NANO-DRIVE®C OPERATION MANUAL

MAD CITY LABS INC.

2524 Todd Drive, Madison, WI 53713

Tel: (608) 298-0855   Fax: (608) 298-9525

IMPORTANT SAFETY INFORMATION

Hazardous voltages and currents present in controller and stage.
Risk of electrocution and death.
Do not open controller enclosure or stage body. There are no user serviceable parts.

Piezoactuators have large capacitance and are capable of storing hazardous amounts of electrical energy over long periods of time. Various conditions such as load and temperature changes can also cause piezoactuators to accumulate charge.

It is recommended that the input command voltage be set to zero and turn off the Nano-Drive®C before disconnecting the nanopositioner from the controller.

The Nano-Drive®C has no user serviceable parts. Only trained service personnel should perform service.

IMPORTANT

All Technical Information, recommendations, and examples related to Mad City Labs, Inc. products made in this manual are based on information believed to be correct. The purchaser or user should determine the suitability of each product before using. The purchaser or user assumes all risks and liability whatsoever in connection with the use of any and all Mad City Labs, Inc. products or services.

CONTENTS

5 TROUBLE SHOOTING

1 INTRODUCTION

The Nano-Drive®C is the compact controller required for operating compatible Mad City Labs, Inc. nanopositioning systems. The Nano-Drive®C includes a low noise, low drift 165 V driver, PicoQ® position sensing electronics, and closed loop proportional-integral feedback control. The driver is capable of supplying 50 mA at 165 V. The closed loop position command may be supplied through the front panel analog interface or, alternatively, by the Universal Serial Bus (USB) digital interface.

Each Nano-Drive®C has been factory adjusted for its complementary nanopositioning device. This adjustment includes setting the gain, bandwidth, and offset of the position sensing electronics. Under stable environmental conditions readjustment of these settings is not necessary. Each nanopositioning stage and each axis on the Nano-Drive®C are clearly labeled for identification.

OUTPUT VOLTAGE RANGE	-20 to 165 V
OUTPUT CURRENT	50 mA
ANALOG INTERFACE * (INPUT BNC)	0.0 to +10.0V
FRONT PANEL ACCESSIBLE OUTPUT SIGNALS Output Voltage ÷ 10 (HV/10) Position Sensor Signal	 -2.0 to 16.5V 0.0 to 10.0 V
FRONT PANEL ADJUSTMENTS	SENSOR OFFSET
CONNECTOR TYPE	DB-9
POWER REQUIREMENT	12V/5.17A
16 BIT DIGITAL INTERFACE *	USB 2.0

TABLE.1 NANO-DRIVE®C SPECIFICATIONS

*The command signal is the sum of the digital and analog interface inputs.

A block diagram of the Nano-Drive®C is shown in Fig.1. The command signal is used to control the displacement of the nanopositioning device. This command signal can be supplied by either the front panel BNC analog input or by the internal DAC of the USB interface. A command signal of minimum to maximum voltage values corresponds to a stage displacement of zero to the maximum value. If the USB interface is to be used alone, it is recommended that the front panel BNC be grounded.

