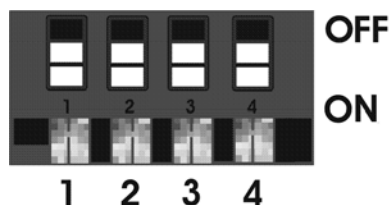


Figure 2-4. Configuration switches on rear of SOLO ROE/Controller unit (switch positions shown are factory defaults).

Switches 1, 2, 3 and 4 set the operating characteristics of the SOLO at power-up.

Table 2-1. Configuration DIP switch settings.

Switch #	Function	Setting	Description
1	Knob Rotation Directionality for Forward (+) Manipulator Movement	OFF (UP)	Counterclockwise
		ON* (DOWN)*	Clockwise
2	Length of Travel of Connected Electromechanical Device	OFF (UP)	50mm
		ON* (DOWN)*	25mm
3	Electromechanical device compatibility	OFF (UP)	MP-285/M series
		ON* (DOWN)*	SOLO/M
4	Calibration Homing on Power On	OFF (UP)	Disabled. Point of Origin stays at physical position of attached electromechanical. All positions forward of Point of Origin are positive, and negative in opposite direction. (**)
		ON* (DOWN)*	Enabled. Moves to position 0 (physical beginning of travel), then advances to 1,000 microns.

* Normal operation (factory default).

** Switch 4 OFF (up) to retain power-off position.

(NOTE: The positions of the switches in the above table were the opposite of what they should have been, starting with Revision 1.08 and prior revisions of this manual. As of Revision 1.09, the positions and factory defaults are now correct.)

3. OPERATIONS

3.1 Main Controls and Indicators on the ROE/Controller

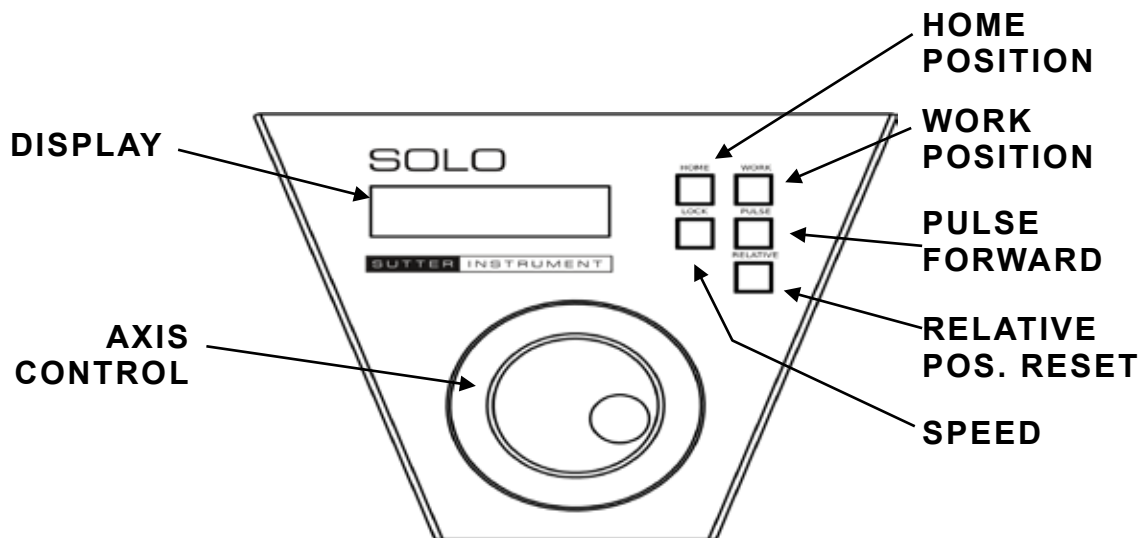


Figure 3-1. Front view of the SOLO ROE/Controller

3.2 Display

3.2.1 Initial Startup



Figure 3-1. Startup screen

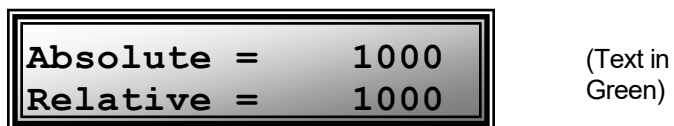


Figure 3-2. Factory default startup calibrated position

When starting the SOLO system for the first time, the value for the single axis will be 1,000 micrometers (microns) for both *Absolute* and *Relative*. *Absolute* always indicates the absolute number of microns from beginning of travel (absolute position 0). *Relative* always indicates the number of microns away from the last-set absolute position (absolute position 0 on startup).

3.3 Control Operations

3.3.1 Maximum Positive Position Values:

Move the dial of an axis clockwise until its position value stops incrementing. The following table lists the maximum position value (in microns) for each axis.

Table 3-1. Maximum positive position value of each axis

Maximum Position Value (in microns)	
25mm	50mm
25,000	50,000

Absolute =	25000
Relative =	25000

(Text in
Green)

Figure 3-3. Maximum positive values

3.3.2 Setting Position for HOME or WORK

To set position, hold down HOME or WORK button for 3 seconds until beep sounds.

3.3.3 Moving to the Home Position

Absolute =	2602
Relative =	2602

(Text in
Red)

Figure 3-4. Moving to Home position (screen is amber while moving)

If the Home position has not yet been defined and saved, the Home position value for the single axis will default to 1,000 microns, as shown in the following figure.

Absolute =	1000
Relative =	1000

(Text in
Green)

Figure 3-5. Factory default Home position

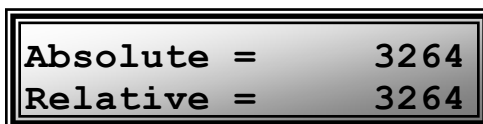
If the Home position has been previously defined (saved), pressing HOME will make a move to the defined home position (see example in the following figure).

Absolute =	2602
Relative =	2602

(Text in
Green)

Figure 3-6. Example Home position defined and saved

3.3.4 Moving to the Work Position



(Text in red while moving, then green)

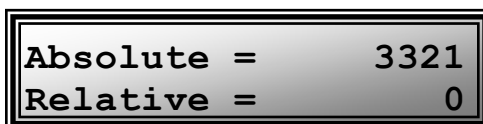
Figure 3-7. Example Work position

To move to the Work position, press the WORK button.

