

# QUAD®

## FOUR-AXIS MOTORIZED MICROMANIPULATOR SYSTEM

WITH THREE 25MM AXES, FOURTH DIAGONAL 30MM  
AXIS, & USB INTERFACE FOR EXTERNAL CONTROL

### OPERATION MANUAL

REV. 3.41 (20220923) (FW v2.74+)



**SUTTER INSTRUMENT®**

ONE DIGITAL DRIVE  
NOVATO, CA 94949

VOICE: 415-883-0128 WEB: [WWW.SUTTER.COM](http://WWW.SUTTER.COM)  
FAX: 415-883-0572 EMAIL: [INFO@SUTTER.COM](mailto:INFO@SUTTER.COM)



Copyright © 2022 Sutter Instrument Company. All Rights Reserved.  
QUAD® is a registered trademark of Sutter Instrument Company.



# 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:** QUAD 4-Axis Motorized Micromanipulator System

**Model(s):** QUAD (system), consisting of  
QUAD/E (controller & ROE (Rotary Optical Encoder) for user control), with Power adapter (100-240 VAC to 24VDC)  
QUAD/M (motorized micromanipulator electromechanical)

---

**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 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](mailto:itcemc@itcemc.com) Web: [www.itcemc.com](http://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](mailto:info@sutter.com) Web: <http://www.sutter.com>



## DISCLAIMER

The **QUAD** 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 QUAD 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 QUAD system.
-   The QUAD 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 electro-mechanical 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 QUAD/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 QUAD 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 QUAD/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

<b>DISCLAIMER</b> .....	<b>3</b>
<b>SAFETY WARNINGS AND PRECAUTIONS</b> .....	<b>3</b>
Electrical .....	3
Avoiding Electrical Shock and Fire-related Injury.....	3
Electromagnetic Interference .....	3
Operational .....	4
Other.....	4
Handling Micropipettes.....	4
<b>1. INTRODUCTION</b> .....	<b>9</b>
1.1 Structure of the QUAD Documentation Package .....	9
1.2 Components of the QUAD System .....	9
1.3 Overview .....	10
1.3.1 Features.....	10
1.3.2 Description.....	10
<b>2. INSTALLATION</b> .....	<b>11</b>
2.1 Mounting Instructions .....	11
2.1.1 Mounting the QUAD/M to the Stand or Platform .....	11
2.2 Headstage Mounting .....	13
2.3 Other Accessories.....	13
2.4 Electrical Connections and Initial Operating Instructions.....	13
2.5 Initial Operating Instructions .....	15
2.6 ROE/Controller Rear Panel Controls and Configuration.....	15
2.6.1 Power Switch .....	15
2.6.2 Configuration Switches .....	15
2.6.2.1 Switches 1, 2, 3, and 4 .....	15
2.6.2.2 Switch 5: Locking Out Y Movement During Homing.....	15
2.6.2.3 Switch 6: Sensor Test.....	16
2.6.2.4 Switches 7 through 10.....	16
<b>3. OPERATIONS</b> .....	<b>17</b>
3.1 Main Controls and Indicators on the ROE/Controller .....	17
3.2 Display.....	17
3.2.1 Initial Startup .....	17
3.3 Control Operations .....	18
3.3.1 Maximum Positive Position Values .....	18
3.3.2 Setting Position for HOME or WORK.....	18
3.3.3 Moving to the Home Position .....	18
3.3.4 Moving to the Work Position .....	19
3.3.5 Setting Absolute/Relative Coordinates Mode.....	20
3.3.6 Mode Indications .....	20
3.3.7 Movement Knobs Disabling and Quiet Mode .....	21
3.3.8 Pausing Home Movements .....	21
3.3.9 Pausing Work Movements .....	21

3.3.10 Pulse Mode and Diagonal Movement .....	21
3.4 Setting the D-Axis Headstage Angle.....	21
3.5 Micropipette/Headstage Exchange .....	22
<b>4. EXTERNAL CONTROL .....</b>	<b>23</b>
4.1 General.....	23
4.2 Virtual COM Port (VCP) Serial Port Settings.....	24
4.3 Protocol and Handshaking .....	24
4.4 Command Sequence Formatting .....	24
4.5 Axis Position Command Parameters.....	25
4.6 Microsteps and Microns (Micrometers).....	25
4.7 Travel Ranges & Bounds .....	26
4.8 Travel Speed.....	26
4.9 Commands .....	26
4.9.1 Get Current Position ('c' or 'C') Command .....	26
4.9.2 Move to Controller-Defined HOME Position ('h') Command.....	27
4.9.3 Move to Controller-Defined WORK Position ('w') Command.....	27
4.9.4 Move to Specified "Home" Position ('H') Command.....	27
4.9.5 Move to Specified "Work" Position ('W') Command.....	28
4.9.6 Move to Specified X-Axis Position ('x' or 'X') Command.....	29
4.9.7 Move to Specified Y-Axis Position ('y' or 'Y') Command.....	29
4.9.8 Move to Specified Z-Axis Position ('z' or 'Z') Command.....	30
4.9.9 Move to Specified D-Axis Position ('d' or 'D') Command.....	30
4.9.1 Set Velocity for all Externally Controlled Movements ('v') Command (Firmware Ver. 2.51+) .....	31
4.10 Notes .....	31
<b>5. MAINTENANCE.....</b>	<b>35</b>
<b>6. RECONFIGURATION.....</b>	<b>35</b>
6.1 Changing the Rotary Knob Functions on the ROE/Controller .....	35
6.2 Changing Handedness.....	35
6.3 Moving the Location of the Control Cable Bracket.....	36
6.4 Moving the Control Cable Connector from Vertical Feed to Horizontal.....	36
6.5 Installing the Z-Axis Extension Plate.....	36
<b>APPENDIX A. LIMITED WARRANTY.....</b>	<b>39</b>
<b>APPENDIX B. ACCESSORIES .....</b>	<b>40</b>
<b>APPENDIX C. TECHNICAL SPECIFICATIONS .....</b>	<b>41</b>
<b>APPENDIX D. QUICK REFERENCE.....</b>	<b>43</b>
D.1. Manual Operation.....	43
D.2. Configuration.....	43
D.3. External Control .....	44
<b>INDEX.....</b>	<b>51</b>

## TABLE OF FIGURES

Figure 1-1. The QUAD system.....	9
Figure 2-1. Angled side view of QUAD/M showing mounting adapter plate.....	12
Figure 2-2. Locations of Mounting Bracket and Connector Bracket lock screws.....	12
Figure 2-3. QUAD/M Adapter Plate.....	13
Figure 2-4. Rear panel of QUAD ROE/Controller cabinet.....	14
Figure 2-5. Configuration switches on rear of QUAD ROE/Controller unit (switch positions shown are factory defaults).....	15
Figure 3-1. LCD Display showing startup screen.....	17
Figure 3-2. Startup screen.....	18
Figure 3-3. Factory default startup (Home) position.....	18
Figure 3-4. Maximum positive values.....	18
Figure 3-5. Moving to Home position (screen is amber while moving).....	18
Figure 3-6. Factory default Home position.....	19
Figure 3-7. Example Home position defined and saved.....	19
Figure 3-8. Example Work position.....	19
Figure 3-9. Relative mode.....	20
Figure 3-10. Relative mode.....	20
Figure 3-11. Absolute mode.....	20
Figure 3-12. Top view of QUAD/M showing lock screw for headstage/pipette angle change.....	22
Figure 3-13. Swinging out the top half of the QUAD/M for side access to headstage mount.....	23
Figure 6-1. Locations of the axis connectors inside the ROE/Controller.....	35

## TABLE OF TABLES

Table 2-1. Configuration Switches 1 – 4: Configuring the direction of each axis.....	15
Table 2-2. Configuration Switch 5: Configuring the Homing Y-Movement Lock Out.....	16
Table 2-3. Configuration Switch 6: Configuring the Sensor Test.....	16
Table 2-4. Configuration Switches 7 – 10: Reserved for future use.....	16
Table 3-1. Maximum positive position value of each axis.....	18
Table 3-2. Screen colors and modes.....	21
Table 4-1. USB-VCP interface serial port settings.....	24

Table 4-2. Microns/microsteps conversion.....	25
Table 4-3. Ranges (travel distances and bounds of each axis for compatible devices).....	26
Table 4-4. Travel speeds.....	26
Table 4-5. Get Current Position command.....	26
Table 4-6. Move to controller-defined HOME position ('h') command.....	27
Table 4-7. Move to controller-defined WORK position ('w') command.....	27
Table 4-8. Move to specified "Home" position ('H') command.....	28
Table 4-9. Move to specified "Work" position ('W') command.....	29
Table 4-10. Move to specified X-axis position ('x' or 'X') command.....	29
Table 4-11. Move to specified Y-axis position ('y' or 'Y') command.....	30
Table 4-12. Move to specified Z-axis position ('z' or 'Z') command.....	30
Table 4-13. Move to specified D-axis position ('d' or 'D') command.....	30
Table 4-14. Set velocity command ('v') command.....	31
Table C-1. QUAD cables and receptacles/connectors.....	41
Table D-1. Configuration Switches 1 through 10 definitions.....	43
Table D-2. USB-VCP interface serial port settings.....	44
Table D-3. Microns/microsteps conversion.....	45
Table D-4. Ranges and bounds.....	45
Table D-5. Travel speeds.....	45
Table D-6. QUAD external control commands.....	45

# 1. INTRODUCTION

## 1.1 Structure of the QUAD Documentation Package

The QUAD 4-Axis Micromanipulator System is comprised of a ROE/controller, a power adapter, and a QUAD/M stepper-motor-based electromechanical micromanipulator. This manual consists of four parts: This chapter, Introduction, which provides an overview and general description of the QUAD 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 QUAD; Chapter 4, Maintenance, describes how to perform routine and other maintenance of the QUAD; and Chapter 5, Reconfiguration, describes the reconfiguration possibilities of the QUAD system.