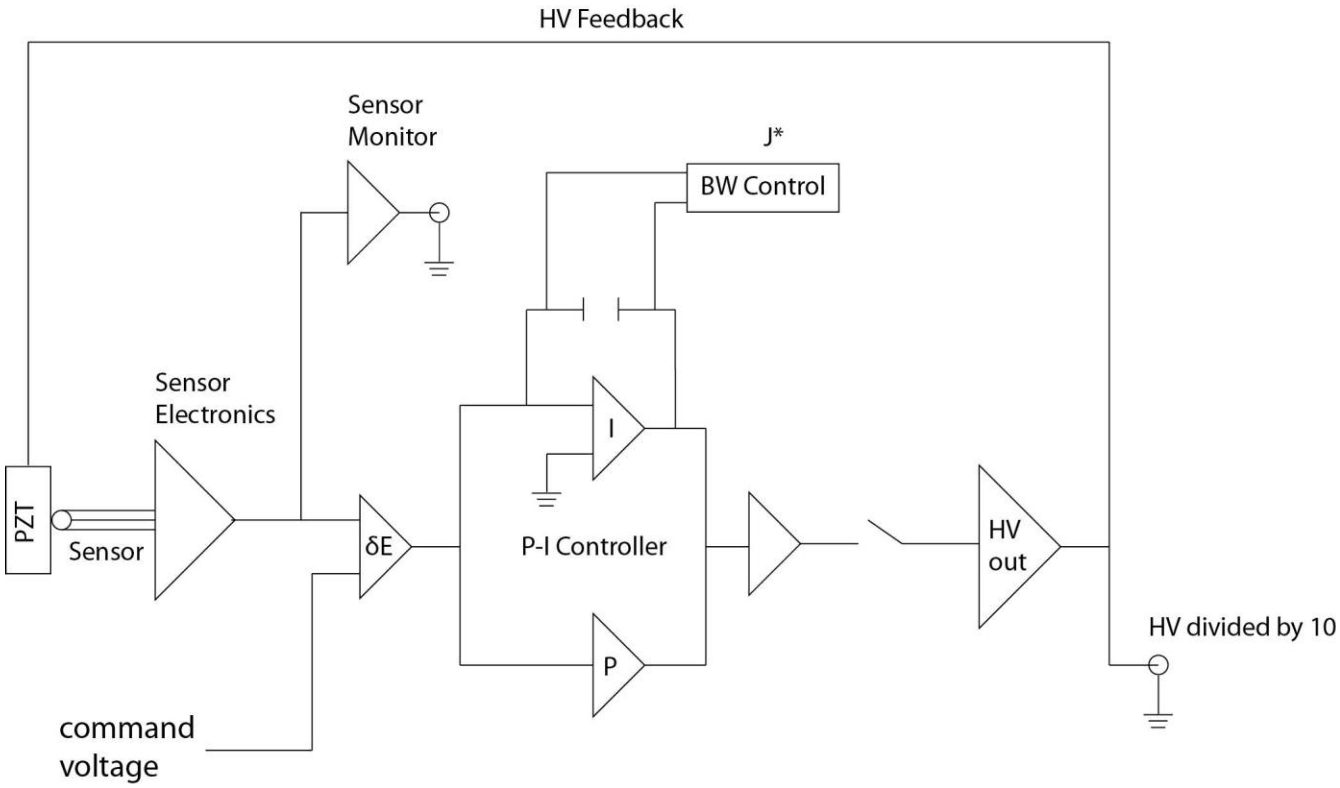

Fig.1 Block diagram of Nano-Drive®

The command signal (cmd) is compared to the sensor signal and an error signal δE is generated. The error signal is passed on to the proportional-integral (PI) controller. The stage displacement is held at the value corresponding to the command signal, eliminating the effects of creep and hysteresis.

1.1 Front panel connections

Fig.2 Nano-Drive®C Controller Front Panel

The front panel of the Nano-Drive®C controller is shown in Fig.2. The front panel connections are:

<u>Input</u>	Analog input for command voltage (0V to 10V)
<u>Offset</u>	adjustment for position sensing circuit
<u>DB-9</u>	9-Pin D-Type connector to nanopositioning stage
<u>Push button</u>	Momentary switch to alternate between the sensor output (default) signal and the HV/10 on the monitor BNC. The switch must be kept depressed when reading the HV/10 signal.

- Sensor output
  0.0 to 10.0 V buffered output on a BNC connector*
- <u>HV/10</u>
  High voltage ÷10 (-2.0V to 16.5V) output on a BNC connector

*The sensor output signal of 0.0 to 10.0 V corresponds to a stage displacement of zero to the maximum displacement value.

1.2 Rear panel connections

Fig.3 Nano-Drive®C Controller Rear Panel

The rear panel connections for the Nano-Drive® controller are shown in Fig.3.

DC-Power	12V @ 5.17A
USB	USB Connector (Blue ready LED to left)

The DC power cable connector locks to the enclosure connector. To remove the power cable, grasp the rectangular housing of the cable and pull away from the enclosure connector. Do not pull the cable directly as this will not release the locking mechanism.

1.3 Recommended operating practice

Irrespective of what configuration of the Nano-Drive®C controller you own, we recommend that you establish a known default setting for your system. For most users we recommend setting the analog input voltages to 0V prior to disconnecting the Nano-Drive®C controller or during extended periods of inactivity. The internal DAC of the USB interface defaults to the zero displacement value at start up, position values are not retained once power is disconnected to the Nano-Drive®C controller. It is recommended that users establish a default setting that is relevant to their needs.

2 INSTALLATION

Prior to commencing installation of your Nano-Drive®C controller, be certain to install your nanopositioning stage as described in its **“Installation and Operation Manual”**. Each DB-9 connector from the stage is clearly identified for a specific axis. It is important that each stage axis is connected to the corresponding Nano-Drive®C axis.