3.3.5 Setting Absolute/Relative Coordinates Mode

The RELATIVE button toggles between Relative and Absolute coordinate systems. The default coordinate system on power up is Absolute, with the coordinates on the screen shown in green. To switch to relative coordinates, press the RELATIVE button once. To reset the current position to all zeroes, depress the RELATIVE for 3 seconds or until a beep is heard, and then release the button. This resets the current position to all zeroes.

Depress RELATIVE for 3 sec. or until beep sounds



(Text in green)

Figure 3-8. Relative mode

3.3.6 Mode Indications

The SOLO system has two modes of operation: Absolute/relative coordinates status and while moving with the rotary knob, and when moving to HOME/WORK or when commanded to move by an external-control command. The display turns color for each specific mode, as shown in the following table.

Table 3-2. Screen colors and modes

Screen Color	Mode	Example
Green	Absolute & relative coordinates and while moving with the rotary knob.	
Red	Move to Home or Work position, or when commanded by external computer control (knob is disabled during the move)	

3.3.7 Speed Control and ROE Knob Movements (SPEED)

The rate at which the ROE axis knob moves the manipulator can be adjusted with the SPEED button. Each press of the button cycles through four speeds: 0 (normal) through 3 (fastest).

3.3.8 Pausing Home Movements

After Move to Home has been initiated, and while the move is in progress, pressing HOME a second time pauses the manipulator. Pressing HOME again resumes movement.

3.3.9 Pausing Work Movements

After Move to Work has been initiated, and while the move is in progress, pressing WORK a second time pauses the manipulator. Pressing WORK again resumes movement.

3.3.10 Pulse Mode

Pulse mode advances the single axis in $2.85 \mu\text{m}$ steps. Each press of the PULSE button increments by one $2.85\text{-}\mu\text{m}$ step beyond the current position. This feature can be used to penetrate tough or resistant tissue.

4. EXTERNAL CONTROL

Controlling the SOLO externally via computer is accomplished by sending commands over the USB interface between the computer and the USB connector on the rear panel of the SOLO controller/ROE. The USB device driver for Windows is downloadable from Sutter Instrument's web site (www.sutter.com). The SOLO requires Sutter Instrument's USB CDM (Combined Driver Model) Version 2.10.00 or higher. The CDM device driver consists of two device drivers: 1) USB device driver, and 2) VCP (Virtual COM Port) device driver. Install the USB device driver first, followed by the VCP device driver. The VCP device driver provides a serial RS-232 I/O interface between a Windows application and the SOLO. Although the VCP device driver is optional, its installation is recommended even if it is not going to be used. Once installed, the VCP can be enabled or disabled.

The CDM device driver package provides two I/O methodologies over which communications with the controller over USB can be conducted: 1) USB Direct (D2XX mode), or 2) Serial RS-232 asynchronous via the VCP device driver (VCP mode). The first method requires that the VCP device driver not be installed, or if installed, that it be disabled. The second method requires that the VCP be installed and enabled.

4.1 Virtual COM Port (VCP) Serial Port Settings

The following table lists the required RS-232 serial settings for the COM port (COM3, COM5, etc.) generated by the installation or enabling of the VCP device driver.

Table 4-1. USB-VCP interface serial port settings.

Property	Setting
Data ("Baud") Rate (bits per second (bps))	57600
Data Bits	8
Stop Bits	1
Parity	None
Flow Control	None

The settings shown in the above table can be set in the device driver's properties (via the Device Manager if in Windows) and/or programmatically in your application.

4.2 Protocol and Handshaking

Command sequences do not have terminators. All commands return an ASCII CR (Carriage Return; 13 decimal, 0D hexadecimal) to indicate that the task associated with the command has completed. When the controller completes the task associated with a command, it sends ASCII CR back to the host computer indicating that it is ready to receive a new command. If a command returns data, the last byte returned is the task-completed indicator.

4.3 Command Sequence Formatting

Each command sequence consists of at least one byte, the first of which is the "command byte". Those commands that have parameters or arguments require a sequence of bytes that follow the command byte. No delimiters are used between command sequence arguments, and command sequence terminators are not used. Although most command bytes can be

expressed as ASCII displayable/printable characters, the rest of a command sequence must generally be expressed as a sequence of unsigned byte values (0-255 decimal; 00 – FF hexadecimal, or 00000000 – 11111111 binary). Each byte in a command sequence transmitted to the controller must contain an unsigned binary value. Attempting to code command sequences as “strings” is not advisable. Any command data returned by the controller should be initially treated as a sequence of unsigned byte values upon reception. Groups of contiguous bytes can later be combined to form larger values, as appropriate (e.g., 2 bytes into 16-bit “word”, or 4 bytes into a 32-bit “long” or “double word”). For the SOLO, all axis position values (number of microsteps) are stored as “unsigned long” 32-bit positive-only values, and each is transmitted and received to and from the controller as four contiguous bytes.

4.4 Axis Position Command Parameters

All axis positional information is exchanged between the controller and the host computer in terms of microsteps. Conversion between microsteps and microns (micrometers) is the responsibility of the software running on the host computer (see *Microns/microsteps conversion* table for conversion factors).

Microsteps are stored as positive 32-bit values (“long” (or optionally, “signed long”), or “unsigned long” for C/C++ ; “I32” or “U32” for LabVIEW). “Unsigned” means the value is always positive; negative values are not allowed. The positive-only values can also be stored in signed type variables, in which case care must be taken to ensure that only positive values are exchanged with the controller.

The 32-bit value consists of four contiguous bytes, with a byte/bit-ordering format of Little Endian (“Intel”) (most significant byte (MSB) in the first byte and least significant (LSB) in the last byte). If the platform on which your application is running is Little Endian, then no byte order reversal of axis position values is necessary. Examples of platforms using Little Endian formatting include any system using an Intel/AMD processor (including Microsoft Windows and Apple Mac OS X).

If the platform on which your application is running is Big Endian (e.g., Motorola PowerPC CPU), then these 32-bit position values must have their bytes reverse-ordered after receiving from, or before sending to, the controller. Examples of Big-Endian platforms include many non-Intel-based systems, LabVIEW (regardless of operating system & CPU), and Java (programming language/environment). MATLAB and Python (script programming language) are examples of environments that adapt to the system on which each is running, so Little-Endian enforcement may be needed if running on a Big-Endian system. Some processors (e.g., ARM) can be configured for specific endianness.

