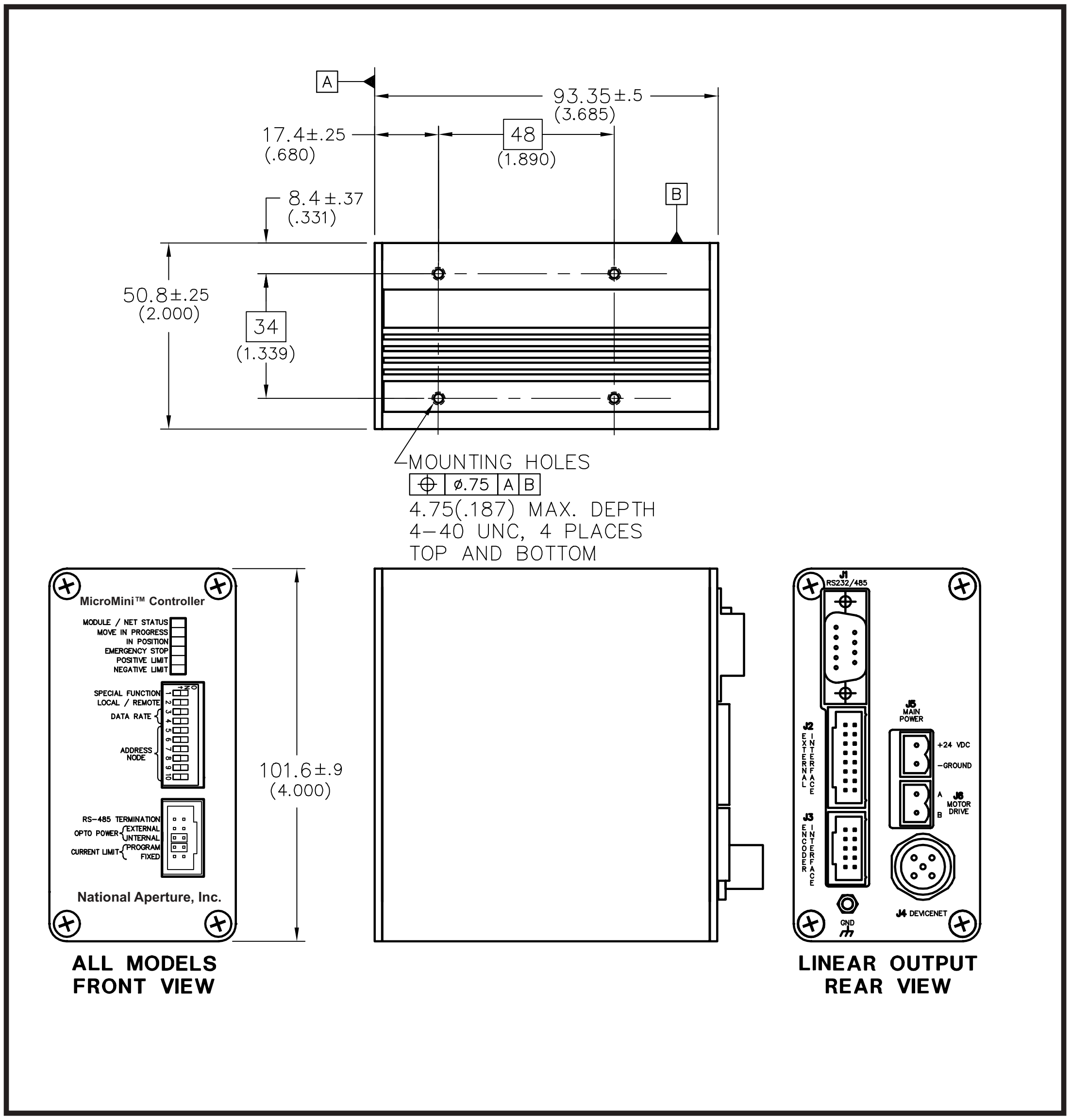
**Manual #002.1**

# USER MANUAL





**National Aperture,Inc.**
**16 Northwestern Drive**
**Salem, NH 03079-4810**
**603-893-7393/FAX 603-893-7857**
**www.nationalaperture.com/www.naimotion.com**


**MICROMINI™ CONTROLLER**

# Table of Contents

# MicroMini™ Controller

The basic MVP® series controllers are miniature, single-axis, “intelligent” drives that can provide speed and position control and can be configured for a wide range of DC motors. Designed to integrate seamlessly into any proprietary or open architecture system, they come complete with built-in motor drive amplifiers. This eliminates the extra amplifier costs, amplifier modules, and wiring costs inherent in traditional motion control topologies. Up to 63 motor axes can be addressed using the RS-485 Multidrop Network interface. MicroMini™ Controller modules can operate in Remote Mode (serially), or Local Mode (stand-alone). Your choice is made entirely through DIP switch adjustments and software configuration: no additional or substitute parts or additional modules are needed.

The **MicroMini™ Controller** (MVP-1) is a highly specialized version of the generic OEM MVP® controller. It has been specially configured and documented to handle National Aperture’s MicroMini™ stages with their limit-homing switch system and specially wired servo motors.
*Warning: A standard generic MVP® should not be used with National Aperture’s MM stages as the unconfigured MVP® controller without specialized cabling will do serious damage to both the motor and the stage body.*

**Use with other motors and stages**
However, the controller can be reconfigured to handle the wide variety of motors as mentioned in the previous paragraph but this requires a knowledge of the commands and the available settings.

## MicroMini™ Controller Module Flexibility

**Stand-alone**
- Local Mode
  - Velocity Control (desired velocity)
    - Absolute (coarse speed setting)
    - Relative (fine tuning)
  - Tracking Mode (Optional) Step/Direction or Encoder Following

**DCS**
- Remote Mode
  - RS-485 or RS-232C
    - Velocity Control
    - Position Control...... Absolute and Relative

MicroMini™ Controller units can be operated as either stand alone modules or as part of a distributed control system (DCS).

## Summary of Features

- •Programmable Current Limit
- •Optically Isolated Inputs including Encoder, Overtravel Limits, External Event (Home) Inputs, and Emergency Stop
- •One Analog Input (10 bit)
- •One Analog Output (12 bit). Also used for programmable current limit.
- •+/-10 Volt DC Linear Control Outputs
- •Linear Drive Amplifier Supplying 10 Watts Continuous @22˚C Ambient
- •Programmable Position Range Limits
- •Programmable Maximum Allowable Following Error
- •RS-232C or RS-485 Operation
- •Panel, Rack or optional DIN rail mounting
- •Stand-alone, Terminal, PC Compatible Operation
- •Interface Software Demo and Example Code
- •Very Compact — only 2x4x4 inches
- •Easy to use Plug and Play operation
- •Flexible configuration through software

### Amplifiers:

One of the distinguishing features of the MicroMini™ Controller modular system is that integrated amplifiers are included inside each module.

### Communication, I/O, & Environmental:

RS-232C or RS-485 (Command sent as ASCII)

### Controller Inputs:

| | |
|---|---|
| Encoder: | Two channel, single-ended +5 VDC TTL compatible<br>4 MHz max. frequency, optically isolated |
| Analog: | One analog input (0-5 VDC, 10 bit) |
| General Purpose: | Two hard limits, one emergency stop,<br>two external event inputs: all optically protected |

### Controller Outputs:

| | |
|---|---|
| Analog | One Analog output (± 10 VDC, 12 bit DAC) |
| Motor Command | One 12-bit DAC (± 10 VDC) |

**Drive Amplifier**

| | |
|---|---|
| Linear Amplifier | Outputs ± 22 VDC @ 1.0 Amp Continuous; 3.5 Amp peak |

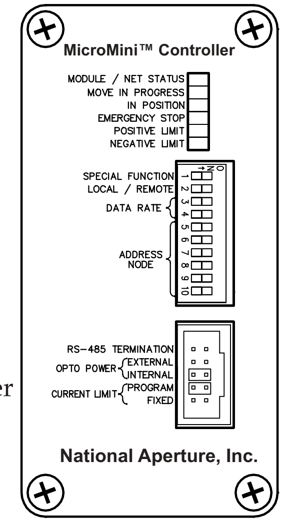
**Environmental (standard version):**

| | |
|---|---|
| Controller Operating Temp | 0 to +70°C (32°F to 158°F) |
| Ambient Operating Temp | 0 to +40°C (32°F to 104°F) |
| Storage Temp | -25°C to +85°C (-13°F to +185°F) |
| Humidity Tolerance | 80% RH, non-condensing |
| Enclosure Protection Level | IP40 |

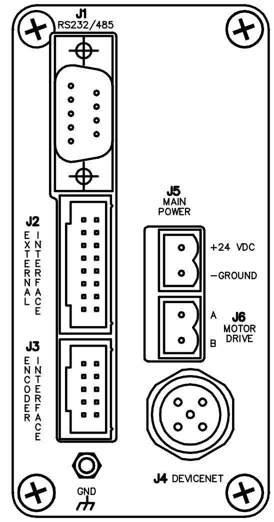
## Unpacking

Examine shipping containers for visible damage. If damage is discovered, file a claim with the carrier immediately. Each MicroMini™ Controller has been 100% inspected and tested prior to shipment in accordance with National Aperture's Quality Assurance Program. Your MicroMini™ Controller comes in an antistatic package. Carefully remove the unit from this package and retain the packaging in case the controller needs to be shipped or stored in the future. Make sure that the unit(s) shipped match the packing list.

***Warning: The MicroMini™ Controller contains ESD sensitive devices.***




FRONT VIEW

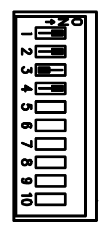



LINEAR OUTPUT
REAR VIEW

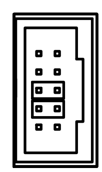
## Hooking up to your MicroMini™ Controller.

### (WIN 3.x, 95, 98, NT)

As damage can occur in transit we recommend that you hook up your module and verify its operation upon receipt. Make sure you have all the parts listed on the packing list.

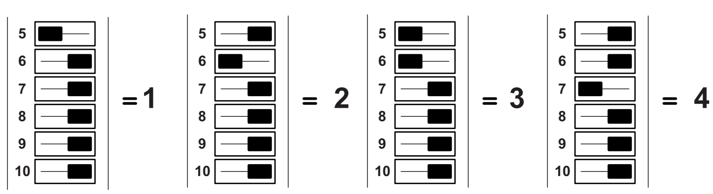


**1. Leave power off**
(Don't plug in MicroMini™ Controller wall supply yet.)



**2. Check front panel.**
If dip switches 1 and 2 and the two jumpers are not as shown, please correct (switches 3 through 10 are to be set by the user), then continue as follows.

**3. Create a binary address other than "0" using switches 5-10 as shown.**
The address (1,2,3,4,...) will be the first entry in every command.[Using the address "0" will address all the modules in your network.]




**4. Make sure baud rate is set**
(typical 38,400 switch 3 LEFT, switch 4 RIGHT)
See "**Configuration Options**" (page 6) for more settings.

**5. Set up rear panel**:

Refer to connection diagrams in Appendix B and proceed as follows:

(a). Plug in J-5 connector (from wall supply).

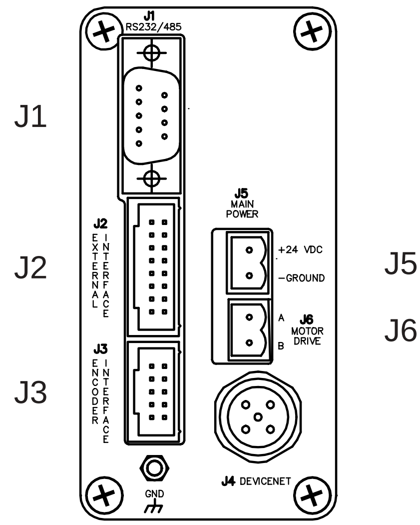
(Don't plug MicroMini™ Controller supply in wall yet.)

(b).Using main (3 to 1) motor cable, CA-MVP-1(See also Appendix C for reference; familiarity with the connection will help in trouble shooting.)

Plug in:

| | |
|---|---|
| MOTOR DRIVE | J-6 |
| ENCODER INTERFACE | J-3 |
| EVENT INTERFACE | J-2 |

Plug 10 pin IDC end of CA-MVP-1 interface cable (male) into stage motor cable (female).




LINEAR OUTPUT
REAR VIEW

**6. Plug SERIAL CONNECTOR in back of MicroMini™ Controller and connect to PC.**

[RS-485 adapter must be plugged into PC for multi-axis systems]. See also "Remote Mode Serial Designation," page 7.

**7. If computer is not already on, turn on now.**

(Do not plug in MicroMini™ Controller wall supply yet.).

**8. Install MicroMini™ Controller Demo software from disks** (check for correct Windows version) **if not already installed.** See "Software Installation", page 7.

**9. Start MicroMini™ Controller Demo program.**

(a) Go to **"Configuration"** (page 8). Set baud rate to match that on MicroMini™ Controller DIP switches (38,400, typical) and select available comport (1,2, etc.) and save.

*Make sure comport is available and working - may require computer tech to verify*.

(b.) Go to **Terminal Emulation.** (See picture on page 9.)

(c.) **Please read entire paragraph before attempting to follow these instructions.**

Plug in **MicroMini™ Controller** wall supply to AC outlet and observe LED's on front panel. Upon plug-in, watch upper LED's for self diagnostics. All lights must in turn appear on. The first (top) green light will stay on and the third (green light) <u>may</u> remain on at this point.The red limit lights will flash once in turn.

Example: If you didn't see this happen type in "node" **DI**, and "enter","node" **RN***, and "enter". The **Terminal Emulation** box should say "MVP... ready." If not, recheck connections, configuration, port. ("See Front Panel Indicators", page 8).

**Note: RN means reset node and is similar to unplugging and replugging the wall supply.*

**10. Choose either Rotary or Linear translation stage configuration.**

– If you have a **Linear** stage, you will not need to change the default Limit switch (**LS**) value, nor will you need to alter the wiring of the interface as the Limit and Homing functions are shared by one switch.

– If you have a **Rotary** stage you will need to disable the Limit action of the Homing switch (+Limit and Homing are combined by the Interface cable). There are two ways to disable the Limit action in a MM-3MR/MM-4MR rotary stage.

a.) Set the Limit action (**LS**) to 99 (see "Examples and Explanations" page 11).

b.) Clip or unplug the jumper on the J-2 connector (eliminate pin 5).

This method will be necessary if you plan to use external limit protection: pins 5 & 6 will be used for positive and negative limits respectively with pin 16 as the common ground. Pin 7 (event 2) will be reserved exclusively for Homing.

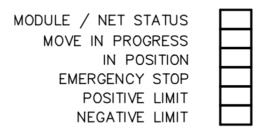
**The correct way to apply hard limit protection to a Rotary stage is externally.**

## Front Panel Indicators

### LED Status Indicators

The following, from top to bottom, are the status indicators and their meaning:

- MODULE/NET STATUS
  - Flashing green = not initialized/Allocated
  - Steady green = Initialized/Allocated
  - Flashing Red = Recoverable Communications Error
  - Steady Red = Unrecoverable Communications Error
- MOVE IN PROGRESS = Active Position or Velocity Profile
- IN POSITION = Motor is in the Commanded Position
- EMERGENCY STOP = ESTOP Input has been Activated
- POSITIVE LIMIT = PLIM Input has been activated
- NEGATIVE LIMIT = NLIM Input has been Activated




## Configuration Options

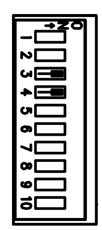
### Node Address (switches 5-10)

The individual node address is determined by the binary coding of these switches. Switch 5 is the least significant bit (LSB). Node 0 = all switches RIGHT. In the Remote Mode (RS232/RS-485) all nodes will respond to a command directed to node 0. This functionality allows synchronization among multiple nodes. This is the reason 63 axes can be addressed serially.