Hazardous voltages and currents present in controller and stage.
Risk of electrocution and death.
Do not open controller enclosure or stage body. There are no user serviceable parts.

2.1 Initial Installation

a) Run the installation program on the Mad City Labs Installation disk that came with your nanopositioner. The USB driver must be installed on your host computer before connecting the Nano-Drive® to the host computer. Your host computer must be using either Windows Vista, 7, 8, or 10. The installation program also installs the manuals and example software for interfacing to the Nano-Drive®C controller.

b) Install the Nanopositioning stage as described in its **“INSTALLATION AND OPERATION MANUAL”**.

c) Be certain the Nano-Drive®C power is disconnected on the rear panel.

d) Connect the DB-9 connector of the nanopositioning stage to the Nano-Drive®C and tighten down the screws.

e) Ensure that there is no voltage to the analog command (BNC). The simplest approach is to leave it disconnected from any external voltage source. The USB internal DAC will produce the only available voltage command (default is zero displacement).

f) Confirm that you have run the software installation disk and USB driver.

g) Connect power supply to the rear panel of the Nano-Drive®C.

h) The USB “ready” light should come on indicating that the USB firmware has been loaded from the onboard EEPROM into the onboard microcontroller.

i) The Nano-Drive®C may now be connected to your host computer with the provided USB cable (standard USB cables may also be used).

j) Check the sensor offset value. Section 2.2 describes the normal operating value and procedure for adjustment.

k) The system is now ready for operation.

l) Never disconnect the DB-9 connector(s) while the power is on.

2.2 Sensor Offset Adjustment

******<u>IMPORTANT</u>******

Before commencing data collection or measurement it is important to check the sensor offset value. We recommend that this be an integral part of your initialization procedure prior to data collection and whenever circumstances have changed in your instrumentation arrangement (e.g. temperature changes, fixture changes etc.).

******<u>IMPORTANT</u>******

The offset of the position sensing electronics have been factory adjusted. It is important to connect the correct nanopositioning stage to each Nano-Drive®C. The nanopositioning stages and Nano-Drive®C have matching serial numbers.

In closed loop mode the high voltage output operates between (approximately) -10.0 V and 155.0V. The 10V overhead at both ends is necessary for the PI controller to work. The PI controller uses this voltage to compensate for minor temperature and environmental changes as well as creep and hysteresis.

A change in room temperature by a few degrees centigrade will shift the offset of the high voltage output. An increase in temperature may shift the output range from -10.0V-155.0V to -5.0V-160.0V. Under such circumstances no adjustment is necessary since the PI circuit has enough overhead to compensate for creep and hysteresis. Typically, environmental changes such as temperature effect only the offset and not the total range.

IT IS ALWAYS BEST TO CONTROL THE ENVIRONMENTAL CONDITIONS RATHER THAN CONTINUOUSLY ADJUSTING THE SENSING ELECTRONICS. USE THE NANOPOSITIONING SYSTEM IN A TEMPERATURE CONTROLLED ROOM AND MINIMIZE THE NUMBER OF TIMES THE STAGES ARE REINSTALLED.

Once the new nanopositioning stage is installed it is advisable to adjust the offset of the position sensing circuit. Once adjusted, no further adjustment should be necessary unless the stage is reinstalled or the temperature changes by more than 5°C.

Use the following procedure to set/adjust the offset:

a) With the nanopositioning stage installed and connected to the Nano-Drive® and the Nano-Drive®C on, monitor the HV/10 output signal using a calibrated DVM. The push switch must be kept depressed to read the HV/10 output signal from the monitor output BNC on the front panel.

b) Ensure that there is no voltage to the analog command (BNC). The simplest approach is to leave it disconnected from any external voltage source. The USB internal DAC will produce the only available voltage command (default is zero displacement).

c) If the HV/10 value is between -0.5V and -1.0V then no further adjustment is necessary. If the HV/10 value is not in the desired range then proceed to step (d). *Please note that HV/10 can be up to 16.5V.*

d) Using an insulated screw driver, adjust the front panel offset screw so that the HV/10 output signal is between -0.5V and -1.0V. Remember to that the push switch must be depressed to read the HV/10 output signal from the monitor output BNC. Adjust very slowly while monitoring the signal. Clockwise increases the voltage.

3 OPERATION USING THE ANALOG INTERFACE

The Nano-Drive®C controller provides an analog interface via a front panel input BNC connector and a USB digital interface. The matching nanopositioning system can be controlled using (a) analog input only, (b) USB digital commands only, and (c) combination of analog and digital commands. Irrespective of which mode of operation you choose it is important to note that the analog input command voltage is summed with the digital command voltage supplied via the USB interface.