4.5 Microsteps and Microns (Micrometers)

All coordinates sent to and received from the controller are in microsteps. To convert between microsteps and microns (micrometers), use the following conversion factors (multipliers):

Table 4-2. Microns/microsteps conversion

Controller with Device	From/To Units	Conversion Factor (multiplier)
SOLO-25/M or SOLO-50/M micromanipulator, or single axis of a TRIO series or QUAD/M electromechanical	$\mu\text{steps} \rightarrow \mu\text{m}$	0.09375
	$\mu\text{m} \rightarrow \mu\text{steps}$	10.666666666667
Single axis of an MP-285/M micromanipulator or derived electromechanical	$\mu\text{steps} \rightarrow \mu\text{m}$	0.125
	$\mu\text{m} \rightarrow \mu\text{steps}$	8

For accuracy in your application, type these conversion factors as “double” (avoid using the “float” type as it lacks precision with large values). When converting to microsteps, type the result as a 32-bit “unsigned long” (C/C++), “uint32” (MATLAB), or “U32” (LabVIEW) integer (positive only) value. When converting to microns, type the result as a “double” (C/C++, MATLAB) or “DBL” (LabVIEW) 64-bit double-precision floating-point value.

4.6 Ranges and Bounds:

Table 4-3. Ranges and bounds

Device	Axis	Len. (mm)	Origin	Microns	Microsteps
SOLO-25/M	(any)	25	BOT	0 – 25,000	0 – 266,667
SOLO-50/M	(any)	50	BOT	0 – 50,000	0 – 533,334

4.7 Travel Speed

The following table shows the travel speeds for supported devices using move commands.

Table 4-4. Travel speeds.

Device	mm/sec or $\mu\text{m}/\text{ms}$
SOLO-25/M or SOLO-50/M, or single axis of a TRIO series or QUAD/M electromechanical *	3
Single axis of an MP-285/M micromanipulator or derived electromechanical **	5

4.1 Commands

4.1.1 Get Current Position ('c' or 'C') Command

This command is used to obtain the current position of the manipulator. The command sequence consists of one byte as shown in the following table. The data received consists of five bytes containing the position (32-bit) value in microsteps (4 bytes) and the completion indicator (1 byte).

Table 4-5. Get Current Position and Angle ('c' or 'C') command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	99 or 67	63 or 43	0110 0011 or 0100 0011	0099 or 0043		`c` or `C`	Command
Rx	All	5	0-3	Current absolute position of the single axis, in microsteps, consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value in Little-Endian bit order.						
			13	13	0D	0000 1101		^M	<CR>	Completion indicator

4.1.2 Move to Controller-Defined HOME Position ('h') Command

moves to the position saved by the controller's HOME button. Table 4-6. Move to controller-defined HOME position ('h') command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	104	68	0110 1000	0104		`h`	Command
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

4.1.3 Move to Controller-Defined WORK Position ('w') Command

moves to the position saved by the controller's WORK button. Table 4-7. Move to controller -defined WORK position ('w') command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx		1	0	119	77	0111 0111	0119		`w`	Command
Rx		1	0	13	0D	0000 1101			<CR>	Completion indicator

4.1.4 Move to Specified "Home" Position ('H') Command

This command instructs the controller to move to specified Home position (see *Ranges* table). The command sequence consists of five bytes: The command byte (1 byte) followed by a 32-bit unsigned (positive) integer value (4 bytes) for the position in microsteps.

Table 4-8. Move to specified “Home” position (‘H’) command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	5	0	72	48	0100 1000	0072		\‘H\’	Command
			1 (4)	Target absolute position, in microsteps, consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value (see <i>Ranges and bounds</i> table) in Little-Endian bit order.						
Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Completion indicator

4.1.5 Move to Specified “Work” Position (‘W’) Command

This command instructs the controller to move to specified Work position (see *Ranges* table). The command sequence consists of five bytes: The command byte (1 byte) followed by a 32-bit unsigned (positive) integer value (4 bytes) for the position in microsteps.

Table 4-9. Move to specified “Work” position (‘W’) command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	5	0	87	57	0101 0111	0087		\‘W\’	Command
			1 (4)	Target absolute position, in microsteps, consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value (see <i>Ranges and bounds</i> table) in Little-Endian bit order.						
Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Completion indicator

4.1.6 Move to Specified Position (‘x’ or ‘X’) Command

This command moves to a specified position for the single axis. The command sequence consists of five bytes: The command byte (1 byte) followed by a 32-bit unsigned (positive) integer value (4 bytes) for the position in microsteps.

Table 4-10. Move to specified X-axis position (‘x’ or ‘X’) command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	5	0	120	78	0111 1000	0120		\‘x\’	Command
				or	or	or	or			
				88	58	0101 1000	0088		\‘X\’	
			1 (4)	Target absolute position, in microsteps, consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value (see <i>Ranges and bounds</i> table) in Little-Endian bit order.						
Rx		1	0	13	0D	0000 1101			<CR>	Completion indicator

4.1.7 Set Velocity for all Externally Controlled Movements ('v') Command (Firmware Ver. 2.55+)

This command sets the speed at which all externally controlled moves are made. The command byte is followed by two bytes representing a 16-bit integer value ranging from 0 (fastest) to 65,535 (slowest).

Table 4-11. Set velocity command ('v') command.

Command	Tx/-Delay/-Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key-pad #	Ctrl-char	ASCII def./-char.	Description
					Dec.	Hex.	Binary				
Set Velocity ('v')	Tx	2.55+	3	0	118	76	0111 0110	0118		'v'	Sets the velocity for all external-control movement commands
				1 (2)	Speed factor consisting of 2 contiguous bytes representing a single 16-bit unsigned (positive) integer value in Little-Endian bit order ranging from 0 (fastest) to 65,535 (slowest)						
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

4.2 Notes

- Task-Complete Indicator:** All commands will send back to the computer the "Task-Complete Indicator" to signal the command and its associated function in controller is complete. The indicator consists of one (1) byte containing a value of 13 decimal (0D hexadecimal), and which represents an ASCII CR (Carriage Return).
- Intercommand Delay:** A short delay (usually around 2 ms) is recommended between commands (after sending a command sequence and before sending the next command).
- Clearing Send/Receive Buffers:** Clearing (purging) the transmit and receive buffers of the I/O port immediately before sending any command is recommended.
- Positions in Microsteps and Microns:** All positions sent to and received from the controller are in microsteps (μ steps). See *Microns/microsteps conversion* table) for conversion between μ steps and microns (micrometers (μ m)).