### Data Rate Switches

DATA RATE (3 and 4) Communication rate determined by the binary encoding of these switches as indicated below:

| **Switch #** | | **Remote Data Rate (kbaud)** |
|---|---|---|
| **3** | **4** | **RS-232C/RS485** |
| LEFT | LEFT | 57.6 |
| RIGHT | LEFT | 19.2 |
| LEFT | RIGHT | 38.4 |
| RIGHT | RIGHT | 9.6 |

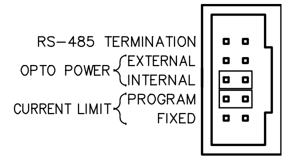


### Local/Remote Mode

Switch number 1 to RIGHT = RS-232C/RS485 mode active (default)
Switch number 1 to LEFT = Local mode (Potentiometer/joystick operated)

### Isolation Power Source

The "opto power" selector determines whether or not the MicroMini™ Controller opto isolation devices will be powered from an internal (5 VDC) or external (5 VDC) power source. The customary (default) setting is "internal". If an "external" power is desired, a source can be accommodated at pin 3 of the external interface connector, J2, on the rear of the module.




**RS-485 Network Termination**
The termination jumper pins are below the front mounted switches (see drawing):

Unless you have specified otherwise, your MicroMini™ Controller unit has been shipped without a termination jumper. In the event you were to use a jumper (typically for very lengthy runs of networked modules), it is advisable to insert the jumper on the MicroMini™ Controller module furthest from the host controller. This keeps the signal levels balanced across the network. Each MicroMini™ Controller has an internal 120 Ohm termination resistor included for this purpose.

## Remote Mode Serial Designation

Remote operating mode allows closed loop speed and position control with trapezoidal profiling via the RS-232C or RS-485 interface. Remember when you operate in any of the serial modes, the communication protocol is pre-set at the factory to either RS-232C or RS-485. Confirm which model you have by checking the label on the side of the MicroMini™ Controller module. If your unit is configured for RS-485 operation, your host computer or terminal must also be capable of RS-485 communication. RS-485 converter modules are available from National Aperture.

## Event Input Overview

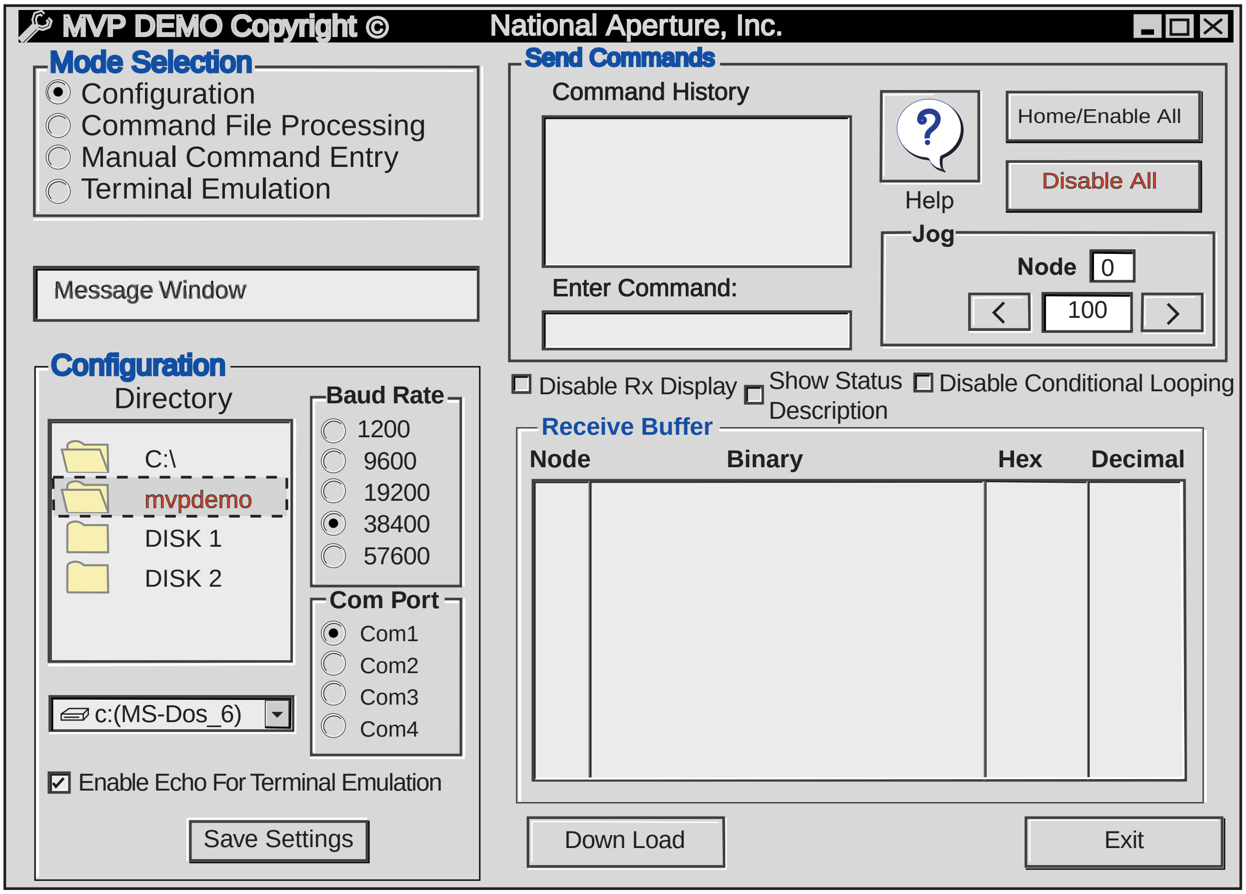
Inputs for analog control or monitoring, overtravel, emergency stop, and HOME or event sensing are also accessible, some of which are used by the CA-MVP-1 interface cable. An amplifier enable output can also be used to automatically select an external amplifier when a control mode is activated. The overtravel (Limit switch) inputs are intended to protect the drive from a hard stop. The status of the HOME or event inputs may be determined by testing bits in the status response. The value of the 10 bit analog input can be obtained by interrogating the controller using the serial ANI command. The location of the external interface connections are detailed in Appendix C.

## Software Installation

Operation in Remote Mode allows you to install the demo software which came with your Startup Package or as an option with your first commercial shipment of MicroMini™ Controller modules. The demo software is designed to highlight the capabilities of Remote Mode Operation. The program disks are for Windows 3.x, Windows 95, Windows 98, or Windows NT operating systems, utilizing a 16550 UART or equivalent on a 386/16 MHz machine or better. Initial program installation will give you an MVPDEMO icon in the program group in your hard drive which you can use for future training or testing. When installing the Windows program for the first time, close all other Windows programs. Insert disk 1 and run “a:\setup” (or”b:\setup” depending on where your floppy disk drive is located). Insert Disk 2 when prompted. This will establish an MVPDEMO icon in your program file group folder.

# Configuration

Click on the MVPDEMO icon to begin setting up your system..



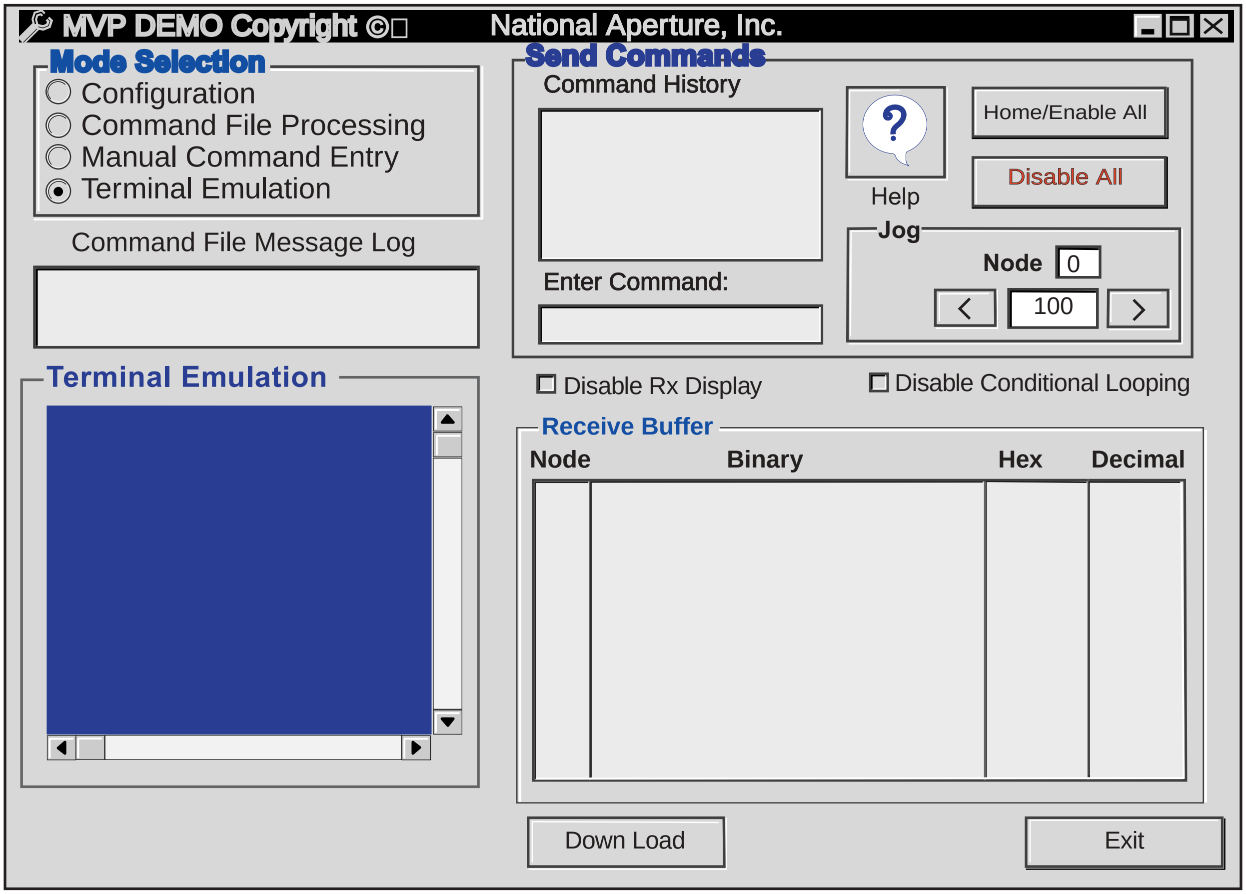

The default screen which you will open looks like this

The “Configuration” button allows you to choose the baud rate at which your system will operate and the com port which you want to use. **Both the software and the MicroMini™ Controller module(s) running on this program must have the same baud rate**.

We recommend configuring to 38,400 baud to provide reasonable bandwidth and information processing speed.

| Switch Number | Remote Data Rate (kbaud) RS232C/RS485 |
|---|---|
| 3. Left<br>4. Left | 57.6 |
| 3. Right<br>4. Left | 19.2 |
| 3. Left<br>4. Right | 38.4 |
| 3. Right<br>4. Right | 9.6 |

## Terminal Emulation



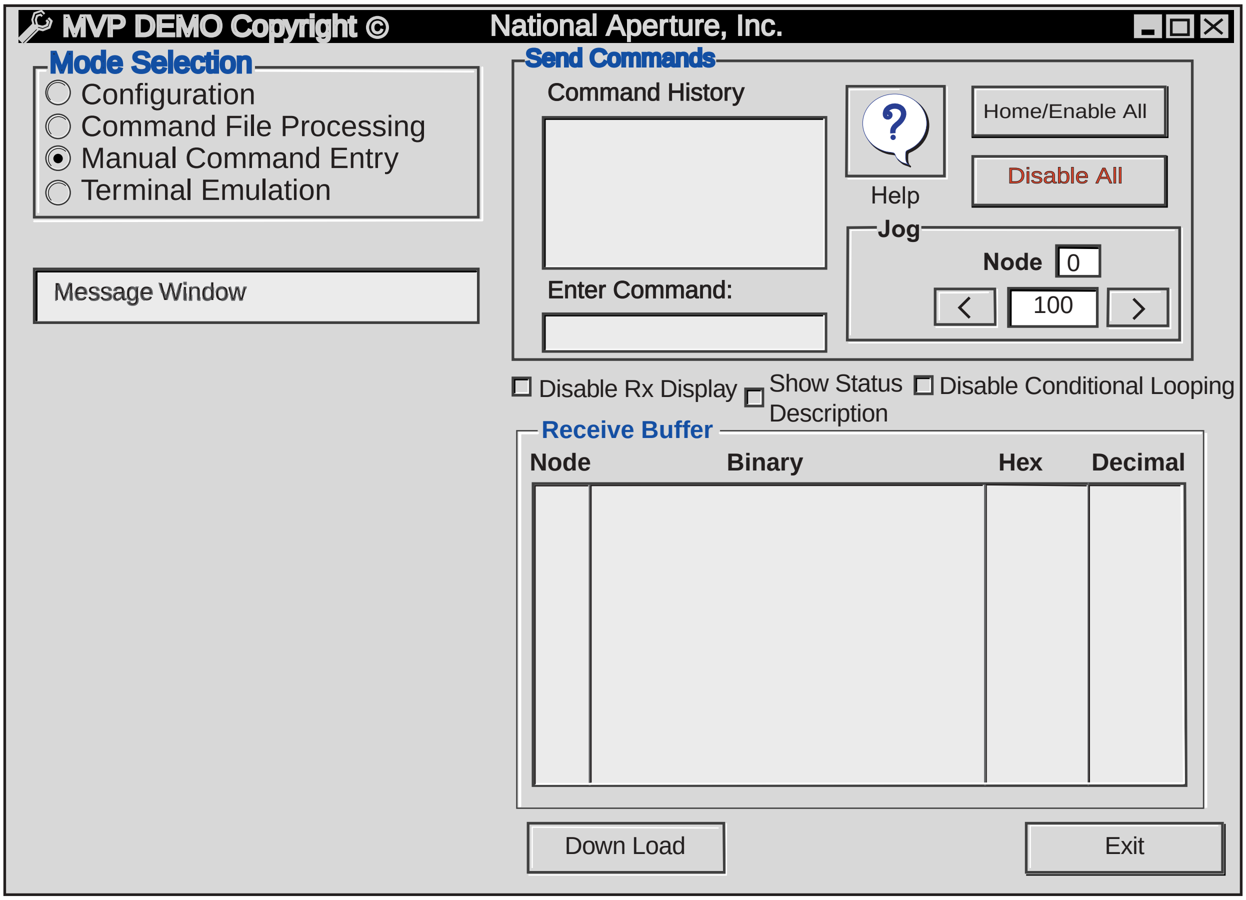

Terminal emulation mode allows direct interaction with the MicroMini™ Controller. Command strings entered in terminal mode are sent directly to the target controller and response data is displayed in the terminal window. Configuration for terminal mode is the same as for all other modes. This provides a means to troubleshoot communications difficulties without the overhead of additional software.

Communication may be verified by typing and entering "node#" **DI** then **RN (**Reset Node) or by unplugging and replugging controller wall supply.

The best way to practice sending commands is by (following page) the " Manual Command Entry" mode.

## Manual Command Entry

Select the "**Manual Command Entry**" button. The following screen will appear. This should be the first means of testing commands and command syntax.




The code you will put into the "Manual Command Entry" screen will perform all functions one command at a time including any status request. It can't be saved or edited. Each instruction is invoked only once before passing into Command History. It will, however, give you a flavor of how the command structure works and will allow you to control a servo motor. But first you will need to know a few generic motion command formalities.**( See "Examples & Explanations", next page.)**

1. In order for a module or series of modules to receive instructions from the host, each module must have a unique "address." If you are using an RS-232C protocol, you will be using only one address, because in RS-232C configuration you can only communicate with one MicroMini™ Controller module. If you are using the RS-485 option, you must give each of your MicroMini™ Controller modules a unique address from 1-63 so each module will know when the host is addressing it. Using the address "0" will address all the modules in your network.

2. Before a motor can move, it must be "enabled." Before enabling it must receive an HO command (not to be confused with homing to a fixed reference). Enabling allows the MicroMini™ Controller to source current by turning on its amplifier. To "enable" your drive, you will use the Command **HO**, then **EN**. Each axis can be enabled independently by using appropriate address with letter commands, or click the Home/Enable button to enable all.

3. After enabling, the motor(s) can be commanded to move, but first use a sample velocity of 200 (slow). **(See "Examples & Explanations", next page.)**

# Examples and Explanations Using Manual Command Entry Mode

## I. Sending Commands (See also "MicroMini™ Controller Commands")

Constructing a motion profile is simply a matter of stringing together a series of compatible commands which make the drive system execute the profile you want to implement. Set the address of your MicroMini™ Controller to 1. Try the following simple sequence to make the motor move, **after** first selecting an available com port. (See "Hooking up to your MicroMini™ Controller" for DIP switch settings.)