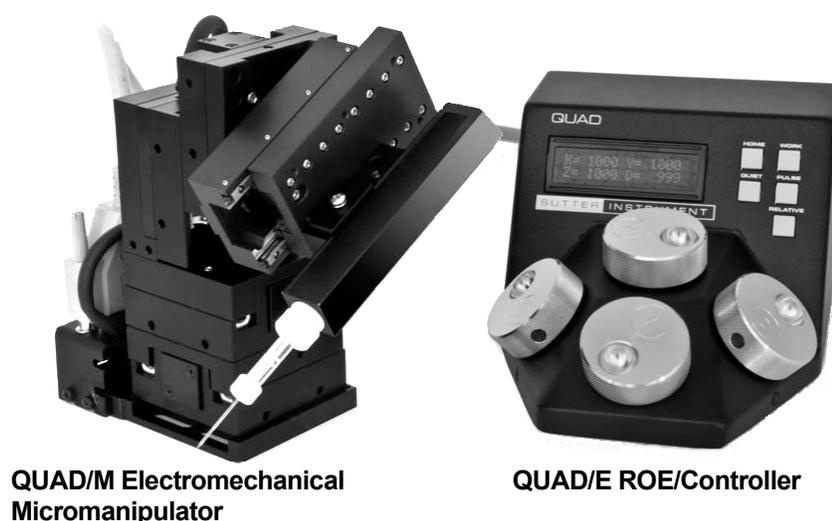


Figure 1-1. The QUAD system.

## 1.2 Components of the QUAD System

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

- QUAD ROE Rotary Optical Encoder input device with built-in controller and external power adapter.
- QUAD/M electromechanical micromanipulator
- 26-pin DSUB cable (connects the ROE/controller to the QUAD/M electromechanical micromanipulator).
- Power adapter
- Power adapter AC mains cable appropriate for your location
- Ground/Earth cable.
- Z-Axis Vertical Extender Plate with mounting hardware
- USB cable for external computer control

## IMPORTANT

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

- Four independent axes – 30mm travel in the diagonal D axis for coaxial pipette movement, and 25mm travel in X, Y, and Z
- 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, quiet, pulse and relative
- Robotic home- and work-position moves for easy automated pipette exchange

#### 1.3.2 Description

**QUAD**, the newest Sutter Instrument motorized manipulator is easy to use and has four independent axes. The X, Y, Z axes provide 25mm range of motion and Diagonal offers 30mm. The ROE controller has a digital display and keys for Home, Work, Pulse, Quiet and Relative. The compact, intuitive controller takes up minimal bench space, is fan-free, and easy to use.

While the three axes provide X, Y and Z orthogonal motion typical of most motorized manipulators, Sutter has introduced a true fourth axis with the **QUAD** so one can move the electrode coaxially at the exact desired angle of approach. The fourth axis also significantly extends the range of travel (30mm) for the system.

The **QUAD'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 100nm. Five conveniently located buttons on the ROE provide control of all the basic functions you will need in normal operation (Work, Home, Quiet, 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. When you are ready to record data, the motor drive electronics can be suppressed by pressing the **QUIET** button. In the **QUIET** mode the display turns red and ROE input is locked out to avoid any accidental motion. As of v2.3, the **QUIET** button has changed to the **SPEED** button, allowing for the settings of how fast movement occurs while rotating the knobs of the ROE, and the **QUIET** function (now called **LOCK**) activated by holding the **SPEED** button for 3 seconds. Four speeds are available (0 (normal) through 3 (slowest)), selected by each press of the **SPEED** button. 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 $\mu$ 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 **QUAD** controller/ROE – no separate controller is required.

External computer control of the **QUAD** is possible via the USB connector mounted on the controller/ROE's rear panel. The controller's internal software is programmed with a defined set of commands allowing for a wide range of micromanipulator/stage movements as programmed in software residing in an external computer connected via USB.

## 2. INSTALLATION

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

### 2.1 Mounting Instructions

The following sections describe how to mount QUAD/M manipulator to a stand using the mounting adapter plate, how to adjust pipette angle and change pipettes and how to mount different headstages. The figure below shows a right-handed QUAD/M. You may have a left-handed unit instead – the setup mirrors that of the right-handed model.

#### 2.1.1 Mounting the QUAD/M to the Stand or Platform

Examine the space of the platform onto which installation is to take place. Attach the control cable to QUAD/M and move the entire unit around on the platform until the precise desired position is determined.

The QUAD/M mounts to the mounting adapter plate using four M3.5x6 hex head locking screws. The figure below shows how the adapter plate attaches to the manipulator:

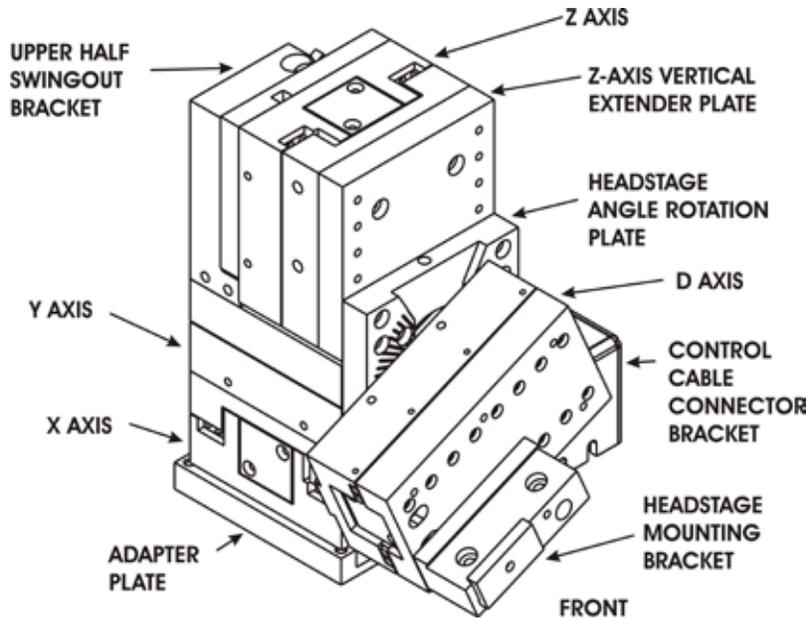


Figure 2-1. Angled side view of QUAD/M showing mounting adapter plate.

Only the front pair of screws is visible in this figure. **The adapter plate is normally shipped in a small plastic bag also containing the necessary hardware.**

The figure above shows the lowermost portion (X axis) of the QUAD/M attached to the adapter plate. Before detaching from the adapter plate, it is a good idea to slide the QUAD/M around on the base plate to determine where you want the QUAD/M positioned for the long term. Once the long-term position has been determined, unscrew the four locking screws from the plate immediately below the X axis and above the adapter plate (see the following figure).

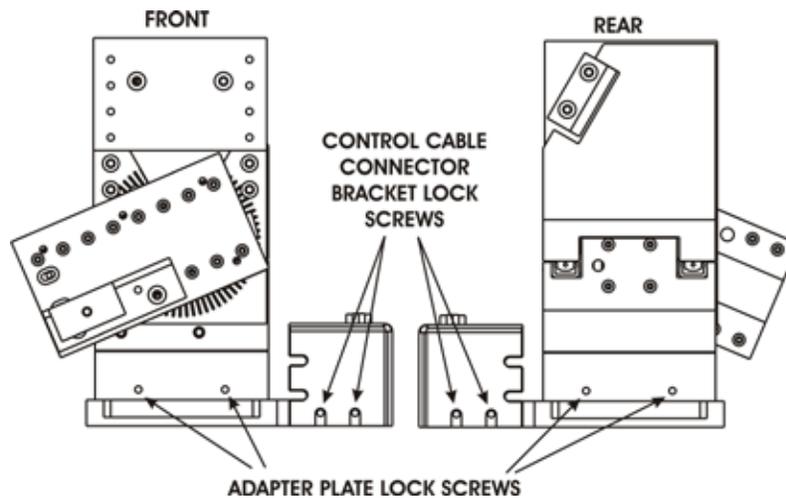


Figure 2-2. Locations of Mounting Bracket and Connector Bracket lock screws.

Once the locking screws are sufficiently loosened, lift the QUAD/M upwards from the adapter plate.

Before attaching the adapter plate to the QUAD/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). A good technique is to set the manipulator on top of the adapter plate and slide it around on your stand until it is in a good location. Then take the manipulator off and fix the adapter plate securely to the stand with the necessary hardware. Finally, align the holes on the bottom corners of the manipulator with the four threaded holes on the corners of the adapter plate and attach the manipulator using the four supplied M3.5x6 hex head locking screws.

**NOTE:** Because of the shortness of each cable between the Control Cable Connector Adapter and to each axis, the detached QUAD/M will need to be held in one hand while using the other hand to fasten the adapter plate to the stand. The process is made easier if the Control Cable Connector Adapter is also detached from the adapter plate, thereby freeing up both hands to securely install the adapter plate to the stand.

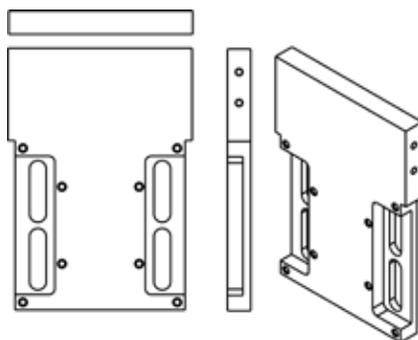


Figure 2-3. QUAD/M Adapter Plate.