Declaring position variables in C/C++:

```
/* current position for X */
unsigned long cp_x_us; /* microsteps */
double cp_x_um; /* microns */
/* specified (move-to) position for X */
unsigned long sp_x_us; /* microsteps */
double sp_x_um; /* microns */
```

Use the same convention for other position variables the application might need.

Declaring the microsteps/microns conversion factors in C/C++:

```
/* conversion factors for the SOLO-xx/M based config. */
double us2umCF = 0.09375; /* microsteps to microns */
double um2usCF = 10.6666666667; /* microns to microsteps */
```

Converting between microsteps and microns in C/C++:

```
/* converting X axis current position */
cp_x_um = cp_x_us * us2umCF; /* microsteps to microns */
cp_x_us = cp_x_um * um2usCF; /* microns to microsteps */
```

Do the same for any other position sets used in the application.

5. **Ranges and Bounds:** See *Ranges and Bounds* table for exact minimum and maximum values for each axis of each compatible device that can be connected. All move commands must include positive values only for positions – negative positions must never be specified. All positions are absolute as measured from the physical beginning of travel of a device’s axis. In application programming, it is important that positional values be checked (≥ 0 and $\leq \text{max.}$) to ensure that a negative absolute position is never sent to the controller and that end of travel is not exceeded. All computational relative positioning must always resolve to accurate absolute positions.

Declaring minimum and maximum absolute position variables in C/C++:

```
/* minimum and maximum positions for X, Y, & Z */
double min_x_um; /* minimum microns */
double max_x_um; /* maximum microns */
```

Set minimum and maximum absolute positions for each axis – see Ranges & Bounds table.

```
/* initialize all minimum positions in microns*/
min_x_um = 0;
/* initialize all maximum positions in microns*/
/* SOLO-25/M */
max_x_um = 25000;
/* SOLO-50/M */
max_x_um = 50000;
```

6. **Absolute Positioning System Origin:** The Origin is set to a physical position of travel to define absolute position 0. The physical Origin position is fixed at beginning of travel (BOT). This means that all higher positions (towards end of travel (EOT)) are positive values; there are no lower positions and therefore no negative values are allowed.
7. **Absolute vs. Relative Positioning:** Current position (‘c’) and move commands always use absolute positions. All positions can be considered “relative” to the Origin (Position 0), but all are in fact absolute positions. Any position that is considered to be “relative” to the current position, whatever that might be, can be handled synthetically by external programming. However, care should be taken to ensure that all relative position calculations always result in correct positive absolute positions before initiating a move command.

Declaring relative position variables in C/C++:

```
/* relative positions for X, Y, Z, & D */
double rp_x_um; /* microns */
/* initialize all relative positions to 0 after declaring them */
rp_x_um = 0;
```

Enter any positive or negative value for each relative position (e.g., rp_x_um = 1000).

For the single axis, check to make sure that the new resultant absolute position (to which to move) is within bounds. Reset the relative position to 0 if not. If relative value is negative, its positivized value must not be greater than the current position. Otherwise, if positive, adding current position with relative position must not exceed the maximum position allowed. If out of bounds, resetting relative position to 0 allow the remaining conversions and movement to resolve without error.

```
/* check to make sure that relative X is within bounds */
if ( ( rp_x_um < 0 && abs(rp_x_um) > cp_x_um ) ||
      (cp_x_um + rp_x_um > max_x_um) ) /* out of bounds? */
    rp_x_um = 0; /* yes, so reset relative pos. to 0 */
```

Calculate new absolute position in microns and then convert to microsteps before issuing a move command.

```
/* convert X relative position to absolute position */
sp_x_um = cp_x_um + rp_x_um; /* add relative pos. to current pos. */
/* convert new absolute X position in microns to microsteps */
sp_x_us = sp_x_um * um2usCF;
```

8. **Position Value Typing:** All positions sent and received to and from the controller are in microsteps and consist of 32-bit integer values (four contiguous bytes). Position values in microsteps are always positive, so data type must be an “unsigned” integer that can hold 32 bits of data. Although each positional value is transmitted to, or received from, the controller as a sequence of four (4) contiguous bytes, for computer application computational and storage purposes each should be typed as an unsigned 32-bit integer (“unsigned long” in C/C++, “uint32” in MATLAB, “U32” in LabVIEW, etc.).
Position values in microns (micrometers or μm) should be data typed as double-precision floating point variables (“double” in C/C++ and MATLAB, “DBL” in LabVIEW, etc.).
Note that in Python, incorporating the optional NumPy package brings robust data typing like that used in C/C++ to your program, simplifying coding and adding positioning accuracy to the application.
9. **Position Value Bit Ordering:** All 32-bit position values transmitted to, and received from, the controller must be bit/byte-ordered in “Little Endian” format. This means that the least significant bit/byte is last (last to send and last to receive). Byte-order reversal may be required on some platforms. Microsoft Windows, Intel-based Apple Macintosh systems running Mac OS X, and most Intel/AMD processor-based Linux distributions handle byte storage in Little-Endian byte order so byte reordering is not necessary before converting to/from 32-bit “long” values. LabVIEW always handles “byte strings” in “Big Endian” byte order irrespective of operating system and CPU, requiring that the four bytes containing a microsteps value be reverse ordered before/after conversion to/from a multibyte type value (I32, U32, etc.). MATLAB automatically adjusts the endianness of multibyte storage entities to that of the system on which it is running, so explicit byte reordering is generally unnecessary unless the underlying platform is Big Endian. If your development platform does not have built-in Little/Big Endian conversion functions, bit reordering can be accomplished by first swapping positions of the two bytes in each 16-bit half of the 32-bit value, and then swap positions of the two halves. This method efficiently and quickly changes the bit ordering of any multibyte value between the two Endian formats (if Big Endian, it becomes Little Endian, and if Little Endian, it becomes then Big Endian).
10. **Travel Lengths and Durations:** “Move” commands might have short to long distances of travel. If not polling for return data, an appropriate delay should be inserted between the sending of the command sequence and reception of return data so that the next command is sent only after the move is complete. This delay can be auto calculated by determining the distance of travel (difference between current and target positions) and rate of travel. This delay is not needed if polling for return data. In either case, however, an appropriate timeout must be set for the reception of data so that the I/O does not time out before the move is made and/or the delay expires.
11. **Movement Speed:** All move commands cause movement to occur at a maximum rate of 3,000 microns/second. With firmware v2.55 and above, speed can be controlled for all

externally controlled movements using the 'v' command followed by a 16-bit value of 0 (fastest) to 65,535 (slowest) (Little-Endian bit order).