This is how the practice sequence looks with a return after each command. Read the explanations as you enter them:

1. **1 HO**
2. **1 EN**
3. **1 LA 2000**
4. **1 SP 200**
5. **1 AC 400**
6. **1 M**

[Normally, you would next set Proportional & Integral Gain, but for this exercise you can leave the gain setting in the default positions which are very tame. The Proportional Gain Setting defines how quickly your system will respond to a deviation from the desired profile (moving). The Integral Gain determines how fast the motor or load integrates into position at the end of the profile (stopped). When you do an actual motion application you will have to change these values to optimize the performance of your system. If you need to brush up on this topic, see Appendix H "Digital Filter Configuration."]

### Explanations:

1. **Home**: A motor's home location will be at a point you define as zero (0). To tell your module number 1 that it is home, or that it is "where it is commanded to be", enter "**1 HO**" (without the quotes) and hit return.

2. Next **Enable** your MicroMini™ Controller because you cannot move a motor that is not enabled. The **EN** command allows the amplifier to respond to commands destined for the MicroMini™ Controller module. Enter "**1 EN**" and return. *Feel the lead screw coupling to make sure you are in servo mode (resists rotation).*

3. The **Destination** position (where the motor/load will move to). This can be either an Absolute or Relative point. If you want to move your load, motor shaft, etc. to a predetermined point from "home," you will need to "**L**oad [the] **A**bsolute[destination position]" which you can do by giving the module(s) the command **LA**.. Putting "**1 LA**" followed by a space and a number will allow the shaft to turn encoder counts in either direction. If you want to go to a position a given distance from the point you are presently located at (other than home), give the module the command LR which will move the motor shaft incrementally relative to its present location (*i.e.* from a point other than home). Repeatedly typing "1 LR 1000", (return) followed by "M" (return) to initiate the motion, will give you consistent steps of 1000 counts each.

4. The **Velocity** at which you wish to move. This is defined by the command **SP** (speed). The speed at which the motor shaft will turn is dependent on the resolution of the encoder you are using (see table ). If you are using a 500 place encoder, each unit = 0.48 rpm If you are using a 10 place encoder, each unit = 24.0 rpm. maximum. Typing "**1 SP** 200" (return) should be adequate for demonstration purposes.

| Encoder Lines/rev | Quadrature cts/rev. | rpm/unit | Example |
|---|---|---|---|
| 1000 | 4000 | 0.24 | 960/4000 = 0.24 rpm/step |
| 500 | 2000 | 0.48 | 960/2000 = 0.48 rpm/step |
| 200 | 800 | 1.2 | 960/800 = 1.2 rpm/step |
| 100 | 400 | 2.4 | 960/400 = 2.4 rpm/step |
| 16 | 64 | 15.0 | 960/64 = 15.0 rpm/step |
| 15 | 60 | 16.0 | 960/60 = 16.0 rpm/step |
| 10 | 40 | 24.0 | 960/40 = 24.0 rpm/step |

*Note: using at 1000μs sampling period (SR), 960 is the conversion factor (at 500μs; 1920, etc.)*

5. The **Acceleration** (command **AC**) which is the number of quadrature counts per sample period. The sample period for the MicroMini™ Controller is 1000μs. Type "**1 AC** 400" (return) [the default].

6. The **MOVE** (command **M**).
   Make it move by using the command "**1 M**" (MicroMini™ Controller box at address node #**1 MOVE**!). If your motor doesn't move, recheck all connections and try reentering the command string again.

   If your motor jumps a little but does not reach destination, try increasing the following error (**FD**) to 130 (default is 110). If your motor does not move at all, recheck all connections and communication configuration and reset the controller if necessary. If still no motion can be achieved, please contact National Aperture, Inc. at (603)893-7393 and ask for "MicroMini™ Controller Technical Support" In this case, you should be able to access the computer and controller while talking to the support engineer.
   **Important: Using the address of "0" will address all the modules in your network.**

## II. Query Status

You can enter any command you wish, without any argument, to verify the current setting. Not having an argument treats the command as a request to send the stored value back to the Host or Master.

1. "**Status**" request **ST**
   The motion and event status of your operations can be determined at any point by invoking the "status"request (**ST**). The **ST** command serves to verify whether the move is happening or has happened. Try typing "**1 ST** " to check the motion status. A binary result of 1010 (decimal 10), according to the status bit code (see **"Status Response Bit Map"**, page 21) signifies "command recognized," "motor in position," and "trajectory is complete". The remaining bits 4-15 are 0, indicating no errors.

   A very useful feature of the status command is the **Status Mask**, used in Command File execution. This involves placing a binary value after the **ST** command. This will prevent further program execution until the desired status is achieved as dictated by that binary value. For example; **1 ST 10** means that unless the binary code of 1010 is achieved (binary 1010 indicates motion is complete and has a decimal value of **10** ). Other statuses such as an active Limit Switch, Velocity Mode, and/or External Event active, also contribute to a binary value which will reflect their existence. For example, if you want to demand three conditions before continuing execution you must add their binary values and place the total value after the **ST.** The command (**1 ST** *XXXX).* must be inserted where needed.

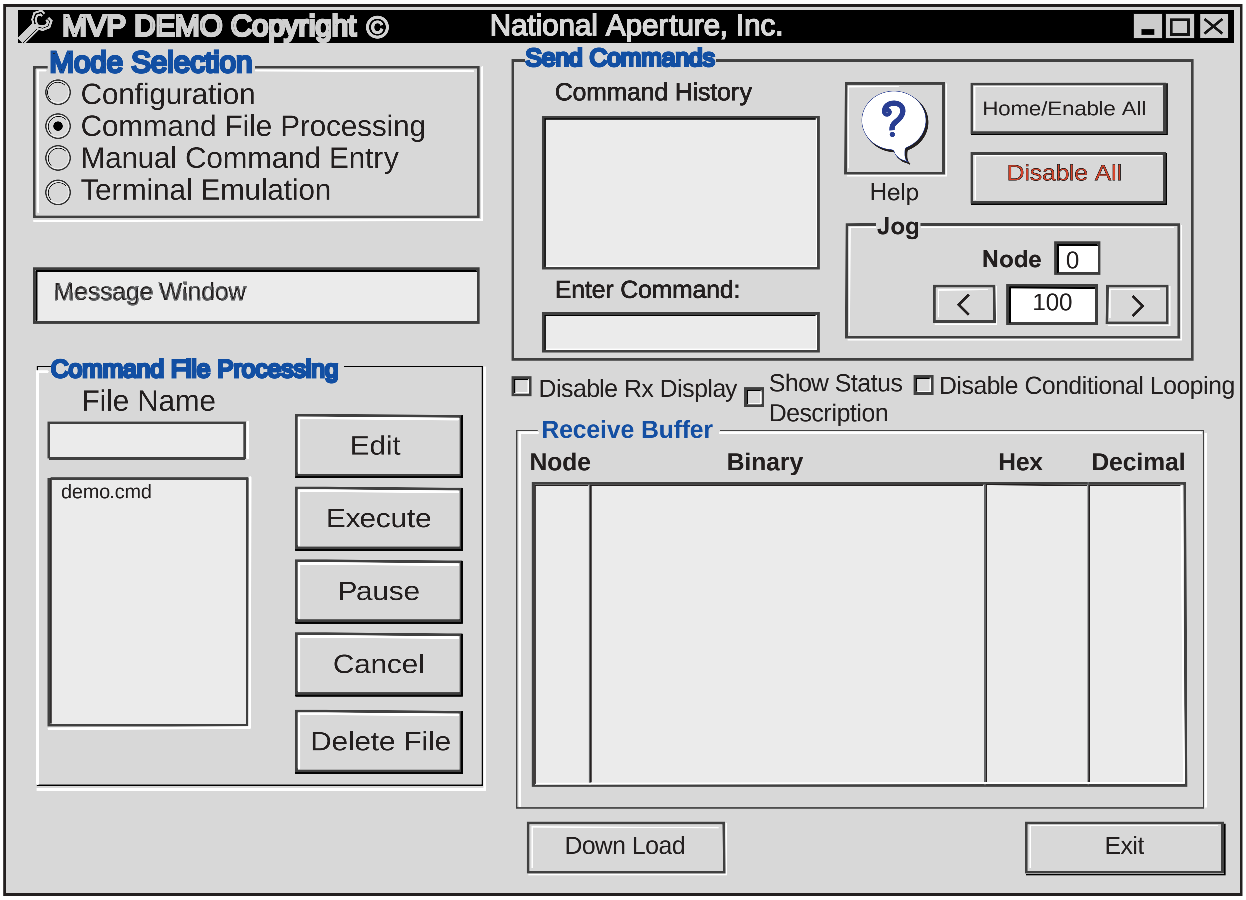
   *Note: If the status mask of over 32,767 is required a minus sign must be placed before the value (i.e. -32,768).*

2. **Value** requests. **All applicable commands.**
   Any value or setting can be retrieved by entering the node #, then the appropriate command. The binary and decimal values will then be displayed in the Received Buffer window.

   Example; **1 POS** *enter* returns the position value of node 1 in the Receive Buffer window.

   One current ***exception*** involves the **HA** (Home Arm) and **HP** (Home Polarity) commands which utilize the **HS** (Home Status) query. The result is displayed as binary, the first (bit 0) and second (bit 1) bits show the **HA** and **HP** values respectively.

# Command File Processing




In order to construct, store, edit and execute motion routines, you must go into the programming screen by selecting the “Command File Processing” button from the Mode Selection box. Here is what it looks like.

Operable programs are provided for you in the form of “files”.cmd. These also serve as examples to show you how the code works. To execute a program, highlight the “file”.cmd” and click the execute button [ It is advisable to check the content of the file by first going to “Edit” mode] and select the “NO” option to execute the file once. Comments are provided at the top of each file to indicate the application.

Write a stored, executable file by typing a new program name in the File Name window and clicking the edit button

When a command file is created and saved it will be stored in the directory with this demo program as a .cmd file. Any “.cmd” files stored in the MVP program directory will be available for execution.

See “**Examples**” (Appendix G).

# Sending Direct Serial Commands without the MVP Demo Software

## Remote Communication with the MVP Controller

1. Start the terminal program you intend to use for communication with your MVP Controller. Open the configuration or setup menu or dialog box for the program and locate the settings concerned with the communication port.

2. Select the port number, such as COM1 or COM2 to use an available serial port on your computer. Set the data word format for 8 data bits, no parity bit, and 1 stop bit. Select the data rate matching the configuration of your MVP by examining the settings of the Data Rate switches 3 and 4 and referring to the following table:

| 3 | 4 | Data Rate |
|---|---|---|
| ON | ON | 9600 |
| ON | OFF | 19200 |
| OFF | ON | 38400 |
| OFF | OFF | 57600 |

3. In your terminal program select the setting that matches that of your MVP. If the MVP's data rate setting is out of range of your terminal program, then configure the terminal program and the MVP to the fastest data rate they both have.

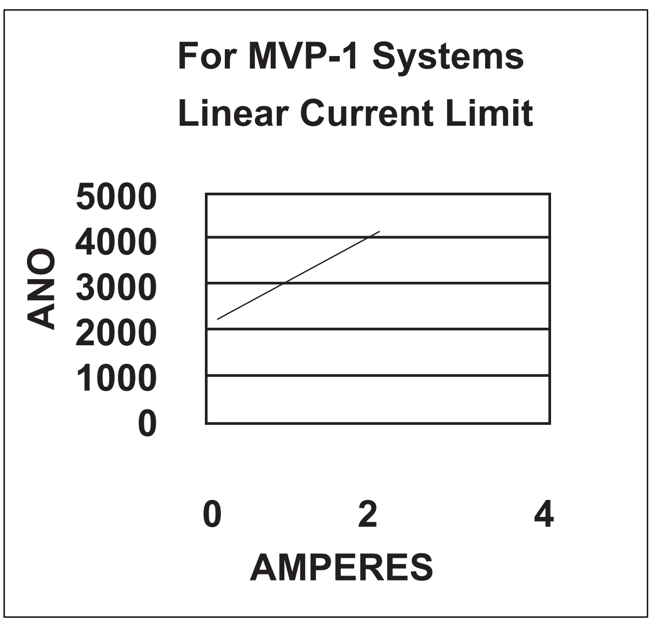
Once this procedure is done, you may run your terminal program, power up the MVP and watch for the sign-on message, "MVP 2001 ready." Then proceed to issue the commands in the "MicroMini™ Controller Commands" section as needed.

*Note: User must be familiar with his own terminal program.*

## Current Limiting and Torque Control

The **MicroMini™ Controller Servo Controller** provides programmable output current protection. The ANO command allows the user to program the maximum allowable continuous output drive current supplied to the motor. This feature provides not only short circuit protection for the internal drive circuitry, but also protects the motor from potentially harmful high currents that can occur during locked rotor conditions. The following table diagrams the programming parameters for the available output range for each device. The power up default setting is about 300 mA (ANO=2350 decimal).

Since the programmable output current capability is used in conjunction with the programmable following error limit feature, torque limiting or torque control mode operation is implemented. In this instance the servoloop should be configured for following error stall **(FA=1**). The default dynamic f value **(FD**) is set to 110 quadrature counts . When the following error exceeds the programmed amount the servo loop stops producing trajectory data and the motor stops. This contains the accumulated following error within the programmed limit. Since the motor current is limited, so is the available torque, or, if the following error allowed (FD) is insufficient to overcome a load on the motor (with MicroMini™ Controller in a given current limit/ANO state), the motor will stall and remain in servo. If the current is insufficient to allow the motor to produce enough torque to stay within the following error setting, the MicroMini™ Controller will stall in servo mode. Setting the following action to FA=1 will allow the motor to stall. If the motor is able to recover the following error, the trajectory generator will continue operation and the motor will servo normally. The dynamic following error is programmable using the FD command and both of these parameters are programmable on the fly to allow flexibility in applications where varying load conditions are a factor. The torque developed by a motor per ampere of input current is known as the torque constant ($K_T$ ). This definition is specific to




| | Amps | ANO | |
|---|---|---|---|
| Micro Mini™ Stage Range | 0.0 | 2048 | |
| | 0.1 | 2190 | |
| | 0.2 | 2277 | |
| | | 2350 | max. for 10 mm default |
| | 0.3 | 2364 | |
| | 0.4 | 2450 | |
| | 0.5 | 2537 | |
| | | 2600 | max for 13 mm motor |
| | 0.6 | 2623 | |
| larger motors as per-missible | 0.7 | 2710 | |
| | 0.8 | 2796 | |
| | 0.9 | 2883 | |
| | 1.0 | 2969 | |
| | 1.1 | 3056 | |
| | 1.2 | 3142 | |
| | 1.3 | 3229 | |
| | 1.4 | 3316 | |
| | 1.5 | 3402 | |
| | 1.6 | 3489 | |
| | 1.7 | 3575 | |
| | 1.8 | 3662 | |
| | 1.9 | 3748 | |
| | 2.0 | 3835 | |
| | 2.1 | 3922 | |
| | 2.2 | 4008 | |
| | 2.3 | 4095 | |

brush type and electronically commutated motors. This constant is temperature dependent and can be mathematically derived from the Back EMF constant ($K_E$) which can be found on the motor data sheet. FA=1 would be the safer drive to start, but be aware of possible stalls which don't allow programs to complete; while reporting a false completion status.

$$K_T[\text{oz-in/amp}] \simeq 1.35\ \text{x}K_E\ [\text{mV/rpm}]$$

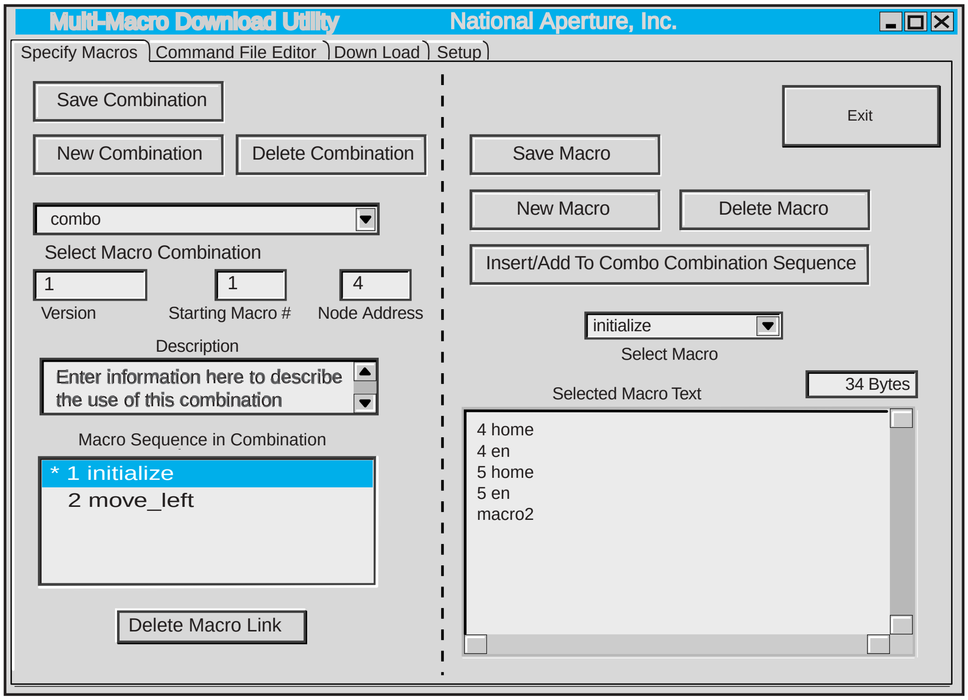
# Macro Module Operation (Optional Device)

The **Macro Module** enhances the stand-alone capabilities of the MicroMini™ Controller Series Controllers. In this configuration, programs developed using the MVP® Demo Software can be downloaded into non-volatile memory in a Macro Module contained within a MVP® Servo Drive. A single Marco Module is capable of controlling up to 63 axes of MVP® nodes. In addition to the ability to download multiple macro routines, individual macros may be "chained" together in any order providing additional flexibility. These "chained" groups of macros are called Macro Combinations. Once downloaded, macro routines may be activated at power up, serially, or through a DeviceNet™ poll connection. Macro routines provide a mechanism to implement configuration sequences, indexing operations, or variable motion profiles with minimal host interaction.