## 2.2 Headstage Mounting

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 QUAD/M. The dovetail slide bracket on the QUAD/M also supports older Axon and Heka headstages

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 QUAD/M and/or modifying the headstage mount to the manipulator (i.e. rotating base, microscope stage mount, gantry stand, dovetail extension). 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 QUAD/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 QUAD/M micromanipulator electromechanical baseplate is firmly bolted down to a breadboard or solidly to a firm surface, the QUAD/M is likely to tip over when fully extending all of its axes, especially if its loaded with a headstage that extends beyond the QUAD/M’s current center of gravity.

Upon deciding to directly install the QUAD 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.

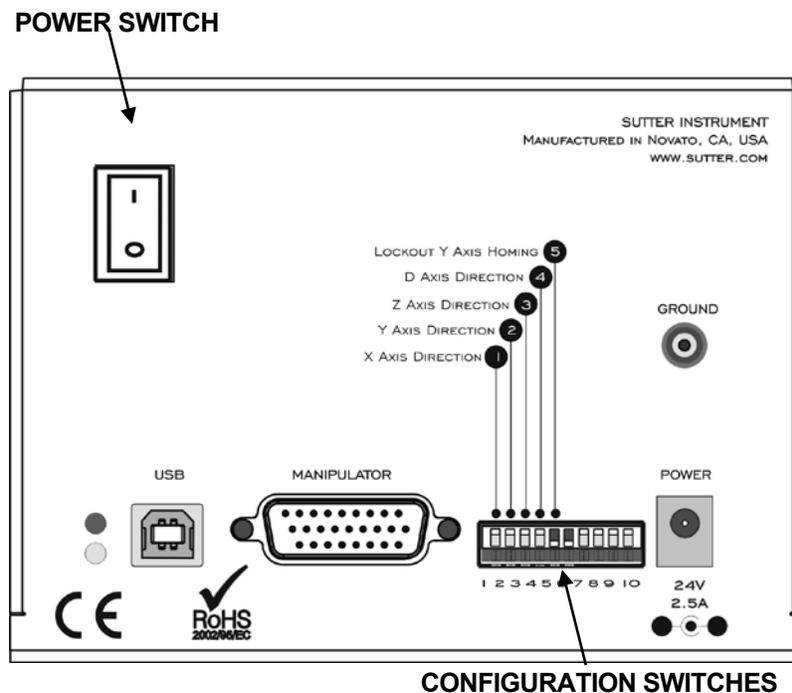


Figure 2-4. Rear panel of QUAD ROE/Controller cabinet.

2. With the power OFF (rear panel switch in the “0” position), connect a well-grounded/earthed wire to the GROUND banana plug receptacle.
3. With the power OFF, connect the male end of the DB-26 cable to the MANIPULATOR connector on the ROE, the other end of which is connected to the QUAD/M micromanipulator electromechanical. **(See cautionary note below.)**
4. Verify that the six 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 QUAD/M WHILE THE POWER IS ON!

## 2.5 Initial Operating Instructions

Please continue to the next chapter, “Operations,” for a description of the controls on the ROE/Controller, and initial power-up and general operating instructions.

## 2.6 ROE/Controller Rear Panel Controls and Configuration

### 2.6.1 Power Switch

The power switch for the QUAD 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.

### 2.6.2 Configuration Switches

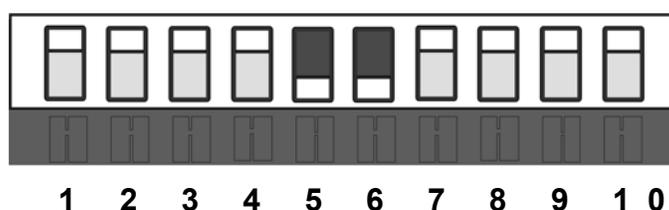


Figure 2-5. Configuration switches on rear of QUAD ROE/Controller unit (switch positions shown are factory defaults).

#### 2.6.2.1 Switches 1, 2, 3, and 4

The first four switches set the directionality for each of the four axes.

Table 2-1. Configuration Switches 1 – 4: Configuring the direction of each axis.

Switch #	Axis	Knob Rotation Directionality for Forward (+) Movement	
		Clockwise	Counterclockwise **
1	X	OFF (Up)*	ON (Down)
2	Y	OFF (Up)*	ON (Down)
3	Z	OFF (Up)*	ON (Down)
4	D	OFF (Up)*	ON (Down)

\* Factory default (typical setting for right-hand-mounted manipulator).

#### 2.6.2.2 Switch 5: Locking Out Y Movement During Homing

Configures whether the Y axis is locked out while homing.

Table 2-2. Configuration Switch 5: Configuring the Homing Y-Movement Lock Out.

Switch #	Homing Y Movement Lock Out	
	Enabled	Disabled
5	OFF (Up)	ON (Down)*

\* Factory default (recommended normal operation setting).

### 2.6.2.3 Switch 6: Sensor Test

Table 2-3. Configuration Switch 6: Configuring the Sensor Test.

Switch #	Sensor Test	
	Enabled	Disabled
6	OFF (Up)	ON (Down)*

\* Factory default (do not change unless requested to by Sutter Instrument Technical Support).

***CAUTION: To avoid damage to the micromanipulator, DIP Switch 6 (Sensor Test) must always be set to ON (DOWN).***

### 2.6.2.4 Switches 7 through 10

These switches must always be in the OFF (UP) position.

Table 2-4. Configuration Switches 7 – 10: Reserved for future use.

Switch #	Definition	Setting
7 - 10	Reserved	OFF (UP)*

## 3. OPERATIONS

### 3.1 Main Controls and Indicators on the ROE/Controller

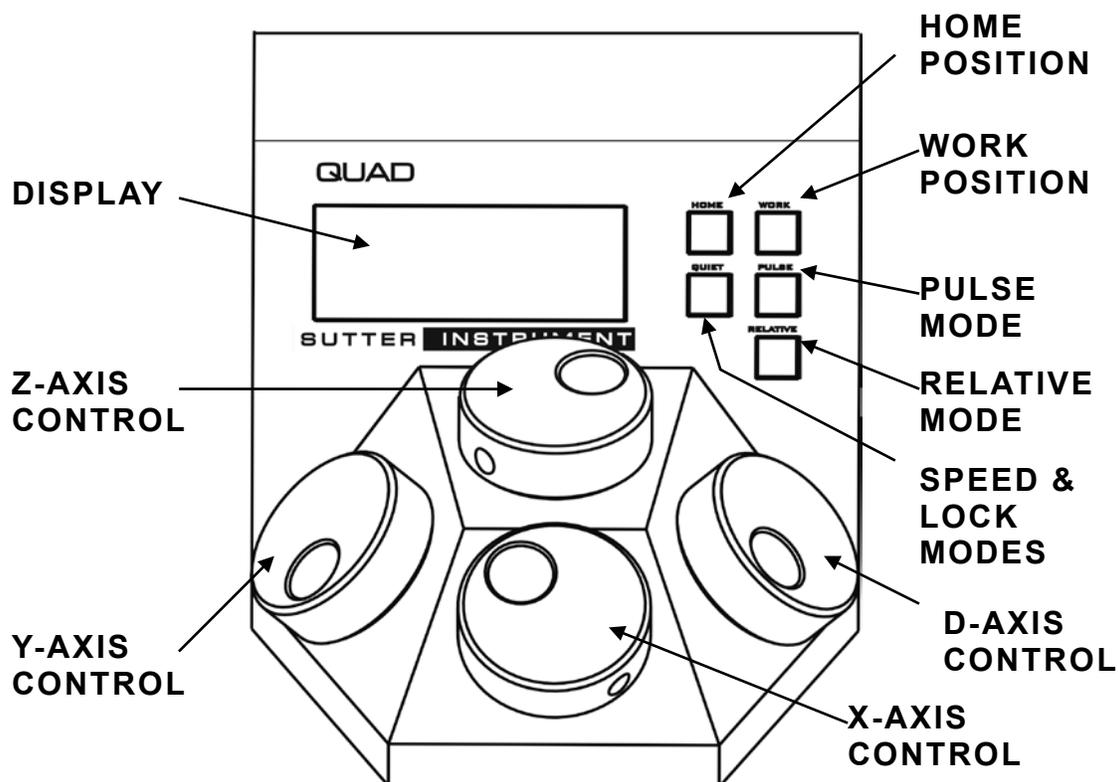


Figure 3-1. Front view of the QUAD ROE/Controller.

### 3.2 Display

#### 3.2.1 Initial Startup



(Text in  
Green)

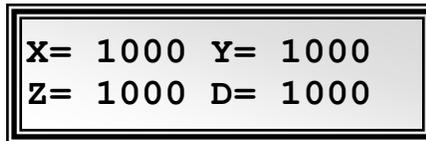
Figure 3-1. LCD Display showing startup screen.

When starting the QUAD system for the first time or if the HOME position has not yet been defined (saved), the values of all four axes will be 1,000 micrometers (microns).



(Text in Green)

Figure 3-2. Startup screen.



(Text in Green)

Figure 3-3. Factory default startup (Home) position.

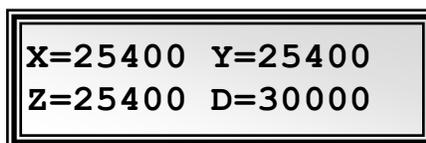
### 3.3 Control Operations

#### 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.

Axis	Maximum Position Value (in microns)
X	25,400
Y	25,400
Z	25,400
D	30,000



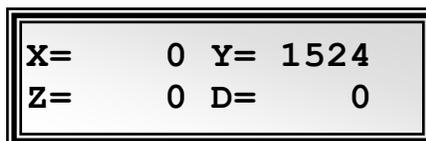
(Text in Green)

Figure 3-4. 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



(Text in Red)

Figure 3-5. Moving to Home position (screen is amber while moving)

If the Home position has not yet been defined and saved, the Home position values for all four axes will be default 1,000 microns, as shown in the following figure.