5. MAINTENANCE

Routine cleaning of the SOLO system is required to prevent excessive dust accumulations. Wipe all exterior surfaces with a dry, soft, cotton cloth.

Periodically inspect all cables and connections to make sure that all connections are made well and that all connectors are well and evenly seated.

APPENDIX A. LIMITED WARRANTY

- Sutter Instrument Company, a division of Sutter Instrument Corporation, limits the warranty on this instrument to repair and replacement of defective components for two years from date of shipment, provided the instrument has been operated in accordance with the instructions outlined in this manual.
- Abuse, misuse, or unauthorized repairs will void this warranty.
- Warranty work will be performed only at the factory.
- The cost of shipment both ways is paid for by Sutter Instrument during the first three months this warranty is in effect, after which the cost is the responsibility of the customer.
- The limited warranty is as stated above and no implied or inferred liability for direct or consequential damages is intended.
- An extended warranty for up to three additional years can be purchased at the time of ordering, or until the original warranty expires. For pricing and other information, please contact Sutter Instrument.

APPENDIX B. ACCESSORIES

W621 150	Ground cable
285204	4-inch dovetail extension
285210	Mounting adapter plate
225RBI	Rotating base
221165	Z-axis vertical extension
BR-AW	Rod holding clamp for 4mm OD rods and Sutter Instrument XenoWorks injectors
MP-ROD	Rod holder (for rod OD 6.25mm or larger)
MD Series	Micromanipulator platform
MT-78-FS	Large fixed-stage platform
MT-78-FS/M6	Large fixed-stage platform with M6 tapped holes
MT-75	Standard gantry-stand 8.7 to 13.4 in (22.1 to 33.9 cm)
MT-75S	Short gantry-stand 6.7 to 9.6 in (16.9 to 24.4 cm)

APPENDIX C. TECHNICAL SPECIFICATIONS



C.1. SOLO/M

Travel	25mm (SOLO-25/M) or 50mm (SOLO-50/M)
Resolution	Minimal microstep size is 62.5 nanometers per microstep. Display has single micron resolution.
Speed	3 mm/sec. maximum
Drift	< 1 micron/hr drive mechanism

C.2. SOLO ROE/Controller

Electrical:

Power Adapter:	Meanwell GS60A24-P1J
Input (Mains)	100 - 240 VAC, 50/60 Hz, 1.4A
Output (fixed cable to ROE/Controller's POWER receptacle)	24V DC, 2.5A, 60W Max. (see following table for cable specs)

SOLO System Power consumption 60-Watts maximum

Mains fuses None replaceable (power protection built into the Power Adapter)

Cables (Refer to the following tables for a description of all possible cables.)

Table C-1. SOLO cables and receptacles/connectors.

Controller Rear Panel Port Connector/Receptacle	Cable Connector Types	Connects to ...	Cable Type	Cable Max. Length
Power Adapter 3-pin male connector	← 3-pin power standard (female) 3-pin male → (Geographical region dependent)	Mains power source.	10A, 250V, with safety ground plug	3 meters (approx. 10 feet)
ROE/Controller Cabinet: MANIPULATOR (9-Pin DSUB female receptacle)	← DB-9 male DB-9 female → (Straight-through)	SOLO/M series	Minimum of 26 awg stranded wire with 500 Volt.	3 meters (approx. 10 feet)
Power Adapter	← (fixed) ID 2.1 x OD 5.5 mm Barrel Plug (male) →	ROE/Controller Cabinet: POWER receptacle (center pin positive)	UL1185 18AWG	1.8 meters (approx. 6 feet)
ROE/Controller Cabinet: GROUND (1-pin Banana-style female receptacle)	← Banana male Alligator clip → (hooded)	a ground/earth source (user determined)		
ROE/Controller Cabinet: USB	← A B →	Computer USB port		

Dimensions:

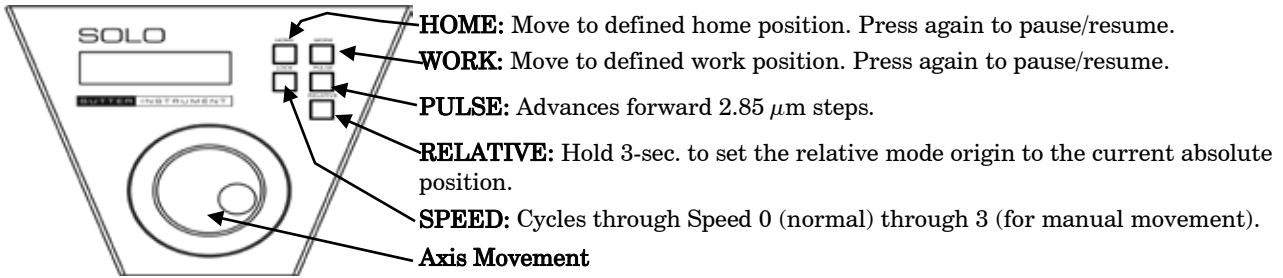
SOLO ROE/controller 5.5 x 5.1 x 3.2 in | 14 x 13 x 8 cm

Weight:

SOLO ROE/controller 1.2 lbs. | 0.53 kg

APPENDIX D. QUICK REFERENCE

Manual Operation



Setting Home/Work Pos. & Relative Mode Origin Pos.: To set position, hold down HOME, WORK, & RELATIVE buttons for 3 seconds until beep sounds.

Screen-color mode indications: Green = Position status; Red = Movement in progress (knob disabled).

Movement Knob Disabling: Movement knob is disabled during movement to Home, Work, or when initiated by external movement command.

Configuration

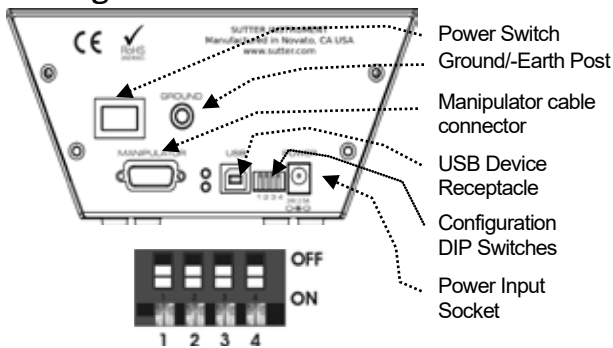


Table D-1. Configuration Switches 1 – 4.

