

XenoWorks® XWM-100

Motorized Micromanipulator System

Operation Manual

Rev. 1.03h (20230629.01)

SUTTER INSTRUMENT

One Digital Drive
Novato, CA 94949

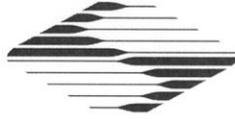
Voice: 415-883-0128 Web: www.sutter.com
Fax: 415-883-0572 Email: info@sutter.com





Copyright © 2023 Sutter Instrument Company. All Rights Reserved.

XenoWorks® is a registered trademark and TRIO™ is a 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: XenoWorks XWM-100 Micromanipulator System

Model(s): XWM-100 (system): Controller/joystick and XWML/M or XWMR/M (micromanipulator mechanical).

Conforms to Standards: EMC Emissions: EN 61326-1:2013, including:
EN 55011: 2009 Class B;
EN 61000-3-2:2015, & EN 61000-3-3:2014
EMC Immunity: EN 61000-4-2:2009, EN 61000-4-3:2011,
EN 61000-4-4:2012, EN 61000-4-5:2014,
EN 61000-4-6:2014, EN 61000-4-8:2010, &
EN 61000-4-11:2004
LVD (Safety): EN 61010-1:2010

Tested/Verified (as a QUAD system on which the XenoWorks XWM-100 is based and part on MPC-200 Series systems) by: ITC Engineering Services, Inc
9959 Calaveras Road, PO Box 543
Sunol, CA 94586-0543 USA
Tel. +1 925 862 2944 Fax: +1 925 862 9013
Email: itcemc@itcemc.com Web: www.itcemc.com
Sutter Instrument

Test Report(s): 20091104-03-CE-Rev2; 20150212-02A;
SI_EMC_XenoWorks_BRM_20160713; 20140120-01R1; 20140120-01;
SI_EMC_QUAD_20160713; 20120511-01A-R1_MC CE Rpt R4;
SI_EMC_MPC-200_20160713;
20200520_REACH and RoHS Compliance Statement

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

Project Engineer:


Mark Flaming
President

SUTTER INSTRUMENT®

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

DISCLAIMER

The **XenoWorks XWM-100** Micromanipulator is designed for the specific use of moving micropipettes in three-dimensional space in increments of micrometers into and out of an optical pathway of a microscope and no other use is recommended.

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

Do not open or attempt to repair the instrument.

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

Any misuse will be the sole responsibility of the user/owner. 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 **XenoWorks** using 110-120 V AC, 60 Hz, or 220-240 V AC., 50 Hz line voltage. This instrument is designed for connection to a standard laboratory power outlet (Overvoltage Category II), and because it is a microprocessor--controlled device, it should be accorded the same system wiring precautions as any 'computer type' system. A surge protector and power regulator are recommended.

Avoiding Electrical Shock and Fire-related Injury

-  Always use the grounded power supply cord set provided to connect the system to a grounded outlet (3-prong). This is required to protect you from injury in the event 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 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 with the unit for the interconnection of the joystick with the manipulator mechanical (see the **TECHNICAL SPECIFICATIONS** appendix 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.

- This unit is not designed for operation at altitudes above 2000 meters nor was it tested for safety above 2000 meters.
- Before making electrical connections, ensure that the instrument is switched off.
-  **DO NOT CONNECT OR DISCONNECT THE CABLES BETWEEN THE JOYSTICK 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 this 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.
-  Operate only in a location where there is a free flow of fresh air on all sides. **NEVER ALLOW THE FREE FLOW OF AIR TO BE RESTRICTED.**
-  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 attempt to operate the instrument with the manipulator shipping screws in place - severe motor damage may result.
-  Do not operate if there is any obvious damage to any part of the instrument. When transporting the mechanical manipulator, be sure to install the shipping screws supplied in their correct 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).
-  As with all microinjection 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 microinjectors.
-  Do not handle the manipulator mechanical while the power is on and take care to ensure no cables pass close to the mechanical manipulator.
-  Sutter Instrument recommends that the operator touch the microscope equipment base (to discharge any static electricity) prior to using the XenoWorks BRM Micromanipulator system.

Other

- Use this instrument only for microinjection purposes in conjunction with the procedures and guidelines in this manual.

- Retain the original packaging for future transport of the instrument.
- Some applications, such as piezo-impact microinjection call for the use of mercury in the micropipette tip. The use of any hazardous materials with any Sutter Instrument's instrument is not recommended and, if undertaken, is done so at the users' own risk.
- This instrument contains no user-serviceable components — do not open the instrument casing. This instrument should be serviced and repaired only by Sutter Instrument or an authorized Sutter Instrument servicing agent.
- Sutter Instrument reserves the right to change specifications without prior notice.
- This device is intended only for research purposes.
- Do not carry the joystick by its joystick paddle because damage may result.

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. Contact with the pulled micropipette tips, therefore, should be avoided to prevent accidentally impaling yourself.
- Always dispose of micropipettes by placing them into a well-marked, spill-proof “sharps” container.

(This page intentionally left blank.)

TABLE OF CONTENTS

DISCLAIMER	iii
SAFETY WARNINGS AND PRECAUTIONS	iii
Electrical	iii
Avoiding Electrical Shock and Fire-related Injury.....	iii
Electromagnetic Interference	iii
Operational	iii
Other.....	iv
Handling Micropipettes.....	v
1. GENERAL INFORMATION	13
1.1 Introduction.....	13
1.2 About This Manual.....	13
1.3 Technical Support.....	13
1.4 Product Description.....	13
1.4.1 Packing List	13
1.4.2 Joystick Components	14
1.4.2.1 Status Display	14
1.4.2.2 Drag Adjustment Ring.....	15
1.4.2.3 Height Adjustment.....	15
1.1.1 Components of the Manipulator Mechanical.....	16
1.1.1.1 Micropipette Holder Clamp.....	16
1.1.1.2 Swing Gate	16
1.1.2 Joystick Controls.....	17
1.1.2.1 Function Buttons.....	17
1.1.3 Joystick Features	18
1.1.3.1 Controlling Horizontal Movements (X- and Y-Axes).....	18
1.1.3.2 Controlling Vertical Movements (Z-Axis)	18
1.1.3.3 Controlling Diagonal Movements (X- and Z-Axes).....	18
1.1.3.4 Speed and Range of Movement.....	19
1.1.3.5 Y-Axis Lock/Unlock	19
1.1.3.6 X-Axis Lock/Unlock.....	19
1.1.3.7 ‘Origin’ Position.....	19
1.1.3.8 ‘Work’ Positions.....	19
1.1.3.9 ‘Home’ Position	19
1.1.3.10 ‘Z-floor’	20
1.1.3.11 ‘Pulse’	20
1.1.3.12 ‘X-Limit’	20
1.1.3.13 ‘User Profiles’	20
1.1.3.14 Joystick Declutch.....	20
1.1.1.1 Configuration DIP Switches.....	21
2. INSTALLATION	23
2.1 Installing the Manipulator Mechanical.....	23
2.2 Setting the Configuration DIP Switches.....	24
2.3 Installing the Joystick Controller	25

3. OPERATING INSTRUCTIONS	27
3.1 General.....	27
Status Display	27
3.1.1 Moving the Micropipette with the Joystick.....	28
3.1.2 Centering the Micropipette with Respect to the Joystick.....	28
3.1.3 Setting the Angle of Approach of the Micropipette.....	28
3.1.3.1 Microinjection Chamber	29
3.1.3.2 Applications and Suggested Setups.....	29
3.1.3.3 Micropipette Type	31
3.1.3.4 Working Distance	31
3.1.3.5 Piezo Impact Drive	31
3.2 Controller Functions	31
3.2.1 Using the ‘Work’ Function.....	31
3.2.2 Using the ‘Home’ Function.....	31
3.2.3 Using the ‘Pulse’ Function.....	32
3.2.3 Changing the Working Range of Movement.....	32
3.2.4 Constraining Pipette Movements.....	32
3.2.4.1 Using the ‘Y-Lock’ and ‘X-Lock’ functions.....	32
3.2.4.2 Setting a ‘Z-Floor’	32
3.2.4.3 Setting an ‘X-Limit’	33
3.3 Common Uses of Controller Functions	33
3.3.1 Cleaning the Micropipette.....	33
3.3.2 Exchanging the Micropipette.....	34
3.4 Starting a Microinjection Session.....	34
3.4.1 Setting up the Microscope.....	34
3.4.2 Setting up the Micromanipulator.....	34
4. EXTERNAL CONTROL	37
4.1 General.....	37
4.2 Virtual COM Port (VCP) Serial Port Settings.....	37
4.3 Protocol and Handshaking	37
4.4 Command Sequence Formatting	37
4.5 Axis Position Command Parameters.....	38
4.6 Micro-steps and Microns (Micrometers)	38
4.7 Range Speed	39
4.8 Travel Speed.....	39
4.9 Commands	39
4.9.1 Get Name and Firmware Version (‘K’) Command	39
4.9.2 Get Current Position (‘C’) Command.....	41
4.9.3 Toggle XY Mode (Diagonal/Orthogonal) (‘D’) Command.....	42
4.9.4 Setting Diagonal Approach Angle (‘A’) Command.....	43
4.9.5 Perform XZ Pulse (‘P’) Command	43
4.9.6 Move to Specified Position at Full Speed (‘M’) Command	43
4.9.7 Move to Origin (‘O’) Command.....	44
4.9.8 Move to Home Position (‘H’) Command	44
4.9.9 Move to Work Position (‘Y’) Command.....	44
4.9.10 Move to Center Position (‘N’) Command.....	44
4.9.11 Get Approach Angle (‘a’) Command.....	45

4.9.12 Report Internal Resolution ('R') Command.....	45
4.9.16 Move to Specified Position at Selected Speed ('m') Command.....	45
4.9.17 Interrupt Move ('^C') Command.....	46
4.9.18 Command Notes.....	46
5. MAINTENANCE.....	51
5.1 General.....	51
5.2 Troubleshooting.....	51
5.3 Transporting the Micromanipulator	54
APPENDIX A. LIMITED WARRANTY.....	57
APPENDIX B. ACCESSORIES.....	59
Extensions.....	59
Rotating Base.....	59
Mounts	59
Adapters	59
Holders	59
APPENDIX C. TECHNICAL SPECIFICATIONS.....	61
APPENDIX D. QUICK REFERENCE.....	63
Manual Operation.....	63
Configuration	63
Status Display	65
External Control	66
INDEX.....	73

TABLE OF FIGURES

Figure 1-1 Joystick front panel controls.	14
Figure 1-2 Joystick rear panel controls.....	15
Figure 1.3 Height adjustment clamp.....	16
Figure 1-4. Manipulator Mechanical with Swing Gate. (MP-285/M shown).....	16
Figure 1.5 Function buttons (right side shown) and their primary functions.	17
Figure 1.6 Function buttons and their primary functions.....	17
Figure 1-7 Configuration DIP Switches.....	21
Figure 2-1. Mounting the right-handed manipulator to the far-left corner of the adapter plate. ...	23
Figure 2-2. Engaging the manipulator base plate in the microscope adaptor.	24
Figure 3-1. Reporting of the controller's operating modes.....	27
Figure 3-2. Status Display when either the X- or Y-axes are locked at their current positions....	27

Figure 3-3. Status Display when the Diagonal (XZ) Mode is engaged for a right-side manipulator. 27

Figure 3-4. Z-Floor function is engaged, with the manipulator at its lower Z-limit (left) and above the Z-limit (right). 27

Figure 3-5.. Status Display for a right-side manipulator when the X-Limit function is engaged, with the manipulator at its X-Limit (left) and away from the X-limit (right). 27

Figure 3-6. Status Display showing the manipulator’s current position. 28

Figure 3-7. Micropipette alignment controls. 29

Figure 3-8 (a and b). Suggested approach for **suspended cells with angled micropipettes** (ICSI, Stem Cell, Nuclear Transfer). 29

Figure 3-9 (a and b). Suggested approach for **suspended cells with straight micropipettes** (Pronuclear Injection). 30

Figure 3-10 (a and b). Suggested approach for **adherent cells ransd straight pipettes**. 30

Figure D-1. Rear of the controller..... 63

Figure D-2. Function buttons (right side shown) and their primary functions. 64

Figure D-3. Button Functions 64

Figure D-4. Reporting of the controller’s operating modes..... 65

Figure D-5. Status Display when either the X- or Y-axes are locked at their current positions... 65

Figure D-6. Status Display when the Diagonal (XZ) Mode is engaged for a right-side manipulator. 65

Figure D-7. Z-Floor function is engaged, with the manipulator at its lower Z-limit (left) and above Z-limit (right). 65

Figure D-8. Status Display for a right-side manipulator when the X-Limit function is engaged, with the manipulator at its X-Limit (left) and away from the X-limit (right). 65

Figure D-9. Status Display showing the manipulator’s current position..... 65

TABLE OF TABLES

Table 1-1. Function Button Commands. See Operating Instructions for details..... 18

Table 1-2. Configuration DIP Switches. See Installation Instructions for details. 22

Table 4-1. USB-VCP interface serial port settings. 37

Table 4-2. Microns/micro-steps conversion..... 38

Table 4-3. Ranges and bounds..... 39

Table 4-4. Range (speed) settings. 39

Table 4-5. Travel speeds..... 39

Table 4-6. Get name and firmware version ('K') command (Firmware <2).	40
Table 4-7. Get name and firmware version ('K') command (Firmware ≥2)	41
Table 4-8. Get current position ('C') command (Firmware <2).	42
Table 4-9. Get current position ('C') command (Firmware ≥2).	42
Table 4-10. Toggle XY Mode (Diagonal/Orthogonal) ('D') Command.	42
Table 4-11. Set the angle ('A') command (Firmware <2).	43
Table 4-12. Set the angle ('A') command (Firmware ≥2).	43
Table 4-13. Perform XZ pulse ('P') command.	43
Table 4-14. Move to specified position at full speed ('M') command.	43
Table 4-15. Move to origin ('O') command.	44
Table 4-16. Move to controller-defined HOME position ('H') command.	44
Table 4-17. Move to controller -defined WORK position ('Y') command.	44
Table 4-18. Move to center position ('N') command.	45
Table 4-19. Get approach angle ('a') command.	45
Table 4-20. Report internal resolution ('R') command.	45
Table 4-24. Move to specified position at selected speed ('m) command.	46
Table 4-25. Interrupt move (' ^ C') command.	46
Table C-1. Range speeds.	61
Table C-2. XenoWorks XWM-100 cables and receptacles/connectors.	62
Table D-3. Configuration Switches 1 – 8.	63
Table D-4. USB-VCP interface serial port settings.	66
Table D-5. Microns/micro-steps conversion.	67
Table D-6. Ranges and bounds.	67
Table D-7. Range (speed) settings.	67
Table D-8. Travel speeds.	67
Table D-9. XenoWorks XWM-100 external-control commands.	68

(This page intentionally blank.)

1. GENERAL INFORMATION

1.1 Introduction

1.2 About This Manual

The XenoWorks XWM-100 is a manipulator system comprised of two basic parts: the joystick/controller (XWM-100), and the manipulator mechanical (XWM/M or MP-845/M).

In the next, few pages you will find a product description to help you become acquainted with operation, followed by installation instructions, and then detailed operating instructions. Please take the time to read these instructions to assure the safe and proper use of this instrument.

This manual is continually being updated. If you encounter any part of the manual that you feel should be covered in expanded detail, we would like to hear from you. Please contact our Technical Support staff with your suggestions (415-883-0128 or info@sutter.com).

1.3 Technical Support

Unlimited technical support is provided by Sutter Instrument Company at no charge to our customers. Our technical support staff is available between the hours of 8:00 AM and 5:00 PM (Pacific Time) at +1 (415) 883-0128. You may also E-mail your queries to info@sutter.com.

1.4 Product Description

NOTE: Unless otherwise noted, all references to the manipulator mechanical are for the right-handed version.

1.4.1 Packing List

The XenoWorks™ XWM-100 Micromanipulator is shipped with the following components:

1. Manipulator mechanical
2. Joystick / Controller
3. Manipulator mechanical base plate
4. AC / DC power supply
5. Mains power cord
6. Cable for connecting the manipulator mechanical with the joystick / controller
7. USB cable
8. Instruction manual
9. Manipulator mechanical mounting screws (4)
10. Hex key (for manipulator mechanical mounting screws)

If any items are missing or appear damaged, contact Sutter Instrument or your dealer/rep immediately.

1.4.2 Joystick Components

The user interface of the XenoWorks™ Micromanipulator is a three-axis joystick. When the micromanipulator is installed correctly, the joystick oriented appropriately on the working surface, and the default axis polarity settings used (see “Axis Polarity”), a displacement of the handle to the left, for example, will result in concomitant movement of the micromanipulator to the left. The movement of the micromanipulator is directly proportional to the movement of the joystick. However, the movement reduction ratio between the joystick and the mechanical can be varied (see “Range” below).