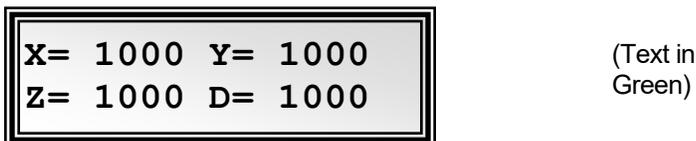


Figure 3-6. Factory default Home position.

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).

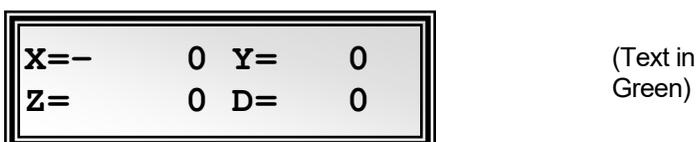


Figure 3-7. Example Home position defined and saved

To move to the Home position, press HOME. If the current position before pressing HOME is greater than the Home position, the movement will be as follows:

**NOTE:** Movement to the Home position works only if X & Z coordinates of the HOME position are less than the WORK position.

1. Movement begins by retracting the D axis (at the angle currently set) away from the sample.
2. Movement then continues along the diagonal of X and Z toward the Home position. If one of the axes reaches its end of travel before the other, the other axis then continues until reaching its end.
3. The final movement is along the Y-axis towards the operator and away from the microscope.

**NOTE:** Step 3 occurs only if “Y-Lockout” is disabled. Otherwise, no movement along the Y-axis occurs.

### 3.3.4 Moving to the Work Position

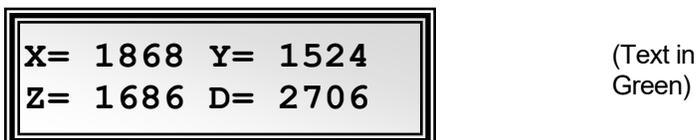


Figure 3-8. Example Work position

To move to the Work position, press the WORK button. If the current position before pressing WORK is less than the Work position, the movement will be as follows:

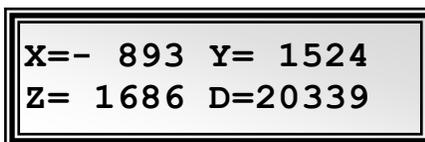
1. Movement travels along the Y-axis away from the operator and towards the microscope.
2. Movement is then made along the diagonal (X & Z) toward the sample. Whichever of the axes is longest, travel begins along that axis until reaching the start point of the diagonal vector. Travel then continues along the diagonal until reaching its end-of-travel point.
3. The final movement is along the D axis (at the angle currently set) towards the sample.

NOTE: Step 1 occurs only if “Y-Lockout” is disabled. Otherwise, movement begins with Step 2.

### 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 red. 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.

Press RELATIVE once (briefly for < 2 sec.)

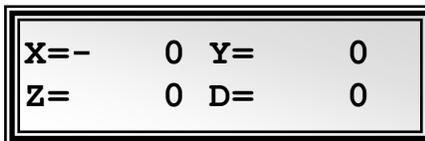


X=- 893 Y= 1524  
Z= 1686 D=20339

(Text in  
Blue)

Figure 3-9. Relative mode.

Depress RELATIVE for 3 sec. or until beep sounds

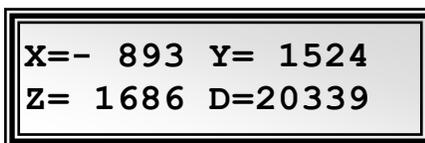


X=- 0 Y= 0  
Z= 0 D= 0

(Text in  
Blue)

Figure 3-10. Relative mode.

Pressing RELATIVE briefly while in Relative mode, returns displayed coordinates back to Absolute mode



X=- 893 Y= 1524  
Z= 1686 D=20339

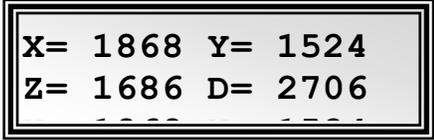
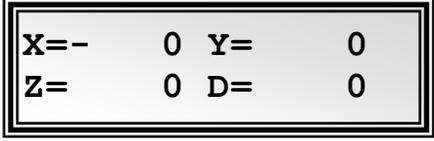
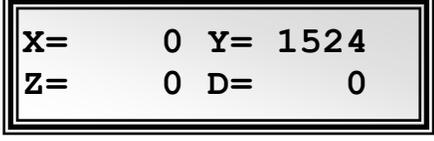
(Text in  
Blue)

Figure 3-11. Absolute mode.

### 3.3.6 Mode Indications

The QUAD system has three modes of operation: Absolute coordinates, Relative coordinates, and Quiet mode. 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 Coordinates	 <pre>X= 1868 Y= 1524 Z= 1686 D= 2706</pre>
Blue	Relative Coordinates	 <pre>X=- 0 Y= 0 Z= 0 D= 0</pre>
Red	Knobs disabled during move to Home or Work position, while in Quiet mode.	 <pre>X= 0 Y= 1524 Z= 0 D= 0</pre>

### 3.3.7 Movement Knobs Disabling and Quiet Mode

Axis-movement knobs are disabled during movements to Home, Work, or while in Quiet Mode.

### 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 and Diagonal Movement

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

## 3.4 Setting the D-Axis Headstage Angle

The angle of the D-axis is adjusted by loosening the hex set screw located on the top-center of angle bracket (attached to the Z axis), as shown in the following figure. Rotate the D axis until reaching the desired angle. Note that angle range is limited only to that which does not cause interference with the control cable leading to the D axis. Once establishing the desired angle, retighten the lock screw to fix the angle.

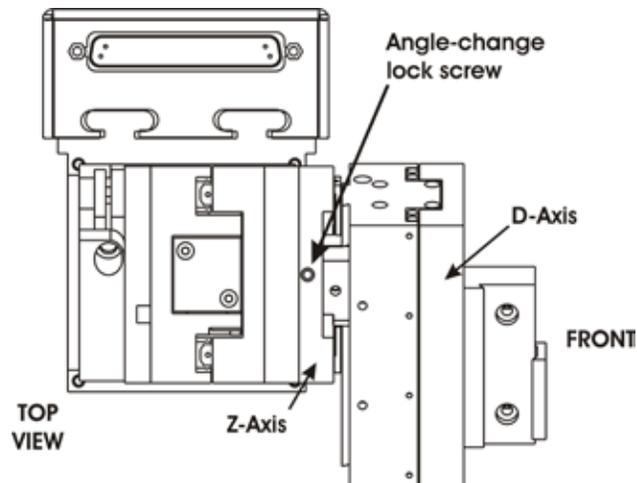


Figure 3-12. Top view of QUAD/M showing lock screw for headstage/pipette angle change.

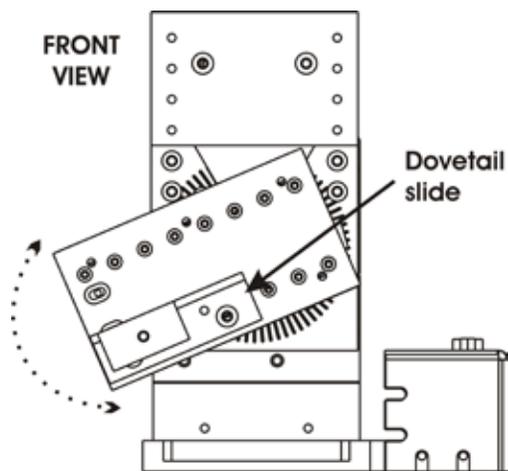


Figure 3-2. Front view of QUAD/M showing D-axis rotation.

### 3.5 Micropipette/Headstage Exchange

Mounted on the front of the Z-axis of the manipulator is the angle-control plate for the D axis/headstage mount. All these components combined (making up the upper half of the QUAD/M), are mounted on the hinge plate that allows this upper half to swing away from the Y and X axes (the lower half), so as to aid in the exchange of headstages or micropipettes. To swing out the upper half of the manipulator, lightly grasp the top diagonally opposite corners of the Z axis, and twist to the right (if right-handed QUAD/M/R) or left (if left-handed QUAD/M/L) until the Z-axis is 90° away from its operational position.

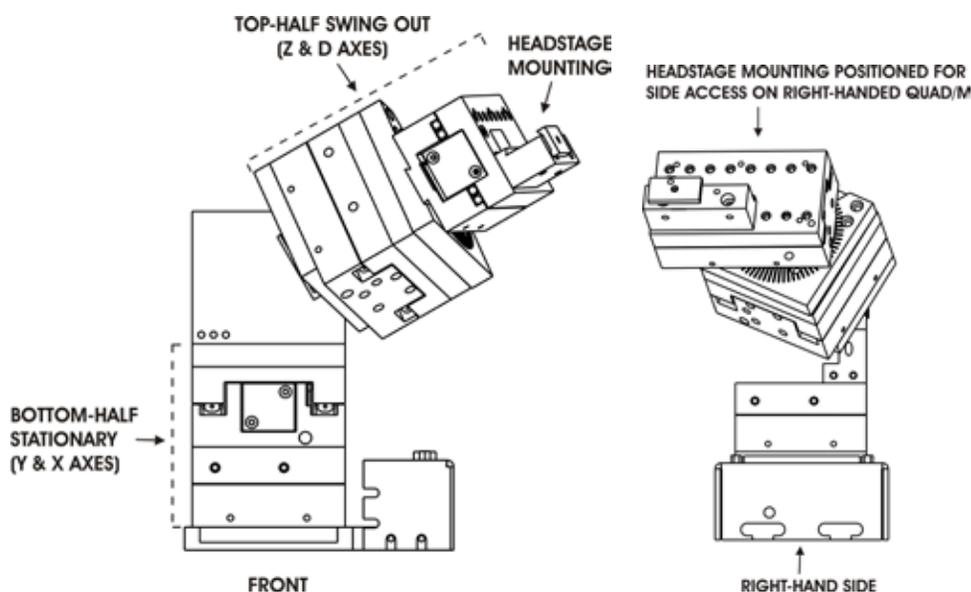


Figure 3-13. Swinging out the top half of the QUAD/M for side access to headstage mount.