Event inputs (from the J2 connector ) can be used to flag the interruption and prompt continuation of macro routines, as these events have binary values that can be used as status masks. The status mask tells the program not to continue operation until it sees the correct status/binary value.

**The following pages will illustrate the Macro Module Setup and Download utilities.**

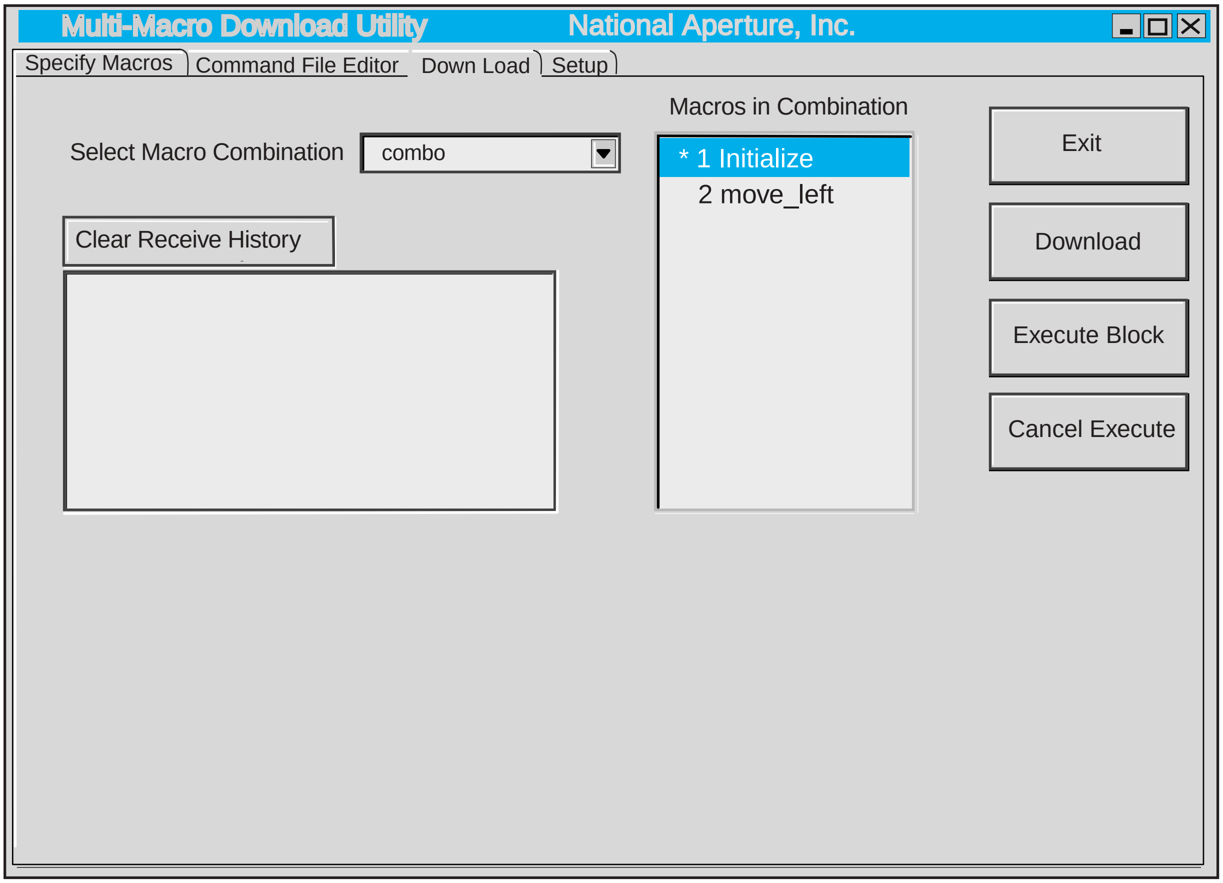
# The "Specify Macros" Screen




This screen is divided into two sections. The right (macro) side of the screen is used to develop the individual macro segments. MVP® Commands may be entered in the 'Selected Macro Text' window, and name the macro in the "Select Macro' window. Clicking on 'Save Macro' will save the series of commands under the selected macro name. Clicking on 'New Macro' will clear the text and macro name area. Each macro code sequence ***must*** end with the keyword 'Macro', followed by a number — (i.e. 'Macro2').[The name you give a Macro ("init" "zig zag", etc.) is **not** the thing that you place at the end of the code sequence; it is just a nickname which appears in the combination list.] The position in that combination is pre-numbered by the software, and *that* number is the suffix that is placed after keyword 'Macro' (Macro2, Macro3) at the end of the content of each macro. The assigned number is a pointer to the *next* macro to be executed. In the Example, 'Macro2' indicates that the second macro in the combination (whatever you named it) will be executed next. A Macro can therefore be renamed without changing the code (content) at all.A macro that ends with it's own position number will therefore loop forever. Specifying Macro 0 will terminate execution after the first pass. The left or combination portion of the screen is used to develop the combination of macros to be downloaded to the module. Type a name for your combination of macros into the 'Select Macro Combination' window, and click 'Save Combination'. This will establish a filename into which you will insert individual macros.

Select the macro(s) using the 'Select Macro' button in the right screen. Insert the macro into the Combination file by clicking on the 'Insert/Add To Combo..' bar. You will see the macro name(s) appear in the 'Macro Sequence..' window, in the correct order. The 'Version' and 'Description' windows are for your use as an optional means of controlling the revisions of the combinations created.The 'Starting Macro#' window contains the initial macro executed in the combination. This should be set to 0 if you do not wish to execute any macro routines at power up. Otherwise, enter the number of the macro you want to execute when the MVP® is turned on (normally, the first macro). The 'Node Address' should be the address of the MVP® module that the macro module is installed in. During initialization, this MVP® module will be defined as the "Network Master" for macro module operations.

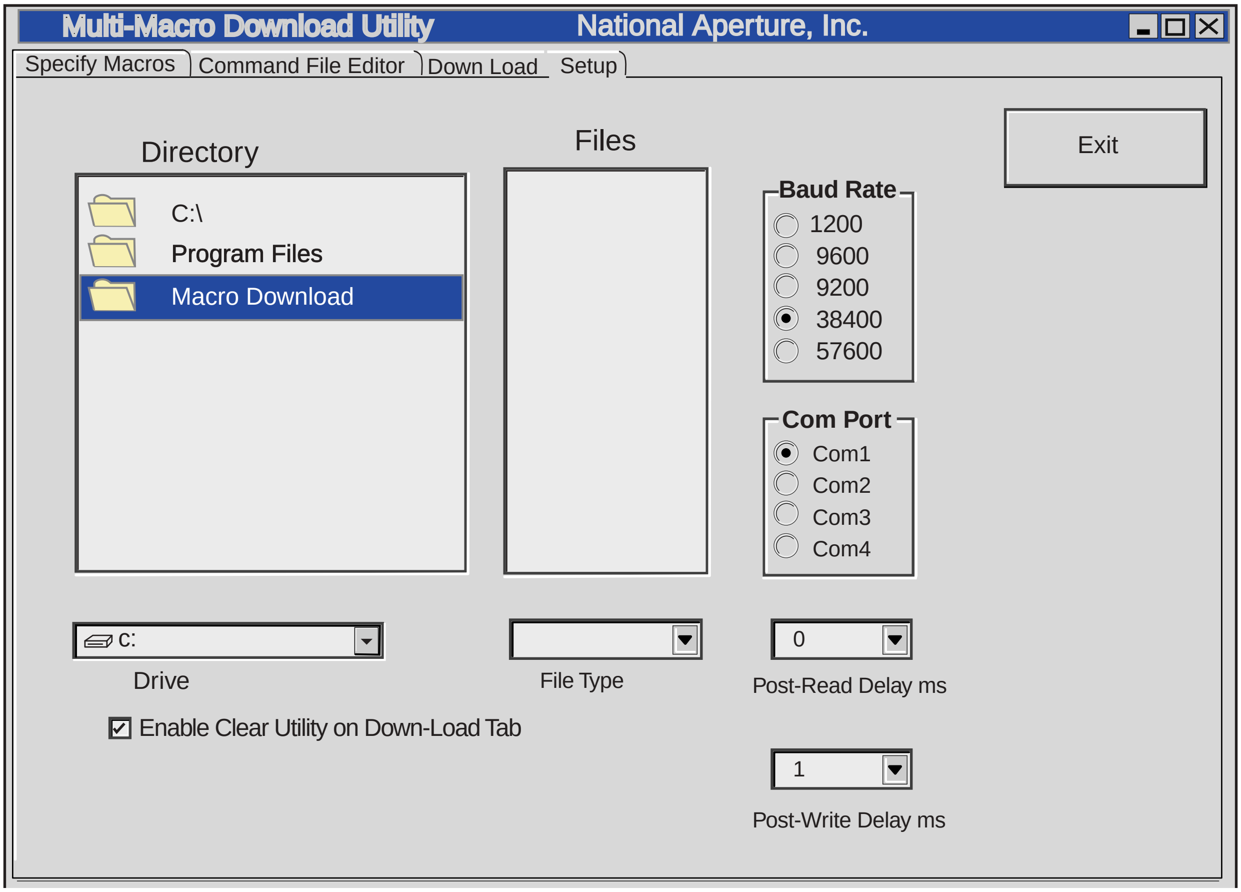
## The “Download” Screen




Selecting the ‘Down Load’ tab will bring you to the macro download screen. Here, you may choose the combination to load into the EPROM section of the Macro Module. Click on the arrow button in the ‘Select Macro Combination’ window , and select the desired macro combination. Now simply click on the ‘Download’ button to send the named combination of macros to the MVP® Macro Module. The progress of the download will be reported at the bottom of this screen, and you will be informed of the success or failure of the downloading process.

This screen also allows an individual macro contained in the selected combination to be activated for testing purposes. If the selected macro is chained, all macros in the chain will execute. The cancel execution button allows the operator to terminate the process.

# The “Setup” Screen




In the Setup screen, select the disk drive desired, the baud rate, and the serial com port you wish to use when downloading macros to the Macro Module. Initially set the ‘Post–Read Delay’ to 0 and the ‘Post–Write Delay’ to 1. The program will retain these settings until you change them in the future.

## The “Read back” Screen

By clicking the ‘Read E2prom’ button, the contents of the Eprom on the Macro Module can be displayed, with all the needed information and values. This is also an extremely useful tool, especially for verifying the success of your download procedure.

# Notes on Actual (real life) Application

The type of terminal serial input device you select will depend on your application, but any device capable of serial communications can be used for remote mode operation. It is the responsibility of the programmer to control access to the communications network in the event that more than one device is connected.
(*Note: The DeviceNet™ protocol defines it's own mechanism to control access arbitration*.)

While the end applications are extremely varied, almost all motion control applications share at least some common traits. The motion system needs to be initialized, configured, and the drive electronics enabled. These activities are normally accomplished at the beginning of the program and usually don't need to be repeated unless some exception has occurred and some possible exceptions will be discussed in a moment.

The initialization phase involves establishing communications with the various network nodes and defining an initial"Home" position for the controllers to operate from. This position usually differs from the actual "Home" position required by the application. Once the initial "Home" position has been established, the servo loop parameters should be loaded with values that provide the system response required by the application. The responsiveness of the control loop is highly application dependent and may even vary to some degree within an application program.This usually occurs when there are wide variations in load such as a vertical lifting application. Discussion regarding the configuration of the servo parameters is contained in Appendix H Once the filter has been configured, the drive can be enabled and the motor should begin to servo in the temporary "Home" position.*

At this point it is usually applicable to initiate a "Homing Routine" to prepare the device for it's intended task. In most cases where absolute system position isn't maintained once power is lost, it is desirable to locate a "Home Sensor" at one end of each axis range of travel. This will allow the "Home Routine" to seek the sensor in a known direction. Initiate a move in velocity mode in the direction of the sensor while polling the node for it's current status. A moderate velocity may be used to locate the sensor as more precise detection can be accomplished later. Testing the status responses will allow detection of a "Home Sensor" that has been connected to one of the External Event inputs provided on the external interface connector located on the rear of the unit. Utilization of these inputs allows this functionality to be implemented with little difficulty and minimizes the resources necessary to develop a complete applications solution. Inputs for hard overtravel limits as well as an emergency stop input are also available on the external interface connector. It is recommended that these inputs be used to detect serious error conditions that can develop from component failure or other fault conditions that require operator intervention to recover from. Once the "Home Sensor" is initially located, it is good practice to reverse the motor until the status input indicates that the axis has cleared the "Home Sensor". By again seeking the "Home Sensor" at a lower velocity, it will be possible to stop in precisely the same position each time the "Home Routine" is executed. Once the node is stopped in the "Home Position", and defined by the "Home" command, any offset can be invoked to position the device for it's intended application, and if necessary, another "Home" instruction can be issued to mitigate the need for maintaining a position offset in the host application program.

This process should be repeated for each axis in the system. Once the initialization is complete, the system is ready to operate in the intended application.

The system status responses provide a wealth of information about the condition of each node in the system. It is useful for controlling operational flow as well as detecting system level errors. When implementing a position move for example, it is normally good practice to ensure that a commanded profile begins to execute before attempting to detect whether or not it has finished. In a distributed control system, there can be a variety of reasons for an instruction being ignored by a node. These can range from a data collision on the network to noise, which may have been induced from an outside source. But without verifying the beginning of a move, and detecting move complete and motor in position instead, the host program controlling the application may be tricked into believing that the motor is in a position it hasn't moved to.