Sw #	Definition	State	Setting	Position
1	Knob Rotation for Forward (+) Movement	Counter	OFF	UP
		Clockwise	ON*	DOWN*
2	Single axis travel length (25 or 50 mm)	50mm	OFF	UP
		25mm	ON*	DOWN*
3	Electromechanical device compatibility	MP-285/M	OFF	UP
		SOLO/M	ON*	DOWN*
4	Calibration Homing on Power On	Disabled: **	OFF	UP
		Enabled.	ON*	DOWN*

* Normal operation (factory default).

** Switch 4 OFF (up) to retain power-off position.

External Control

Controlling the SOLO externally via computer is accomplished by sending commands over the USB interface between the computer and the USB controller/ROE. The USB device driver for Windows is downloadable from Sutter Instrument’s web site (www.sutter.com). The SOLO requires USB CDM (Combined Driver Model) Version 2.10.00 or higher. The CDM device driver for the SOLO consists of two device drivers: 1) USB device driver, and 2) VCP (Virtual COM Port) device driver. Install the USB device driver first, followed by the VCP device driver. The VCP device driver provides a serial RS-232 I/O interface between a Windows application and the SOLO. Although the VCP device driver is optional, its installation is recommended even if it is not going to be used. Once installed, the VCP can be enabled or disabled.

The CDM device driver package provides two I/O methodologies over which communications with the

controller over USB can be conducted: 1) USB Direct (D2XX mode), or 2) Serial RS-232 asynchronous via the VCP device driver (VCP mode). The first method requires that the VCP device driver not be installed, or if installed, that it be disabled. The second method requires that the VCP be installed and enabled.

Virtual COM Port (VCP) Serial Port Settings: The following table lists the required RS-232 serial settings for the COM port (COM3, COM5, etc.) generated by the installation or enabling of the VCP device driver.

Table D-2. USB-VCP interface serial port settings.

Property	Setting
Data (“Baud”) Rate (bits per second (bps))	57600
Data Bits	8
Stop Bits	1
Parity	None
Flow Control	None

The settings shown in the above table can be set in the device driver's properties (via the Device Manager if in Windows) and/or programmatically in your application.

Protocol and Handshaking: Command sequences do not have terminators. All commands return an ASCII CR (Carriage Return; 13 decimal, 0D hexadecimal) to indicate that the task associated with the command has completed. When the controller completes the task associated with a command, it sends ASCII CR back to the host computer indicating that it is ready to receive a new command. If a command returns data, the last byte returned is the task-completed indicator.

Command Sequence Formatting: Each command sequence consists of at least one byte, the first of which is the "command byte". Those commands that have parameters or arguments require a sequence of bytes that follow the command byte. No delimiters are used between command sequence arguments, and command sequence terminators are not used. Although most command bytes can be expressed as ASCII displayable/printable characters, the rest of a command sequence must generally be expressed as a sequence of unsigned byte values (0-255 decimal; 00 – FF hexadecimal, or 00000000 – 11111111 binary). Each byte in a command sequence being transmitted to the controller must contain an unsigned binary value. Attempting to code command sequences as "strings" is not advisable. Any command data being returned from the controller must also be received and initially treated as a sequence of unsigned byte values. Groups of contiguous bytes can later be combined to form larger values, as appropriate (e.g., 2 bytes into 16-bit "word", or 4 bytes into a 32-bit "long" or "double word"). For the SOLO, all axis position values (number of microsteps) are stored as "unsigned long" (32-bit) values, and each is transmitted and received to and from the controller as four contiguous bytes.

Axis Position Command Parameters: All axis positional information is exchanged between the controller and the host computer in terms of microsteps. Conversion between microsteps and microns (micrometers) is the responsibility of the software running on the host computer (see *Microns/microsteps conversion* table for conversion factors).

Microsteps are stored as positive 32-bit values ("long" (or optionally, "signed long"), or "unsigned long" for C/C++; "I32" or "U32" for LabVIEW). "Unsigned" means the value is always positive; negative values are not allowed. The positive-only

values can also be stored in signed type variables, in which case care must be taken to ensure that only positive values are exchanged with the controller.

The 32-bit value consists of four contiguous bytes, with a byte/bit-ordering format of Little Endian ("Intel") (most significant byte (MSB) in the first byte and least significant (LSB) in the last byte). If the platform on which your application is running is Little Endian, then no byte order reversal of axis position values is necessary. Examples of platforms using Little Endian formatting include any system using an Intel processor (including Microsoft Windows and Apple Mac OS X).

If the platform on which your application is running is Big Endian (e.g., Motorola PowerPC CPU), then these 32-bit position values must have their bytes reverse-ordered after receiving from, or before sending to, the controller. Examples of Big-Endian platforms include many non-Intel-based systems, LabVIEW (regardless of operating system & CPU), and Java (programming language/environment). MATLAB adapts to the system on which it is running, so Little Endian may need to be enforced if running on a Big-Endian system.

Microsteps and Microns (Micrometers): All coordinates sent to and received from the controller are in microsteps. To convert between microsteps and microns (micrometers), use the following conversion factors (multipliers):

Table D-3. Microns Microns/microsteps conversion.

Controller with Device	From/To Units	Conversion Factor (multiplier)
SOLO with SOLO-xx-M series micromanipulator	$\mu\text{steps} \rightarrow \mu\text{m}$	0.09375
	$\mu\text{m} \rightarrow \mu\text{steps}$	10.6666666667
SOLO with a single axis of an MP-285-M or MP-225-M micromanipulator	$\mu\text{steps} \rightarrow \mu\text{m}$	0.125
	$\mu\text{m} \rightarrow \mu\text{steps}$	8

For accuracy in your application, type these conversion factors as "double" (avoid using the "float" type as it lacks precision with large values). When converting to microsteps, type the result as a 32-bit "unsigned long" (C/C++), "uint32" (MATLAB), or "U32" (LabVIEW) integer (positive only) value. When converting to microns, type the result as a "double" (C/C++, MATLAB) or "DBL" (LabVIEW) 64-bit double-precision floating-point value.

Ranges and Bounds:

Table D-4. Ranges and bounds

Device	Axis	Len. (mm)	Origin	Microns	Microsteps
SOLO-25/M	(any)	25	BOT	0 – 25,000	0 – 266,667
SOLO-50/M	(any)	50	BOT	0 – 50,000	0 – 533,334

Table D-5. Travel speeds.