To change the headstage, loosen the screw at the top edge of the holding bracket. Lightly tighten the holding bracket screw and then carefully swing the top half of the QUAD/M back into the normal position for operation, all the time making sure that the headstage does not collide with any part of the microscope or sample. Once the swing is complete into the position for operation, make any adjustments needed of the headstage within the holding bracket, and then tighten down (but do not over tighten) the lock screw at the top of the holding bracket.

## 4. EXTERNAL CONTROL

### 4.1 General

Controlling the QUAD 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 QUAD controller/ROE. The USB device driver for Windows is downloadable from Sutter Instrument's web site ([www.sutter.com](http://www.sutter.com)). The QUAD 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 QUAD. 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.

## 4.2 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.

## 4.3 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.

## 4.4 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 QUAD, 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.

## 4.5 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.

## 4.6 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.

System/Device	From/To Units	Conversion Factor (multiplier)
QUAD with QUAD/M micromanipulator	$\mu\text{steps} \rightarrow \mu\text{m}$	0.09375
	$\mu\text{m} \rightarrow \mu\text{steps}$	10.66666666667

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.

## 4.7 Travel Ranges & Bounds

The following table shows the travel lengths, ranges, and bounds for each axis of each supported device.

Table 4-3. Ranges (travel distances and bounds of each axis for compatible devices).

Device	Axis	Millimeters	Microns	Microsteps
QUAD/M	X	0 - 25	0 – 25,000	0 – 266,667
	Y	0 - 25	0 – 25,000	0 – 266,667
	Z	0 - 25	0 – 25,000	0 – 266,667
	D	0 - 30	0 – 30,000	0 – 320,000

*NOTE: Origin is a physical position of travel that defines the beginning of the absolute position coordinate system (i.e., absolute position 0).*

*Physical Positions: BOT (Beginning Of Travel), COT (Center Of Travel), & EOT (End Of Travel).*

*In the QUAD, the Origin is fixed at BOT.*

## 4.8 Travel Speed

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

Table 4-4. Travel speeds.

Device	mm/sec or $\mu\text{m}/\text{ms}$	
	Single Axis	Dual Axis (x 1.4)
QUAD/M	3	4.2

## 4.9 Commands

### 4.9.1 Get Current Position ('c' or 'C') Command

This command is used to obtain the current position (X, Y, Z, & D coordinates) of the manipulator. The command sequence consists of one byte as shown in the following table. Seventeen bytes are returned: X, Y, Z, & D position values in microsteps (4 bytes each), and completion indicator (1 byte).

Table 4-5. Get Current Position command.

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 positions ( $\mu\text{steps}$ ) of X, Y, Z, & D axes.
	Rx	All	17		Current absolute position of the X, Y, Z, & D axes, in microsteps, each consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value in Little-Endian bit order. See <i>Ranges</i> table for						

Command	Tx/ Delay/ Rx	Ver .	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
					Dec.	Hex.	Binary				
					minimum and maximum values.						
				0 (4)	X pos. in $\mu$ steps						
				4 (4)	Y pos. in $\mu$ steps						
				8 (4)	Z pos. in $\mu$ steps						
				12 (4)	D pos. in $\mu$ steps						
				13	13	0D	0000 1101		^M	<CR>	Completion indicator

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

Table 4-6. Move to controller-defined HOME position ('h') command.

Command	Tx/ Delay/ Rx	Ver .	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
					Dec.	Hex.	Binary				
Move to HOME Position ('h')	Tx	All	1	0	104	68	0110 1000	0104		'h'	Moves to the position saved for the controller's HOME button. <b>D</b> moves <u>first</u> to the target, followed by <b>Z</b> , and then <b>X &amp; Y</b> together <u>last</u> (simultaneity determined by shape of 2D area of travel).
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

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

Table 4-7. Move to controller-defined WORK position ('w') command.

Command	Tx/ Delay/ Rx	Ver .	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
					Dec.	Hex.	Binary				
Move to WORK Position ('w')	Tx	All	1	0	119	77	0111 0111	0119		'w'	Moves to the position saved for the controller's WORK button. <b>X &amp; Y</b> move together to the target <u>first</u> (simultaneity determined by shape of 2D area of travel), followed by <b>Z</b> , and then <b>D</b> <u>last</u> .
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

#### 4.9.4 Move to Specified "Home" Position ('H') Command

This command instructs the controller to move all four axes of the manipulator to the position specified, moving D axis first, followed by Z, and finally X & Y together. The command sequence consists of seventeen bytes.

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

Command	Tx/ Delay/ Rx	Ver .	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description			
					Dec.	Hex.	Binary							
Move to Specified “Home” Position (‘H’)	Tx	All	17	0	72	48	0100 1000	0072		‘H’	Move all 4 axes to the specified position as if away from a work position. <b>D</b> moves <u>first</u> to the target, followed by <b>Z</b> , and then <b>X &amp; Y</b> together <u>last</u> (simultaneity determined by shape of 2D area of travel) (see <i>Ranges</i> table)			
					Target absolute position of X, Y, Z, & D axes, in microsteps, each 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.									
					1 (4)	X $\mu$ steps								
					5 (4)	Y $\mu$ steps								
					9 (4)	Z $\mu$ steps								
					13 (4)	D $\mu$ steps								
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Completion indicator			

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

This command instructs the controller to move all four axes of the manipulator to the position specified, moving X & Y together first, then Z, and finally D. The command sequence consists of seventeen bytes.

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

Command	Tx/ Delay/ Rx	Ver .	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description			
					Dec.	Hex.	Binary							
Move to Specified “Work” Position (‘W’)	Tx	All	17	0	87	57	0101 0111	0087		‘W’	Move all 4 axes to the specified position, as if away from the home position and approaching a work position. <b>X &amp; Y</b> move together to the target <u>first</u> (simultaneity determined by shape of 2D area of travel), followed by <b>Z</b> , and then <b>D last</b> (see <i>Ranges</i> table)			
					Target absolute position of X, Y, Z, & D axes, in microsteps, each 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.									
					1 (4)						X $\mu$ steps			
					5 (4)						Y $\mu$ steps			
					9 (4)						Z $\mu$ steps			
13 (4)						D $\mu$ steps								
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Completion indicator			

#### 4.9.6 Move to Specified X-Axis Position (‘x’ or ‘X’) Command

This command moves to a specified position for only the X-axis.

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

Command	Tx/ Delay/ Rx	Ver .	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
					Dec.	Hex.	Binary				
Move to specified X axis Position (‘x’ or ‘X’)	Tx	All	5	0	120 or 90	78 or 5A	0111 1000 or 0101 1010	0120 or 0090		‘x’ or ‘X’	Move X axis to specified position (see <i>Ranges</i> table)
					1 (4)	Target absolute position of X axis, 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

#### 4.9.7 Move to Specified Y-Axis Position (‘y’ or ‘Y’) Command

This command moves to a specified position for only the Y-axis.

Table 4-11. Move to specified Y-axis position ('y' or 'Y') command.

Command	Tx/ Delay/ Rx	Ver .	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
					Dec.	Hex.	Binary				
Move to specified Y axis Position (‘y’ or ‘Y’)	Tx	All	5	0	121 or 91	79 or 5B	0111 1001 or 0101 1011	0121 or 0091		‘y’ or ‘Y’	Move Y axis to specified position (see <i>Ranges</i> table)
				1 (4)	Target absolute position of Y axis, 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

#### 4.9.8 Move to Specified Z-Axis Position ('z' or 'Z') Command

This command moves to a specified position for only the Z-axis.

Table 4-12. Move to specified Z-axis position ('z' or 'Z') command.

Command	Tx/ Delay/ Rx	Ver .	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
					Dec.	Hex.	Binary				
Move to specified Z axis Position (‘z’ or ‘Z’)	Tx	All	5	0	122 or 92	7A or 5C	0111 1010 or 0101 1100	0122 or 0092		‘z’ or ‘Z’	Move Z-axis to specified position (see <i>Ranges</i> table)
				1 (4)	Target absolute position of Z axis, 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

#### 4.9.9 Move to Specified D-Axis Position ('d' or 'D') Command

This command moves to a specified position for only the D-axis.

Table 4-13. Move to specified D-axis position ('d' or 'D') command.

Command	Tx/ Delay/ Rx	Ver .	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
					Dec.	Hex.	Binary				
Move to specified D axis Position (‘d’ or ‘D’)	Tx	All	5	0	100 or 68	64 or 44	0110 0100 or 0100 0100	0100 or 0068		‘d’ or ‘D’	Move D-axis to specified position (see <i>Ranges</i> table)
				1 (4)	Target absolute position of D axis, 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

#### 4.9.1 Set Velocity for all Externally Controlled Movements ('v') Command (Firmware Ver. 2.51+)

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 4-14. 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	2.51+	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

#### 4.10 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 ( $\mu$ steps). See *Microns/microsteps conversion* table) for conversion between  $\mu$ steps and microns (micrometers ( $\mu$ m)).