By following good installation practices and taking a logical approach to program development, most of the pitfalls normally associated with integrating an application can be avoided. The time and effort spent planning and developing a sound installation can eliminate many hours of debugging later.
* *National Aperture MicroMini™ Stages are equipped with home and limit devices along with appropriate wiring for use with the MicroMini™ Controller. Sample homing routines are also available for demo software supplied.*

## Status Response Bit Map

The MicroMini™ Controller provides you with 16 status bits when you monitor or request the system's status. This feature cannot be used in DeviceNet™ Mode. They are read from right to left with the following values and meanings:

| Binary Value | Bit # | Description |
|---|---|---|
| 1 | Bit 0: | 1=move in progress<br>0=not commanded to move |
| 2 | Bit 1: | 1=motor is in position<br>0=motor is not in position |
| 4 | Bit 2: | 1=MicroMini™ Controller is in Velocity Mode<br>0=MicroMini™ Controller is in Position Mode |
| 8 | Bit 3: | 1=indicates trajectory percentage defined by the "T" command is complete<br>0=Trajectory complete percentage not yet achieved |
| 16 | Bit 4: | 1=an error has occurred<br>0=everything is o.k. |
| 32 | Bit 5: | 1=module is in DeviceNet™ Mode<br>0=module is not in DeviceNet™ Mode |
| 64 | Bit 6: | 1=a DeviceNet™ message error has occurred in one or more packets<br>0=the DeviceNet™ message packets are o.k. |
| 128 | Bit 7: | 1=the current move is off its program trajectory by more than the allowed amount (which is set by the **FD** command)<br>0=current move is going o.k. |
| 256 | Bit 8: | 1=motor is not enabled or has been disabled by some other error<br>0=motor is enabled |
| 512 | Bit 9: | 1=you have reached the program range limit (set by the LL command)<br>0=move is within the range limits |
| 1024 | Bit 10: | 1=Local Mode is active<br>0=Remote Mode is active |
| 2,048 | Bit 11: | Emergency stop flag (1=active) |
| 4,096 | Bit 12: | External Event #1 (1=active) |
| 8,192 | Bit 13: | Positive Limit Flag (1=active) |
| 16,384 | Bit 14: | External Event #2 (1=active)* |
| -32,768 | Bit 15: | Negative Limit Flag (1=active)‡ |

} These flags signal the status of event inputs (Bits 11–15)

*This means Home switch is tripped and held, not home armed status.
‡The binary representation of values over 32,767 require negative sign (ex, -32,768)

**Input Events**
The MicroMini™ Controller is capable of reacting to, and on, external events which you provide to it as inputs. The inputs can be used for any particular event you define. The event sets a bit in the status bit indicator only; for example, using a "home" sensor to turn the motor on or off depending on the position of a mechanism. The bit flag does not perform an event on its own. An event happens only if you order it in your motion profile program.

**Positive and Negative** I-limits , hard are also recognized by the controller. These are typically error inputs. For example, one could use these inputs to stop a motor if an encoder were broken or if the encoder wires became damaged or disconnected. The hard limits are designed to remedy errors or alert the operator to events that are not planned or scheduled to occur.

# MicroMini™ Controller Commands

**See Examples and Explanations before attempting to use commands.**

## Motion Related Commands

| | | |
|---|---|---|
| **AA** | **Abort Action Codes** | **4 AA 2**<br>The Abort Action Codes permit selection of the behavior of the controller during an abort situation. Action code 0 disables the drive immediately. Setting the code to 1 causes the MicroMini™ Controller to perform a Hard Stop and attempt to servo in the present position. If action code 2 is selected, the MicroMini™ Controller will abort the profile at the programmed abort deceleration rate. |
| **AB** | **Abort Motion Command** | **4 AB**<br>Execution of this command terminates the present motion and the motor will ramp to a stop at the rate determined by the **AD** parameter. |
| **AC** | **Load Profile Acceleration** | **4 AC 100**<br>This parameter determines the acceleration and deceleration rate to be calculated by the profiler during execution. |
| **AD** | **Abort Deceleration Parameter** | **4 AD 200**<br>This parameter allows the user to define the abort deceleration rate. This capability permits a profile to be interrupted without loss of servo control by selecting abort action code 2, which is the power up default setting. (The higher the value the quicker the stop) |
| **DI** | **Disable Drive** | **4 DI**<br>Execution of this command disables the drive electronics and removes all current from the motor. With the exception of Local Mode Operation, the MicroMini™ Controller defaults to the disabled state on power up. |
| **EN** | **Enable Drive** | **4 EN**<br>This instruction enables the drive electronics and allows the servo controller to become active, providing motor current as required. |
| **FA** | **Configure Following Error Action** | **4 FA 1**<br>This parameter determines the action taken when the following error exception occurs. Action code 0 will disable the drive current. Action code 1 will cause the MicroMini™ Controller to attempt to servo in the present position, and action code 2 causes the MicroMini™ Controller to attempt to recover the accumulated error, but prevents any additional error accumulation. |
| **FD** | **Set Max Dynamic Following Error** | **4 FD 140**<br>This parameter allows the user to limit the amount of following error in a profile to a predetermined maximum. Excess following error is treated as defined by the **FA** parameter. |

**HA** **Arm the "HOME" input**

**4 HA 1**
The external event input on the J2 connector, pin 7 can be used as a "HOME" input that can be triggered by an external sensor or proximity switch. When used in conjunction with **HP** command, an automated homing sequence can be easily implemented. The current position will be set to "0" the instant the home switch is triggered. Setting the **HA** parameter argument to "0" will disarm the function. The input is also disarmed once it has been triggered.T*he Status Query in this particular command is achieved with the* **HS** *command*

**HF** **Homing Sequence Action Code**

**4 HF 1**
Configuration of this parameter provides the user the opportunity to select which behavior the controller implements during the "Homing" process. Action code 0 disables the drive amplifier when the home input is triggered. Setting the code to 1 will cause the MicroMini™ Controller to execute a "Hard Stop", assigning the present position as the HOME position when the input is triggered, and code 2 is the selection required if a "Soft Stop" is desired. The deceleration rate is defined using the **AD** parameter.

**HO** **Define the present position (Home)**

**4 HO (Argument optional)**
The HOME parameter provides a mechanism to define a known mechanical position a specific numerical position within the MicroMini™ Controller's operational space. Using this command without an argument will set the position to 0. If an argument is used, the new position will be defined by the numerical value of the argument.[1 HO means set pos to "0". 1 HO 242 means set pos to"242".]

**HP** **Define Home Arming Polarity**

**4 HP 1 (1 = Positive Logic, normally open**
The HP command allows the user to select positive or negative logic input signals to trigger the HOME **and** Error inputs. This flexibility eases installation into existing applications where the state of these signals is predetermined. The Home operation is performed upon the appropriate change of state of the selected input. *The Status Query in this particular case is achieved with the* **HS** *command*

**HS** **Query Home Arming Status**

**4 HS**
This allows the user to verify the status of the Homing Sequence. Bit 0 indicates the present status of the Arming function, and bit 1 reflects the current polarity setting. It is **important** to note that the polarity configuration affects the event inputs, as well as the overtravel and emergency stop inputs. *This an **exception** to the general rule of Status Query*

**J** **Set Local Velocity Range Multiplier**

**4 J 2**

This parameter is used to allow the user to select the <u>local mode</u> speed range. Since the encoder resolution and the value of the command voltage multiplied by a scaling factor determine the velocity, the velocity ranges indicated are using a 500 ppr encoder. (Not currently used by National Aperture MM-series stages)

**0 = 250 rpm (default)**
**1 = 1000 rpm**
**2 = 5000 rpm**
**3 = 10000 rpm**
**4 = 20000 rpm**

**K** **Select Continuous Integration**

**4 K 1**
Setting this parameter to 0 permits the user to disable the integral term of the digital filter.

| | | |
|---|---|---|
| **LA** | **Load Absolute Target Position** | **4 LA 1000**<br>Allows programming of the desired target position relative to the present zero or "home" position |
| **LL** | **Set Position Range Limits** | **4 LL 2000 or 4 LL -1300**<br>The range limit mechanism provides an additional layer of protection in applications where the network master calculates position data on the fly. Both positive and negative range limits are determined independently using the sign of the limit parameter. If the MicroMini™ Controller receives an instruction from the host to move to a position which is not within the range limit window, it will respond as directed by the **SA** parameter. |
| **LR** | **Load Relative Target Position** | **4 LR 1000**<br>Allows programming of the desired target position relative to the present position. |
| **LS** | **Limit Sequence Enable** | **4 LS 1**<br>The limit sequence defines the controllers response to the activation of one of the overtravel limit inputs. Setting the LS flag to 0 causes the MicroMini™ Controller to disable the drive amplifier upon limit input activation. Setting the LS flag to 1 causes the MicroMini™ Controller to permit motion only in the opposite direction until the activated input is cleared. Setting this flag to 99 disables limit action (used for National Aperture rotary stages . Rotary stages need to be hard-limit protected externally.) |
| **M** | **(Move) Initiate Motion** | **4 M**<br>Receipt of the "M" Move command initiates motion using the values presently loaded to determine the profile characteristics. |
| **N** | **Define the "In Position" Range** | **4 N 1 (0 not allowable)**<br>The value of the N parameter determines the size of the position window that the MicroMini™ Controller considers to be "In Position". |
| **O** | **Local/Remote Mode Flag** | **4 O 0**<br>This flag provides a mechanism for the user to switch from remote mode to local velocity mode, which provides velocity control using the analog input voltage as the velocity command signal. When used in conjunction with the **J** parameter, this functionality provides a convenient means of switching from programmable to manual control. Setting O to a 0 selects remote mode, and setting o to 1 selects local velocity control mode. |
| **SA** | **Select Range Limit Action** | **4 SA 2 (Code = 0 - 2)**<br>The Range Limit Action Code determines the MicroMini™ Controller response to a command directing motion outside of the defined range limits. Action code 0 disables the drive immediately. Setting the code to 1 causes the MicroMini™ Controller to attempt to servo in the present position. If action code 2 is selected, the MicroMini™ Controller will move to the programmed range limit position. |
| **S P** | **Load Maximum Commanded Velocity** | **4 SP 900**<br>This parameter should be loaded with the desired maximum profile velocity. This value should be selected to **not** exceed the capabilities of the mechanical system**.** |

| | | |
|---|---|---|
| **T** | **Set Percentage Trajectory Parameter** | **4 T 45**<br>This parameter allows the user to determine the percentage of the profile to be completed prior to the **Percent Complete Flag, (Status bit 4)** being set. If this parameter is set to 0, the Percent Complete Flag will not become set during profile execution. |
| **V** | **Constant Velocity Command** | **4 V 700**<br>Execution of this command causes the controller to switch into the Velocity control mode. The motor will ramp up or down at the selected velocity as defined by the **AC** parameter. Velocity mode can be used interactively with position mode. |

# I/O related commands

Serial I/O responses consist of the responding node address followed by the ASCII representation of the requested data parameter. All numerical responses are provided in hexadecimal notation.

Note that the node address is separated from the data by a "space" character. The string is terminated with a carriage return (13) and line feed(10) characters. **Example Status Response 0004 000A**

| | | |
|---|---|---|
| **ANI** | **Query Analog Input Value** | **4 ANI**<br>The selected MicroMini™ Controller node will report the current value of the analog input immediately upon receipt of this instruction. (16 bit signed response) |
| **ANO** | **Set Current Limit** | **4 ANO 2350**<br>The MicroMini™ Controller has programmable current limit protection. Programming the desired value to the analog output sets the current level. Refer to the charts diagramming the output current levels for the corresponding output voltages. If the fixed current limit option is selected on the MicroMini™ Controller front panel, the analog output is available for alternative application. |
| **POS** | **Query Present Position** | **4 POS**<br>The selected MicroMini™ Controller node will report the current position of the motor immediately upon receipt of this instruction. (32 bit signed response) |
| **ST** | **Query Present Node Status** | **4 ST**<br>The selected MicroMini™ Controller node will report the current node status immediately upon receipt of this instruction. (16 bit signed responses-see "Status Response Bit Map"). |

# Configuration Related Commands

| | | |
|---|---|---|
| **DER** | **Set the Derivative Loop Gain** | **4 DER -25536 (1K-32K)**<br>The derivative gain balances the loop error compensation against the rate of change of the loop error. Adjusting this parameter may be necessary to achieve optimal stability when using larger motors with longer mechanical time constants. |
| **I** | **Set Integral Loop Gain** | **4 I 600 (1 - 32K)**<br>Unlike the proportional gain, where the response remains constant if there is no change in error, the integral term continues to increase the effects of it's response until it becomes effective. This parameter determines the rate of change of this response. |

| | | |
|---|---|---|
| **POR** | **Set Proportional Loop Gain** | **4 POR 28000 (4K-32K)**<br>The proportional gain determines the systems' proportional response to a given amount of positional error. Increasing this parameter provides a tighter and more dynamically responsive system. |
| **RD** | **Reverse Operational Direction** | **4 RD 1**<br>Often the direction of rotation of the motor when provided a positive command position needs to be reversed. This parameter allows the user to reconfigure the system without the need to rewire the motor and encoder connections. |
| **RE** | **Reverse Encoder Phasing** | **4 RE 1**<br>This parameter allows the encoder phasing to be reversed to facilitate closure of the control loop. |
| **RN** | **Reset Node** | **4 RN**<br>This command causes the MicroMini™ Controller to perform a power on reset operation. |
| **SR** | **Set MicroMini™ Controller** | **4 SR 1000** |
| | **Loop Sample Period** | This command allows the user to program the sampling rate of the digital filter. Increased sample periods are useful in cases where a low-resolution feedback encoder is used or where very low velocities are desired. |
| **X** | **Synchronize Nodes** | **0 X**<br>This parameter arms the synchronization sequence in multiple MicroMini™ Controller devices. The process is triggered when the next global move (0 M) command is issued. |

## Indexing Commands

| | | |
|---|---|---|
| **IA** | **Indexing Acceleration** | **4 IA 100**<br>This parameter permits configuration of the indexing profile acceleration. |
| **ID** | **Index Destination** | **4 ID 2000**<br>his parameter permits the user to configure the destination position for an indexing operation. |
| **IE** | **Indexing Enable** | **4 IE 1**<br>The IE command provides a means of activating and deactivating indexing mode. Setting this flag to 1 activates indexing, and setting the flag to 0 restores normal operation. |
| **IM** | **Single/Continuous Indexing** | **4 IM 1**<br>Indexing operation may be either single step operation in response to an external pulse applied to J2, pin 8, or continuous operation. Setting this flag to a 1 selects continuous mode. |
| **IS** | **Indexing Velocity** | **4 IS 2000**<br>This parameter permits configuration of the indexing profile velocity. |
| **ITD** | **Index Destination Delay** | **4 ITD 2000**<br>This parameter allows the user to determine the dwell time at the index destination position. Each unit is equal to a 500$\mu$s-time delay. |
| **ITZ** | **Index Zero Delay** | **4 ITZ 2000**<br>This parameter allows the user to determine the dwell time at the index zero position. Each unit is equal to a 500$\mu$s-time delay. |

# Position Capture Commands

**CA** **Capture A Input** — **4 CA 1**

Setting the input capture flag enables capture operations on external event input #1, (J2-7). Setting the flag to 0 terminates capture operation.

**CB** **Capture B Input** — **4 CB 1**

Setting the input capture flag enables capture operations on external event input #2, (J2-8). Setting the flag to 0 terminates capture operation.

**POSCA** **Report Capture Position A** — **4 POSCA**

This command causes the MicroMini™ Controller to return the position captured when capture input A (J2-7) was last triggered.

**POSCB** **Report Capture Position B** — **4 POSCB**

This command causes the MicroMini™ Controller to return the position captured when capture input B (J2-8) was last triggered.

# Demo Software Command File Conventions

**ST mask** **Status response mask field** — **4 ST 10**

The inclusion of a status mask parameter following the **ST** command allows the demo software to compare the responses received with the anticipated response. The demo software program will continue to poll the selected node until the desired response is received, providing a means of implementing flow control in the demo environment.

**ST mask>label** — **4 ST 10>mainloop ST 9>loop2**

Adding a user created label name to the mask parameter (separated with a >) provides a means of implementing a logical goto operation for structured control. If the system status matches the mask, or any one of the masks, if two or more (as shown), a jump to the appropriate label will be performed. The words “mainloop” and “loop2” are user created and not embedded commands. The “mask” is a binary value from the Status Response Bit Map. The program will continue only after seeing that value, and it will go where pointed.

**Delay** **Timing Delay Function** — **Delay 100**

This command causes the demo program to pause for the specified time period. Each unit represents a 10-millisecond delay.

**Message** **Message Field** — **Message Start here**

The text in the message statement (“Start here” as in the above example) will be displayed in the message window during execution of the command file. This feature provides a useful means of following program execution.