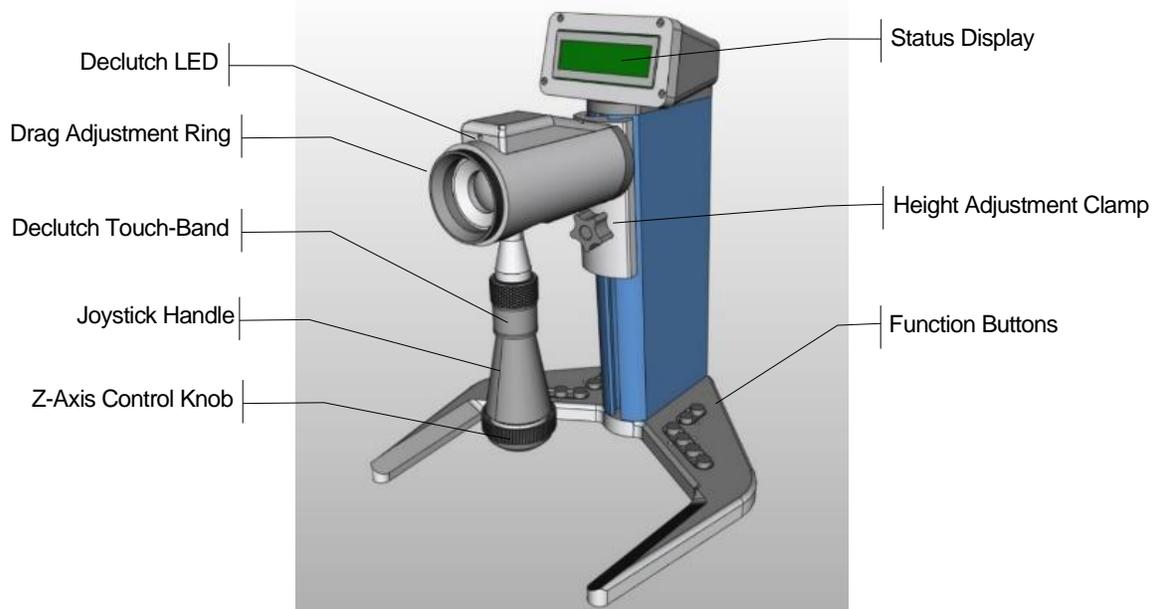


Figure 1-1 Joystick front panel controls.

1.4.2.1 Status Display

The Status Display at the top of the XWM-100 controller informs the user of the operation and state of the joystick. The Status Display operates in three modes. In its standard mode, the Status Display reports the operational state (ex. locked, unlocked, diagonal vs. orthogonal, range limits, etc.) of each axis of the manipulator. In its alternative mode, the display reports the position of the micromanipulator (in microns) relative to the Origin (see below). During other operations, the display prompts users for specific actions.

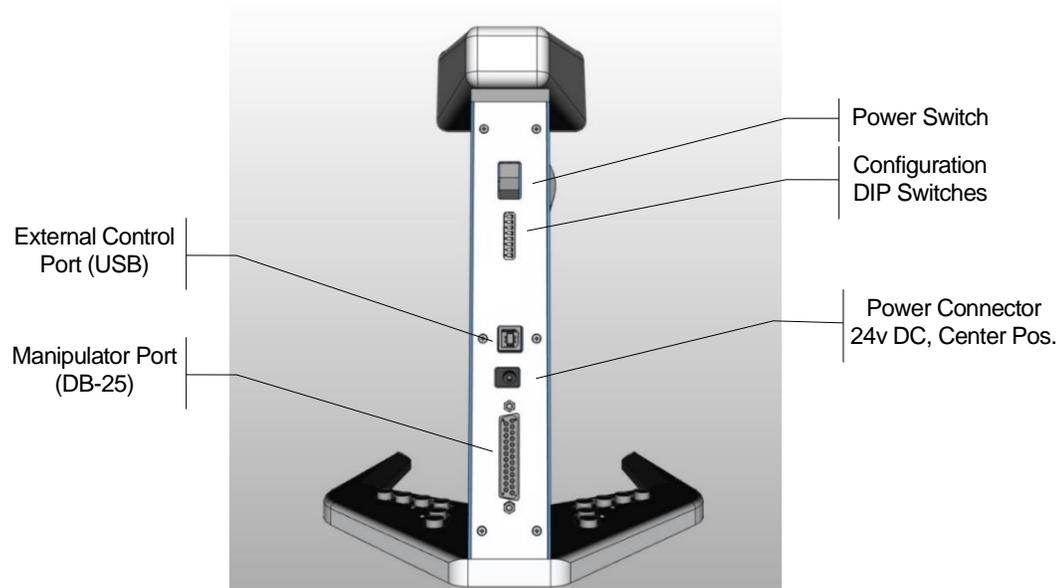


Figure 1-2 Joystick rear panel controls.

1.4.2.2 Drag Adjustment Ring

The adjustment ring at the front of the joystick permits users to adjust the joystick's 'drag' (its resistance to horizontal movements). Gently tightening the ring increases drag, while loosening the ring reduces drag. This adjustment allows users to match the joystick's mechanical drag to their needs: perhaps free movement (very little drag) while coarsely positioning a pipette, then moderate drag while finely adjusting the pipette's position, or even strong drag to hold (or 'park') the pipette at a fixed position.

Although there is no single recommended setting for this adjustment, users should note that if there were no friction at all, the joystick would always fall to its center position under the effects of gravity, which may not be desirable. Similarly, if the friction were set too tight, the joystick would become too stiff to use or may be locked in place and could be damaged if the dial is tightened further. With a little practice, users generally find a suitable range of tensions.

1.4.2.3 Height Adjustment

To maintain good ergonomics and operator comfort, the height of the joystick can be adjusted, as shown in the figure below. The bottom of the joystick handle can be placed anywhere from zero to approx. 45 mm above the desktop. The adjustment is made by carefully loosening the black Height Adjustment Clamp located on the front of the joystick column. Gently move the joystick assembly up and down along its vertical track until its position is comfortable, and then lock that position by re-tightening the Height Adjustment Clamp.

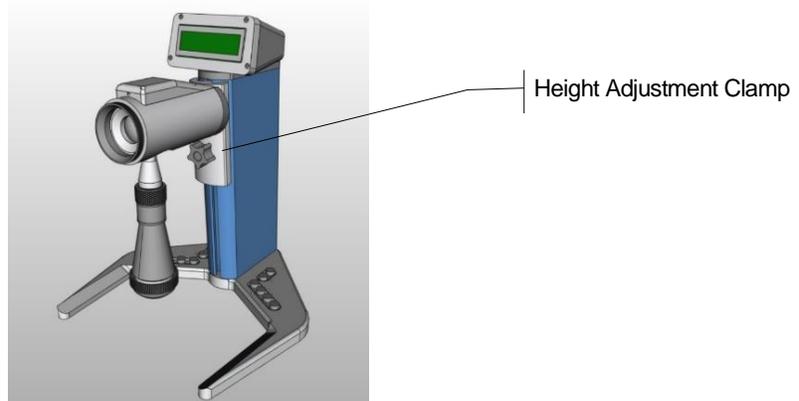


Figure 1.3 Height adjustment clamp.

1.1.1 Components of the Manipulator Mechanical

1.1.1.1 Micropipette Holder Clamp

The micropipette holder clamp on the front of the micromanipulator is designed to allow many different micropipette types to be used with the maximum stability possible. The clamp can be adjusted for height, reach and approach angle. See Operating Instructions for details on setting the micropipette holder clamp for the optimum angle of attack of the micropipette.

1.1.1.2 Swing Gate

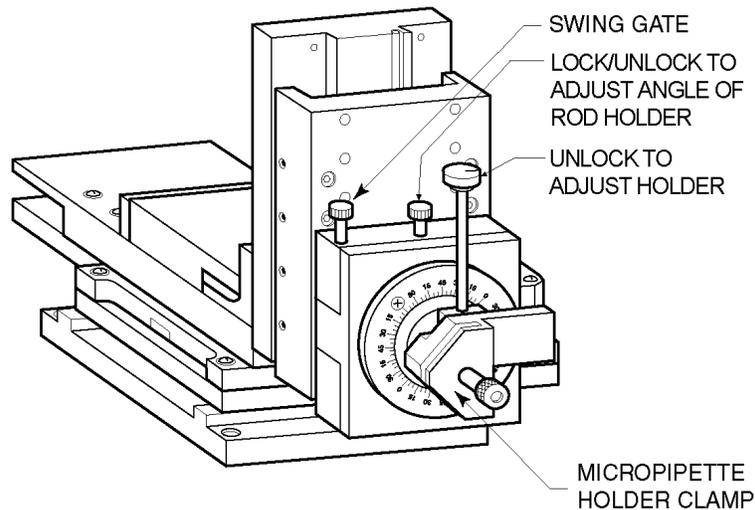


Figure 1-4. Manipulator Mechanical with Swing Gate. (MP-285/M shown)

The swing gate on the front of the mechanical allows the micropipette to be swung out toward the front of the microscope for better access when changing micropipettes. The gate can be released by loosening the fixing screw. The screw should be tightened fully during use.

1.1.2 Joystick Controls

1.1.2.1 Function Buttons

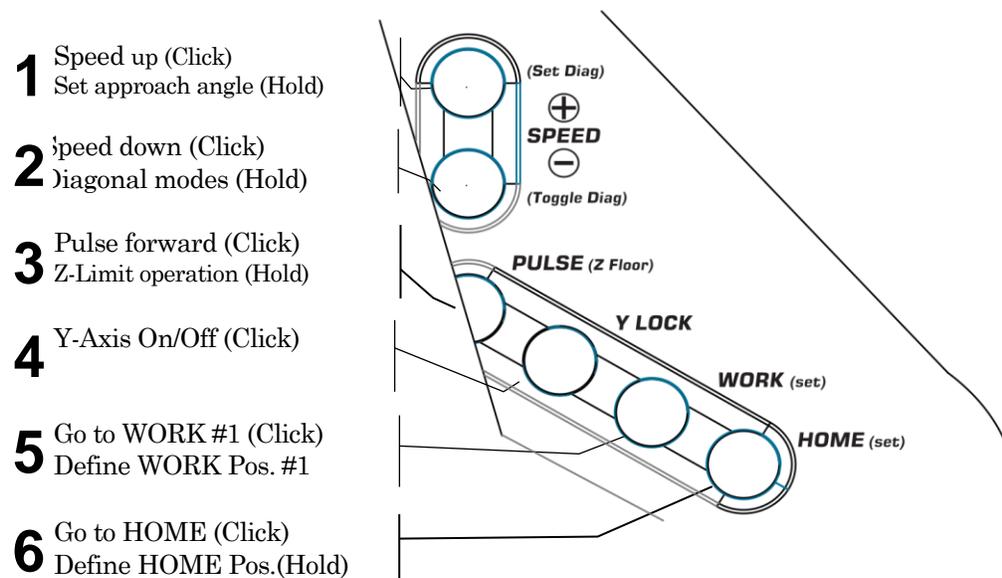


Figure 1.5 Function buttons (right side shown) and their primary functions.

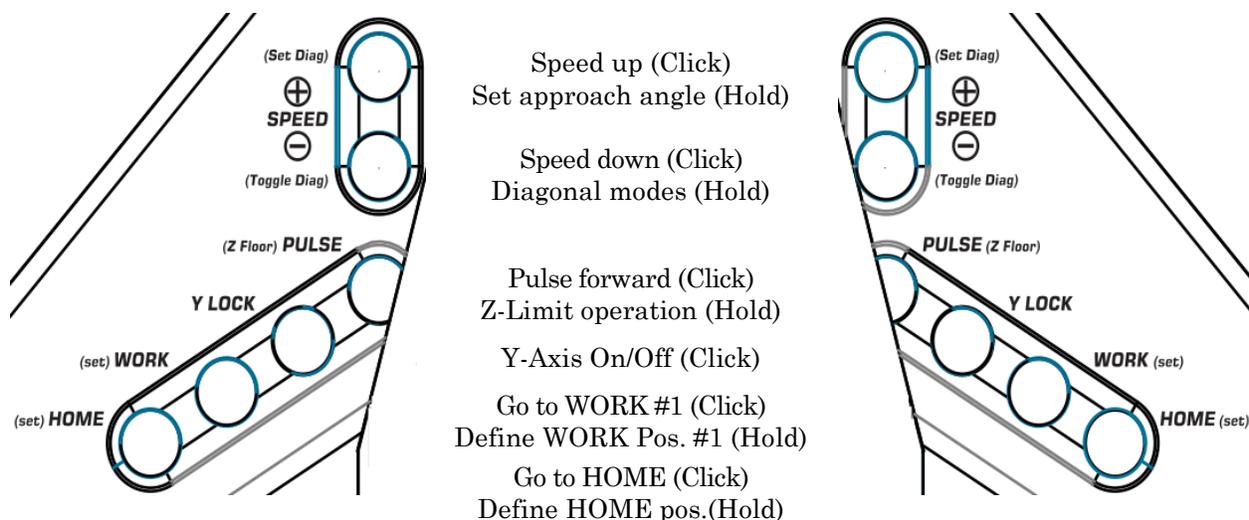


Figure 1.6 Function buttons and their primary functions.

The figure above shows the function buttons on the right side of the baseplate. The buttons on the left side of the baseplate are numbered similarly. Corresponding buttons on the two sides of the baseplate function identically. Most buttons serve at least two primary functions: one function when clicked (pressed briefly and released), and a second function when held (pressed continuously until a beep sounds, about 2 seconds). The primary click functions are labeled in bold on the baseplate; the hold functions are labeled in parentheses.

Several buttons also provide secondary functions when clicked or held while touching the Declutch Touch-Band. In this way, the Declutch Touch-Band can serve like the shift key on a computer keyboard and gives the user extra control over the XWM-100. Activating the secondary functions is described in these notes as ‘Touch-Clicking’ or ‘Touch-Holding’.

The table below lists the primary and secondary functions of each button in the XWM-100 controller.

Table 1-1. Function Button Commands. See Operating Instructions for details.

Button	Primary Functions		Secondary Functions	
	Click (Tap)	Hold (3 sec.)	Touch-Click (Touch-Tap)	Touch-Hold (3 sec.)
1	Speed up	Set approach angle	Display mode	Choose User Profile
2	Speed down	Diagonal mode		
3	Pulse forward	Z-Limit operation		X-Limit operation
4	Y-Axis On/Off		X-Axis EOT/Limit	
5	Go to WORK pos. #1	Define WORK pos. #1	Go to WORK pos. #2	Define WORK pos. #2
6	Go to HOME pos.	Define HOME pos.	Go to Origin	Go to Setup

1.1.3 Joystick Features

1.1.3.1 Controlling Horizontal Movements (X- and Y-Axes)

The micromanipulator is moved horizontally along its X- and Y-axes, by moving the handle left and right, and forward and backward, respectively. The joystick is configured for either right- or left-handed manipulator mechanicals by setting the DIP switches on the joystick’s rear-panel.

1.1.3.2 Controlling Vertical Movements (Z-Axis)

The micromanipulator is moved vertically along its Z-axis, by rotating the control knob on the bottom of the joystick handle. Looking down on the joystick from above and using the default axis settings, rotating the knob in a clockwise direction will move the micromanipulator downward. Rotating the knob counterclockwise will move the micromanipulator upward.

1.1.3.3 Controlling Diagonal Movements (X- and Z-Axes)

The XWM-100 controller can command the micromanipulator mechanical such that the X- and Z-axes are moved in step with each other, creating a diagonal and axial motion of the pipette tip. Such axial movements can reduce cellular damage and injury during injections. The stepping of the X- and Z-axes is performed with sub-micron precision, providing a nearly perfect axial movement of the pipette tip. By defining the Diagonal Approach Angle, the user

can set the angle at which the X- and Z-axes move when Diagonal mode is engaged. When the user matches the manipulator's diagonal movements to the physical angle of the injection pipette, Diagonal Mode moves the pipette tip in a nearly perfect axial direction.

1.1.3.4 Speed and Range of Movement

The speed and range of the micromanipulator's movements can be adjusted to accommodate different microscope magnifications and the differing needs of operators and applications. There are eight different settings, each with unique sensitivities of micromanipulator movement. Speed settings C ('Coarse'), 1, and 2 are generally good for moving the micropipette into the field of view. Speeds 3, 4, and 5 are ideal for micromanipulation and microinjection. Speeds 6 and 7 are for ultra-fine movements. Note that lower settings (e.g., C, 1, and 2) drive the manipulator mechanical at *higher* speeds, while higher settings (e.g., 5, 6, and 7) drive the manipulator mechanical at *lower* speeds.

1.1.3.5 Y-Axis Lock/Unlock

Movements along the Y-axis can be disabled, locking the manipulator's position along the Y-axis, and limiting horizontal movements to only the X-axis. Such movements may be preferred for difficult injections/transfers, or when the cell or egg is more sensitive to the withdrawal of the pipette.

1.1.3.6 X-Axis Lock/Unlock

Movements along the X-axis can be disabled similarly, locking the manipulator's position along the X-axis, and limiting horizontal movements to only the Y-axis.

1.1.3.7 'Origin' Position

Moving the micromanipulator to its 'Origin' position moves the mechanical as close as possible to the user and to its outermost and uppermost position with respect to the preparation. The 'Origin' position of a right-hand micromanipulator will be as high and as far to the right as possible, while the 'Origin' position of a left-hand micromanipulator will be as high and as far to the left as possible. For this reason, a right-hand micromanipulator should not be used on the left side of the microscope and vice-versa. The 'Origin' position is factory preset and cannot be re-configured.

1.1.3.8 'Work' Positions

The user's defined 'Work' positions can be assigned to any point within the range of travel of the micromanipulator. The 'Work' positions are customarily used to quickly place the micropipette tip near its ultimate working position. It is recommended that the 'Work' positions be refreshed at regular intervals during the course of an injection experiment. The micropipette can be returned to the memorized 'Work' position by clicking the 'Work' button. The user sets the Work position by pressing the button and holding for 3 seconds.