*Declaring position variables in C/C++:*

```
/* current position for X, Y, Z, & D */
unsigned long cp_x_us, cp_y_us, cp_z_us, cp_d_us; /* microsteps */
double cp_x_um, cp_y_um, cp_z_um, cp_d_um; /* microns */
/* specified (move-to) position for X, Y, Z, & D */
unsigned long sp_x_us, sp_y_us, sp_z_us, sp_d_us; /* microsteps */
double sp_x_um, sp_y_um, sp_z_um, sp_d_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 QUAD/M based config. */
double us2umCF = 0.09375; /* microsteps to microns */
double um2usCF = 10.666666666667; /* 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 Y, Z, & D, and 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 ( $\geq 0$  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, min_y_um, min_z_um, min_d_um; /* minimum microns */
double max_x_um, max_y_um, max_z_um, max_d_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;
min_y_um = 0;
min_z_um = 0;
min_d_um = 0;
/* initialize all maximum positions in microns*/
/* QUAD/M */
max_x_um = 25000;
max_y_um = 25000;
max_z_um = 25000;
max_d_um = 30000;
```

- 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, rp_y_um, rp_z_um, rp_d_um; /* microns */
/* initialize all relative positions to 0 after declaring them */
rp_x_um = rp_y_um = rp_z_um = rp_d_um = 0;
```

*Enter any positive or negative value for each relative position (e.g., rp\_x\_um = 1000; rp\_y\_um = 500; rp\_z\_um = -200; rp\_d\_um = -500 ... etc.*

*For each 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 positimized 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 */
```