**/* */ Comment Fields** — **4 HO /* Reset Axis 4 to “0”*/**

The slash asterisk allows comments to be inserted into command files. (Not macro downloads.)

# Appendices

# BASIC STARTUP PACKAGE CONTENTS

**1 Axis Configuration**
1 RS-232 Motion Controller
1 RS-232 Null Modem Cable
1 24VDC Power Supply
1 Stage Interface Cable (single axis)
Software Demo Diskette
MicroMini™ Controller User Manual

**2 Axis Configuration**
2 RS-485 Motion Controller
1 RS-485 Converter Module
1 2-axis Communication Cable
2 24VDC Power Supply
2 Stage Interface Cable (single axis)
Software Demo Diskette
MicroMini™ Controller User Manual

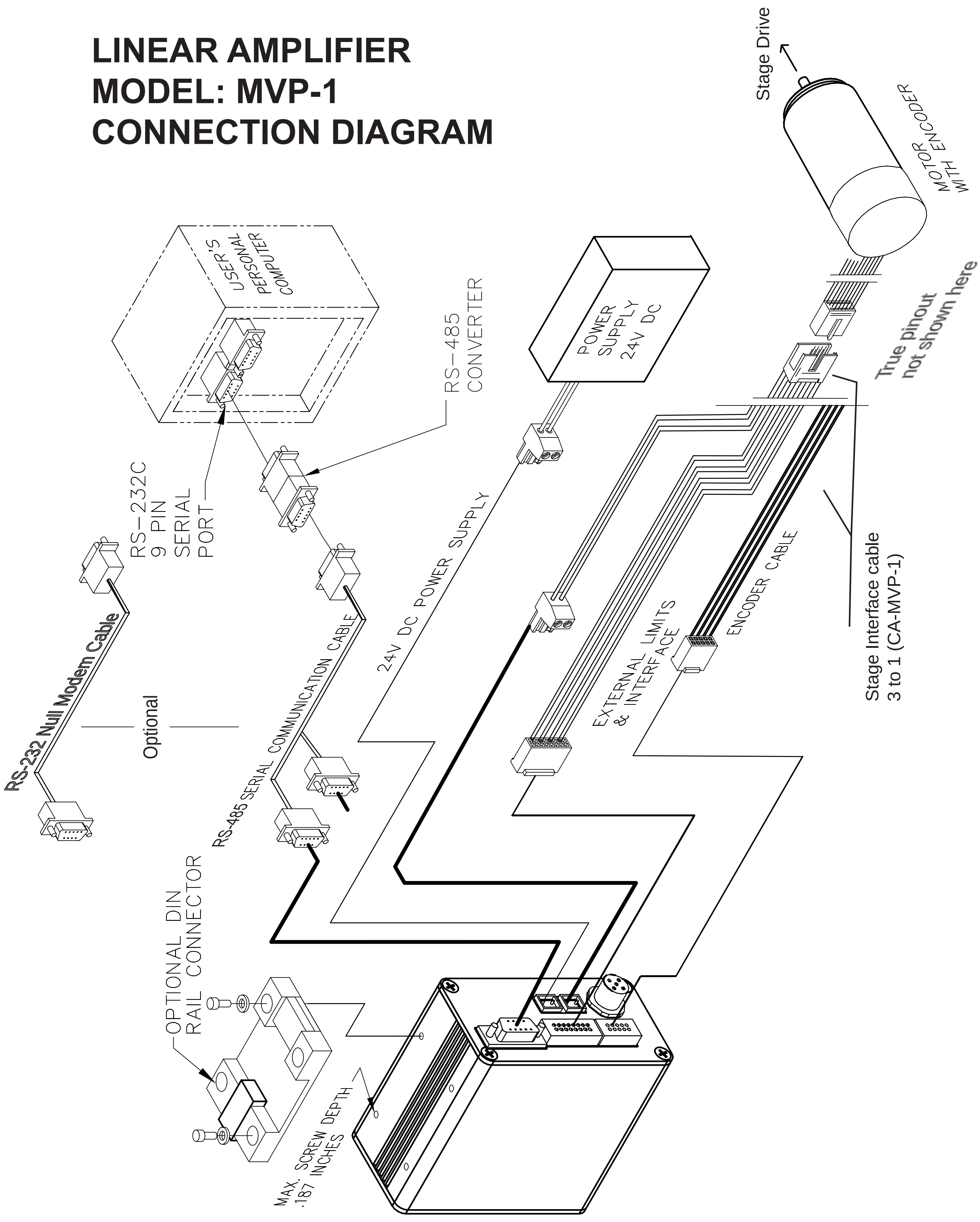
**3 Axis Configuration**
3 RS-485 Motion Controller
1 RS-485 Converter Module
1 3-axis Communication Cable
3 24VDC Power Supply
3 Stage Interface Cable (single axis)
Software Demo Diskette
MicroMini™ Controller User Manual

**4 Axis Configuration**
4 RS-485 Motion Controller
1 RS-485 Converter Module
1 4-axis Communication Cable
4 24VDC Power Supply
4 Stage Interface Cable (single axis)
Software Demo Diskette
MicroMini™ Controller User Manual

**OPTIONS**

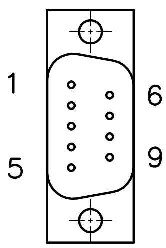
Macro Module (For stand alone applications)
RS-485 Serial Converters
DIN Rails (includes mounting screws and washers)
Custom Cabling
Connector Packs
Custom Software/Firmware
- Electronic Gearing
- Step-Direction Control
- LabView™ Virtual Instruments
- Example Source Code

## LINEAR AMPLIFIER MODEL: MVP-1 CONNECTION DIAGRAM



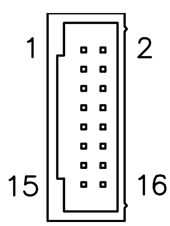
# External Connections

**J1 RS232C/485 Remote Communications Interface**




| | | |
|---|---|---|
| Pin | 1 | Not Connected |
| | 2 | RxD/RS485- |
| | 3 | TxD/RS485+ |
| | 4 | Not Connected |
| | 5 | Ground |
| | 6 | Not Connected |
| | 7 | Not Connected |
| | 8 | Not Connected |
| | 9 | Not Connected |

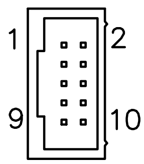
**J2 External Interface Connector**
**Micro Mini™ stage connector cable uses applicable pins**..




| | | | | |
|---|---|---|---|---|
| Pin | 1 | Analog Input Signal (0 to 5 VDC, 10 bits) | | |
| | 2 | Analog Reference Ground | | |
| | 3 | External Opto Power (+5 VDC) | | |
| | 4 | Emergency Stop Input | | |
| | 5 | Positive Hard Limit Input | Status bit 13 | |
| | 6 | Negative Hard Limit Input | Status bit 15 | |
| | 7 | External Event #2, Homing | Status bit 14 | also: position capture input, Status mask input, index input |
| | 8 | External Event #1, | Status bit 12, | also: position capture input, index trigger, etc. |
| | 9 | External Drive Enable (not applicable)<br>0 = disable<br>1 = enable | | |
| | 10 | DAC A Output (External Amplifier Command Signal)(not applicable) | | |
| | 11 | DAC B Output (Programmable Analog Output)(not applicable) | | |
| | 12 | (not applicable) | | |
| | 13 | (not applicable) | | |
| | 14 | (not applicable) | | |
| | 15 | +5 VDC | | |
| | 16 | Ground | | |

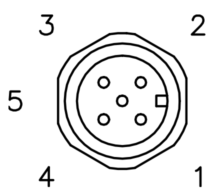
# Appendix C-2

**J3 Encoder Interface (HEDS 5000 Series Compatible)**



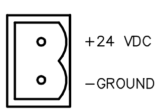

| | | |
|---|---|---|
| Pin | 1 | Encoder Channel A Input |
| | 2 | Encoder Power (+5 VDC) |
| | 3 | Encoder Return (Ground) |
| | 4 | Not Connected |
| | 5 | Encoder Return (Ground) |
| | 6 | Encoder Return (Ground) |
| | 7 | Encoder Power (+5VDC) |
| | 8 | Encoder Channel B Input |
| | 9 | Encoder Power (+5 VDC) |
| | 10 | Encoder Channel Z (index) Input |

**J4 DeviceNet™ Interface:** ***Not Applicable without Special DeviceNet™ setup*.**



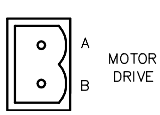

| | | |
|---|---|---|
| Pin | 1 | Drain (Case) |
| | 2 | V+ (Power Input) |
| | 3 | V- (Power Return) |
| | 4 | CAN_H (Communications Interface) |
| | 5 | CAN_L (Communications Interface) |

**J5 Main Power**




| | | |
|---|---|---|
| Pin | 1 | +24 VDC |
| | 2 | Ground |

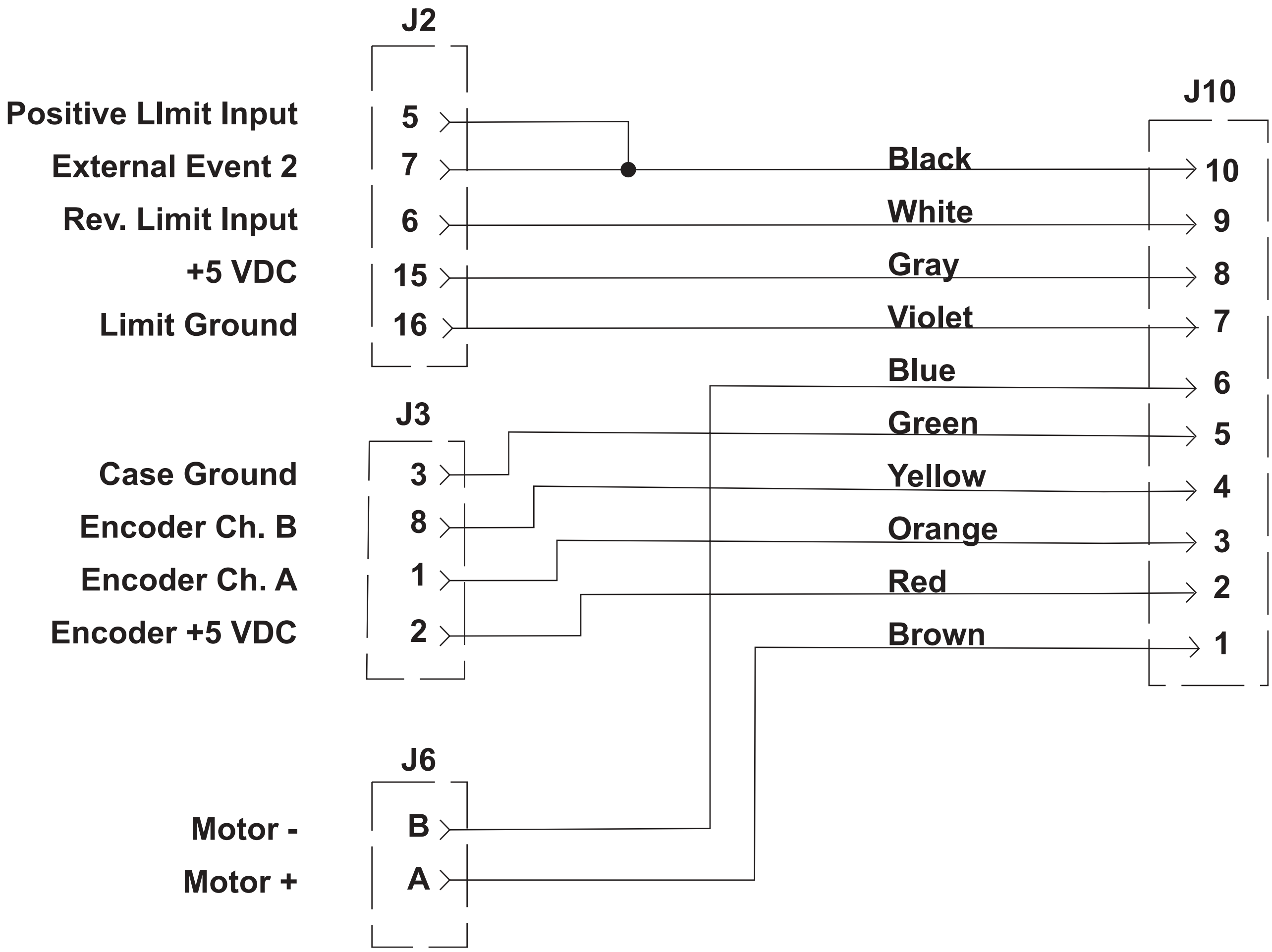
**J6 Motor Drive**

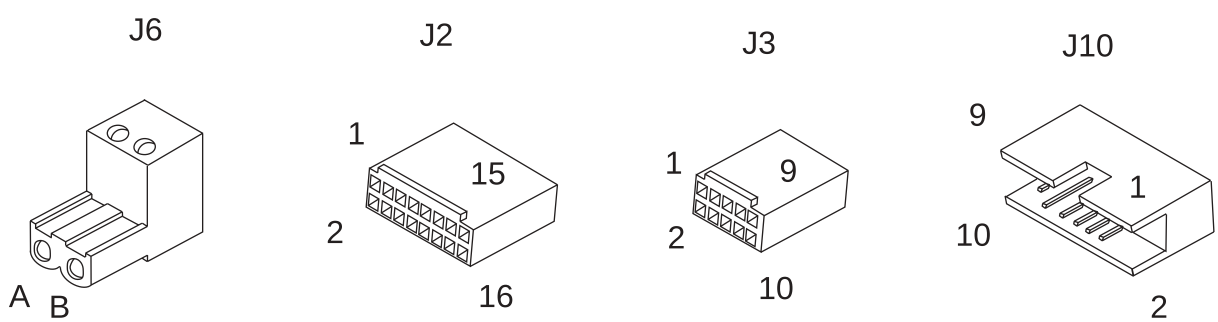



A

B

# MicroMini™ Stage Controller Interface Cable



# Troubleshooting

Should you encounter difficulties with the operation of your MicroMini™ Controller, remove power from the unit and verify all connections as outlined in Appendix C of this manual and reapply power.

Other Troubleshooting instructions cannot be compiled until further troubles are encountered. At this point there are no “typical” problems besides inactive serial port or wrong selection of RS-232C or RS-485 operation.

If a generic MVP® other than National Aperture, Inc. MicroMini™ Controller is used, or if correct NA cabling is not used, there will be unsupportable trouble.

# FREQUENTLY ASKED QUESTIONS (FAQ'S)

**What does a MicroMini™ Controller do?**

Any MicroMini™ Controller module can provide single axis motion, position, and velocity control as well as torque limiting of DC motors. Using the on-board integral amplifier, motors up to about 100 Watts can be controlled. Using external amplifiers motors into the integral horsepower range can be controlled regardless of who they are made by. Communication between the module and the host can be accomplished via serial RS-232, RS-485 Multidrop, or DeviceNet™ protocols.

**Who do I call to discuss a specific application?**

Contact National Aperture at 1(603)893-7393 and ask for MicroMini™ Controller Technical Assistance.

**How reliable is a MicroMini™ Controller?**

Every MicroMini™ Controller unit built undergoes stringent burn-in testing to assure high quality and reliability. National Aperture's engineers have years of experience in the motion control business, ensuring that the MicroMini™ Controller design incorporates state of the art, proven, reliable technology. If you require detailed information before specifying, including results of our thermal, vibration and mechanical shock, electrical and accuracy tests, contact National Aperture's Marketing Department for a copy of the complete validation report.

**What sort of accuracy can I expect with a MicroMini™ Controller?**

A system's potential accuracy is determined by a number of factors in a servo system. The encoder resolution and the ratio of the reduction gearing, if any, as well as the mechanical design of the drive system, are the primary considerations. Servo loop bandwidth should also be addressed when incorporating a high resolution encoder in a high speed application. National Aperture has done applications using MicroMini™ Controller Series controllers that provide 0.5$\mu$ accuracy and 0.25$\mu$ repeatability using our low resolution magnetic encoders with our coreless motors. Testing in similar micropositioning applications with our new high resolution magnetic encoders yields results which are an order of magnitude more precise. The primary obstacle to precise positioning is usually mechanical backlash in the gearing system or torsional resonance at the interface of the moving parts in the system. National Aperture offers some gearhead series in zero backlash versions for sensitive applications. Backlash should also be compensated for in software.