1.1.3.9 'Home' Position

The users' 'Home' position can be set to any position within the range of travel of the micromanipulator. The 'Home' position is used primarily to position the micropipette tip at a safe position, often near the edge of the microscope's field of view permitting the user to place the other components needed for the injection (ex. chambers, cells, etc.). The micropipette can be returned to the memorized 'Home' position by clicking the 'Home' button. The user sets the Work position by pressing the button and holding for 3 seconds.

1.1.3.10 'Z-floor'

The 'Z-floor' allows the user to define a lower limit (or "floor") of the Z-axis (vertical) travel of the micromanipulator. In practice, it is used to protect a micropipette from being broken by preventing the user from 'crashing' the micropipette into the bottom of the microinjection chamber.

1.1.3.11 'Pulse'

The 'Pulse' function rapidly advances the pipette tip diagonally 3 micrometers (microns), moving along the X- and Z-axes. The 'Pulse' function is used primarily to advance the micropipette tip rapidly and axially through cellular membranes or barriers. The 'Pulse' function is activated by clicking the 'Pulse' button.

1.1.3.12 'X-Limit'

The 'X-Limit' feature allows the user to set a maximum horizontal excursion along the X-axis, beyond which the manipulator will not advance. When engaged, 'X-Limit' mode prevents the manipulator mechanical from moving along the X-axis beyond the limiting position. 'X-Limit' mode can be used to prevent the injection pipette from advancing further than the desired limit, preventing the pipette tip from running through ('skewering') cells being injected. The 'X-Limit' may also help speed injections of multiple cells aligned along the Y-axis.

1.1.3.13 'User Profiles'

The Xenoworks Joystick Controller provides many programmable features (ex. Home and Work positions, approach angle, X- and Y-axis locks, display mode, etc.) all designed to make its use more convenient. It is very easy to imagine a single Joystick Controller being shared among many users or scientific applications, each of which have their own preferred settings for these programmable features. Beginning with firmware v.2.06, the Joystick Controller allows users to define, store, reload, and modify up to 100 different sets of these programmable features.

Users select the desired profile by holding a finger against the Declutch Touch-Band, then pressing and holding Function Button 1. After the controller beeps, the display will prompt the user to select their desired profile by rotating the Z-Axis knob at the bottom of the Joystick Handle. Once the ID number for the desired profile is shown on the display, the user should stop rotating the Z-Axis knob. After a brief pause, the joystick controller will load the chosen profile from its memory. Subsequent changes to the profile's programmable features are automatically stored to the chosen profile.

1.1.3.14 Joystick Declutch

The electronic coupling between the joystick and the manipulator mechanical can be temporarily interrupted, allowing users to move the joystick handle without changing the position of the micropipette. All movements of the manipulator mechanical are blocked while the dedicated function is engaged.

This 'declutch' function is activated by touching the Declutch Touch-Band, just above the joystick's rubber handgrip. The red Declutch LED lights whenever the Declutch function is activated. The declutch function is deactivated, and the coupling between joystick and manipulator is reestablished, by releasing the Touch-Band.

Declutching is especially helpful when it's necessary to move the pipette long distances. When the handle reaches the limit of its swing, the user touches the declutch band (isolating the manipulator from movements of the handle), swings the handle to the opposite end of its range, then releases the declutch band and continues moving the manipulator toward the destination. With practice, this 'ratcheting' operation becomes easy, natural, and quick.

1.1.1.1 Configuration DIP Switches

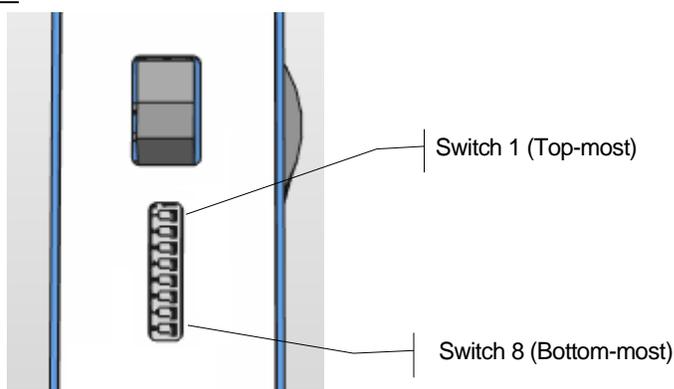


Figure 1-7 Configuration DIP Switches.

The figure above shows the Configuration DIP Switches as they appear in their 'Switch Left' positions on the rear panel of the XWM-100 controller. The following table summarizes the function of each Configuration Switch. The Configuration Switches are typically set when the XenoWorks system is installed and are changed only rarely afterwards. Please refer to the Installation Instructions for more details on setting the Configuration DIP Switches.

Table 1-2. Configuration DIP Switches. See Installation Instructions for details.

Switch #	Definition	Left (ON)	Right (OFF)
1	X-Axis Polarity	Manipulator on <u>right</u> of microscope	Manipulator on <u>left</u> of microscope
2	Y-Axis Polarity	Forward motion moves <u>towards</u> user (normal)	Forward motion moves <u>away from</u> user
3	Z-Axis Polarity	Clockwise rotations <u>lowers</u> manipulator	Clockwise rotations <u>raises</u> manipulator
4	Electromechanical Device Compatibility	XWM/M or MP-285/M 8 micro-steps/micron)	TRIO MP-845/M 10.6667 micro-steps/micron)
5	Power-ON Auto calibration	Auto calibration OFF	Auto calibration ON
6	Pulse Mode	Ramped-Pulse (<u>slower</u> ; minimizes vibration)	Standard-Pulse (<u>faster</u> ; slightly more vibration)
7	<reserved for future use>		
8	Touch-Pulse enable	Enable	Disable

2. INSTALLATION

2.1 Installing the Manipulator Mechanical

The directions below describe the installation procedure common to both left- and right-handed micromanipulators. Although the figures illustrate a right-hand micromanipulator, a left-hand micromanipulator would be illustrated as a mirror image of these figures. It is extremely important that these installation instructions are followed closely.

1. Unpack and assemble the microscope adapter and attach it firmly to the microscope according to the instructions included with the adapter.
2. Unpack the mechanical manipulator adapter base plate. The Y-axis slide on the bottom of the adapter plate can be adjusted so that the mechanical is positioned correctly with respect to the microscope's optical axis. Each model microscope will require special Y-axis slide settings. To determine the appropriate setting for your microscope, refer to the lookup table contained in the instruction sheet supplied with the microscope adapter.
3. Align the back edge of the slide in the numbered channel with the number in the instructions that corresponds with the microscope being used. Lock the Y-axis slide screws firmly with the hex (Allen) key provided.

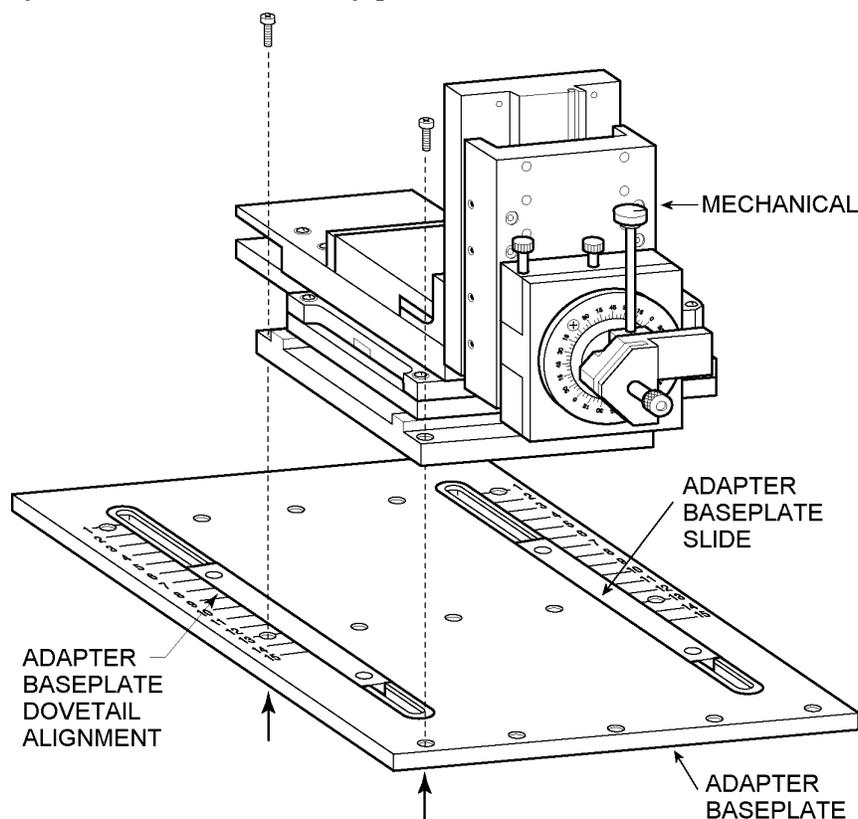


Figure 2-1. Mounting the right-handed manipulator to the far-left corner of the adapter plate.

4. Unpack the manipulator mechanical. Take extra care while handling the mechanical; before it is mounted on the microscope adapter, it is vulnerable to impacts which can damage the stepper motors.
5. Remove the six (6) shipping screws (indicated by the red stripes)
IMPORTANT: Retain the six (6) shipping screws and keep them with the instrument packaging if the instrument needs to be shipped to another location.
6. Carefully place the manipulator on the base plate so that the micropipette holder clamp faces toward the front of the microscope, as shown above for a right-hand

micromanipulator. The two innermost screw holes (closest to the microscope – arrowed) on the base of the mechanical should be visible. Attach the manipulator to the adapter plate in the position shown, using the screws and tool provided. Do not use shipping screws for this purpose - they are too long.

- Slide the dovetail on the bottom of the adapter plate into the slide of the microscope adapter (it may be necessary to loosen the locknut on the rear of the microscope adapter). Do not tighten the locknut at this time.

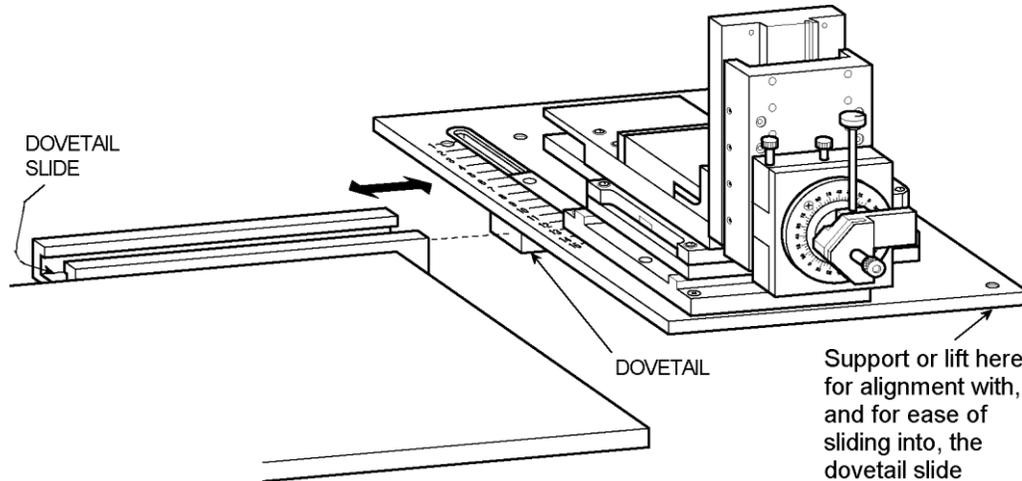


Figure 2-2. Engaging the manipulator base plate in the microscope adaptor.

2.2 Setting the Configuration DIP Switches

- Set the Movement Polarity for each Axis** – Occasionally, it may be helpful to reverse the direction of one, two, or all three motors so movements of the joystick yield opposing movements in the manipulator mechanical. This might be needed for example, when using a microscope or camera where the images are inverted up-down or right-left.

The polarity of movement along each axis is controlled by DIP switches 1, 2, and 3 on the rear of the joystick column. Each DIP-switch control affects one axis: Switch 1 the X-axis, Switch 2 for Y-axis, and Switch 3 for Z-axis.

- Set the Movement Step Size** – The XenoWorks Joystick/Controller is compatible with several different models of Sutter’s micromanipulator mechanicals. Because the compatible mechanicals move in steps of different sizes, users must inform the joystick which step-size is used by the mechanical connected to the joystick. The user should set DIP switch 4, for the step-size of the manipulator mechanical controlled by the XWM-100.
- Power-On Auto Calibration** – The XWM-100 Joystick/Controller remembers the latest position of its micromanipulator mechanical, even when power is turned off. Still, it is useful to periodically recalibrate the controller’s memorized positions. The best way to recalibrate the controller is to move the manipulator to its ‘Origin’, redefining the reference point for all measurements and defined positions for subsequent movements.

The XenoWorks Joystick can automatically recalibrate the controller each time power is turned on. This feature is controlled by the fifth DIP switch.

- Set the Pulse-Mode** – The XWM-100 controller can perform its Pulse function in two alternative ways. The first is optimized for minimizing unwanted lateral vibrations of the pipette tip. The second is optimized for pulsing rapidly. The two alternatives are likely to be useful for injections into different kinds of cells: the first for small or fragile cells and the second for larger cells, or cells surrounded by tougher membranes. The user should set DIP switch 6 to choose the pulse-mode best suited to their preparation.

2.3 Installing the Joystick Controller

Complete the system installation after the manipulator mechanical is installed on the microscope and the Configuration DIP Switches have been set.

1. Connect the manipulator mechanical to the controller
 - **CAUTION:** Be sure power to the controller is turned OFF before changing the connections of its cables.
 - Using the 25-conductor cable, connect one end to the DB-25 connector on the manipulator mechanical, and the other end to the Manipulator Port on the rear of the joystick unit. Be sure to securely tighten the mechanical cinch screws on either side of the connectors at both ends of the cable.
2. Set the joystick height adjustment for comfortable use and adjust the drag adjustment ring to achieve a comfortable motion. Ensure the joystick handle is centered (hanging straight down) and that the joystick unit is oriented with the main cylinder pointing to the front and the blue column at the rear.
3. Connect the power supply to the controller:
 - **CAUTION:** Be sure the Power Switch on the XWM-100 unit is OFF (bottom side pressed in) before connecting the power supply.
 - Insert the barrel plug from the power supply into the Power Connector on the rear of the XWM-100 controller. Be sure the plug is fully seated in the socket.
4. Connect the power supply to the wall (AC mains):
 - **Caution:** Be sure the chosen wall outlet will deliver power at either 60Hz 100-120v AC, or at 50Hz 200-220vAC. Double-check too that the Power Switch on the XWM-100 unit is OFF (bottom side pressed in), before connecting to the AC power.
 - Connect the AC line cord to the Power Supply and then to a grounded (3 pin) wall outlet.
5. Turn the power ON:
 - Turn the joystick's Power Switch ON (top side pressed in). After a few seconds, the Status Display should show it sign-on messages, and the controller should beep briefly. The display will then switch to its default 'Status Mode' display.

The XenoWorks™ micromanipulator is now set up and ready for use!

See the next section for details on how to align a micropipette at the start of an injecting session.

(This page intentionally left blank.)

3. OPERATING INSTRUCTIONS

3.1 General

Status Display

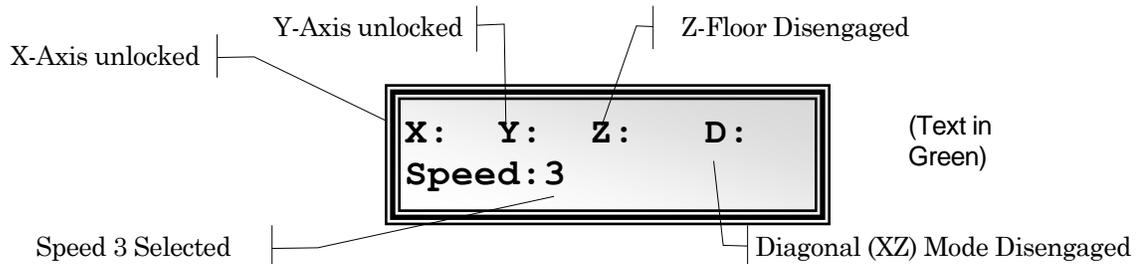


Figure 3-1. Reporting of the controller's operating modes.



Figure 3-2. Status Display when either the X- or Y-axes are locked at their current positions.



Figure 3-3. Status Display when the Diagonal (XZ) Mode is engaged for a right-side manipulator.

The XZ Mode icon will be slightly different for a left-side manipulator.



Figure 3-4. Z-Floor function is engaged, with the manipulator at its lower Z-limit (left) and above the Z-limit (right).



Figure 3-5. Status Display for a right-side manipulator when the X-Limit function is engaged, with the manipulator at its X-Limit (left) and away from the X-limit (right).

The X-Limit icon will be slightly different for a left-side manipulator



Figure 3-6. Status Display showing the manipulator's current position.

The above figure illustrates the Status Display when showing the manipulator's current position. Positions are always reported in microns (micrometers; 10^{-6} m), from the origin, where the origin is defined as the point furthest from the preparation along the X- and Z-axes, and closest to the user along the Y-axis.