Device	mm/sec or $\mu\text{m/ms}$
SOLO-25/M or SOLO-50/M, or single axis of a TRIO series or QUAD/M electromechanical *	3
Single axis of an MP-285/M micromanipulator or derived electromechanical **	5

Travel Speed: The following table shows the travel speeds for supported devices using move commands.

Command Reference: The following table lists all the external-control commands for the SOLO.

Table D-6. SOLO external control commands.

Command	Tx/- Delay/- Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key-pad #	Ctrl-char	ASCII def./-char.	Description
					Dec.	Hex.	Binary				
Get Current Position ('c' or 'C')	Tx	All	1	0	99 or 67	63 or 43	0110 0011 or 0100 0011	0099 or 0043		'c' or 'C'	Returns the current position (μsteps) of the single axis
	Rx	All	5	0-3	Current absolute position of the single axis, in microsteps, consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value in Little-Endian bit order.						
				4	13	0D	0000 1101		^M	<CR>	Completion indicator
Move to HOME Position ('h')	Tx		1	0	104	68	0110 1000	0104		'h'	Moves to the position saved for the controller's HOME button
	Rx		1	0	13	0D	0000 1101		^M	<CR>	Completion indicator
Move to WORK Position ('w')	Tx		1	0	119	77	0111 0111	0119		'w'	Moves to the position saved for the controller's WORK button
	Rx		1	0	13	0D	0000 1101		^M	<CR>	Completion indicator
Move to Specified "Home" Position ('H')	Tx	All	5	0	72	48	0100 1000	0072		'H'	Move single axis to specified position (see <i>Ranges</i> table)
				1-4	Target absolute position, in microsteps, consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value (see <i>Ranges and bounds</i> table) in Little-Endian bit order.						
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Completion indicator
Move to Specified "Work" Position ('W')	Tx	All	5	0	87	57	0101 0111	0087		'W'	Move single axis to specified position (see <i>Ranges</i> table)
				1-4	Target absolute position, in microsteps, consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value (see <i>Ranges and bounds</i> table) in Little-Endian bit order.						
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Completion indicator
Move to specified X axis Position ('x' or 'X')	Tx		5	0	120 or 88	78 or 58	0111 1000 or 0101 1000	0120 or 0088		'x' or 'X'	Move X axis to specified position (see <i>Ranges</i> table)
				1 - 4	Target absolute position, in microsteps, consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value (see <i>Ranges and bounds</i> table) in Little-Endian bit order.						
	Rx		1	0	13	0D	0000 1101		^M	<CR>	Completion indicator

Command	Tx/- Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
					Dec.	Hex.	Binary				
Set Velocity (‘v’)	Tx	2.55 +	3	0	118	76	0111 0110	0118		‘v’	Sets the velocity for all external-control movement commands
				1 (2)	Speed factor consisting of 2 contiguous bytes representing a single 16-bit unsigned (positive) integer value in Little-Endian bit order ranging from 0 (fastest) to 65,535 (slowest)						
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

NOTES:

- Task-Complete Indicator:** All commands will send back to the computer the “Task-Complete Indicator” to signal the command and its associated function in controller is complete. The indicator consists of one (1) byte containing a value of 13 decimal (0D hexadecimal), and which represents an ASCII CR (Carriage Return).
- Intercommand Delay:** A short delay (usually around 2 ms) is recommended between commands (after sending a command sequence and before sending the next command).
- Clearing Send/Receive Buffers:** Clearing (purging) the transmit and receive buffers of the I/O port immediately before sending any command is recommended.
- Positions in Microsteps and Microns:** All positions sent to and received from the controller are in microsteps (μ steps). See *Microns/microsteps conversion* table for conversion between μ steps and microns (micrometers (μ m)).

Declaring position variables in C/C++:

```
/* current position for X */
unsigned long cp_x_us; /* microsteps */
double cp_x_um; /* microns */
/* specified (move-to) position for X */
unsigned long sp_x_us; /* microsteps */
double sp_x_um; /* microns */
```

Use the same convention for other position variables the application might need.

Declaring the microsteps/microns conversion factors in C/C++:

```
/* conversion factors for the SOLO-xx/M based config. */
double us2umCF = 0.09375; /* microsteps to microns */
double um2usCF = 10.66666666667; /* microns to microsteps */
```

Converting between microsteps and microns in C/C++:

```
/* converting X axis current position */
cp_x_um = cp_x_us * us2umCF; /* microsteps to microns */
cp_x_us = cp_x_um * um2usCF; /* microns to microsteps */
```

Do the same for any other position sets used in the application.

- Ranges and Bounds:** See *Ranges and Bounds* table for exact minimum and maximum values for each axis of each compatible device that can be connected. All move commands must include positive values only for positions – negative positions must never be specified. All positions are absolute as measured from the physical beginning of travel of a device’s axis. In application programming, it is important that positional values be checked (≥ 0 and \leq max.) to ensure that a negative absolute position is never sent to the controller and that end of travel is not exceeded. All computational relative positioning must always resolve to accurate absolute positions.

Declaring minimum and maximum absolute position variables in C/C++:

```
/* minimum and maximum positions for X, Y, & Z */
double min_x_um; /* minimum microns */
double max_x_um; /* maximum microns */
Set minimum and maximum absolute positions for each axis – see Ranges & Bounds table.
/* initialize all minimum positions in microns */
min_x_um = 0;
/* initialize all maximum positions in microns */
/* SOLO-25/M */
max_x_um = 25000;
/* SOLO-50/M */
max_x_um = 50000;
```

- Absolute Positioning System Origin:** The Origin is set to a physical position of travel to define absolute position 0. The physical Origin position is fixed at beginning of travel (BOT). This means that all higher positions (towards end of travel (EOT)) are positive values; there are no lower positions and therefore no negative values are allowed.
- Absolute vs. Relative Positioning:** Current position (‘c’) and move commands always use absolute positions. All positions can be considered “relative” to the Origin (Position 0), but all are in fact absolute positions. Any position that is considered to be “relative” to the current position, whatever that might be, can be handled synthetically by external programming. However, care should be taken to ensure that all relative position calculations always result in correct positive absolute positions before initiating a move command.

Declaring relative position variables in C/C++:

```
/* relative positions for X, Y, Z, & D */
double rp_x_um; /* microns */
/* initialize all relative positions to 0 after declaring them */
rp_x_um = 0;
```

Enter any positive or negative value for each relative position (e.g., rp_x_um = 1000).