**My system requires that on startup it must be in a defined position.**
**Can I do this with a MicroMini™ Controller?**

Yes. The MicroMini™ Controller provides a "homing" feature that allows the user to always return to a predetermined position when commanded to do so.

**How fast will a MicroMini™ Controller respond to an external stimulus?**

The MicroMini™ Controller provides two external event inputs, two hard limit inputs, and one emergency stop input. The MicroMini™ Controller can respond to any of these inputs in approximately 500 microseconds.

**How can I synchronize my MicroMini™ Controller to external events?**

The MicroMini™ Controller provides two external event inputs. These inputs are monitored and their states are reported to you. When one of these inputs changes, you can then command the MicroMini™ Controller to perform any function.

**What safety features does the MicroMini™ Controller have?**

The MicroMini™ Controller provides several safety features. The MicroMini™ Controller offers software configurable range limits and three hardwired inputs for overall travel limits and emergency stop. The MicroMini™ Controller can also be configured with action codes to provide "servo-off", "hard stop", and "soft stop" functions when limits are exceeded or when following error occurs. The MicroMini™ Controller also has programmable current limit (ANO command).

**At what point is it cheaper for me to buy a multiaxis system instead of stringing a number of MicroMini™ Controllers together?**

The cost per axis curves never really cross. The decision to move to a centralized multi-axis solution is usually driven by performance and architectural considerations. If the application requires tightly coordinated two axis motion, a high performance multi axis solution may be in order. But if the requirement is simply to operate 20 conveyer lines, an X-Y stage, to coordinate the processing of products with PLC-based I/O, or coordinate sequenced activities, then a distributed single axis implementation using 1 or more MicroMini™ Controllers should be very competitive.

**Can you provide me with a packaged, preconfigured solution?**

In cases involving systems using National Aperture MicroMini™ stages, yes. Our Technical Department can configure for special applications as well.

**What operating mode choices do I have?**

In the "Local" operating mode (with no remote host), the MicroMini™ Controller provides velocity or position control using potentiometer input. Both relative and absolute operation are supported. Additionally, encoder following with scaling is also available. In the "Remote" mode (using a remote host), the MicroMini™ Controller operates as a slave to the master device and depends on the host for instructions. This behavior is consistent with the design constructs for distributed control architectures. Local Mode applications range from simple pump operations, to tracking an analog input signal to position a target in an ion beam. Remote Mode allows networked operation of up to 64 MicroMini™ Controller modules in an RS-485 Multidrop or DeviceNet™ environment.

**When using "Local Velocity" mode, how can I change the MicroMini™ Controller's velocity Range?**

In "Local Velocity" mode, the MicroMini™ Controller's range is set using dip switches 7 through 10. Each switch has a particular range "scale" associated with it when it is turned LEFT. The scaling is as follows:

Switches7SWITCHES8 Switches9Switches10
100050001000020000(RPM)

*ALL SWITCHES **RIGHT** IS 250 RPM

**examples**

To set a range of 5000 rpm, only Switches8 would be turned LEFT.
To set a range of 20,000 rpm, only Switches10 would be turned LEFT.

**What is the difference between absolute and relative positioning?**

Absolute positioning implies that you want to move a load, motor shaft, etc. to a predetermined point from "home" or "base" position (usually from position 0). Relative position implies that you want to move a given distance from your current position.

**I'm confused. When should I use RS-232, RS-485 or DeviceNet™?**

Which way you decide to communicate and control the MicroMini™ Controller is dictated by the type of application you need to do. Traditionally, RS-232 is used in applications involving only two devices (usually a PC and a MicroMini™ Controller) over relatively short distances (up to 32 feet). RS-485 is used when many devices (up to 63 devices) need to communicate over the same medium — usually a twisted pair cable — and the distance between devices is relatively long (up to 4,000 feet). DeviceNet™ is a network that provides connections between simple industrial devices (like sensors and actuators) and higher level devices (PLCs and controllers). All MicroMini™ Controllers are able to communicate with other DeviceNet™ modules but do so only when the MicroMini™ Controller is part of a DeviceNet™ network.

**Can a MicroMini™ Controller control linear motors?**

Yes. As with any closed loop system, however, an encoder or other type of feedback device is still required, and the brushless version of the controller.

**Does the MicroMini™ Controller accommodate resolver feedback?**

Yes. No modifications are required to use the MicroMini™ Controller in a resolver-based system or with other types of feedback devices as long as the feedback device used provides quadrature feedback. This is not uncommon with most types of high performance feedback devices.

**How do I know if the MicroMini™ Controller is operating?**

One indication that a MicroMini™ Controller module is working properly is the diagnostic sequence performed when power is first applied to the unit. You will know the unit is performing this diagnostic sequence by observing the sequential turning ON and OFF of all LED indicators at power up.

**I've hooked up the MicroMini™ Controller but the motor just runs away. What do I do?**

Recheck all motor connections and make sure the encoder is powered and properly connected. Recheck mode configuration dip switches.

**How do I know if the MicroMini™ Controller is communicating with my serial interface and the proper baud rate has been selected?**

If remote mode or serial communication is being used, a serial message "MVP2001 READY" will be received after the diagnostic sequence is performed.

**How do I know if I need custom software for my application?**

The MicroMini™ Controller was designed to be flexible and to provide many useful features, but if you feel your application requirements cannot be met with the present MicroMini™ Controller functions, please contact National Aperture at 1(603)893-7393 and ask for "MicroMini™ Controller Technical Support". Our engineering staff will be happy to go over your application in detail, and provide effective solutions to your problems.

**How do I perform a diagnostic sequence if the MicroMini™ Controller is not performing the commands I just sent it?**

The MicroMini™ Controller provides systems information when the "ST" or status command is used. A status response looks something like this:

0004 010A

The first part of the message (0004) is the node address, in this case the node address is 4. The second half of the message (010A is the system status response represented by a hex number. The 16 bit representation would be 0000000100001010 Binary. Page 21 of the operator's manual describes the meaning of each bit. In this example, the "motor is in position,""command was recognized," but the "motor is not enabled"(since bit 8 is one).

**How can I determine the present system settings?**

You can enter any command without an argument to verify the present setting.

**How do I set the Node address in remote mode?**

In remote mode, switches 5 through 10 set the node address for serial communications and DeviceNet™ communications. the address becomes the total sum of all switches selected. Each switch has a particular "scale" associated with it when it is turned LEFT

The scaling is as follows:

| Switch 5 | Switch 6 | Switch 7 | Switch 8 | Switch 9 | Switch 10 |
|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 32 |

examples:

-to set address 1, only, Switch 5 would be turned LEFT
-to set address 4, only, Switch 7 would be turned LEFT
-to set address 10, only, Switch 8 and Switch 6 would be turned LEFT
-to set address 20, only, Switch 9 and Switch 7 would be turned LEFT

**What is the difference between "V" and "SP" velocity commands?**

The "V" command activates the velocity mode and sets the speed at which the motor will run. The "SP" command sets the velocity at which the motor will run when a "move" in position mode is performed.

# Examples

**Applications Examples**
The following programs illustrate the proper mechanism for initializing and operating the MicroMini™ Controller using example node address = 4.

**The "4 st 10" is a status mask for executable files; not manual entries.**

/* indicates comments that follow this sign will not be sent or executed.
The recommended initialization sequence can be implemented as follows:

```
4 ano 2350      /*Sets a maximum current
4 por 28000     /*Configure Proportional Filter Term
4 i 600         /*Configure Integral Filter Term
4 ho            /*Define temporary "HOME" location
4 en            /*Enable the drive amplifier
4 st 10         /*Verify System Status (refer to status bit definitions)
```

Once the initialization has occurred and operation has been verified, the MicroMini™ Controller is ready for operation under host control. The following instruction sequence demonstrates a motion operation where the motor is commanded to move 1000 encoder counts and then return to the "Home" location.

```
4 la 1000       /*Load an absolute target position of 1000 counts
4 sp 300        /*Set the target velocity
4 ac 400        /*Set the target acceleration
4 m             /*Initiate the motion
4 st 10         /*Use status to confirm that the motion has completed
4 la 0          /*Load an absolute target position of 0 counts
4 m             /*Initiate motion using previously defined SP & AC
4 st 10         /*Use status to confirm that the motion has been completed
```

Note that the mask values associated with the status requests are not sent to the controller, but are used by the demo program to test for the desired response. The demo software will continue to poll the node until the desired mask response is received, indicating that the anticipated conditions have been met. This same testing mechanism should be implemented in the users application program in order to control instruction flow. Multiple masks may be used with labels to separate events.

**Page 1 of 3, Rotary Stage Homing and explanations**

/*Limit switches disabled: release the jumper at pin 5 on the J2 connector for external limits - for MM-3MR 64:1 gear 10mm mtr. axis1*/
message Set servo values for axis 1
**1 ano 2350** sets the current limit to the max allowable for a 1016 motor
**1 k 1** maintains integral gain during move: lowers the following error by strengthening "keep-up" force
**1 ad 200** abort deceleration: lower numbers give slower deceleration
**1 aa 1** abort action: hard stop & remains in servo
**1 fa 1** following error action: hard stop &remains in servo
**1 fd 130** following error allowance in positive rotary direction (position mode, speed set by sp)
**1 sr 1000** sample period in u-sec: longer for lower resolution encoders. The encoder won't update until the next sampling: Longer periods will allow slower speeds. 1000 is optimum for 10-place encoder at higher speeds.
**1 sp 1200** speed: sp[1200] x 960,000 ˜ sample period[sr] ˜ counts/rev[40, here] = motor rpm[28.800]
**1 ac 400** set acceleration
**1 por 28000** set proportional gain
**1 i 600** integral gain, that which brings the motor into target position
**1 ls 99** limit action: 99=limit disabled for rotary stage homing
**1 hp 1** homing polarity 1 = positive logic, normally open
**1 n 5** dead window = 5 for quicker action between stops; n = 1, minimum for maximum accuracy at short-duration stops
**1 ho** sets target position which must be equal to current position: "0" if no argument
**1 en** enable servo (ho must be called first)
**1 st 16394>pullaway st 16392>pullaway st 16648>pullaway st 14>home st 10>home** wait for correct status before proceeding: 16394 if in position mode and in position, 16392 if motor is not yet in position, 16648 just in case rebooted while sitting at switch. Velocity mode ends when home is hit.
**pullaway:** an arbitrary name for the section of code causing release of the homing switch if execution started while resting on switch
message Pull away from home switch
**1 v -100** pull away from switch slowly in the negative direction
**1 st 13** wait for switch release by seeking status of motion in v-mode without other events active
**delay 25** time to get a little away from the switch to avoid chatter
**1 v 0** stop v-mode motion
**home:** an arbitrary name for the commencement of the homing routine after correct switch-release, if needed
message Arm home switch
**1 ha 1** arm home switch event #2, pin 7, J-2 connector
**delay 100** time (1 second) to read the last message
message Go to home in v-mode
**1 fd 100** /*for v-mode, you need a different following error ñ optional for maximum protection*/
**1 v 700** this velocity mode setting gives maximum speed
**delay 150** time to read last message if starting out already close to home switch
message Wait for st 16394(trip home switches) /* 16394 means home event switches active, motor in position, all else ok */
**delay 100**
**1 st 16394>section2 st 16392>section2** 16394 if in position mode and in position, 16392 just in case motor is not yet in position
**delay 100**
**section2:** /* go here after tripping home switch */
message st 16394/2 rec'd; proceed to ctr.
**1 fd 130** following error needs to be larger to prevent stall in pos mode
**1 la -102000** set dest in reverse 102000 encoder counts
**1 m** go there
**1 st 10** finish the last move before you go to the next command
message Arrived at home destination
**delay 100**
**1 ho** reset the current position to ì0î
**1 la 0** reset the current destination to ì0î (just in case someone hits a move command it won't go flying another 102000 counts back)
message Controller has reset to zero
**1 pos** display/verify the current position
**1 pos** a little time to settle into position, and display again
**delay 100**
message Home axis 1 successful
**delay 150** 1.5 seconds to read the last message

---

NOTES:
**Bolded black is used to designate code allowable for macro module download.**
Blue indicates user comments actually allowable in code in the MVP Demo .cmd files, but not allowable in the Macro module.
Red is used for comments not allowable in any running code file. It is strictly for tutoring and/or explanation.
Green indicates messages. They will work in command files, but will not work in the Macro module.

# Appendix G-3

**Page 2 of 3 Rotary Stage Homing code for .cmd demo execution**

```
/*Homing for MM-3MR 64:1 gear 10mm mtr. axis1 10-24-99*/
/*No limit switches involved: release the jumper at pin 5 on the J2 connector for external limit application*/
message Set servo values for axis 1
1 ano 2350
1 k 1
1 ad 200
1 aa 1
1 fa 1
1 fd 130
1 sr 1000
1 sp 1200
1 ac 400
1 por 28000
1 i 600
1 ls 99
1 hp 1
1 n 5
1 ho
1 en
```

servo-settings and basic configurations

```
1 st 16394>pullaway st 16392>pullaway st 16648>pullaway st 14>home st 10>home
pullaway:
message Pull away from home switch
1 v -100
1 st 13
delay 25
1 v 0
home:
message Arm home switch
1 ha 1
delay 100
message Go to home in v-mode
1 fd 100 /*for v-mode, you need a different following error ñ optional for maximum protection*/
1 v 700
delay 150
message Wait for st 16394(trip home switches) /* 16394 means home event switches active, motor in position, all else ok */
delay 100
1 st 16394>section2 st 16392>section2
delay 100
section2: /* go here after tripping home switch */
message st 16394/2 rec'd; proceed to ctr.
1 fd 130
1 la -102000
1 m
1 st 10
message Arrived at home destination
delay 100
1 ho
1 la 0
message Controller has reset to zero
1 pos
1 pos
delay 100
message Home axis 1 successful
delay 150
```

---

NOTES:

**Bolded black is used to designate code required for .cmd file execution.**

Blue indicates user comments allowable in the MVP Demo .cmd files. [not allowable in the Macro module]

Red is used for text not allowable in any running code file.

Green indicates messages. They will work in command files, but will not work in the Macro module. These commands can be eliminated, along with the delays that allow time to read them.

<u>(cont'd)  Page 3 of 3      Rotary Stage Homing for Macro Module</u>

Homing for MM-3MR 64:1 gear 10mm mtr. axis1 10-24-99  For use with Macro Module.
No limit switches involved: release the jumper at pin 5 on the J2 connector for external limit application
The servo-settings alone, or the following entire code, may be used as the boot-up file (“start” macro).

```
1 ano 2350
1 k 1
1 ad 200
1 aa 1
1 fa 1
1 fd 130
1 sr 1000
1 sp 1200
1 ac 400
1 por 28000
1 i 600
1 ls 99
1 hp 1
1 n 5
1 ho
1 en
```

**servo-settings and basic configurations**

```
1 st 16394>pullaway st 16392>pullaway st 16648>pullaway st 14>home st 10>home
pullaway:
1 v-100
1 st 13
delay 25
1 v 0
home:
1 ha 1
1 fd 100
1 v 700
1 st 16394>section2 st 16392>section2
section2:
1 fd 130
1 la -102000
1 m
1 st 10
1 ho
1 la 0
```

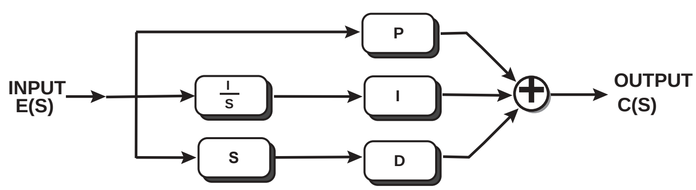
---

NOTES:
**Bolded black is used to designate code used for macro module download.**
Red is used for text not allowable in any running code file.