NOTE: When the display is showing the manipulator's position and a special mode is engaged for one or more axes, the equal signs for the corresponding axes will be replaced by the special-mode icons shown in figures.

3.1.1 Moving the Micropipette with the Joystick

Once the micromanipulator has been installed correctly, the micropipette should follow the movement of the handle. For best results, be sure to check the directions of movements of the pipette tip along both the X- and Y-axes, making sure that the tip moves in the same direction as the joystick handle (ex. leftward swings of the joystick move the tip to the left and rearward swings of the handle move the tip to the rear).

Note too that the higher the setting on the range (speed) control, the smaller the pipette's movement will be for a given movement of the handle. When in range settings '3' through '7', the micropipette will follow the movements of the joystick without delay. Settings 'C', '1', and '2' are for coarse positioning and should generally not be used for precision micromanipulation.

3.1.2 Centering the Micropipette with Respect to the Joystick

The Declutch mechanism can be used to re-center the handle with respect to the micropipette tip. This is useful when the micropipette is at the edge of the microscope field of view, but the handle is centered, or when the micropipette is in the center of the field of view, but the handle is at the edge of its travel.

To center or reposition the handle without affecting the micropipette, simply touch the Declutch Touch-Band above the rubber grip. Once the red Declutch LED lights, you can recenter the handle without moving the manipulator mechanical. When the handle is in its final position, simply remove your finger from the Touch-Band.

3.1.3 Setting the Angle of Approach of the Micropipette

The micropipette holder clamp is capable of rotating around its own axis. Its position can be locked with the thumbscrew (1) shown here. It is also possible to slide the clamp and lock it in place with a separate thumbscrew (2). The clamp itself is spring-loaded, allowing a 4 mm diameter micropipette holder (not included) to be advanced or retracted easily without removing it or risk of it, and without falling out inadvertently.

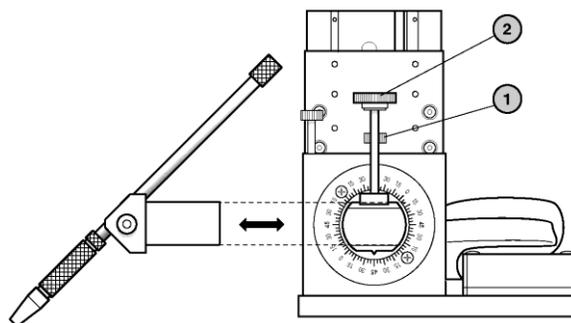


Figure 3-7. Micropipette alignment controls.

These degrees of freedom allow for angles of attack from zero to greater than 45 degrees, while at the same time allowing the micropipette holder to be clamped at any point along its length. The design accommodates many different micropipettes, angled or straight, and accessories such as piezo impact drills.

When combined with the three fixing patterns on the top plate of the mounting adapter and the sliding action of the mounting adapter, the XenoWorks™ Micromanipulator can accommodate almost any application on a large number of microscope platforms.

The following paragraphs describe factors that affect the setting of the micropipette holder.

3.1.3.1 Microinjection Chamber

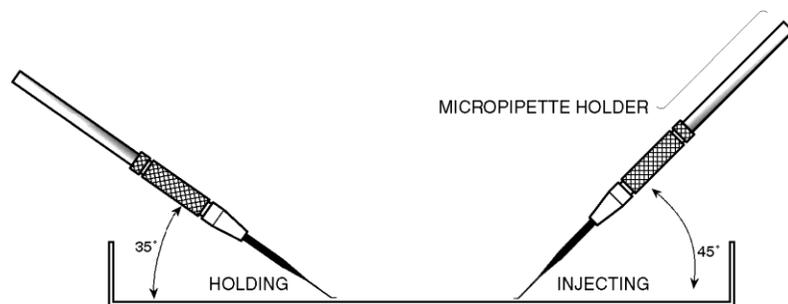
Some types of microinjection chamber, plastic culture dishes for example, require a steep angle of approach (up to 45 degrees), so that the micropipette clears the sides of the dish. Other chambers, depression slides for example, can accommodate a much shallower angle of approach (typically around 10–15 degrees for a depression slide).

3.1.3.2 Applications and Suggested Setups

In general, any application in which suspended cells are to be injected or manipulated (ex. embryonic stem cell transfer, intracytoplasmic sperm injection, pronuclear zygote injection, embryo reconstruction) will require two micropipettes (and thus two micromanipulators), one for holding and one for injecting. In this instance, the micropipette tips must approach the tissue side-on to prevent the injection micropipette from pushing the injected cell away from the holding pipette. When using a deep dish and angled micropipettes, a 35–45-degree angle of attack is appropriate (shown in the figures on the page opposite). If using a depression slide and straight micropipettes, an almost flat angle is required as shown in the figures below.

Recommended Setups for Applications

a.



b.

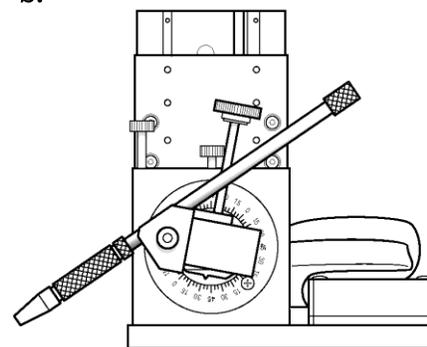
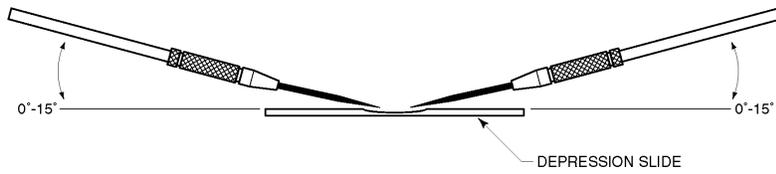


Figure 3-8 (a and b). Suggested approach for **suspended cells with angled micropipettes** (ICSI, Stem Cell, Nuclear Transfer).

a.



b.

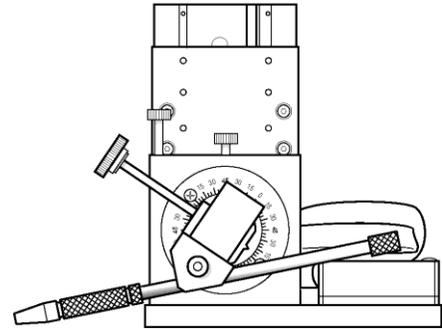
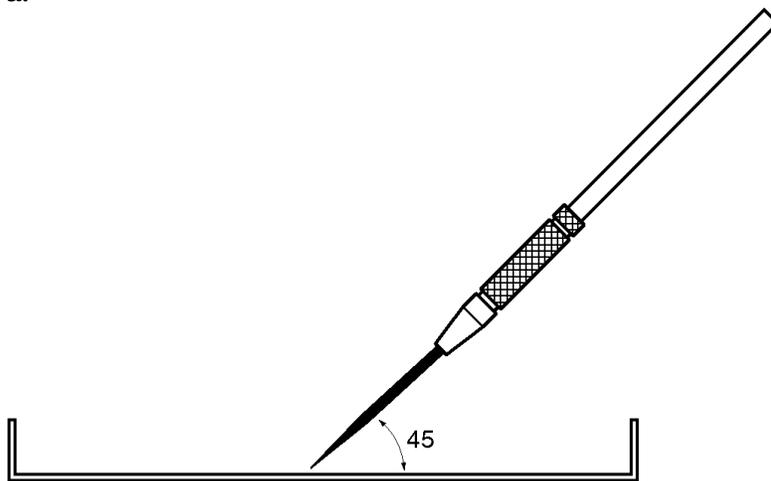


Figure 3-9 (a and b). Suggested approach for **suspended cells with straight micropipettes** (Pronuclear Injection).

a.



b.

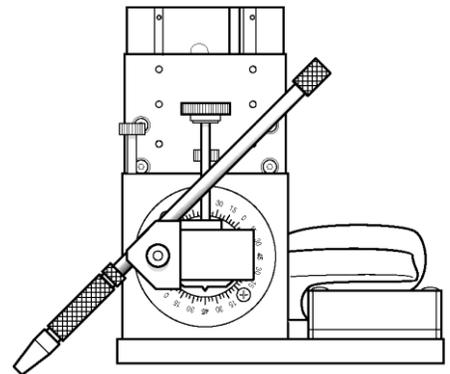


Figure 3-10 (a and b). Suggested approach for **adherent cells and straight pipettes**.

3.1.3.3 Micropipette Type

Some micropipettes (lab-made or purchased ICSI pipettes, for example) often include an angle close to the tip. This is to allow the micropipette tip to approach the injected cell along a path parallel to the bottom of the chamber, and to allow the use of a high-sided dish. Other micropipettes may not have a bend, in which case they must be used with a flat injection chamber (a depression slide for example) of a low angle of approach.

3.1.3.4 Working Distance

Most microinjection procedures are carried out on an inverted microscope where the condenser is located above the stage and the objective lenses below. If the microscope was purchased specifically for microinjection, it was probably supplied with a “long” or “very long” working distance condenser that should provide adequate space above the stage to set up the micropipette and accommodate any type of chamber. When installing the manipulator adapter plate, the manipulator mechanical, and the micropipette holder, carefully observe the relative positions of the micromanipulator, the microscope stage, and the condenser to avoid collision by the manipulator.

3.1.3.5 Piezo Impact Drive

For those applications that require the use of a piezo impact drive, the position of the piezo must be taken into account when aligning the micropipette holder (and also when attaching the mechanical to the microscope mount). Typically, a piezo device is mounted at the back of the micropipette holder, and the holder is held in place at the very front. A Prime Tech piezo device can be mounted anywhere along the length of the micropipette holder, either in front, or behind the micropipette holder clamp.

3.2 Controller Functions

3.2.1 Using the ‘Work’ Function

This is accomplished by first using the joystick handle to manually move the micropipette to the desired ‘Work’ position. Then press and hold down the ‘Work’ button on the controller’s baseplate for approximately three seconds, until a tone is heard, indicating that the ‘Work’ position has been memorized. Any previously memorized ‘Work’ position is overwritten each time a new work position is memorized. The current ‘Work’ position is retained, even when power is switched off.

The micropipette can be returned to the currently set ‘Work’ position by clicking the ‘Work’ button. The movement towards the work position can be halted by clicking the ‘Work’ key a second time while the movement is underway. Clicking the ‘Work’ button again resumes movement towards the ‘Work’ position. This process can be repeated until the micropipette reaches the ‘Work’ position. Because pipettes often differ in length and in their exact position in the holder, it is generally advisable to reset the ‘Work’ position each time the micropipette is switched.

3.2.2 Using the ‘Home’ Function

As for the ‘Work’ position, the ‘Home’ position should be set by first using the joystick handle to move the pipette to the desired safe position. Then press and hold down the ‘Home’ button on the controller’s baseplate for approximately 3 seconds, until a tone is heard indicating that the ‘Home’ position has been memorized. Any previously memorized ‘Home’ position is overwritten each time a new home position is memorized. The current ‘Home’ position is retained, even when power is switched off.

This ‘Home’ position is generally used to retract the micropipette from the injection chamber to a safe position (to change dishes, for example), or to exchange micropipettes. Note that a

‘Work’ position must be set (see above) if the micropipette is to be returned to the position it occupied before the ‘Home’ function was activated. When the ‘Home’ function is activated, the micropipette will move to the ‘Home’ position. This movement can be halted at any time by clicking ‘Home’ again, or by briefly pressing ‘Work’. Once the movement is halted, pressing ‘Home’ again will resume the movement of the micropipette to the ‘Home’ position. Clicking the ‘Work’ key will move the micropipette to the ‘Work’ position (In this way, a micropipette can be partially raised out of the injection chamber and returned, without having to move all the way out to the ‘Home’ position – see “Cleaning the Micropipette” below). Note that the micropipette tip moves out at approximately 45°, no matter what its angle of incidence.

1.4.1 Using the ‘Pulse’ Function

The pulse function has proven valuable in rapidly injecting cells, similar to the way a piezo impact drive aids injecting cells. See the Installation Instructions in Chapter 1 for more details about the two methods of Pulse function provided by the XWM-100.

3.2.3 Changing the Working Range of Movement

The range of movement of the micropipette can be varied depending on the user’s desired movement (large, fast movements vs. small, slow movements) and the size of the microscope’s field of view. The range can be changed by simply clicking the ‘Speed +’ and ‘Speed –’ until the desired speed and range selected. The ranges and maximum speeds of each movement setting are listed under Specifications.

3.2.4 Constraining Pipette Movements

3.2.4.1 Using the ‘Y-Lock’ and ‘X-Lock’ functions

It is sometimes useful to limit the pipette’s movement to a single horizontal axis, creating movements purely parallel to the X- or Y-axis. For example, movements along only the X-axis may be helpful to reduce lateral vibration of the pipette tip when injecting or retracting from small or delicate cells. Clicking Function Button 3 will alternately lock and unlock the manipulator’s movement along its Y-axis. When movements along the Y-axis are locked, swinging the joystick handle will move the pipette tip parallel to the X-axis; any handle movements along the Y axis will be ignored.

Similarly, Touch-Clicking Function Button 3 will alternately lock and unlock movements along the X-axis. When movements along the X-axis are locked, swinging the joystick handle will move the pipette tip purely along the Y-axis; any handle movements along the X axis will be ignored.

NOTE: Locking the X- and Y-axis are independent operations. It is possible to simultaneously lock both the X- and Y-axes. While this may be useful in some circumstances, it may also confuse an unwary user. If the manipulator mechanical moves in the vertical (Z) direction, but not in one or both horizontal directions, be sure to check the state of the X- and Y-axis locks.

Note too that locking the X- or Y-axis takes precedence over the go to change in spec Home and Work commands. That is, if the X- or Y-axis is locked, the manipulator mechanical will not move its position along the locked axis when commanded to go to Home or Work.

3.2.4.2 Setting a ‘Z-Floor’

Follow these steps to set the ‘Z-floor’:

1. Locate the tip of the micropipette in the field of view of the microscope.

2. Slowly rotate the Z-axis knob to lower the micropipette towards the bottom of the injection chamber, following the micropipette tip using the microscope's fine focus. Use slower speeds (3 – 7) when close to the bottom of the dish.
3. When the micropipette touches the bottom of the injection chamber, its tip will move inward (sideways, away from the pipette's shaft) as it bends in contact with the bottom of the dish. (**CAUTION: Sharp micropipettes, such as cell injection micropipettes or pipettes with high angles of approach, may break if they are pressed too hard against the bottom of the injection chamber.**) Raise the micropipette up slightly so that it just clears of the bottom of the chamber.
4. Press and hold the 'Z-Floor' button for about three (3) seconds. A tone will be heard, and the Status Display will update, indicating that the Z-Floor has been set and that the function is active. The 'Z-floor' function is disabled by again pressing and holding the 'Z-floor' button. Each time the Z-Floor function is engaged, the Z-floor position is reset to the current Z position.
5. While the 'Z-floor' is active, the Z-axis will not move any lower than the memorized position, protecting the micropipette from inadvertently breaking against the bottom of the chamber.

CAUTION: If a 'Home' or 'Work' position is set below the 'Z-floor' limit, the micropipette will halt vertical movement at the Z-Floor, before reaching the Z-axis position of the memorized 'Work' position.

3.2.4.3 Setting an 'X-Limit'

Follow these steps to set an 'X-Limit':

1. Locate the tip of the micropipette in the field of view of the microscope.
2. Maneuver the tip of the micropipette until it is just above the cell to be injected (or the x-axis position to be used as the X-Limit).
3. While holding a finger against the Declutch Touch-Band, press and hold Function Button 3 until the tone is heard, about 3 seconds. The status display will update, indicating that the X-Limit function is engaged.
4. The micropipette can now be moved as needed, but it will not advance beyond the X-Limit so long as the function is engaged.
5. The X-Limit remains engaged until the user again touches the declutch band and holds button 3 for approximately 3 seconds, until the tone is heard, and the display indicates the X-Limit has been disengaged.

3.3 Common Uses of Controller Functions

3.3.1 Cleaning the Micropipette

The 'Work' and 'Home' functions can often be used together to clean material from the tip of the micropipette, by passing the micropipette back and forth through the surface of the injecting medium.

Record the 'Work' position with the tip of the pipette in the center of the field of view, then click the 'Home' button to start moving the tip towards the 'Home' position. As soon as the micropipette tip has passed out of the injection medium, click 'Home' again to stop the micropipette. Finally, click 'Work' to return the micropipette to the 'Work' position.

3.3.2 Exchanging the Micropipette

During a microinjection procedure, it may be necessary to exchange micropipettes. The 'Home' and 'Work' functions provide one effective way to do this.

1. Center the micropipette in the microscope field of view and above the dish by 3 mm. Press and hold down the 'Work' key to memorize the coordinates.
2. Click 'Home', and the manipulator mechanical will move to the upper and outer limits of travel. The movement from 'Work' to 'Home' can be paused at any time by clicking 'Home' or 'Work' again. Movement to 'Home' can be resumed by clicking 'Home' yet again.
3. Unlock and open the swing gate to access the top of the micropipette holder.
4. Taking care not to disturb the position of the micropipette holder in its clamp, loosen the chuck on the front of the micropipette holder and remove and discard the micropipette safely.
5. Fit a new micropipette into the holder and lock the chuck in place.
6. Return the swing gate to the working position and lock it in place.
7. Press 'Work' to move the micropipette above the injection chamber and in the field of view.