*Repeat the above bounds check for each of the remaining axes.*

*For each axis, 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;
```

*Repeat for each of the remaining axes as required before issuing a move command.*

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  $\mu\text{m}$ ) 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 v2.51 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 QUAD 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.

## 6. RECONFIGURATION

### 6.1 Changing the Rotary Knob Functions on the ROE/Controller

The axis motor assignment of each axis control knob on the ROE can be changed by opening the ROE/Controller cabinet as seen in the figure below and changing cables to appropriate connectors.

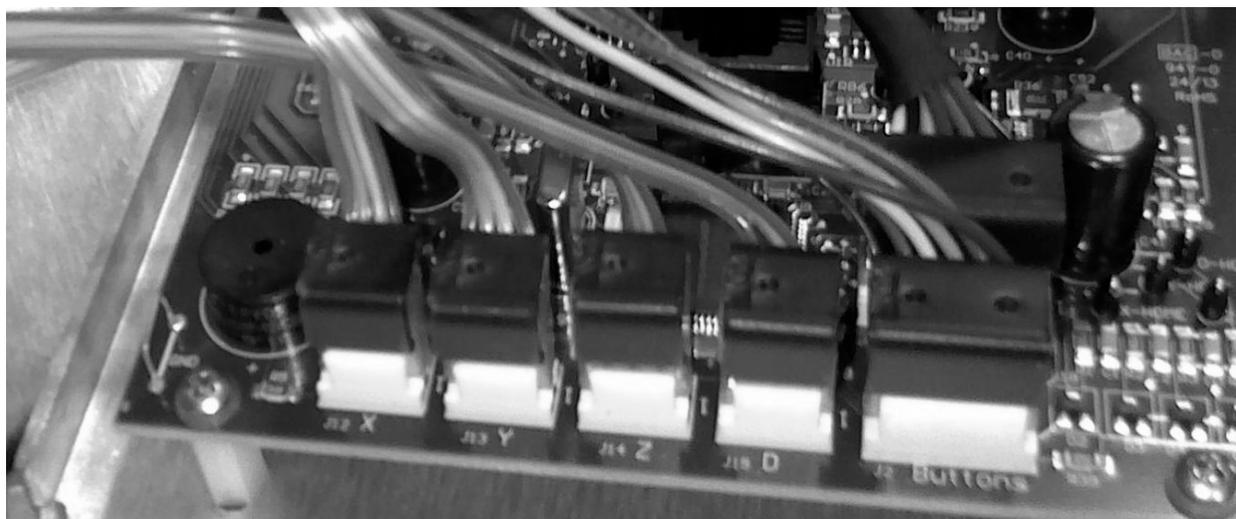


Figure 6-1. Locations of the axis connectors inside the ROE/Controller.

### 6.2 Changing Handedness

When the QUAD system is ordered, the handedness of the QUAD/M micromanipulator electromechanical and the operation of the ROE/Controller can be specified for right (default) or left. If the handedness of a system already shipped to you requires changing, the system must be returned to Sutter Instrument and the handedness will be changed as specified for a nominal charge. Please contact Sutter Instrument Customer Service for details.

### 6.3 Moving the Location of the Control Cable Bracket

For right-handed QUAD/M, the Control Cable Connector Bracket is located on the right side of the QUAD/M, and on the left side for left-handed versions. In either case, the bracket can be moved to the rear of the QUAD/M. When making this change, the entire QUAD/M must be separated from the Adapter Plate first, rotated a quarter of turn, and then repositioned on the four posts of Adapter Plate. Note that if the front of the QUAD/M is to remain facing in the same direction, the Adapter Plate must also be rotated and then refastened to the base plate. Note also that when moving the cable bracket to the rear of the QUAD/M, the bracket blocks access to two of the locking screws on X-axis component that locks the QUAD/M in place with the Adapter Plate, thereby requiring removing the cable bracket before separating the QUAD/M from the Adapter Plate.

### 6.4 Moving the Control Cable Connector from Vertical Feed to Horizontal

The orientation of the control cable connector can be changed between vertical and horizontal feeds. Loosen the four lock screws of the Connector Bracket, lift the bracket upwards, tilt up or down, make a half turn, remount to the Adapter Plate, and finally tighten the four locking screws. While making the change, be careful not to tug the wires tightly or bend them.

### 6.5 Installing the Z-Axis Extension Plate

If the QUAD's Diagonal Axis is too tall or the angle of approach is too steep for your application, install the Z-Axis Extension Plate provided. Once the plate is installed, the D-Axis Rotator Plate can be installed in a lower position.

The Rotator and D-axis are normally installed in the center position on the Z-axis vertical plate. If the entire Rotator and D-Axis mechanism must be installed at a lower aspect, an adapter plate must be installed that allows the Z axis to move the Rotator/D-Axis mechanism to down past the ridge.

#### **Step by step instructions:**

Using an M1.5mm Allen wrench, loosen the D-Axis Rotator lock screw, and rotate the D-Axis to approximately the horizontal position to expose the two screws that hold the Rotator Plate in place.

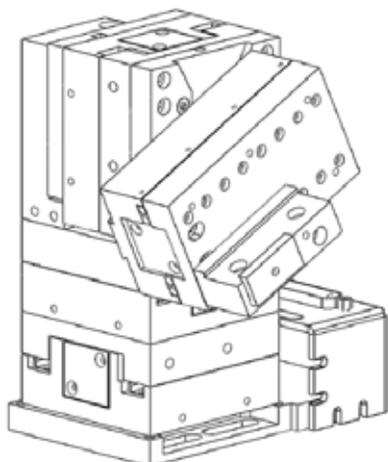


Figure 6-1. Default Rotator/D-Axis mechanism mounting position.

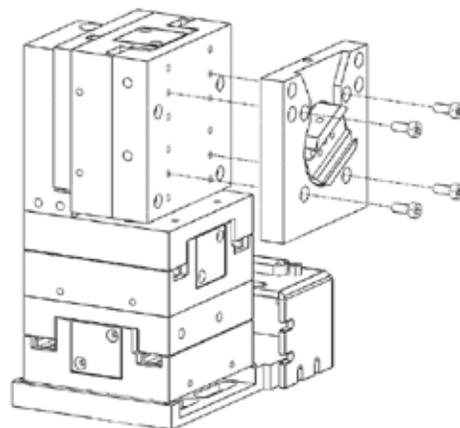


Figure 6-2. Removing the Rotator Plate from the Z-Axis Plate.

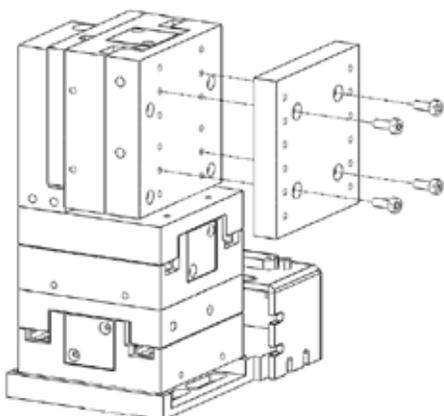


Figure 6-3. Installing the Vertical Extender Plate to the Z-Axis Plate.

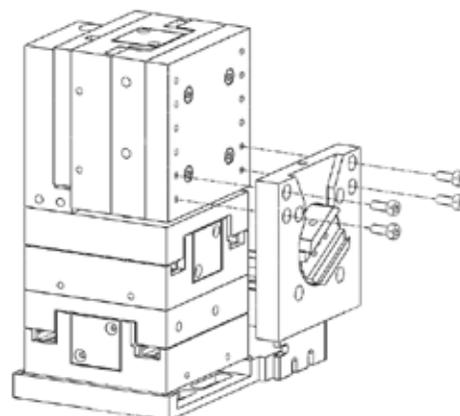


Figure 6-4. Installing the Rotator to the lower position on the Vertical Extender Adapter.

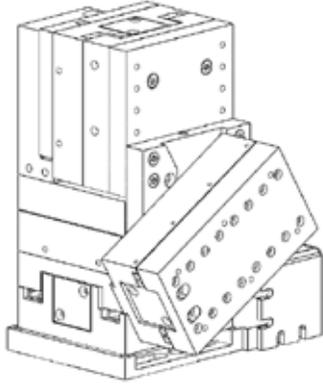


Figure 6-5. The QUAD/M showing the Rotator and D-Axis at a lower position on the Z-Axis.

## 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.
- Consumables, PMTs, galvanometers, and Uniblitz<sup>®1</sup> shutters are exempt from this warranty.
- 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.

---

<sup>1</sup> Uniblitz<sup>®</sup> is a registered trademark of Vincent Associates.

## APPENDIX B. ACCESSORIES

<b>W621150</b>	Ground cable
<b>285204</b>	4-inch dovetail extension
<b>285210</b>	Mounting adapter plate
<b>225RBI</b>	Rotating base
<b>221165</b>	Z-axis vertical extension
<b>BR-AW</b>	Rod clamp for XenoWorks injectors (for rod OD 2-4 mm)
<b>BR-AW-L</b>	Long rod clamp for XenoWorks injectors (for rod OD 2-4 mm)
<b>MP-ROD</b>	Rod holder (for rod OD 6.25mm or larger)
<b>MD Series</b>	Micromanipulator platform
<b>MT-78-FS</b>	Large fixed-stage platform
<b>MT-78-FS/M6</b>	Large fixed-stage platform with M6 tapped holes
<b>MT-75</b>	Standard gantry-stand 8.7 to 13.4 in (22.1 to 33.9 cm)
<b>MT-75S</b>	Short gantry-stand 6.7 to 9.6 in (16.9 to 24.4 cm)
<b>MP-RISER-0.5</b>	½-inch riser (risers can be combined to achieve desired height)
<b>MP-RISER-1.0</b>	1-inch riser (risers can be combined to achieve desired height)

## APPENDIX C. TECHNICAL SPECIFICATIONS



<b>Travel</b>	25mm on X, Y, and Z axes 30mm on D (diagonal) axis
<b>Resolution</b>	Minimal microstep size is 62.5 nanometers per microstep. Display has single micron resolution.
<b>Speed</b>	3 mm/sec.
<b>Long Term Stability</b>	< 1 micron/hour drive mechanism
<b>Electrical:</b>	
<b>Power Adapter:</b>	Meanwell GS60A24-P1J
<b>Input (mains)</b>	100 - 240 VAC, 50/60 Hz, 1.4A
<b>Output (to controller)</b>	24V DC, 2.5A, 60W Max. (see following table for cable specs)
<b>System Power consumption</b>	60-Watts maximum
<b>Mains fuses</b>	None replaceable (power protection built into the Power Adapter)
<b>Cables</b>	(Refer to the following tables for a description of all possible cables.)

Table C-1. QUAD 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 (26-Pin HD DSUB female receptacle)	← HD DB-26 male   HD DB-26 female → (Straight-through)	QUAD/M	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	ROE/Controller Cabinet: POWER receptacle	UL1185 18AWG	1.8 meters (approx. 6 feet)

Controller Rear Panel Port Connector/Receptacle	Cable Connector Types	Connects to ...	Cable Type	Cable Max. Length
	Barrel Plug (male) →	(center pin positive)		
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:**

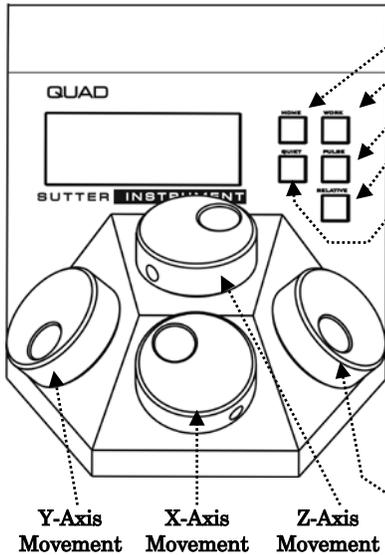
QUAD ROE/controller                      5.5 x 7.5 x 4 in | 14 x 19 x 10.2 cm

**Weight:**

QUAD ROE/controller                      2.2 lbs. | 1.0 kg

# APPENDIX D. QUICK REFERENCE

## D.1. Manual Operation



**HOME:** Move to defined home position. Press again to pause/resume.

**WORK:** Move to defined work position. Press again to pause/resume.

**PULSE:** Advances diagonal axis in 2.85 μm steps.

**RELATIVE:** Toggles between **Relative** and **Absolute** position moves. Hold 3-sec. to set the relative mode origin to the current absolute position.

**SPEED:** Cycles through Speed 0 (normal) through 3. Hold 3 sec. to enter **LOCK** mode (Ver. 2.3+).

**Setting Home, Work, or Relative Mode Origin Position:** To set position, hold down HOME, WORK, or RELATIVE button for 3 seconds until beep sounds.

**Screen-color mode indications:** Green = Absolute position; Blue = Relative position; Red = Movement in progress or in quiet (**LOCK**) mode; knobs disabled.

**Movement Knobs Disabling and Quiet (**LOCK**) Mode:** Movement knobs are disabled during movement to Home, Work, external movement command, or while in quiet (**LOCK**) mode.

## D.2. Configuration

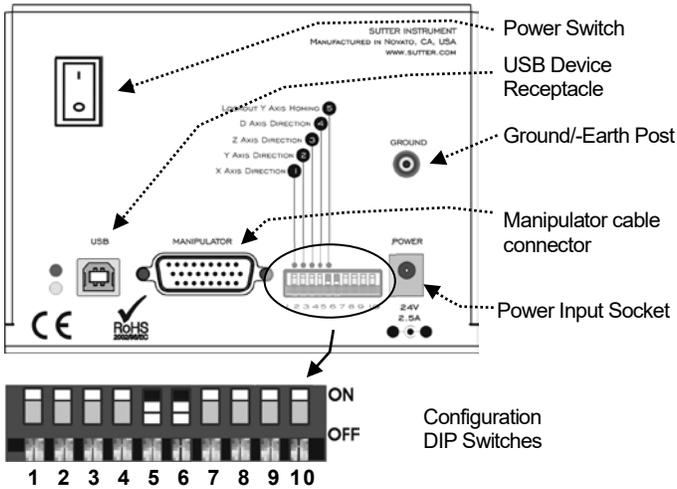


Table D-1. Configuration Switches 1 through 10 definitions

Sw #	Definition	State	Setting	Position
1	X –Axis Knob Directionality for Forward (+) Movement	Clockwise	OFF*	UP*
		Counter	ON	DOWN
2	Y –Axis Knob Directionality for Forward (+) Movement	Clockwise	OFF*	UP*
		Counter	ON	DOWN
3	Z –Axis Knob Directionality for Forward (+) Movement	Clockwise	OFF*	UP*
		Counter	ON	DOWN
4	D –Axis Knob Directionality for Forward (+) Movement	Clockwise	OFF*	UP*
		Counter	ON	DOWN
5	Y Axis Lock Out for Homing	Enabled	OFF	UP
		Disabled	ON*	DOWN*
6	Sensor Test** (see Caution)	Enabled	OFF	UP
		Disabled	ON*	DOWN*
7 - 10	Reserved	Disabled	OFF*	UP*

\*Normal operation (factory default).

\*\*CAUTION: To avoid damage to the micromanipulator/stage, DIP Switch 6 (Sensor Test) must always be set to ON (DOWN).

### D.3. External Control

Controlling the QUAD 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 QUAD controller/ROE. The USB device driver for Windows is downloadable from Sutter Instrument’s web site ([www.sutter.com](http://www.sutter.com)). The QUAD 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 QUAD. 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 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 QUAD, 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.

**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.

**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/microsteps conversion.

System/Device	From/To Units	Conversion Factor (multiplier)
QUAD with QUAD/M micromanipulator	$\mu\text{steps} \rightarrow \mu\text{m}$	0.09375
	$\mu\text{m} \rightarrow \mu\text{steps}$	10.6666666667

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.

**Travel Ranges & Bounds:** The following table shows the travel lengths, ranges, and bounds for each axis of each supported device.

Table D-4. Ranges and bounds

Device	Axis	Millimeters	Microns	Microsteps
QUAD/M	X	0 - 25	0 - 25,000	0 - 266,667
	Y	0 - 25	0 - 25,000	0 - 266,667
	Z	0 - 25	0 - 25,000	0 - 266,667
	D	0 - 30	0 - 30,000	0 - 320,000

*NOTE: Origin is a physical position of travel that defines the beginning of the absolute position coordinate system (i.e., absolute position 0).*

*Physical Positions: BOT (Beginning Of Travel), COT (Center Of Travel), & EOT (End Of Travel).*

*In the QUAD, the Origin is fixed at BOT.*

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

Table D-5. Travel speeds.

Device	mm/sec or $\mu\text{m}/\text{ms}$	
	Single Axis	Dual Axis (x 1.4)
QUAD/M	3	4.2

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

Table D-6. QUAD 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 positions ( $\mu\text{steps}$ ) of X, Y, Z, & D axes.
	Rx	All	17		Current absolute positions of the X, Y, Z, & D axes, in microsteps, each consisting of 4 contiguous bytes representing a single 32-bit unsigned (positive) integer value in Little-Endian bit order. See <i>Ranges</i> table for minimum and maximum values.						
					0 (4)	X pos. in $\mu\text{steps}$					
					4 (4)	Y pos. in $\mu\text{steps}$					
					8 (4)	Z pos. in $\mu\text{steps}$					
					12 (4)	D pos. in $\mu\text{steps}$					
				13	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					
Move to HOME Position ('h')	Tx	All	1	0	104	68	0110 1000	0104		'h'	Moves to the position saved for the controller's HOME button. <b>D</b> moves <u>first</u> to the target, followed by <b>Z</b> , and then <b>X &amp; Y</b> together <u>last</u> (simultaneity determined by shape of 2D area of travel).	
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator	
Move to WORK Position ('w')	Tx	All	1	0	119	77	0111 0111	0119		'w'	Moves to the position saved for the controller's WORK button. <b>X &amp; Y</b> move together to the target <u>first</u> (simultaneity determined by shape of 2D area of travel), followed by <b>Z</b> , and then <b>D</b> <u>last</u> .	
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator	
Move to Specified "Home" Position ('H')	Tx	All	17	0	72	48	0100 1000	0072		'H'	Move all 4 axes to the specified position as if away from a work position. <b>D</b> moves <u>first</u> to the target, followed by <b>Z</b> , and then <b>X &amp; Y</b> together <u>last</u> (simultaneity determined by shape of 2D area of travel) (see <i>Ranges</i> table)	
											Target absolute positions of X, Y, Z, & D axes, in microsteps, each 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.	
											1 (4)	X $\mu$ steps
											5 (4)	Y $\mu$ steps
											9 (4)	Z $\mu$ steps
	13 (4)	D $\mu$ steps										
Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Completion indicator		
Move to Specified "Work" Position ('W')	Tx	All	17	0	87	57	0101 0111	0087		'W'	Move all 4 axes to the specified position, as if away from the home position and approaching a work position. <b>X &amp; Y</b> move together to the target <u>first</u> (simultaneity determined by shape of 2D area of travel), followed by <b>Z</b> , and then <b>D</b> <u>last</u> (see <i>Ranges</i> table)	
											Target absolute positions of X, Y, Z, & D axes, in microsteps, each 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.	
											1 (4)	X $\mu$ steps
											5 (4)	Y $\mu$ steps
											9 (4)	Z $\mu$ steps
	13 (4)	D $\mu$ steps										
Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Completion indicator		
Move to specified X axis Position ( 'x' or 'X' )	Tx	All	5	0	120	78	0111 1000	0120		'x' or 'X'	Move X axis to specified position (see <i>Ranges</i> table)	
					or 90	or 5A	or 0101 1010	or 0090				
				1 (4)	Target absolute position of X axis, 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.							

Command	Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./ char.	Description
					Dec.	Hex.	Binary				
	Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator
Move to specified Y axis Position (‘y’ or ‘Y’)	Tx	All	5	0	121	79	0111 1001	0121		‘y’	Move Y axis to specified position (see <i>Ranges</i> table)
					or 91	or 5B	or 0101 1011	or 0091	or ‘Y’		
	1 (4)	Target absolute position of Y axis, 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			<CR>	Completion indicator
Move to specified Z axis Position (‘z’ or ‘Z’)	Tx	All	5	0	122	7A	0111 1010	0122		‘z’	Move Z-axis to specified position (see <i>Ranges</i> table)
					or 92	or 5C	or 0101 1100	or 0092	or ‘Z’		
	1 (4)	Target absolute position of Z axis, 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			<CR>	Completion indicator
Move to specified D axis Position (‘d’ or ‘D’)	Tx	All	5	0	100	64	0110 0100	0100		‘d’	Move D-axis to specified position (see <i>Ranges</i> table)
					or 68	or 44	or 0100 0100	or 0068	or ‘D’		
	1 (4)	Target absolute position of D axis, 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			<CR>	Completion indicator
Set Velocity (‘v’)	Tx	2.51+	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>

## 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 ( $\mu$ steps). See *Microns/microsteps conversion* table) for conversion between  $\mu$ steps and microns (micrometers ( $\mu$ m)).

*Declaring position variables in C/C++:*