For the single axis, check to make sure that the new resultant absolute position (to which to move) is within bounds. Reset the relative position to 0 if not. If relative value is negative, its positivized value must not be greater than the current position. Otherwise, if positive, adding current position with relative position must not exceed the maximum position allowed. If out of bounds, resetting relative position to 0 allow the remaining conversions and movement to resolve without error.

```
/* check to make sure that relative X is within bounds */
if ( ( rp_x_um < 0 && abs(rp_x_um) > cp_x_um ) ||
```

```

    (cp_x_um + rp_x_um > max_x_um) ) /* out of
bounds? */
    rp_x_um = 0; /* yes, so reset relative pos.
to 0 */

```

Calculate new absolute position in microns and then convert to microsteps before issuing a move command.

```

/* convert X relative position to absolute
position */
sp_x_um = cp_x_um + rp_x_um; /* add relative
pos. to current pos. */
/* convert new absolute X position in microns to
microsteps */
sp_x_us = sp_x_um * um2usCF;

```

8. **Position Value Typing:** All positions sent and received to and from the controller are in microsteps and consist of 32-bit integer values (four contiguous bytes). Position values in microsteps are always positive, so data type must be an “unsigned” integer that can hold 32 bits of data. Although each positional value is transmitted to, or received from, the controller as a sequence of four (4) contiguous bytes, for computer application computational and storage purposes each should be typed as an unsigned 32-bit integer (“unsigned long” in C/C++, “uint32” in MATLAB, “U32” in LabVIEW, etc.).

Position values in microns (micrometers or μm) should be data typed as double-precision floating point variables (“double” in C/C++ and MATLAB, “DBL” in LabVIEW, etc.).

Note that in Python, incorporating the optional NumPy package brings robust data typing like that used in C/C++ to your program, simplifying coding and adding positioning accuracy to the application.

9. **Position Value Bit Ordering:** All 32-bit position values transmitted to, and received from, the controller must be bit/byte-ordered in “Little Endian” format. This means that the least significant bit/byte is last (last to send and last to receive). Byte-order reversal may be required on some platforms. Microsoft Windows, Intel-based Apple Macintosh systems running Mac OS X, and most Intel/AMD processor-

based Linux distributions handle byte storage in Little-Endian byte order so byte reordering is not necessary before converting to/from 32-bit “long” values. LabVIEW always handles “byte strings” in “Big Endian” byte order irrespective of operating system and CPU, requiring that the four bytes containing a microsteps value be reverse ordered before/after conversion to/from a multibyte type value (I32, U32, etc.). MATLAB automatically adjusts the endianness of multibyte storage entities to that of the system on which it is running, so explicit byte reordering is generally unnecessary unless the underlying platform is Big Endian. If your development platform does not have built-in Little/Big Endian conversion functions, bit reordering can be accomplished by first swapping positions of the two bytes in each 16-bit half of the 32-bit value, and then swap positions of the two halves. This method efficiently and quickly changes the bit ordering of any multibyte value between the two Endian formats (if Big Endian, it becomes Little Endian, and if Little Endian, it becomes then Big Endian).

10. **Travel Lengths and Durations:** “Move” commands might have short to long distances of travel. If not polling for return data, an appropriate delay should be inserted between the sending of the command sequence and reception of return data so that the next command is sent only after the move is complete. This delay can be auto calculated by determining the distance of travel (difference between current and target positions) and rate of travel. This delay is not needed if polling for return data. In either case, however, an appropriate timeout must be set for the reception of data so that the I/O does not time out before the move is made and/or the delay expires.
11. **Movement Speed:** All move commands cause movement to occur at a maximum rate of 3,000 microns/second. With firmware v2.55 and above, speed can be controlled for all externally controlled movements using the ‘v’ command followed by a 16-bit value of 0 (fastest) to 65,535 (slowest) (Little-Endian bit order).

NOTES:

NOTES:

INDEX

A	
accessories	30
C	
cleaning.....	28
configuration	33
Configuration	13
Configuration switches	14
controller	
cable specs	32
Controls	
SOLO	13
Power switch.....	13
D	
disclaimer	3
E	
Electrical Connections	12
external control.....	33
absolute positioning system origin.....	25, 36
absolute vs. relative positioning.....	25, 36
axis position command parameters	20
clearing send/receive buffers	24, 36
intercommand delay.....	24, 36
microsteps and microns (micrometers)	20
Move to specified.....	23
Move to specified position command	23
position value bit ordering.....	26, 37
position value typing	26, 37
positions in microsteps and microns.....	24, 36
protocol and handshaking.....	19
ranges and bounds	21, 25, 34, 36
task-complete indicator.....	24, 36
travel lengths and durations.....	26, 37
Virtual COM Port (VCP) serial port settings	19
external-control commands	
Move to specified.....	22
F	
fuses, mains.....	31
fuses, replacement	
mains.....	3
G	
glassware	
precautions.....	4
I	
input	
voltage.....	31
Installation	
electrical connections	12
General	11
headstage mounting.....	12
initial operating instructions.....	12
other accessories	12
ROE/Controller rear panel controls and	
configuration.....	13
configuration switches	14
calibration at power up	14
connected device length of travel	14
scaling factor	14
setting axis directionality.....	14
power switch	13
SOLO/M mounting.....	11
Installation	11
Introduction	
Components	9
Overview	
Description	10
Features.....	10
Overview.....	10
Introduction	9
M	
mains	
fuses	3, 31
voltage.....	31
Maintenance.....	28
manual operation	33
Mounting	
headstage.....	12
N	
notes	
user.....	40
O	
Operations	
control operations.....	16
maximum positive position values	16
mode indications.....	17
moving to the Home Position.....	16
moving to the Work Position.....	17
pausing Home movements	18
pausing Work movements	18

Pulse Mode and diagonal movement	18	electrical	3
ROE axis knob movement speed control ...	18	Setting axis directionality	14
setting Absolute/Relative coordinates mode		SPEED button	18
.....	17	T	
setting position for HOME and WORK.....	16	technical specifications.....	31
display	15	dimensions	32
initial startup	15	electrical.....	31
R		SOLO ROE/Controller	31
Remote Control Commands		weight	32
Set Velocity for all Externally Controlled		V	
Movements ('v') (Firmware Ver. 2.55+)	24	voltage	
S		input.....	31
safety warnings		mains.....	31
mains fuse.....	3	W	
safety warnings & precautions		warranty	29
operational.....	3		
SAFETY WARNINGS & PRECAUTIONS	3		

NOTES