CAUTION: It is important to remember that when activating the 'Work' function, the micromanipulator records the positions of the micromanipulator motors – not the position of micropipette tip. Therefore, if one micropipette is replaced by one that is longer or shorter, the micropipette tip will not return to the original spot when returning to the 'Work' position. If another pipette was shorter than the replacement pipette, the new pipette may crash into the bottom of the injection chamber upon returning to 'Work'. Several measures, taken alone or together, can prevent this: When returning to 'Work' from 'Home', approach the injection chamber with the new micropipette in stages by repeatedly clicking the 'Work' key. Once the micropipette tip is close to the bottom of the injection chamber, use small movements on low speeds with the joystick handle, to reach the desired position. Reset Work position to conform to the new pipette.

3.4 Starting a Microinjection Session

3.4.1 Setting up the Microscope

1. Locate the correct focal plane for the microinjection chamber in the following manner:
2. Place a representative microinjection dish, chamber, or slide on the stage of the microscope, set the microscope to its lowest possible magnification, and focus on the uppermost surface of the bottom of the chamber (an effective method is to draw a pen mark in the center of the chamber to be used, and focus on that). This establishes the correct microscope focal plane for microinjection.
3. The microinjection chamber can now be removed and, if the microscope focus is not disturbed, the micropipette tip can be manipulated into the optical axis, above the bottom of the injection chamber.

3.4.2 Setting up the Micromanipulator

1. Remove the micropipette holder (rod) from the clamp of the mechanical.
2. Load a micropipette appropriate to the chosen application into the micropipette holder and install the holder in the clamp of the mechanical so that the tip of the micropipette is pointing in and down towards the optical axis of the microscope.

3. Adjust the angle setting of the micropipette clamp on the mechanical so that the angle of approach of the pipette is appropriate for the chosen application, as described in Operating Instructions.
4. Once an appropriate angle of approach has been set, loosen the micropipette holder clamp slightly and slide the holder axially inward and downward, along its own axis, until the micropipette tip is level with (almost touching) the plane of the microscope stage.
NOTE: Do this by eye – it is not necessary to look through the microscope for this step.
5. Place the manipulator in ‘Coarse’ mode ‘C’, and carefully manipulate the joystick until the tip of the micropipette can be seen through the microscope eyepieces.
6. Change the speed to 1 or 2, center the micropipette tip in the microscope field of view, and change microscope to a higher magnification. Remember not to adjust the microscope focus at this point. To “focus” the micropipette tip, move the pipette vertically (along the Z-axis) by rotating the Z-axis dial at the bottom of the joystick.
7. Once the desired magnification has been reached with the micropipette tip still centered in the field of view, set the ‘Work’ position.
8. Be sure you have set the ‘Work’ position. Then to replace the microinjection dish, click ‘Home’ to move the micromanipulator towards the ‘Home’ position. Once the micropipette tip has stopped, clear of the microscope stage, replace the microinjection dish. Pressing the ‘Work’ button now will cause the micropipette tip to return to the ‘Work’ position previously set.
9. At this point, it may be desirable to set the ‘Z-floor’ function to prevent accidentally breaking the micropipette during use. To do this, lower the tip to the bottom of the dish until you see the micropipette tip start to “kick-out” slightly as it touches the dish (this is not recommended for sharp micropipettes with a high angle of attack, because the micropipette tip may break). Press and hold the ‘Z-floor’ button until a tone is heard. The ‘Z-floor’ is now active and will stay in the micromanipulator memory until it is overwritten with a new limit or until the instrument is powered down.
10. The XenoWorks™ Micromanipulator is now ready for microinjection.

Note that, with experience, the user may choose to skip some these steps (like the Z-limit).

(This page intentionally left blank.)

4. EXTERNAL CONTROL

4.1 General

Users can control the XenoWorks XWM-100 system from an external computer by sending commands over a USB interface between the computer and the connector on the rear of the joystick. When using a Windows computer, the external interface for the XenoWorks XWM-100 requires use of Sutter's CDM (Combined Driver Model) device driver, version 2.12.18 or higher, available from the Technical Support page on Sutter Instrument's web site (www.sutter.com). The CDM driver for the XenoWorks XWM-100 consists of two components: 1) a USB device driver, and 2) a VCP (Virtual COM Port) device driver. Install the USB driver first, followed by the VCP driver. The VCP driver provides a serial I/O interface between a Windows application and the XenoWorks controller. 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). Using the first, USB Direct, method requires either that the VCP device driver not be installed, or if installed, that it be disabled. The second (Serial) method requires that the VCP be both 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))	9,600
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 sent to the XenoWorks system do not have explicit terminators. All responses from the XenoWorks system end with 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 it is ready to receive a new command. If a response returns other data as well, 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 sent to the XenoWorks system must be an unsigned binary value. Attempting to encode command sequences as “strings” is not advisable. Any data returned from the XenoWorks system should be treated upon reception, as a sequence of unsigned byte values. Groups of contiguous bytes can later be combined to form larger values, as appropriate (e.g., 2 bytes into 16-bit “word”, or 4 bytes into a 32-bit “long” or “double word”). For the XenoWorks BRM, all axis position values (number of micro-steps) 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 micro-steps. Conversion between micro-steps and microns (micrometers) is the responsibility of the software running on the host computer (see *Microns/micro-steps conversion* table for conversion factors).

Micro-steps are stored and reported 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 byte-order reversed after receiving from, or before sending to, the XenoWorks system. 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 Micro-steps and Microns (Micrometers)

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

Table 4-2. Microns/micro-steps conversion.

System/Device	From/To Units	Conversion Factor (multiplier)
XenoWorks XWM/M, MP-285/M	$\mu\text{steps} \rightarrow \mu\text{m}$	0.125
	$\mu\text{m} \rightarrow \mu\text{steps}$	8
MP-845/M	$\mu\text{steps} \rightarrow \mu\text{m}$	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 micro-steps, type the result as a 32-bit “long”, “signed long”, or “I32” integer. When converting to microns, type the result as “double” (64-bit double-precision floating-point values).

Table 4-3. Ranges and bounds.

Device	Axes	Millimeters	Microns	Micro-steps
XenoWorks XWM/M MP-285/M	X,Y,Z	25	0 – 25,000	0 – 200,000
MP-845/M	X,Y,Z	25	0 – 25,000	0 – 266,667

4.7 Range Speed

The following table shows the range speeds.

Table 4-4. Range (speed) settings.

Range (Speed) Setting	X & Y Axes (1 joystick swing)	Z Axis (1 rotation)
C	13,700 μm	3,200 μm
1	3,560 μm	800 μm
2	950 μm	400 μm
3	432 μm	200 μm
4	216 μm	100 μm
5	108 μm	50 μm
6	100 μm	30 μm
7	50 μm	10 μm

4.8 Travel Speed

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

Table 4-5. Travel speeds.

Device	mm/sec or $\mu\text{m}/\text{ms}$		
	Single Axis	Dual Axis (x 1.4)	Triple Axis (x 1.7)
XenoWorks XWM/M MP-285/M	3.0	4.2	5.1
MP-845/M	2.5	3.5	4.2

4.9 Commands

4.9.1 Get Name and Firmware Version ('K') Command

This command is used to obtain the name of the product vision of the firmware.

Table 4-6. Get name and firmware version ('K') command (Firmware <2).

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	75	4B	0100 1011	0075		'K'	Command
Rx.	<2	34	0-29	A sequence of ASCII characters describing the name of the product: "Sutter Inst. XenoWorks XWM-100"						
			30	Build version number coded in BCD (e.g., if ver. = 3.15.16, then byte 30 = 0x16; (upper nibble = 1; lower nibble = 6)						
			31	Minor version number coded in BCD (e.g., if ver. = 3.15.16, then byte 31 = 0x15; upper nibble = 1; lower nibble = 5)						
			32	Major version number coded in BCD (e.g., if ver. = 3.15.16, then byte 32 = 0x03; upper nibble = 0; lower nibble = 3)						
			33	13	0D	0000 1101		^M	<CR>	Completion indicator

Extracting the XenoWorks XWM-100 Firmware Version Number (Firmware <2): The firmware version number returned by the 'K' command is encoded in BCD (Binary Coded Decimal) in three bytes, with build version byte first, followed by minor version byte and major version byte, each of which contains a two-digit pair, the first digit of which is in the upper nibble and the second in the lower nibble. For example, if the complete version is 3.15.16, then the bytes at offsets 30, 31, and 32 will contain (in hexadecimal) as 0x16, 0x15, and 0x03 (ret[30], ret[31], and ret[32] below). The following snippet of code shows how to extract and convert the 6 BCD digits into usable forms for later comparison without altering the original return data (written in C/C++ but is easily portable to Python, Java, C#, MATLAB script, etc.).

```

/* "ret" is the array of bytes containing the 'K' command's return data
*/
/* define variables */
unsigned char verbyte; /* temp work byte */
int buildver, minver, majver, majminver;
float version;

```

Build version number as an integer (e.g., 16):

```

verbyte = ret[30]; /* build ver. digits */
buildver = (verbyte & 0x0F) +
  ((verbyte >>4 & 0x0F) * 10);

```

Minor version number as an integer (e.g., 15):

```

verbyte = ret[31]; /* minor ver. digits */
/* get 1's digit & then get & add 10's digit */
minver = (verbyte & 0x0F) +
  ((verbyte >>4 & 0x0F) * 10);

```

Major version number as an integer (e.g., 3):

```

verbyte = ret[32]; /* major ver. digits */
majver = (verbyte & 0x0F) +
  ((verbyte >>4 & 0x0F) * 10);

```

Complete version as an integer (e.g., 31516):

```

majminbldver = majver * 10000 + minver * 100 + buildver;

```

Complete version as a floating-point number (e.g., 3.1516):

```

version = majminbldver * .0001;

```

Table 4-7. Get name and firmware version ('K') command (Firmware ≥2) .

Tx/- Delay/- Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	75	4B	0100 1011	0075		κ	Command
Rx	≥2	31	0-27	A sequence of ASCII characters describing the name of the product: "Sutter XenoWorks XWM-100 "						
			28	Minor version number coded in BCD (e.g., if ver. = 3.15, then byte = 0x15 (upper nibble = 1; lower nibble = 5))						
			29	Major version number coded in BCD (e.g., if ver. = 3.15, then byte = 0x03 (upper nibble = 0; lower nibble = 3))						
			30	13	0D	0000 1101		^M	<CR>	Completion indicator

Extracting the XenoWorks XWM-100 Firmware Version Number (Firmware ≥2): The firmware version number returned by the 'K' command is encoded in BCD (Binary Coded Decimal) in two bytes, with minor version byte first, followed by major version byte, each of which contains two digits with first in the upper nibble and the next in the lower nibble. For example, if the complete version is 3.15, then the bytes at offsets 28 and 29 will show (in hexadecimal) as 0x15 and 0x03 (ret[28] and ret[29] as shown in the following code snippets). The following code shows how to extract and convert the 4 BCD digits into usable forms for later comparison without altering the original command return data (written in C/C++ and easily portable to Python, Java, C#, MATLAB script, etc.).

```
/* "ret" is the array of bytes containing the 'K' command's return
data */
/* define variables */
unsigned char verbyte; /* temp work byte */
int minver, majver, majminver;
float version;
```

Minor version number as an integer (e.g., 15):

```
verbyte = ret[28]; /* get minor ver. digits */
/* get 1's digit & then get & add 10's digit */
minver = (verbyte & 0x0F) +
((verbyte >>4 & 0x0F) * 10);
```

Major version number as an integer (e.g., 3):

```
verbyte = ret[29]; /* get major ver. digits */
majver = (verbyte & 0x0F) +
((verbyte >>4 & 0x0F) * 10);
```

Complete (thousands) version as an integer (e.g., 315):

```
majminver = majver * 100 + minver;
```

Complete version as a floating-point number (e.g., 3.15):

```
version = majminver * .01;
```

4.9.2 Get Current Position ('C') Command

This command is used to obtain the current position (X, Y, & Z coordinates) of the manipulator or stage. The command sequence consists of one byte as shown in the following table. The data received consists of seventeen bytes containing X, Y, & Z position (32-bit) values in micro-steps (4 bytes each), and the completion indicator (1 byte).

Table 4-8. Get current position ('C') command (Firmware <2).

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	67	43	0100 0011	0043		'C'	Command
Rx	<2	17		Three 4-byte (32-bit) values (current positions in μ steps of X, Y, & Z) + one 2-byte value of approach angle + one 2-byte value of resolution + 1 byte for completion indicator (see <i>Ranges</i> table for minimum/maximum values)						
			0-3 (4)	X pos. in μ steps						
			4-7 (4)	Y pos. in μ steps						
			8-11 (4)	Z pos. in μ steps						
			12-13 (2)	Approach angle in degrees [1,45]						
			14-15 (2)	Internal resolution (see Table 2)						
			12	13	0D	0000 1101		^M	<CR>	Completion indicator

Table 4-9. Get current position ('C') command (Firmware \geq 2).

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	67	43	0100 0011	0043		'C'	Command
Rx	\geq 2	13		Three 4-byte (32-bit) values (current positions in μ steps of X, Y, & Z) + 1 byte for completion indicator (see <i>Ranges</i> table for minimum/maximum values)						
			0-3 (4)	X pos. in μ steps						
			4-7 (4)	Y pos. in μ steps						
			8-11 (4)	Z pos. in μ steps						
			12	13	0D	0000 1101		^M	<CR>	Completion indicator

NOTE: See *Microns/micro-steps conversion table* for conversion from μ steps to μ m (microns).

NOTE: All positions are in micro-steps (μ steps): 32-bit (4 bytes) positive (unsigned) integer values, in Little Endian bit order (see Notes).

4.9.3 Toggle XY Mode (Diagonal/Orthogonal) ('D') Command

This command toggles XY Mode between diagonal and orthogonal.

Table 4-10. Toggle XY Mode (Diagonal/Orthogonal) ('D') Command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	68	44	0100 0100	0068		D	Command
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

4.9.4 Setting Diagonal Approach Angle ('A') Command

Sets the angle value, in degrees.

Table 4-11. Set the angle ('A') command (Firmware <2).

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All <2	3	0	65	41	1010 1001	0065		'A'	Command
			1-2 (2)							
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

Table 4-12. Set the angle ('A') command (Firmware ≥2).

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All ≥2	2	0	65	41	1010 1001	0065		'A'	Command
			1	1	01	0000 0001	0001			Approach angle in degrees (1 – 45 (1 byte))
				-	-	-	-			
			45	2D	0010 1101	0045				
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

4.9.5 Perform XZ Pulse ('P') Command

This command performs XZ pulse.

Table 4-13. Perform XZ pulse ('P') command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	80	50	0101 0000	0080		P	Command
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

4.9.6 Move to Specified Position at Full Speed ('M') Command

This command instructs the controller to move all 3 axes to specified position.

Table 4-14. Move to specified position at full speed ('M') command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description	
				Dec.	Hex.	Binary					
Tx	All	13	0	77	4D	0101 1101	0077		M	Command	
				X, Y, & Z target positions, in micro-steps from the origin, each consisting of 4 contiguous bytes representing a single 32-bit, positive, integer (see <i>Ranges</i> table).							
			1-4 (4)	X μ steps							
			5-8 (4)	Y μ steps							

Tx/- Delay/- Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
≥30ms			9-12 (4)							Z μ steps
				Delay between send and next command or timeout. To be calculated based on distance of travel.						
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

NOTE: All positions are in micro-steps (μ steps): 32-bit (4 bytes) positive (unsigned) integer values, in Little Endian bit order (see Notes).

4.9.7 Move to Origin ('O') Command

This command moves to the origin position.

Table 4-15. Move to origin ('O') command.

Tx/- Delay/- Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	79	4F	0100 1111	0079		o	Command
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

4.9.8 Move to Home Position ('H') Command

This command moves to the position saved by the controller's HOME button.

Table 4-16. Move to controller-defined HOME position ('H') command.

Tx/- Delay/- Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	72	48	0100 1000	0072		H	Command
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

4.9.9 Move to Work Position ('Y') Command

This command moves to the position saved by the controller's WORK button.

Table 4-17. Move to controller -defined WORK position ('Y') command.

Tx/- Delay/- Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	89	59	0101 1001	0089		Y	Command
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

4.9.10 Move to Center Position ('N') Command

This command moves to the center position.

Table 4-18. Move to center position ('N') command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	78	4E	0100 1110	0078		N	Command
Rx	All	1	0	13	0D	0000 1101			<CR>	Completion indicator

4.9.11 Get Approach Angle ('a') Command

This command returns the approach angle setting.

Table 4-19. Get approach angle ('a') command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	≥2	1	0	97	61	0110 0001	0097		a	Command
Rx	≥2	2	0							Approach angle in degrees
			1	13	0D	0000 1101			<CR>	Completion indicator

4.9.12 Report Internal Resolution ('R') Command

This command returns an integer flag indicating the current resolution.

Table 4-20. Report internal resolution ('R') command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	≥2	1	0	82	52	0101 0010	0082		R	Command
Rx	≥2	3	0-1							Integer flag indicating resolution: 8000 or 10667 (unsigned short (2 bytes), Little-Endian)
			2	13	0D	0000 1101			<CR>	Completion indicator

4.9.16 Move to Specified Position at Selected Speed ('m') Command

This command instructs the controller to move all 3 axes to specified position at the selected speed (0 - 7).