Section 3.1 describes how to operate the Nano-Drive®C using analog commands. Section 4 describes operating a USB enabled Nano-Drive®C using digital commands.

When not in use it is advisable to leave the Nano-Drive®C on with the command signal voltage at 0V or your chosen default value. Never disconnect the DB-9 connector when the Nano-Drive®C is on.

3.1 Operating the USB enabled Nano-Drive®C via analog commands

Nano-Drive®C controllers with the USB interface can be operated via (1) analog interface only, (2) USB digital interface only or, (3) analog and digital commands. Operating the Nano-Drive®C via the USB interface only is covered in sections 4 (USB 16 bit). Irrespective of which mode of operation you choose it is important to note that the analog input command voltage is summed with the digital command voltage supplied via the USB interface.

For analog commands an input signal, sourced from a stable power supply or digital to analog converter (DAC) on a data acquisition card, is supplied to the input BNC connector on the front panel. The input range is 0.0V to 10.00V corresponding to a zero to maximum stage displacement. Nanopositioning stage movement is determined by the analog input voltage. Do not exceed the maximum value on the analog input and do not go below the minimum voltage.

The analog inputs, when not connected to an external voltage source, are held at 0V. The USB interface automatically defaults to the zero position at startup. The default start up position is the zero position for the stage.

When not in use it is advisable to leave the command signal voltage at the minimum (0V) and the Nano-Drive®C on. Never disconnect the DB-9 connector when the Nano-Drive®C is on.

4 OPERATION USING THE USB INTERFACE

The USB interface in the Nano-Drive®C controller allows the user to digitally control the position via user generated digital commands or the supplied LabView based Nano-Route®3D software. The position of each axis is controlled independently by a dedicated DAC and read independently by a dedicated ADC. Communication with the host PC is controlled by the installed USB driver, and the driver is accessed by function calls to the MADLib.dll library. The position, in micrometers, of the nanopositioner can be changed and monitored under software control by using the supplied dynamically linked library (.dll).

Product	***ADC & DAC Rates***
1-Axis	5ms/point to 33µs/point (in 1µs steps) **(33µs/point is the default rate)**

Table 2

4.1 Introduction to MADLib.dll

Windows allows programs to link to function calls in either static or dynamic libraries. For a program to link to a static library, it must be incorporated when the program is compiled, while for a dynamically linked library, the program is linked to the library when it is run. Programs written in LabView or Visual Studio (C++, C#, VB etc.) can make function calls to dynamically linked libraries such as MADLib.dll. In LabView, these function calls to an external DLL are referred to as “call library functions”. The user may wish to study the many LabView examples provided and read the LabView documentation to get a better understanding of how to used “call library functions”. In order to view the LabView examples you must have LabView installed (not supplied).

The USB driver provided with the Nano-Drive®C is designed to mediate communications between Windows and the Nano-Drive®C. Communication between the USB driver and the user is controlled by MADLib.dll When a user wishes to access a USB device such as the Nano-Drive®C, a “handle” to the device must be opened in Windows as a means for the user and Windows to identify a specific device the user wishes to communicate with. In this sense, a “handle” is simply a unique identifier.

All programs should be of the form:

<...>

Initialize handle(s).

Program portion that relies on the Nano-Drive®C

Release handle(s).

<...>

4.1.1 Using a single Nano-Drive®C controller

Using a single controller three functions are required to properly manage handles:

- MCL_InitHandle will obtain control of the controller. Should a LabView program stop/crash/finish prior to releasing the handle subsequent calls to MCL_InitHandle will fail since the DLL still has the handle.
- MCL_InitHandleOrGetExisting will obtain control of the controller. This function will return control of the controller even if the LabView program did not properly release the handle.
- MCL_ReleaseHandle will relinquish control of the controller.

4.1.2 Using multiple Nano-Drive®C controllers

Using multiple Nano-Drives MCL_InitHandle and MCL_GrabHandle may have ambiguous results. MCL_InitHandle will obtain a handle to one of the Nano-Drive®C (s) attached. MCL_GrabHandle may be used to differentiate between different classes of Nano-Drive® (s), however, if multiple devices of the same class are attached the result will be ambiguous. To deal with this ambiguity each Nano-Drive®C has one unique feature, namely, its serial number. MCL_GrabAllHandles may be used to obtain handles to all Nano-Drive®C (s) attached but not yet under control. MCL_GetHandleBySerial may be used to search all of the attached and controlled Nano-Drive®C (s) for a Nano-Drive®C with a matching serial number.