```

/* current position for X, Y, Z, & D */
unsigned long cp_x_us, cp_y_us, cp_z_us,
cp_d_us; /* microsteps */
double cp_x_um, cp_y_um, cp_z_um,
cp_d_um; /* microns */
/* specified (move-to) position for X, Y, Z, & D */
unsigned long sp_x_us, sp_y_us, sp_z_us,
sp_d_us; /* microsteps */

```

```

double sp_x_um, sp_y_um, sp_z_um,
sp_d_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 QUAD/M based
config. */
double us2umCF = 0.09375; /* microsteps
to microns */
double um2usCF = 10.666666666667; /* 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 Y, Z, & D, and 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 ( $\geq 0$  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, min_y_um, min_z_um, min_d_um;
/* minimum microns */
double max_x_um, max_y_um, max_z_um, max_d_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;
min_y_um = 0;
min_z_um = 0;
min_d_um = 0;
/* initialize all maximum positions in microns*/
/* QUAD/M */
max_x_um = 25000;
max_y_um = 25000;
max_z_um = 25000;
max_d_um = 30000;
```

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, rp_y_um, rp_z_um, rp_d_um; /*
microns */
/* initialize all relative positions to 0 after
declaring them */
rp_x_um = rp_y_um = rp_z_um = rp_d_um = 0;
```

*Enter any positive or negative value for each relative position (e.g., rp\_x\_um = 1000; rp\_y\_um = 500; rp\_z\_um = -200; rp\_d\_um = -500 ... etc.*

*For each 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 */
```

*Repeat the above bounds check for each of the remaining axes.*

*For each axis, 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;
Repeat for each of the remaining axes as required before issuing
a move command.
```

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  $\mu\text{m}$ ) 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 v2.51 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

<b>A</b>	
accessories .....	40
<b>C</b>	
cleaning.....	35
<b>configuration</b> .....	43
Configuration .....	15
Configuration switches .....	15
controller	
cable specs .....	41
Controls .....	17
<b>QUAD</b> .....	15
Power switch.....	15
<b>ROE</b> .....	17
Buttons .....	17
Dials .....	17
Switches.....	17
<b>D</b>	
disclaimer .....	3
<b>E</b>	
Electrical Connections .....	13
external control	
microsteps and microns (micrometers) ....	25, 45
external control	
axis position command parameters .....	25
protocol and handshaking.....	24
external control	
task-complete indicator .....	31
external control	
intercommand delay.....	31
external control	
clearing send/receive buffers .....	31
external control	
positions in microsteps and microns.....	31
external control	
ranges and bounds.....	32
external control	
absolute positioning system origin.....	32
external control	
absolute vs. relative positioning .....	32
external control	
position value typing .....	33
external control	
position value bit ordering.....	33
external control	
travel lengths and durations.....	33
external control	
Virtual COM Port (VCP) serial port settings	44
external control	
protocol and handshaking.....	44
external control	
axis position command parameters .....	44
external control	
task-complete indicator .....	47
external control	
intercommand delay .....	47
external control	
clearing send/receive buffers .....	47
external control	
positions in microsteps and microns.....	47
external control	
ranges and bounds.....	47
external control	
absolute positioning system origin.....	48
external control	
absolute vs. relative positioning .....	48
external control	
position value typing .....	48
external control	
position value bit ordering.....	48
external control	
travel lengths and durations.....	48
External control	
Move to specified X-axis position command...	29
Move to specified Y-axis position command...	29
Move to specified Z-axis position command ...	30
notes.....	31
External-control	
virtual COM port (VCP) serial port settings..	24
External-control commands	
notes.....	31
<b>F</b>	
<b>fuses, mains</b> .....	41
fuses, replacement	
mains.....	3
<b>G</b>	
<b>glassware</b>	
<b>precautions</b> .....	4
<b>H</b>	
headstage Exchange .....	22
<b>I</b>	
Initial Operating Instructions .....	15
<b>input</b>	

<b>voltage</b> .....	41	moving to the Home Position.....	19
Installation		moving to the Work Position.....	19
electrical connections .....	13	pausing Home movements .....	21
General .....	11	pausing Work movements .....	21
headstage mounting .....	13	Pulse Mode and diagonal movement.....	21
initial operating instructions.....	13	setting Absolute/Relative coordinates mode	
Initial Operating Instructions .....	15	.....	20
other accessories .....	13	setting position for HOME and WORK.....	18
QUAD/M mounting		setting the D-axis headstage angle.....	21
to a stand or platform.....	11	display .....	17
QUAD/M mounting.....	11	initial startup .....	17
ROE/Controller rear panel controls and		ROE controls & indicators.....	17
configuration .....	15		
configuration switches .....	15	<b>R</b>	
locking out Y-movement homing.....	15	Reconfiguration .....	35
reserved for future use .....	16	changing handedness .....	35
sensor test.....	16	changing rotary knob functions on the ROE.	35
setting axis directionality.....	15	Installing the Z-Axis Extension Plate.....	36
power switch .....	15	moving Control Cable Connector from vertical	
Installation .....	11	feed to horizontal .....	36
Introduction .....	9	moving location of Control Cable Bracket ....	36
Components .....	9	Remote Control Commands	
Overview.....	10	Get Current Position Command.....	26
Description .....	10	Go to Position Command.....	27, 28
Features.....	10	Move to Specified D-Axis Position Command	30
<b>L</b>		Set Velocity for all Externally Controlled	
Locking out Y-movement homing.....	15	Movements ('v') (Firmware Ver. 2.51+) ....	31
<b>M</b>		<b>S</b>	
mains		safety warnings	
fuses .....	3, 41	mains fuse.....	3
<b>voltage</b> .....	41	<b>safety warnings &amp; precautions</b>	
Maintenance.....	35	<b>operational</b> .....	4
<b>manual operation</b> .....	43	<b>SAFETY WARNINGS &amp; PRECAUTIONS</b> .....	3
Micropipette Exchange .....	22	<b>electrical</b> .....	3
Mounting		Sensor test.....	16
headstage.....	13	Setting axis directionality .....	15
QUAD/M to a Stand or Platform .....	11	Setting the D-axis headstage angle.....	21
<b>N</b>		Special installations	
notes		Right and left handedness .....	35
user.....	53, 54	<b>T</b>	
<b>O</b>		technical specifications.....	41
Operations.....	17	<b>dimensions</b> .....	42
control operations.....	18	<b>drift</b> .....	41
headstage exchange .....	22	<b>electrical</b> .....	41
maximum positive position values .....	18	<b>cables</b> .....	41
micropipette exchange .....	22	<b>mains fuses</b> .....	41
mode indications.....	20	<b>power adapter</b> .....	41
movement knobs disabling and Quiet Mode		<b>input (mains)</b> .....	41
.....	21	<b>output (to controller)</b> .....	41
		<b>system power consumption</b> .....	41
		<b>resolution</b> .....	41

speed .....	41	input.....	41
travel.....	41	mains.....	41
weight .....	42	<i>W</i>	
<i>V</i>		warranty.....	39
voltage			

## NOTES

**NOTES**