Table 4-24. Move to specified position at selected speed ('m) command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description		
				Dec.	Hex.	Binary						
Tx	≥2	14	0	109	6D	0110 1101	0109		m	Command		
			1	0	00	0000 0000	0000			Selected speed: 0 through 7		
				-	-	-	-					
			7	07	0000 0111	0007						
			X, Y, & Z target positions, in micro-steps from the origin, each consisting of 4 contiguous bytes representing a single 32-bit, positive, integer (see <i>Ranges</i> table).									
			2-5 (4)									X μsteps
			6-9 (4)									Y μsteps
10-13 (4)									Z μsteps			
30ms				Delay between send and next command or timeout. To be calculated based on distance of travel and velocity.								
Rx	≥2	1	0	13	0D	0000 1101			<CR>	Completion indicator		

NOTE: All positions are in micro-steps (μsteps): 32-bit (4 bytes) positive (unsigned) integer values, in Little Endian bit order (see Notes).

4.9.17 Interrupt Move ('^C') Command

This command interrupts a move in progress. The command sequence consists of one byte.

Table 4-25. Interrupt move ('^C') command.

Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt-key- pad #	Ctrl- char	ASCII def./- char.	Description
				Dec.	Hex.	Binary				
Tx	All	1	0	3	03	0000 0011	0003	^C	<ETX>	Command
Rx		1	0	13	0D	0000 1101			<CR>	Completion indicator

4.9.18 Command Notes

The following list of notes apply to all the external control commands described in this chapter.

- 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 the I/O Send & Receive Buffers:** Clearing (purging) the transmit and receive buffers of the I/O port immediately before sending any command is recommended. Note that this clearing of the buffers affects only the computer-side I/O; it does not (necessarily) clear the buffers on the controller side, requiring, when necessary, to reset/power-cycle the controller. Following the rules described will generally avoid problems with getting garbage data in the I/O buffers of both the computer and controller (i.e., using exact number of bytes for both command sequences and return data (as per the *Commands*

table), never sending a command before the previous command is finished with its task, etc.).

4. **Positions in Micro-steps and Microns:** All positions sent to and received from the controller are in micro-steps (μ steps). See Microns/micro-steps conversion table) for conversion between μ steps and microns (micrometers (μ m)).

Declaring position variables in C/C++:

```
/* current position for X, Y, & Z */
unsigned long cp_x_us, cp_y_us, cp_z_us; /* micro-steps */
double cp_x_um, cp_y_um, cp_z_um; /* microns */
/* specified (move-to) position for X, Y, & Z */
unsigned long sp_x_us, sp_y_us, sp_z_us; /* micro-steps */
double sp_x_um, sp_y_um, sp_z_um; /* microns */
```

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

Declaring the micro-steps/microns conversion factors in C/C++:

```
/* conversion factors for the Xenoworks XWM/M and MP-285/M based
config. */
double us2umCF = 0.125; /* micro-steps to microns */
double um2usCF = 8; /* microns to micro-steps */
/* conversion factors for the MP-845[S]/M based config. */
double us2umCF = 0.09375; /* micro-steps to microns */
double um2usCF = 10.666666666667; /* microns to micro-steps */
```

Converting between micro-steps and microns in C/C++:

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

Do the same for Y and Z, and for any other position sets used in the application.

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

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

```
/* minimum and maximum positions for X, Y, & Z */
double min_x_um, min_y_um, min_z_um; /* minimum microns */
double max_x_um, max_y_um, max_z_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;
/* initialize all maximum positions in microns*/
/* Xenoworks XWM/M, MP-285/M, MP-845[S]/M, etc. */
max_x_um = 25000;
max_y_um = 25000;
max_z_um = 25000;
```

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 the end of travel (EOT)) are positive

values; there are no positions ‘lower’ than the origin, 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 */
double rp_x_um, rp_y_um, rp_z_um; /* microns */
/* initialize all relative positions to 0 after declaring them */
rp_x_um = rp_y_um = rp_z_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 ... 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 micro-steps 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 micro-steps */
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 micro-steps and consist of 32-bit integer values (four contiguous bytes). Position values can be either positive or negative, so type must be “signed”. 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 a signed integer (“long” or “signed long” in C/C++; “I32” 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 micro-steps 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. **Orthogonal Move Speed:** Full speed for the “Orthogonal Move” (‘M’) command is 5000 microns/sec. (5 mm/sec. or microns/millisecond) for single-axis movements (XWM/M or MP-285/M) and 3000 microns/sec. (3 mm/sec. or $\mu\text{m}/\text{ms}$) for TRIO MP-845/M).
12. **Multi Axis Movement Speed Increase:** Specified travel speeds are for single-axis movements. When travel traverses a 45° diagonal within a dual-axis square, speed is increased by 40% (x 1.4), and by 70% (x 1.7) within a triple-axis cube.
13. **Move Interruption:** A command should be sent to the controller only after the task of any previous command is complete (i.e., the task-completion terminator (CR) is returned). One exception is the “Interrupt Move” (^C) command, which can be issued while a command-initiated move is still in progress.
14. **Extracting the XenoWorks XWM-100 Firmware Version Number (Firmware <2)::** The firmware version number returned by the ‘K’ command is encoded in BCD (Binary Coded Decimal) in three bytes, with build version byte first, followed by minor version byte and major version byte, each of which contains a two-digit pair, the first digit of which is in the upper nibble and the second in the lower nibble. For example, if the complete version is 3.15.16, then the bytes at offsets 30, 31, and 32 will contain (in hexadecimal) as 0x16, 0x15, and 0x03 (ret[30], ret[31], and ret[32] below). The following snippet of code shows how to extract and convert the 6 BCD digits into usable forms for later comparison without altering the original return data (written in C/C++ but is easily portable to Python, Java, C#, MATLAB script, etc.).

```
/* Assume "ret" is the array of bytes
containing the 'K' command's return data */
unsigned char verbyte; /* temp work byte */
int buildver, minver, majver, majminver;
float version;
```

Build version number as an integer (e.g., 16):

```
verbyte = ret[30]; /* build ver. digits */
buildver = (verbyte & 0x0F) +
((verbyte >>4 & 0x0F) * 10);
```

Minor version number as an integer (e.g., 15):

```
verbyte = ret[31]; /* minor ver. digits */
/* get 1's digit & then get & add 10's
digit */
minver = (verbyte & 0x0F) +
        ((verbyte >>4 & 0x0F) * 10);
```

Major version number as an integer (e.g., 3):

```
verbyte = ret[32]; /* major ver. digits */
majver = (verbyte & 0x0F) +
        ((verbyte >>4 & 0x0F) * 10);
```

Complete version as an integer (e.g., 31516):

```
majminbldver = majver * 10000 + minver * 100 + buildver;
```

Complete version as a floating-point number (e.g., 3.1516):

```
version = majminbldver * .0001;
```

Extracting the XenoWorks XWM-100 Firmware Version Number (Firmware ≥2): The firmware version number returned by the 'K' command is encoded in BCD (Binary Coded Decimal) in two bytes, with minor version byte first, followed by major version byte, each of which contains two digits with first in the upper nibble and the next in the lower nibble. For example, if the complete version is 3.15, then the bytes at offsets 28 and 29 will show (in hexadecimal) as 0x15 0x03 (ret[28] and ret[29] as shown in the following code snippets). The following code shows how to extract and convert the 4 BCD digits into usable forms for later comparison without altering the original command return data (written in C/C++ and easily portable to Python, Java, C#, MATLAB script, etc.).

```
/* "ret" is the array of bytes containing
the 'K' command's return data */
/* define variables */
unsigned char verbyte; /* temp work byte */
int minver, majver, majminver;
float version;
```

Minor version number as an integer (e.g., 15):

```
verbyte = ret[28]; /* get minor ver. digits */
/* get 1's digit & then get & add 10's digit */
minver = (verbyte & 0x0F) +
        ((verbyte >>4 & 0x0F) * 10);
```

Major version number as an integer (e.g., 3):

```
verbyte = ret[29]; /* get major ver. digits */
majver = (verbyte & 0x0F) +
        ((verbyte >>4 & 0x0F) * 10);
```

Complete (thousands) version as an integer (e.g., 315):

```
majminver = majver * 100 + minver;
```

Complete version as a floating-point number (e.g., 3.15):

```
version = majminver * .01;
```

5. MAINTENANCE

5.1 General

The XenoWorks Micromanipulator requires no regular maintenance.

- To avoid limits on the movement of the mechanical, make sure its cables are not taught, and the cables can move freely across the mechanicals range of movement.
- Do not oil the mechanicals.
- If the mechanical must be removed for storage or shipping, use the joystick to move the mechanical so the holes for the shipping screw are properly aligned. Then turn power OFF to the controller and install the shipping screws. If needed, use a screwdriver in the center slot of each motor to fine-tune the alignment of screw holes.

5.2 Troubleshooting

Problem	Possible Cause	Solution
There is excessive vibration at the micropipette tip during movement.	1. The micromanipulator is set on a coarse range of movement setting (“C,” “1,” or “2”).	Change the range of movement setting (Speed) to “3”, “4”, “5” or “6”.
	2. The micropipette is too long.	Use a micropipette that is no longer than about 50mm. Ideally, no more than about 40mm of micropipette should protrude from the front of the chuck. In practice, micropipettes with wide tip diameters (such as holding pipettes) will be less prone to vibration, and so can extend further from the chuck than sharp micropipettes such as those used for intracellular microinjection. Push the micropipette further into the holder if vibration is occurring, but do not seat the back end of the micropipette on the back surface of the holder - pull it back out slightly.
	3. The micropipette has not been pushed far enough into the micropipette holder.	
	4. The O-rings in the micropipette holder chuck are missing or damaged.	The micropipette holder has been designed to minimize vibration when used with three O-rings in the chuck, and when the O-rings are the only thing contacting the back of the micropipette.

Problem	Possible Cause	Solution
There is still excessive vibration at the micropipette tip during movement.	5. The micropipette is dragging on the bottom or sides of the microinjection chamber.	Using fewer than three O-rings (or letting the back of the micropipette touch the back of the holder) may lead to excessive vibration. Replace all missing or damaged O-rings.
	1. The micropipette holder clamp has not been tightened down sufficiently.	If the tip is touching the bottom of the chamber, raise the micropipette slightly by turning the Z-axis control knob. If the shaft of the micropipette is touching the side of the chamber, consider repositioning the micropipette with respect to the micromanipulator as described in Section 4.
	2. There is too great a distance between the micropipette tip and the point where the micropipette holder is gripped by the holder clamp on the micromanipulator.	Tighten the micropipette holder clamp.
	3. The fixing screws of the mounting adapter are not tight.	This is a different problem to the one described in 2 and 3 above. Here the micropipette holder's shaft projects too far from the clamp on the micromanipulator. Shorten the micropipette tip-to-clamp distance by moving the holder outboard and clamping the micropipette holder closer to its tip.
	4. The swing gate thumbscrew is loose.	Tighten the fixing screws.
	5. The mechanical is not screwed securely to the mechanical base plate.	Tighten the swing gate thumbscrew.
	6. The mechanical base plate is not securely attached to the microscope adapter.	Ensure all four screws fixing the mechanical to its base plate are tight.
7. The micromanipulator is at the extreme limit of	Lock the mechanical base plate is to the microscope adapter.	
	7. The micromanipulator is at the extreme limit of	Remove the micropipette holder from the

Problem	Possible Cause	Solution
	one or more of its axes.	micromanipulator. Move the manipulator mechanical to the approximate center of its range of movement. If necessary, adjust the mechanical base plate with respect to the microscope adapter so the microscope objective intersects the center of the micromanipulator's Y-axis. Realign the micropipette holder clamp arrangement (X- and Z-axes), so that when the micropipette holder is replaced in the clamp, the micropipette tip will come as close as possible to the microscope's optical axis. Only after these steps should you use the joystick to fine position the tip of the micropipette in the microscope field of view.
	8. The cable attached to the mechanical is binding.	Release any tension by coiling the cable once on the bench behind, or to the side of the microscope.
	9. The center fitting of the micropipette holder is loose.	Tighten the fitting.
The micropipette has been lost from the field of view.	A too-coarse range of movement setting has been selected, or the joystick resistance is too loose.	Select a finer range setting (Speed) or tighten the joystick friction dial.
One or more motors move in the opposite direction to that commanded by the joystick.	One or more of the axis polarity DIP switches are in the wrong position.	Flip the appropriate DIP switches (see the axis guide in section 2.4). It is not necessary to restart the instrument.
The micropipette will not move any lower.	1. The end of motor travel has been reached.	Set up the micropipette alignment again.
The micropipette will not move any lower.	2. A 'Z-limit' has been set.	Deactivate the 'Z-limit' (press the 'Z-limit' key once briefly. The green LED will go out).
The micropipette appears to move in a direction opposite to that commanded by the joystick, but only in the Y-axis.	1. The Y-axis inversion DIP switch is in the wrong position.	Reverse the Y-axis DIP switch.
The micropipette appears to move in a direction opposite to that commanded by the joystick, but only in the Y-axis.	2. A micropipette with an angled tip has been	Angled micropipettes should be adjusted from the start so

Problem	Possible Cause	Solution
The micropipette does not return to the assigned coordinates memorized by the 'Work' setting.	aligned such that the bend or "heel" of the micropipette is touching the dish, but the tip is not. This is usually only noticeable at higher magnifications.	that they are slightly "toe-down", in other words, the tip should touch the bottom of the microinjection chamber before the bend does.
	1. No new 'Work' setting was assigned, and so the micropipette moved to a different, previously assigned position.	Relocate the micropipette tip and assign a new 'Work' position.
	2. Repeated rapid or violent movements of the joystick may cause the instrument to miss information coming from the joystick to the processing circuitry, and so miscalculate the correct position of 'Work'. Normally, the error is no more than a few tens of microns.	Avoid violent movements of the joystick; they not only confuse the electronics but can also damage the joystick assembly. To ensure that the 'Work' position is always current, save a new one periodically during an experiment.
	3. The joystick was inadvertently moved during the automatic return to the 'Work' position.	Press 'Work' again.
The motors "buzz" abnormally during movement, particularly during movements to/from "work" or "home" and particularly at the ends of axis travel.	There is some resistance preventing the motors from moving smoothly.	Do not touch the mechanical during automatic movement. Ensure that the cable connecting the mechanical to the controller is not under tension – coil the cable a few times on the bench surface behind the microscope.
Manipulator moves vertically, but not along one or the other (or both) of the horizontal axes.	The X- or Y-axis lock (or both) is (are) engaged.	Check the Status Display to see if either axis is locked. If so, use the Function Buttons to unlock the locked axis (axes).

5.3 Transporting the Micromanipulator

When moving the instrument, the entire system should be disassembled and placed in its original packaging. Shipping screws are supplied to prevent inadvertent vibrations from damaging the delicate manipulator mechanical. To install the shipping screws, follow this procedure:

1. With the instrument switched on and the mechanical still installed on the microscope adaptor, place the joystick in the coarse ('C') range setting.
2. Use the joystick to move the mechanical INWARDS, towards the center of the microscope, exposing the outer two screws that fix the mechanical to the base plate.
3. Loosen, remove, and retain these two screws.
4. Use the joystick to move the mechanical to the center of its range of motion, along the X, Y, and Z axes.
5. Without touching the joystick further, switch off the power to the instrument.
6. Carefully unplug the connecting cable from the rear of the mechanical.
7. Remove the mechanical from the adapter plate (remove retaining screws and ensure they are kept in a safe place).
8. Locate the shipping screw holes. There are four through-holes in each axis slide, but only two of the holes will line up with the correct tapped holes on the inner slide surface when the instrument is in the 'Setup' position.
9. Locate the original shipping screws that should have been retained with the original packaging and insert them into the appropriate locations. Do not over tighten the screws - turn them until they are lightly seated in their holes.
10. Place the mechanical in a plastic bag.
11. The mechanical is now ready for transport in the original packaging.
Take care to ensure that the shipping screws are removed before the instrument is operated again (see Section 3 for the correct installation procedure).

(This page intentionally left blank.)

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[®] 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.

^{*} Uniblitz[®] is a registered trademark of Vincent Associates.

(This page intentionally blank.)

APPENDIX B. ACCESSORIES

The following accessories are available for the XenoWorks Micromanipulator system.

Extensions

4-in (10.16-cm) Dovetail Extension (X285204)

Z-axis vertical extension (X285305)

Z-axis horizontal extension (X285310)

Rotating Base

Rotating base (225RBI)

Mounts

Hinged headstage mount (285HEA)

Adapters

Mounting adapter plate (X285210). *For use with MT and MD series stands and platforms, or any surface with 1-inch centered holes.*

Right angle adapter (X285300)

HOLDERS

Rod holder (MP-ROD)

(This page intentionally blank.)

APPENDIX C. TECHNICAL SPECIFICATIONS



Manipulator Mechanical:

3-axis stepper motors:	X, Y, and Z
Resolution:	125 nm/microstep (XWM/M, MP-285/M) 93.75 nm/microstep (MP-845/M)
Travel:	25 mm (1 in) full travel for each of X, Y, and Z axes

Joystick:

- Inverted joystick with integrated declutch.
- Height and resistance adjustable
- Independent axis-polarity inversion
- 8-step range-of-movement control
- Three user-defined set points ('Home', 'Work 1', 'Work 2'),
- Two user-defined axis limits ('X-Limit', 'Z-floor'),
- One preset point ('Origin')
- X-axis lock, Y-axis lock

Maximum Speed:	3.25 mm/sec (3,250 microns/sec) (XWM/M, MP-285/M) 2.5 mm/sec (2,500 microns/sec) (MP-845/M)
----------------	--

Ranges of Movement:

Table C-1. Range speeds.