More detailed documentation on these functions can be found in the document Madlib_*_*.doc (supplied on CD).

4.2 Position Commands

Position command data coming into the Nano-Drive®C from an attached computer is read by the DAC and applied to the nanopositioner at the chosen (or default) rate. Since the ADC and DAC have independent clocks, the read/write rates do not have to be the same. If the DAC rate is changed from the default value, the new DAC rate will remain in effect until it is changed again or until a data logging procedure is initiated (see below).

It is important to note that a latency exists between the time when a computer program issues a position command and the time when that command is communicated to the Nano-Drive®C via the USB interface. The Windows operating system "talks to" each USB port on a regular 1 millisecond interval. This access timing can become important if the nanopositioner is controlled in a point-by-point manner and the desired motion is fairly fast (roughly 100Hz or faster). Fast motions should be controlled and monitored using the built-in data logging facility.

4.3 Reading Position

Under normal operation, the position of each axis is constantly recorded at the chosen (or default) ADC rate. Internal memory in the Nano-Drive®C is used to hold 30 position values which are accessed and averaged together in the MADLib.dll software. As each new ADC value is stored, the oldest value is removed from memory to maintain a buffer of only 30 values. This process reduces the effects of random noise and produces highly reliable position data. It should be noted, however, that reducing the chosen ADC rate too much can create an apparent “phase shift” in the position data since the total time span of all 30 values may be excessive. Leaving the ADC at the default rate (highest speed) is probably the best choice for most applications. If the ADC rate is changed from the default value, the new ADC rate will remain in effect until it is changed again or until a data logging procedure is initiated (see below).

4.4 Built-In Data Logging

The Nano-Drive®C controller has internal memory to store up to 10,000 position command values and up to 10,000 position sensor values. Rather than reading/writing a single position value at a time (as described above), the USB interface can be used to read/write 10,000 data values to/from these internal memory locations. This method of data logging eliminates the 1ms/point Windows latency which limits nanopositioner data rates when the USB interface reads/writes a single point at a time. Software trigger commands initiate the flow of data in and out of the ADC/DAC circuits. The ADC and DAC clock rates are independently selectable from the range of values from 5ms/point to 33(s/point. After completing a data logging procedure, the Nano-Drive®C reverts back to the default ADC/DAC rates and resumes the normal monitoring of the position by reacquiring a new array of 30 position values (as described above). A new single position read is available 3ms after completing a data logging procedure – this time is required to fill the 30 data point array.

Since a single memory area is available for data storage, the data logging feature can only be used on one axis at a time. This apparent limitation of the data logging method is rarely an issue in actual scanning applications. Typical scanning routines sweep the nanopositioner across the area of interest in one axis, step ahead in another axis, and then repeat. Software “triggers” initiate the process of reading/writing data by the ADC/DAC. In the simplest case, a single trigger can be used to initiate the downloading of position data from the Nano-Drive®C memory into the DAC to provide a signal into the nanopositioner control circuit. Simultaneous writing and reading of position data with a single trigger can also be done. While ADC and DAC default rates are the same – their clocks are completely independent and their rates can be set at different values.

4.5 LabView Examples

Several examples of software control using LabView have been installed in the LabView Examples folder of the root directory, which by default is C:\Program Files\Mad City Labs\NanoDrive. Documentation for each VI can be found on its Block Diagram and/or Front Panel. The supplied VIs are outlined below along with a brief description.

4.5.1 Simple Position Control VI's

- NANOXYZ_MONITOR.VI uses a LabView function generator to generate commonly used waveforms such as sine wave, square wave, triangle wave, and sawtooth on all axes.
- READWRITEN.VI reads the current position, writes a new position, then waits.
- WRITEREADN.VI writes a position, waits, then reads the current position.

READWRITEN.VI and WRITEREADN.VI perform sequences that can be used as the basis for a scanning program. Both VI's are set up to function as sub-VI's. WRITEREADN.VI is used in all of our provided scanning examples. Consult National Instruments documentation for information about using sub-VI's in LabView.