Range (Speed) Setting	X & Y Axes (1 joystick swing)	Z Axis (1 rotation)
C	13,700 μm	3,200 μm
1	3,560 μm	800 μm
2	950 μm	400 μm
3	432 μm	200 μm
4	216 μm	100 μm
5	108 μm	50 μm
6	100 μm	30 μm
7	50 μm	10 μm

Electrical:

Input voltage (Mains):	100 – 120 VAC, 50/60 Hz, 200 – 240 VAC, 50/60 Hz
------------------------	---

Power consumption: 60W max
Cables (Refer to the following two tables for a description of all possible cables.)

Table C-2. XenoWorks XWM-100 cables and receptacles/connectors.

Controller Rear Panel Port Connector/Receptacle	Cable Connector Types	Connects to ...	Cable Type	Cable Max. Length
(Power entry) 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)
DRIVE MODULE (25-Pin DSUB female receptacle)	←DB-25 male DB-25 female→ (straight-through)	MP-285M (25-Pin DSUB male connector)	Minimum of 26 awg stranded wire with 500 Volt (see note)	
USB USB "A" (Device) female receptacle (full-sized)	←A connector B connector→	Host computer's USB "B" receptacle (full-sized)	Dielectric separation of circuits. Foil shielding.	

Storage Environment:

Temperature: 0 – 70 °C (32 – 158 °F)

Humidity: 0 – 95% (non-condensing)

Operating Environment:

Temperature: 3.5 – 35 °C (38.3 – 95 °F)

Humidity: 0 – 80% (non-condensing; 80% @ 31°C (87.8°F) decreasing linearly to 67% @ 35°C (95°F))

Regulatory:

Safety - EN61010-1, CE
 EMC - EN61326, CE

Cleaning:

70% alcohol (or e.g., UV)

Dimensions (H x W x D):

Mechanical: 112 x 145 x 185 mm (4.4 x 5.7 x 7.3 in)

Joystick: 286 x 235 x 238 mm (11.25 x 9.25 x 9.4 in)

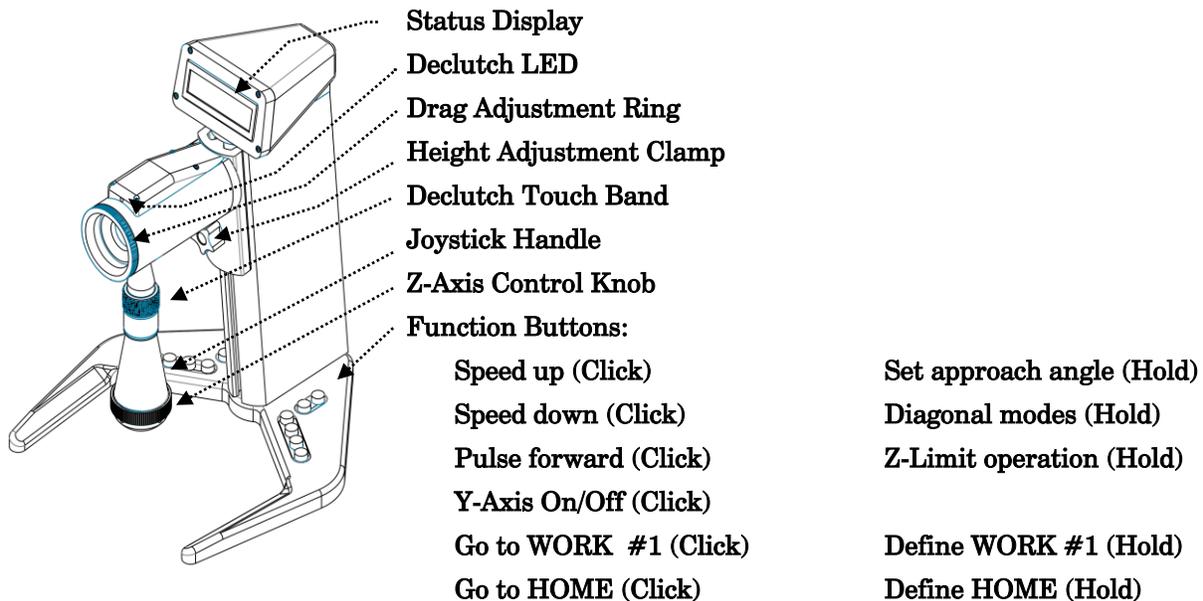
Weight:

Mechanical: 174 g (0.38 lb)

Joystick: 3018 g (6.65 lb)

APPENDIX D. QUICK REFERENCE

Manual Operation



Configuration

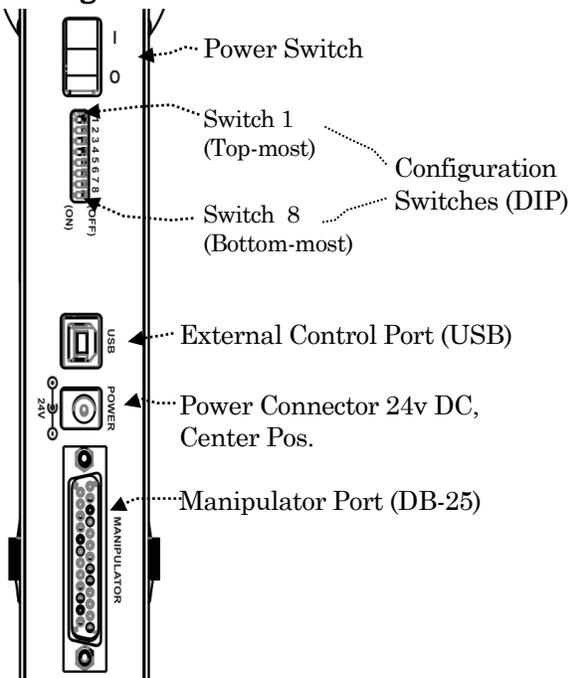


Figure D-1. Rear of the controller.

All the switches are shown in their 'Switch Right' positions. These switches are typically set during installation and are changed only rarely afterwards.

Table D-3. Configuration Switches 1 – 8.

Sw #	Definition	Left (ON)	Right (OFF)
1	X-Axis Polarity	Manipulator on <u>right</u> of preparation	Manipulator on <u>left</u> of preparation
2	Y-Axis Polarity	Forward motion moves <u>towards</u> user (normal)	Forward motion moves <u>away from</u> user
3	Z-Axis Polarity	Clockwise rotations <u>lowers</u> manipulator	Clockwise rotations <u>raises</u> manipulator
4	Electro-mechanical Device Compatibility	XWM/M or MP-285/M (8 micro-steps/micron)	TRIO MP-845/M (10.6666666667 micro-steps/micron)
5	Power-ON auto calibration	Auto calibration OFF	Auto calibration ON
6	Pulse Mode	Ramped-Pulse (<u>slower</u> ; minimizes vibration)	Standard-Pulse (<u>faster</u> ; slightly more vibration)
7	<reserved>	<for future use>	<for future use>
8	Touch-Pulse Enable	Enable	Disable

If the manipulator will be used on the right side of the preparation, make sure DIP switch 1 is pushed to the left (as viewed from the rear). Alternatively, if the manipulator will be used on the left side of the preparation, make sure DIP switch 1 is pushed to the right.

If the manipulator is XWM/M or MP-285/M, make sure DIP switch 4 is pushed to the left. If the manipulator is an MP-845/M, MP-865/M or other member of TRIO family of manipulators, be sure DIP switch 4 is pushed to the right.

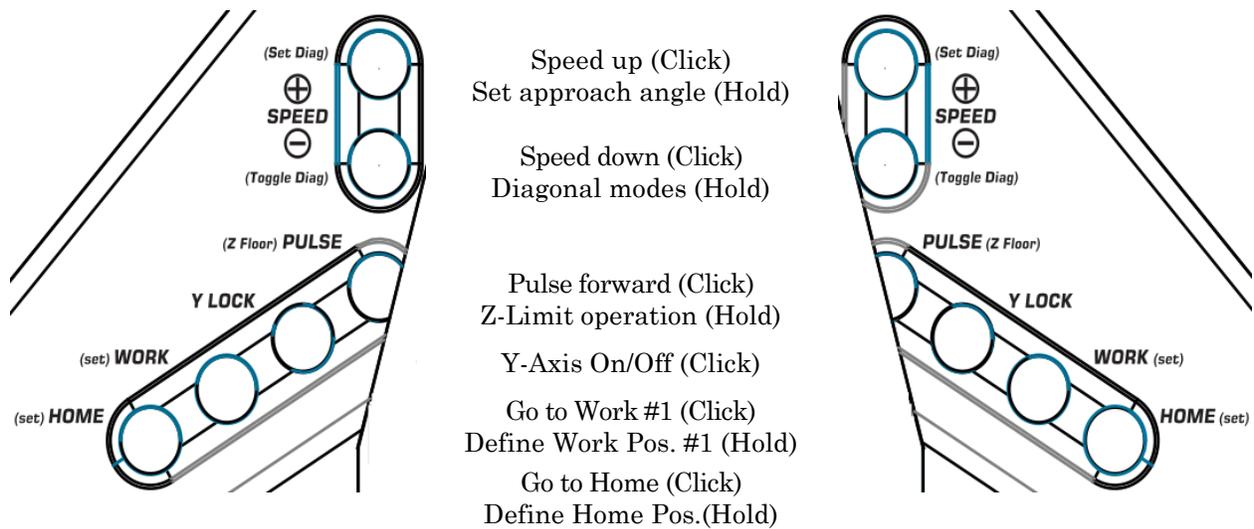


Figure D-2. Function buttons (right side shown) and their primary functions.

Several buttons also provide extra functions when clicked (tapped) or held while touching the declutch. In this way, the declutch can function like the computer shift key.

Figure D-3. Button Functions

Button	Primary Function		Secondary Function	
	Click (Tap)	Hold (3 sec.)	Touch-Click (Touch-Tap)	Touch-Hold (3 sec.)
1	Speed up	Set approach angle	Toggle display mode (?)	
2	Speed down	Diagonal mode		
3	Pulse forward	Z-Limit operation		X-Limit operation (?)
4	Y-Axis On/Off		X-Axis EOT/Limit	
5	Go to Work pos. #1	Define Work pos. #1	Go to Work pos. #2	Define Work pos. #2
6	Go to Home pos.	Define Home pos.	Go to Origin	Setup

Status Display

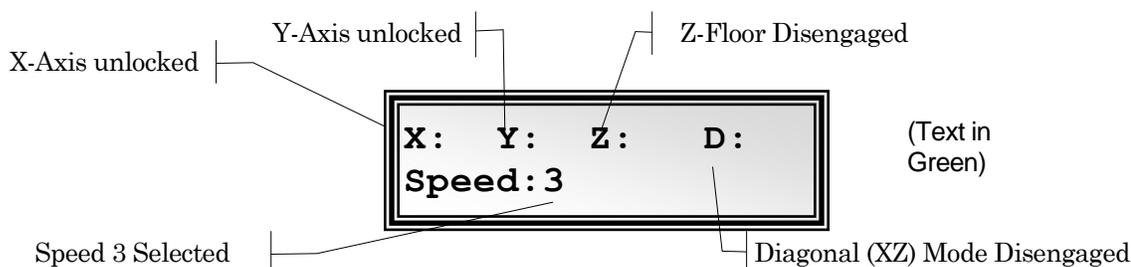


Figure D-4. Reporting of the controller's operating modes.



Figure D-5. Status Display when either the X- or Y-axes are locked at their current positions.



Figure D-6. Status Display when the Diagonal (XZ) Mode is engaged for a right-side manipulator.

The XZ Mode icon will be slightly different for a left-side manipulator.



Figure D-7. Z-Floor function is engaged, with the manipulator at its lower Z-limit (left) and above Z-limit (right).



Figure D-8. Status Display for a right-side manipulator when the X-Limit function is engaged, with the manipulator at its X-Limit (left) and away from the X-limit (right).

The X-Limit icon will be slightly different for a left-side manipulator



Figure D-9. Status Display showing the manipulator's current position.

The above figure illustrates the Status Display when showing the manipulator's current position. Positions are always reported in microns (micrometers; 10^{-6} m), from the origin, where the origin is defined as the point furthest from the preparation along the X- and Z-axes, and closest to the user along the Y-axis.

NOTE: When the display is showing the manipulator's position and a special mode is engaged for one or more axes, the equal signs for the corresponding axes will be replaced by the special-mode icons shown in figures.

External Control

Users can control the XenoWorks XWM-100 system from an external computer by sending commands over a USB interface between the computer and the connector on the rear of the joystick. When using a Windows computer, the external interface for the XenoWorks XWM-100 requires use of Sutter’s CDM (Combined Driver Model) device driver, version 2.12.18 or higher, available from the Technical Support page on Sutter Instrument’s web site (www.sutter.com). The CDM driver for the XenoWorks XWM-100 consists of two components: 1) a USB device driver, and 2) a VCP (Virtual COM Port) device driver. Install the USB driver first, followed by the VCP driver. The VCP driver provides a serial I/O interface between a Windows application and the XenoWorks controller. 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). Using the first, USB Direct, method requires either that the VCP device driver not be installed, or if installed, that it be disabled. The second (Serial) method requires that the VCP be both 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-4. USB-VCP interface serial port settings.

Property	Setting
Data (“Baud”) Rate (bits per second (bps))	9,600
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 sent to the XenoWorks system do not have explicit terminators. All responses from the XenoWorks system end with 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 it is ready to receive a new command. If a response returns other data as well, 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 sent to the XenoWorks system must be an unsigned binary value. Attempting to encode command sequences as “strings” is not advisable. Any data returned from the XenoWorks system should be treated upon reception, as a sequence of unsigned byte values. Groups of contiguous bytes can later be combined to form larger values, as appropriate (e.g., 2 bytes into 16-bit “word”, or 4 bytes into a 32-bit “long” or “double word”). For the XenoWorks BRM, all axis position values (number of micro-steps) 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 micro-steps. Conversion between micro-steps and microns (micrometers) is the responsibility of the software running on the host computer (see *Microns/micro-steps conversion* table for conversion factors).

Micro-steps are stored and reported 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 byte-order reversed after receiving from, or before sending to, the XenoWorks system. 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.

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

Table D-5. Microns/micro-steps conversion

System/Device	From/To Units	Conversion Factor (multiplier)
XenoWorks XWM/M MP-285/M *	$\mu\text{steps} \rightarrow \mu\text{m}$	0.125
	$\mu\text{m} \rightarrow \mu\text{steps}$	8
MP-845/M **	$\mu\text{steps} \rightarrow \mu\text{m}$	0.09375
	$\mu\text{m} \rightarrow \mu\text{steps}$	10.6666666667

* Same applies to BRM/M and MP-865 micromanipulator, 3DMS or MT-78 stage, and MOM or SOM objective mover.

** Same applies to MP-845U/M (or MP-245[S]/M (discontinued) with DB25/DB26HD adapter) micromanipulator.

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 micro-steps, type the result as a 32-bit “long”, “signed long”, or “I32” integer. When converting to microns, type the

result as “double” (64-bit double-precision floating-point values).

Table D-6. Ranges and bounds.

Device	Axes	Milli-meters	Microns	Micro-steps
XenoWorks XWM/M MP-285/M	X,Y,Z	25	0 – 25,000	0 – 200,000
MP-845/M	X,Y,Z	25	0 – 25,000	0 – 266,667

Range Speed: The following table shows the range speeds.

Table D-7. Range (speed) settings.

Range (Speed) Setting	X & Y Axes (1 joystick swing)	Z Axis (1 rotation)
C	13,700 μm	3,200 μm
1	3,560 μm	800 μm
2	950 μm	400 μm
3	432 μm	200 μm
4	216 μm	100 μm
5	108 μm	50 μm
6	100 μm	30 μm
7	50 μm	10 μm

Travel Speed: The following table shows the travel speeds for single-, double-, and triple-axis movements for supported devices using orthogonal move commands.

Table D-8. Travel speeds.

Device	mm/sec or $\mu\text{m}/\text{ms}$		
	Single Axis	Dual Axis (x 1.4)	Triple Axis (x 1.7)
XenoWorks XWM/M MP-285/M	3.0	4.2	5.1
MP-845/M	2.5	3.5	4.2

Command Reference: The following table lists the external-control commands for the XenoWorks XWM-100 Controller.

Table D-9. XenoWorks XWM-100 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 Name & Firmware Version (‘K’) (FW <2)	Tx	All	1	0	75	4B	0100 1011	0075		K	Command
	Rx	<2	34	0-29	A sequence of ASCII characters describing the name of the product: “Sutter Inst. XenoWorks XWM-100”						
				30	Build version number coded in BCD (e.g., if ver. = 3.15.16, then byte 30 = 0x16; (upper nibble = 1; lower nibble = 6)						
				31	Minor version number coded in BCD (e.g., if ver. = 3.15.16, then byte 31 = 0x15; upper nibble = 1; lower nibble = 5)						
				32	Major version number coded in BCD (e.g., if ver. = 3.15.16, then byte 32 = 0x03; upper nibble = 0; lower nibble = 3)						
			33	13	0D	0000 1101		^M	<CR>	Completion indicator	
Get Name & Firmware Version (‘K’) (FW ≥2)	Tx	All	1	0	75	4B	0100 1011	0075		K	Command
	Rx	≥2	31	0-27	A sequence of ASCII characters describing the name of the product: “Sutter XenoWorks XWM-100 ”						
				28	Minor version number coded in BCD (e.g., if ver. = 3.15, then byte = 0x15 (upper nibble = 1; lower nibble = 5))						
				29	Major version number coded in BCD (e.g., if ver. = 3.15, then byte = 0x03 (upper nibble = 0; lower nibble = 3))						
			30	13	0D	0000 1101		^M	<CR>	Completion indicator	
Get Current Position (‘C’) (FW <2)	Tx	All	1	0	67	43	01000 011	0067		C	Command (see note)
	Rx	<2	17	Three 4-byte (32-bit) values (current positions in μsteps of X, Y, & Z) + one 2-byte value of approach angle + one 2-byte value of resolution + 1 byte for completion indicator (see <i>Ranges</i> table for minimum/maximum values)							
				0-3 (4)	X pos. in μsteps						
				4-7 (4)	Y pos. in μsteps						
				8-11 (4)	Z pos. in μsteps						
				12-13 (2)	Approach angle in degrees [1,45]						
				14-15 (2)	Internal resolution						
			16	13	0D	0000 1101		^M	<CR>	Completion indicator	
Get Current Position (‘C’) (FW ≥2)	Tx	All	1	0	67	43	01000 011	0067		C	Command (see note)
	Rx	≥2	13	Three 4-byte (32-bit) values (current positions in μsteps of X, Y, & Z) + 1 byte for completion indicator (see <i>Ranges</i> table for minimum/maximum values)							
				0-3 (4)	X pos. in μsteps						
				4-7 (4)	Y pos. in μsteps						
				8-11 (4)	Z pos. in μsteps						
			12	13	0D	0000 1101		^M	<CR>	Completion indicator	
Toggle XZ Mode (Diagonal/ Orthogonal) (‘D’)	Tx	All	1	0	68	44	0100 0100			D	Command
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.

Command	Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description
					Dec.	Hex.	Binary				
Set Diagonal Approach Angle (‘A’) (FW <2)	Tx	All	3	0	65	41	0100 0001	0065		A	Command
		<2	1-2 (2)								Approach angle in degrees (1 – 45 (unsigned short (2 bytes), Little- Endian))
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.
Set Diagonal Approach Angle (‘A’) (FW ≥2)	Tx	All	3	0	65	41	0100 0001	0065		A	Command
		≥2	1	1 – 45	01 – 2D	0000 0001 – 0010 1101	0001 – 0045				Approach angle in degrees (1 – 45 (1 byte))
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.
Perform XZ Pulse (‘P’)	Tx	All	1	0	80	50	0101 0000	0080		P	Command
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.
Move to Specified Position at Full Speed (‘M’)	Tx	All	13	0	77	4D	0100 1101	0077		M	Moves X, Y, and Z to specified position (stereotypic at fastest speed)
				X, Y, & Z target positions, in micro-steps from the origin, each consisting of 4 contiguous bytes representing a single 32-bit, positive, integer (see <i>Ranges</i> table).							
				1-4 (4)	X μsteps						
				5-8 (4)	Y μsteps						
				9-12 (4)	Z μsteps						
	≥30ms	Delay between send and next command or timeout. To be calculated based on distance of travel.									
Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.	
Move to Origin (‘O’)	Tx	All	1	0	79	4F	0100 1111	0079		O	Command
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.
Move to Home Position (‘H’)	Tx	All	1	0	72	48	0100 1000	0072		H	Command
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.
Move to Work Position* (‘Y’)	Tx	All	1	0	89	59	0101 1001	0089		Y	Command
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.
Move to Center Position (‘N’)	Tx	All	2	0	78	4E	0100 1110	0078		N	Command
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.

* The “Home” and “Work” positions are defined manually, using the function buttons on the XenoWorks XWM-100 Joystick.

Command	Tx/ Delay/ Rx	Ver.	Total Bytes	Byte Offset (Len.)	Value			Alt- key- pad #	Ctrl- char	ASCII def./- char.	Description		
					Dec.	Hex.	Binary						
Get Approach Angle (‘a’)	Tx	≥2	1	0	97	61	0110 0001	0097		a	Command		
	Rx	≥2	2	0							Approach angle in degrees		
				1	13	0D	0000 1101		^M	<CR>	Task-completion ind.		
Report Internal Resolution (‘R’)	Tx	≥2	1	0	97	61	0110 0001	0097		a	Command		
	Rx	≥2	1	0-1							Integer flag indicating resolution: 8000 or 10667 (unsigned short (2 bytes), Little- Endian)		
				2	13	0D	0000 1101		^M	<CR>	Task-completion ind.		
Move to Destination at Selected Speed (‘m’)	Tx		14	0	109	6D	0110 1101	0109		m	Command		
				1	0	00	0000 0000	0000			Selected speed: 0 through 7		
					-	-	-	-					
					7	07	0000 0111	0007					
				X, Y, & Z target positions, in micro-steps from the origin, each consisting of 4 contiguous bytes representing a single 32-bit, positive, integer (see <i>Ranges</i> table).									
				2-5 (4)	X μsteps								
				6-9 (4)	Y μsteps								
	10-13 (4)	Z μsteps											
30ms	Delay between send and next command or timeout. To be calculated based on distance of travel and velocity.												
Rx	=>2	1	2	13	0D	0000 1101			^M	<CR>	Task-completion ind.		
Interrupt Move (^C)	Tx	All	2	0	3	03	0000 0111	0003	^C	<ETX>	Command		
	Rx	All	1	0	13	0D	0000 1101		^M	<CR>	Task-completion ind.		

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 the I/O Send & Receive Buffers:** Clearing (purging) the transmit and receive buffers of the I/O port immediately before sending any command is recommended. Note that this clearing of the buffers affects only the computer-side I/O; it does not (necessarily) clear the buffers on the controller side, requiring, when necessary, to reset/power-cycle the controller. Following the rules described will generally avoid problems with getting garbage data in the I/O buffers of both the computer and controller (i.e., using exact number of bytes for both command sequences and

return data (as per the *Commands* table), never sending a command before the previous command is finished with its task, etc.).

- Positions in Micro-steps and Microns:** All positions sent to and received from the controller are in micro-steps (μsteps). See Microns/micro-steps conversion table for conversion between μsteps and microns (micrometers (μm)).

```
Declaring position variables in C/C++:
/* current position for X, Y, & Z */
unsigned long cp_x_us, cp_y_us, cp_z_us;
/* micro-steps */
double cp_x_um, cp_y_um, cp_z_um;
/* microns */
/* specified (move-to) position for X, Y, &
Z */
unsigned long sp_x_us, sp_y_us, sp_z_us;
/* micro-steps */
double sp_x_um, sp_y_um, sp_z_um;
/* microns */
Use the same convention for other position
variables the application might need.
```

```

Declaring the micro-steps/microns
conversion factors in C/C++:
/* conversion factors for the Xenoworks
XWM/M and MP-285/M based config. */
double us2umCF = 0.125;          /* micro-
steps to microns */
double um2usCF = 8;             /* microns
to micro-steps */
/* conversion factors for the MP-845[S]/M
based config. */
double us2umCF = 0.09375;       /* micro-
steps to microns */
double um2usCF = 10.6666666667; /* microns
to micro-steps */

Converting between micro-steps and microns
in C/C++:
/* converting X axis current position */
cp_x_um = cp_x_us * us2umCF; /* micro-
steps to microns */
cp_x_us = cp_x_um * um2usCF; /* microns to
micro-steps */
Do the same for Y and Z, and for any other
position sets used in the application.

```

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

```

Declaring minimum and maximum absolute position
variables in C/C++:
/* minimum and maximum positions for X, Y, & Z
*/
double min_x_um, min_y_um, min_z_um; /*
minimum microns */
double max_x_um, max_y_um, max_z_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;
/* initialize all maximum positions in
microns*/
/* Xenoworks XWM/M, MP-285/M, MP-845[S]/M,
etc. */
max_x_um = 25000;
max_y_um = 25000;
max_z_um = 25000;

```

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 the end of travel (EOT)) are positive values; there are no positions ‘lower’ than the origin, 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 */
double rp_x_um, rp_y_um, rp_z_um; /* microns
*/
/* initialize all relative positions to 0
after declaring them */
rp_x_um = rp_y_um = rp_z_um = 0;

```

Enter any positive or negative value for each relative position (e.g., $rp_x_um = 1000$; $rp_y_um = 500$; $rp_z_um = -200$... 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 positized 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 micro-steps 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 micro-steps */
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 micro-steps and consist of 32-bit integer values (four contiguous bytes). Position values can be either positive or negative, so type must be “signed”. 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 a signed integer (“long” or “signed long” in C/C++; “I32” 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 micro-steps 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. **Orthogonal Move Speed:** Full speed for the “Orthogonal Move” (‘M’) command is 5000 microns/sec. (5 mm/sec. or microns/millisecond) for single-axis movements (XenoWorks XWM/M or MP-285/M) and 3000 microns/sec. (3 mm/sec. or $\mu\text{m}/\text{ms}$) for TRIO MP-845/M).
12. **Multi Axis Movement Speed Increase:** Specified travel speeds are for single-axis movements. When travel traverses a 45° diagonal within a dual-axis square, speed is increased by 40% (x 1.4), and by 70% (x 1.7) within a triple-axis cube.
13. **Move Interruption:** A command should be sent to the controller only after the task of any previous command is complete (i.e., the task-completion terminator (CR) is returned). One exception is the “Interrupt Move” (^C) command, which can be issued while a command-initiated move is still in progress.
14. **Extracting the XenoWorks XWM-100 Firmware Version Number (Firmware <2):** The firmware version number returned by the ‘K’ command is encoded in BCD (Binary Coded Decimal) in three bytes, with build version byte first, followed by minor version byte and major version byte, each of which contains a two-digit pair, the first digit of which is in the upper nibble and the second in the lower nibble. For example, if the complete version is 3.15.16, then the bytes at offsets 30, 31, and 32 will contain (in hexadecimal) as 0x16, 0x15, and 0x03 (ret[30], ret[31], and ret[32] below). The following snippet of code shows how to extract and convert the 6 BCD digits into usable forms for later comparison without altering the original return data (written in C/C++ but is easily portable to Python, Java, C#, MATLAB script, etc.).


```
/* Assume "ret" is the array of bytes
containing the 'K' command's return data */
unsigned char verbyte; /* temp work byte */
int buildver, minver, majver, majminver;
float version;
```

Build version number as an integer (e.g., 16):

```
verbyte = ret[30]; /* build ver. digits */
buildver = (verbyte & 0x0F) +
((verbyte >>4 & 0x0F) * 10);
```

Minor version number as an integer (e.g., 15):

```
verbyte = ret[31]; /* minor ver. digits */
/* get 1's digit & then get & add 10's
digit */
minver = (verbyte & 0x0F) +
((verbyte >>4 & 0x0F) * 10);
```

Major version number as an integer (e.g., 3):

```
verbyte = ret[32]; /* major ver. digits */
majver = (verbyte & 0x0F) +
((verbyte >>4 & 0x0F) * 10);
```

Complete version as an integer (e.g., 31516):

```
majminbldver = majver * 10000 + minver *
100 + buildver;
```

Complete version as a floating-point number (e.g., 3.1516):

```
version = majminbldver * .0001;
```

Extracting the XenoWorks XWM-100 Firmware Version Number (Firmware ≥2): The firmware version number returned by the ‘K’ command is encoded in BCD (Binary Coded Decimal) in two bytes, with minor version byte first, followed by major version byte, each of which contains two digits with first in the upper nibble and the next in the lower nibble. For example, if the complete version is 3.15, then the bytes at offsets 28 and 29 will show (in hexadecimal) as 0x15 0x03 (ret[28] and ret[29] as shown in the following code snippets). The following code shows how to extract and convert the 4 BCD digits into usable forms for later comparison without altering the original command return data (written in C/C++ and easily portable to Python, Java, C#, MATLAB script, etc.).

```
/* "ret" is the array of bytes containing
the 'K' command's return data */
/* define variables */
unsigned char verbyte; /* temp work byte */
int minver, majver, majminver;
float version;
```

Minor version number as an integer (e.g., 15):

```
verbyte = ret[28]; /* get minor ver. digits */
/* get 1's digit & then get & add 10's digit
*/
minver = (verbyte & 0x0F) +
((verbyte >>4 & 0x0F) * 10);
```

Major version number as an integer (e.g., 3):

```
verbyte = ret[29]; /* get major ver. digits
*/
majver = (verbyte & 0x0F) +
((verbyte >>4 & 0x0F) * 10);
```

Complete (thousands) version as an integer (e.g., 315):

```
majminver = majver * 100 + minver;
```

Complete version as a floating-point number (e.g., 3.15):

```
version = majminver * .01;
```

NOTES:

INDEX

- A**
- accessories 59
 - adapters** 59
 - extensions** 59
 - holders** 59
 - mounts** 59
 - rotating base** 59
- C**
- Centering the Micropipette with Respect to the Joystick 28
 - Changing the Working Range of Movement
 - Setting a ‘Z-Floor’ 32
 - Changing the Working Range of Movement 32
 - Cleaning the Micropipette 33
 - commands notes 46
 - configuration** 63
 - controller
 - cable specs 62
- D**
- disclaimer iii
- E**
- electrical
 - specifications 61
 - external control 37, 66
 - absolute positioning system origin 47, 71
 - absolute vs. relative positioning 48, 71
 - axis position command parameters 38, 66
 - clearing send/receive buffers 46, 70
 - command reference 67
 - command sequence formatting 37, 66
 - extracting the firmware version number 49, 50, 72
 - intercommand delay 46, 70
 - Interrupt move (‘^C’) command 46
 - micro-steps and microns (micrometers) .. 38, 42, 67
 - move interruption 49, 72
 - multi axis movement speed increase 49, 72
 - orthogonal move speed 49, 72
 - position value bit ordering 48, 71
 - position value typing 48, 71
 - positions in micro-steps and microns 47, 70
 - protocol and handshaking 37, 66
 - range speed 39, 67
 - ranges and bounds 47, 71
 - Setting Diagonal Approach Angle (‘A’)
 - command 43
 - task-complete indicator 46, 70
 - travel lengths and durations 49, 72
 - travel speed 67
 - Virtual COM Port (VCP) serial port settings 37, 66
 - External control
 - command notes 46
 - external control commands
 - ‘^C’ command 46
 - ‘a’ command 45
 - ‘A’ command 43
 - ‘C’ command 41
 - ‘D’ command 42
 - ‘H’ command 44
 - ‘K’ command 39
 - ‘m’ command 45
 - ‘M’ command 43
 - ‘N’ command 44
 - ‘O’ command 44
 - ‘P’ command 43
 - ‘R’ command 45
 - ‘Y’ command 44
 - Get Current Position 41
 - Get Name and Firmware Version 39
 - Move to origin 44
 - Move to specified position at full speed 43
 - Move to specified position at selected speed... 45
 - Perform XZ Pulse 43
 - Toggle XY Mode (Diagonal/Orthogonal) 42
 - external control commands Get approach angle 45
 - external control commands Report internal resolution 45
- F**
- Frequently used Functions and Features
 - ‘Home’ Position 19
 - ‘Origin’ Position 19
 - ‘Work’ Position 19
 - Frequently-used Functions and Features 20
 - Height Adjustment 15
 - Joystick Declutch Mechanism 20
 - Joystick Friction 15
 - Micropipette Holder Clamp 16
 - Pulse 20
 - Speed 19
 - Swing Gate 16
 - Z-floor 20
- G**
- General Information 13

about this manual.....	13
technical support	13
General Safety Information.....	iii
glassware	
precautions	v
I	
Infrequently-used Functions and Features	
Axis Polarity.....	24
Y-Axis On/Off.....	19
Installation Procedure	23
Instrument Controls and Features.....	14
X- and Y-axis control	18
Z-axis control.....	18
M	
mains	
power consumption	62
voltage.....	61
Maintenance.....	51
general	51
transporting the micromanipulator.....	54
troubleshooting	51
manual operation	63
Memorizing the ‘Home’ Position	31
Memorizing the ‘Work’ Position	31
mounts.....	59
Moving the Micropipette with the Joystick	28
N	
notes	
user.....	75, 76
O	
Operating Instructions.....	28
P	
Packing List	13
power	
consumption.....	62
Product Description	13
R	
resolution.....	61

S

safety warnings.....	iii
safety warnings & precautions.....	iii
operational.....	iii
Setting the Angle of Approach of the Micropipette	
Applications and Suggested Setups	29
Microinjection chamber	29
Micropipette type.....	31
Piezo impact drive	31
Working distance	31
Setting the Angle of Attack of the Micropipette	28
specifications	
cleaning.....	62
dimensions.....	62
operating environment	62
regulatory	62
weight	62
specifications	61
electrical.....	61
joystick	61
manipulator mechanical	61
storage environment	62

T

technical specifications.....	61
technical support	13
transporting the micromanipulator.....	54
travel	61
Troubleshooting	51

U

User Profiles.....	20
using the ‘Pulse’ function	32

V

voltage	
mains.....	61

W

warranty	57
----------------	----

X

X-Limit	20
---------------	----

NOTES

NOTES