4.5.2 Common VIs

- ErrorLookup.vi translates error codes into a readable form. It can be used as a SubVI.
- GetCalibration.vi returns the calibration in microns (μm) of a given axis. It can be used as a SubVI.
- Nano-Drive Attached.vi returns a True or False value based on whether the Nano-Drive® is attached.
- RampGenerator.vi generates a linear ramp between two positions with a specified number of points. It is used as a SubVI in all of the scanning examples.
- ReadWriteN.vi reads the current position, writes a new position, then waits. Works well as a SubVI within scanning programs.
- StandardInit.vi initializes a Nano-Drive®C so it can be used by software. Used as a SubVI
- StandardRelease.vi frees an initialized Nano-Drive® so it can be used by other programs. Used as a SubVI.
- WriteReadN.vi writes a position, waits, then reads the current position. Works well as a SubVI within scanning programs.

4.5.3 Device Information Examples

- Multiple Nano-Drives® Information.vi provides information on all available attached Nano-Drive®C.
- Nano-Drive Information.vi displays information about a single Nano-Drive®C.

4.5.4 Handle Management Examples

- GetHandleBySerial.vi initializes the Nano-Drive®C that has the specified serial number.
- GetHandleByType.vi initializes the Nano-Drive®C that has the specified type.

4.5.5 Multiple Device Examples

- Distinguished by serial.vi controls two separate Nano-Drive®C simultaneously using the serial number to distinguish between the two.
- Distinguished by type.vi controls two separate Nano-Drive®C simultaneously using the type of the Nano-Drive®C to distinguish between the two.

4.5.6 Scan Examples

- SawScan(One axis).vi steps the X axis through a specified linear ramp and upon completion returns the X axis to its starting position.
- SawScan(Two axis).vi performs a scan such that for each step in a specified linear ramp on the Y axis the X axis is completely stepped through a specified linear ramp and upon completion returned to its starting position.
- TriangleScan(Two axis).vi performs a scan such that for each step in a specified linear ramp on the Y axis the X axis is completely stepped through a specified linear ramp and upon completion the X axis remains in its position and has its specified linear ramp inverted.

4.5.7 Simple Device Movement Examples

- Nano_Monitor.vi uses a LabView function generator to generate commonly used waveforms (such as sine, square, triangle, and sawtooth waveforms) on a single axis.
- NanoXYZ_Monitor.vi uses a LabView function generator to generate commonly used waveforms (such as sine, square, triangle, and sawtooth waveforms) on all axes.
- Read.vi reads the position of the nanopositioner in a continuous loop

4.5.8 Waveform Acquisition Examples

- LoadWaveform (subVI).vi stores position commands in the Nano-Drive®C memory. This VI triggers the Nano-Drive®C to use these command positions immediately
- LoadWaveformSetupTrigger.vi stores position commands in the Nano-Drive®C memory with one function call and triggers the Nano-Drive®C to use them with a another function call.
- ReadWaveform(subVI).vi can read up to 1000 consecutive position sensor values from the Nano-Drive®C internal memory. This VI informs the Nano-Drive®C to immediately begin recording data and the number (points) and rate at which data should be logged.
- ReadWaveformSetupTrigger.vi describes a two step process of describing how to log data (as described immediately above) and then triggering the Nano-Drive®C to begin recording data.
- WaveformAcquisitionSynchronous.vi describes how to trigger the Nano-Drive®C to use stored position control values and record consecutive position sensor values synchronously.
- WaveformAcquisitionSynchronous (subVI).vi

4.5.9 Other

- Raster Scan.vi provides step-by-step instructions for setting up and performing a multi-axis raster scan.
- Step Response (16 bit).vi reads the nanopositioner at a fast rate while the nanopositioner moves to a position.

4.6 Programming in other languages

Use the Madlib.h, Madlib.lib, and Madlib.dll to develop software for the Nano-Drive®C USB interface. All three files can be found in the root directory, which by default is C:\Program Files\Mad City Labs\NanoDrive.

5 TROUBLE SHOOTING

The Nano-Drive®C is a high precision instrument and therefore must be operated in a quiet, both physical and electrical, environment. Both electrical and vibration noise can be picked up in the Nano-Drive®C and/or the nanopositioning stage. This noise can be observed on the HV/10 BNC output.

1-5 Hz Low frequency noise is typically due to building vibrations and can be caused by foot traffic or vehicle traffic. This type of noise is often time dependant. It can be eliminated through careful vibration isolation of the nanopositioning stage.

60,120 Hz. This noise is caused by ground loop interference. Ground loop interference results from a slight difference in the ground potentials of the nanopositioning stage and the Nano-Drive®C. This problem can be identified by electrically isolating the nanopositioning stage, for example by placing it on a piece of paper. If the noise level decreases the grounds are at a different potential.