# The Servo Loop

## Digital Filter Configuration

Since the proper configuration of the digital filter parameters are crucial to achieving a table systems, it is necessary to have some understanding of the operation of PID control topologies. A block diagram of an analog PID control structure is illustrated below. The PID transfer function (as a function of **s**) is:



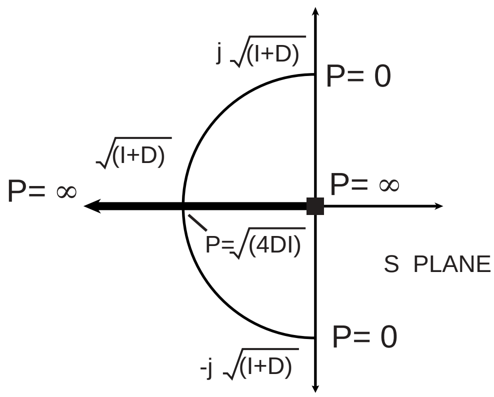

$$\frac{C(s)}{E(s)} = \frac{Ps + I + Ds^2}{s}$$

where **C**(s) is the output of the PID section
**E**(s) is the input of the PID section (usually servo error)
**P** is the multiplier for the servo error
**I** is the multiplier for the integral of the servo error
**D** is the multiplier for the derivative of the servo error
**s** is the laplace complex frequency variable

From the previous equation, it can be seen that the PID controller has a pole at **s**=0, and two zeros at:

$$s = \frac{-P \pm \sqrt{P^2 - 4DI}}{2D}$$

The two zeros are real-valued when 4**DI** ≥$\mathbf{P}^2$. A bode plot of the PID transfer function with real valued zeros reveals that one of the zeros is used to brake the 20 dB/decade descent associated with the integrator, and the other one is used to provide a 20dB/decade rise and positive phase lead required to stabilize the system.

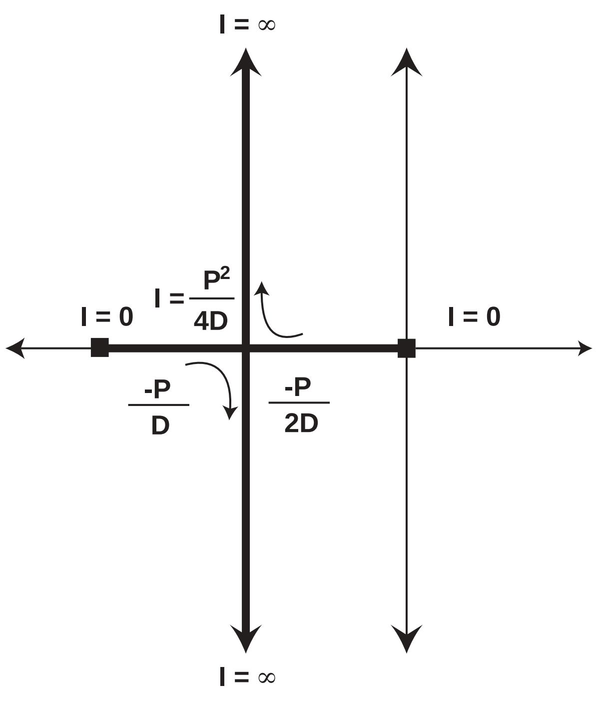


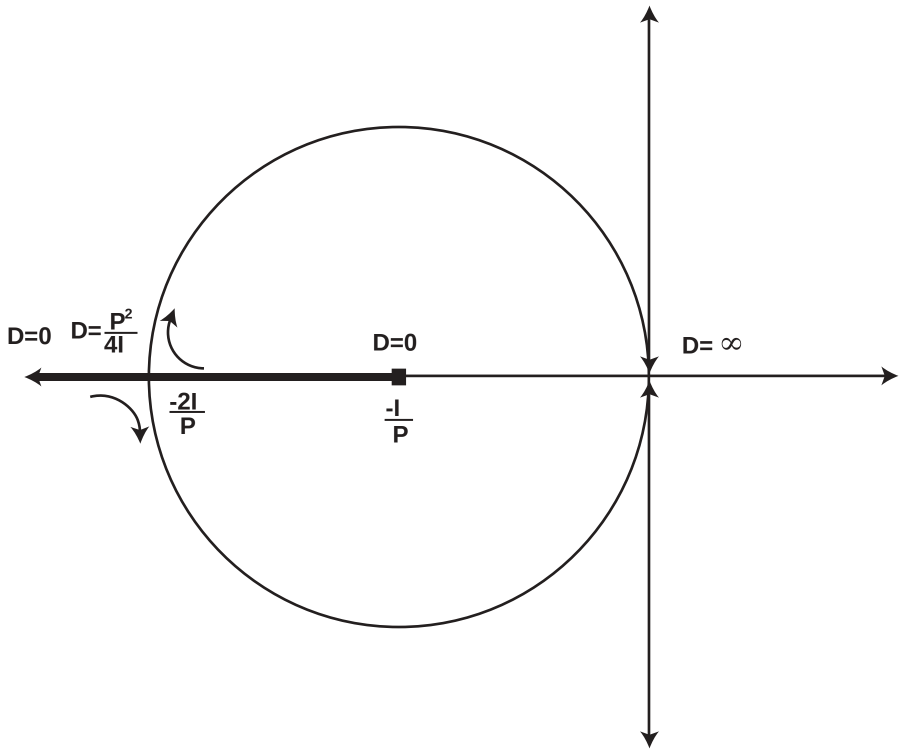

The **Proportional** term amplifies the error signal by a constant amount. However, the **P** term is not in series, but in parallel with **I** and **D**, which implies that **P** cannot be used to scale the transfer function amplitude at all frequencies. Instead, the **P** term interacts with the **I** and **D** terms to determine the placement of the zeroes in the controller open-loop transfer function. A root locus solution to the numerator of the previous equation as **P** is varied with respect to **I** and **D** is illustrated here.

The **Integral** term gives the servo loop that inflexible, stubborn feel. Since the **I** term adjusts the amount of integrated error mixed to the output of the filter, and the value other than zero implies that **no** steady state error can be tolerated by the servo loop. Given sufficient time, the PID control loop will eventually servo the output to the exact value of the commanded input.

In the frequency domain, the **I** term also affects the placement of the zeros as demonstrated below. For **I** = 0, one of the zeros is at **s** =-**P/D**, and the other zero is at **s**=0, which means that it will cancel the integrator pole at **s** = 0. This makes sense intuitively since the integrator is turned off if **I** equals zero. As **I** increases, the servo loop becomes "snappier" and responds more quickly to steady state error.

While adding an integrator does address the issue of steady state error, it can also have a negative impact on the system dynamics. The effect is most easily seen in the time domain. In a linear PID system that performs servo control, the controlled motor is initially at rest, with a zero position error. When torque is applied to the motor shaft changing it's position, the control system senses a steady state error and tries to return the shaft to the commanded position. Since in this example the system is linear, the control voltage will continue to increase as a result of integrated error. While increasing the control voltage could cause the motor to overheat if enough torque cannot be generated to overcome the error, this is not the only possible detrimental effect. If applied torque is suddenly removed while the integrator output is large, the motor shaft will spin past the desired shaft position while the control voltage is "dumped". Eventually a zero steady state condition is achieved, but in an underdamped (and potentially unacceptable) manner. Because this situation is similar to winding up a spring and then letting it go, the term wind-up is often used to describe it.






The **Derivative** term has the greatest effect on servo loop damping and stability. As demonstrated below, increasing the value of **D** from 0 to $\mathbf{P}^2/4\mathbf{I}$ causes both zeros to move toward -2**I**/**P**. As this happens, the higher frequency zero takes on a value that can provide useful phase lead to offset the phase lag introduced by poles elsewhere in the system. In a position servo, the feedback position signal is differentiated (either directly or indirectly) to create a signal proportional to the output velocity. In systems that use digital feedback mechanisms such as shaft encoders, velocity information is also quantized, typically in encoder counts per sample period. At low velocities, the effects of quantization on system performance is pronounced because each quantization step represents a large portion of the velocity signal amplitude. The effects of this quantization error can be mitigated to some degree by extending the filters amplifying period.

Digital PID controller transfer functions are calculated in much the same way as in analog systems, but because they are sampled systems, the Laplace transform of the **s** domain cannot be directly used as in analog calculations. To mitigate this problem, a separate frequency space called the **z** domain has been developed for sampled systems. Using the **z** domain, sampled approximations of many common functions can be represented using the variable "**z**" just as "**s**" is used to represent linear analog functions. Consider the following definitions.

$$\text{Integrator} = \frac{T_z}{z-1}$$

where: **z** is the complex sampled frequency variable
**T** is the sampling frequency period, in seconds.

This form is derived from a step-invariant analysis..... The filter is constructed by dividing the **Z** transform of a specified input (a step function) into the **Z** transform of the desired output for that input (a ramp function).

The differentiator is simply the inverse, or:

$$\text{Differentiator} = \frac{z-1}{T_z}$$

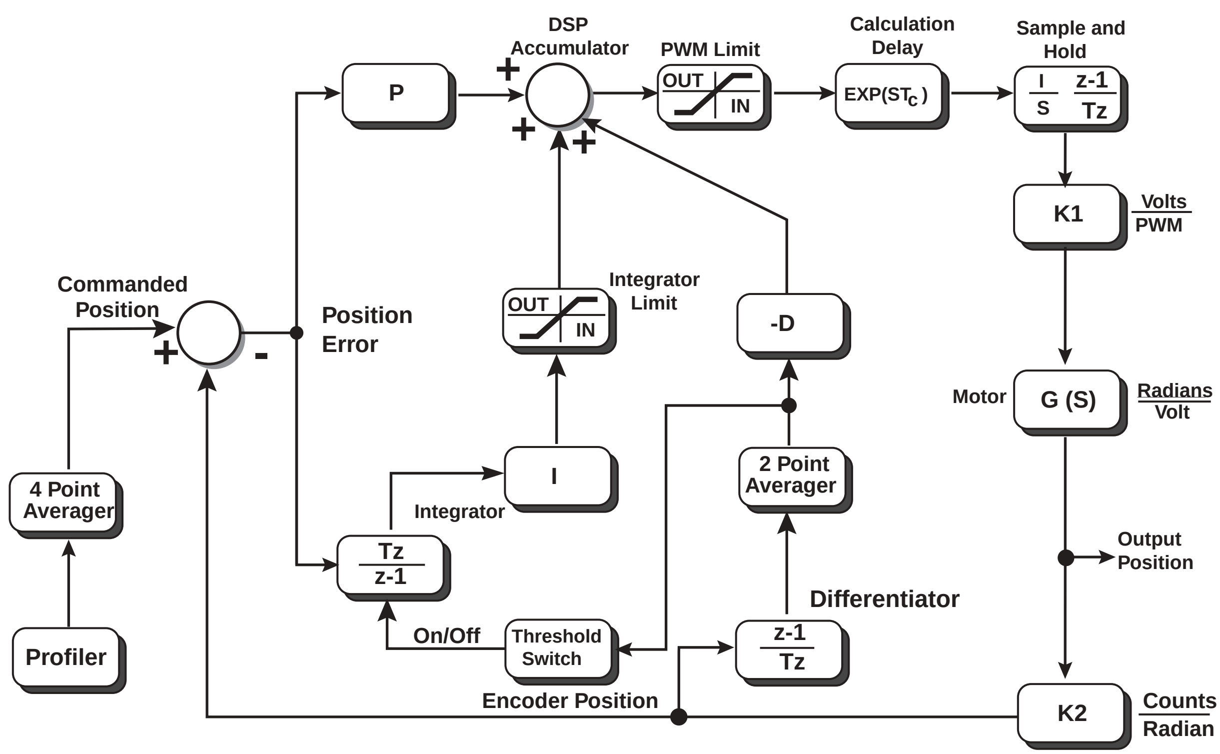
Although these are not the only **z**-domain representations of these functions, they are widely used in control applications.

To mitigate encoder velocity quantization noise, the derivative function is followed by an “n-point averager”, which averages velocity information over a range of samples to provide finer resolution. However, this low pass filter also introduces phase lag proportional to **n** that counteracts the desired phase lead generated by the differentiator. To balance these two constraints, **n** is set equal to two, which effectively doubles encoder resolution per sample interval. The derivative stage transfer function is:

$$\text{VELOCITY} = \left(\frac{z-1}{Tz}\right)\left(\frac{1}{2} + \frac{1}{2z}\right)$$

The following diagram illustrates the PID controller as well as the transfer functions of parasitic effects found in the system.

## PID System Block




The default parameters for the digital filter have been calculated to provide stable operation in most applications. It should not be necessary to adjust the derivative term. In many cases, especially when using small, high speed motors, the proportional and integral terms are set to relatively low values and should be increased to provide the desired system response.

# Appendix I

## General Description

The MicroMini™ Controller is a low cost, miniature, single axis "intelligent drive" that provides speed and position control of DC motors with an on-board integrated amplifier. Up to 63 motor axes can be controlled using an RS-485 Multidrop interface or the DeviceNet™ protocol.

- Programmable Current Limit
- No external amplifier required
- RS-232, RS-485 Multidrop or DeviceNet™ (CAN) compatible interfaces
- Linear drive capable of 10 Watts continuous at 22°C ambient
- Local mode speed range up to 20,000:1
- Short Circuit Protection
- Accepts Step/Direction Input (Optional)
- Torque Limiting Capability

### Modes of Operation

Local/Stand alone (for speed/position control), or Remote via RS-232 or RS-485 , or DeviceNet™.

### Performance (Remote Mode)

| | |
|---|---|
| Position Range | ±8,388,607 quadrature counts |
| Velocity Range | up to 2 million quadrature counts/sec |
| Velocity Accuracy | ±0.1% of max. rate |
| Positional Accuracy | 0.18° (with a 500 CPR encoder) |
| Servo Loop Update Time | 1000$\mu$sec |
| Servo Tuning | Programmable PID filter |
| Acceleration Range | 32,767 quadrature counts/sec/sec |

### Power Requirement

Single supply 24 VDC at 300mA, plus current to drive selected motor

| | | |
|---|---|---|
| Input voltage range | Linear 18-40 VDC | PWM 12-40 VDC |

### Communication

RS-232, RS-485, Multidrop, DeviceNet™

### Controller Inputs

| | |
|---|---|
| Encoder | Two channel, Single-ended +5VDC TTL compatible<br>4 MHz maximum frequency, Optically isolated |
| Analog | One analog input (0-5 VDC, 10 bit) |
| General Purpose | Two external event inputs, Two hard limits, One emergency stop.<br>All inputs are optically protected (except analog input) |
| Opto Coupler Power | Input to supply external Power to Opto Couplers |

### Controller Outputs

| | |
|---|---|
| Sign Bit | One bit digital output (TTL) for external amplifier (direction) |
| Analog | One 12 bit signed output for external amplifier (± 10 VDC) |
| Analog | One 12 bit output (± 10 VDC) |

### Drive Amplifier Specifications @ 22°C (72° )F

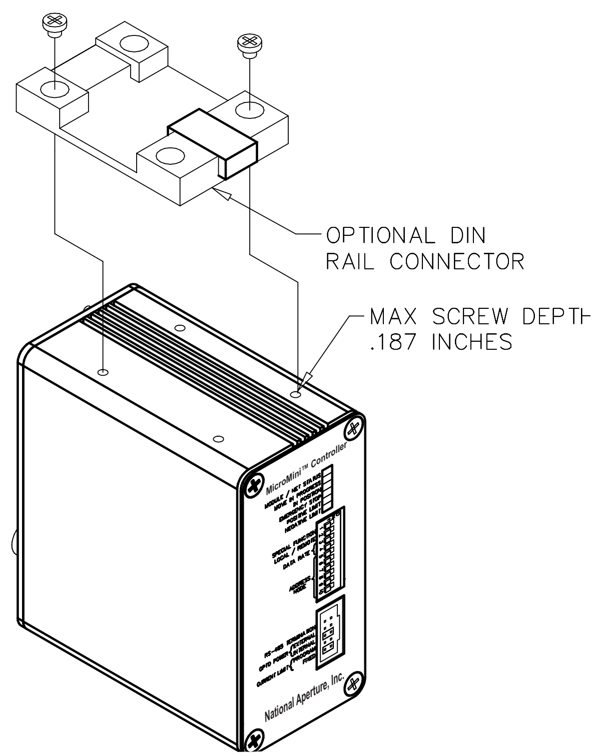
| | |
|---|---|
| **MicroMini™ Controller** Linear Amplifier | Out puts ±22 Volts DC at 1.0 Amp continuous; 3.5 Amps peak |

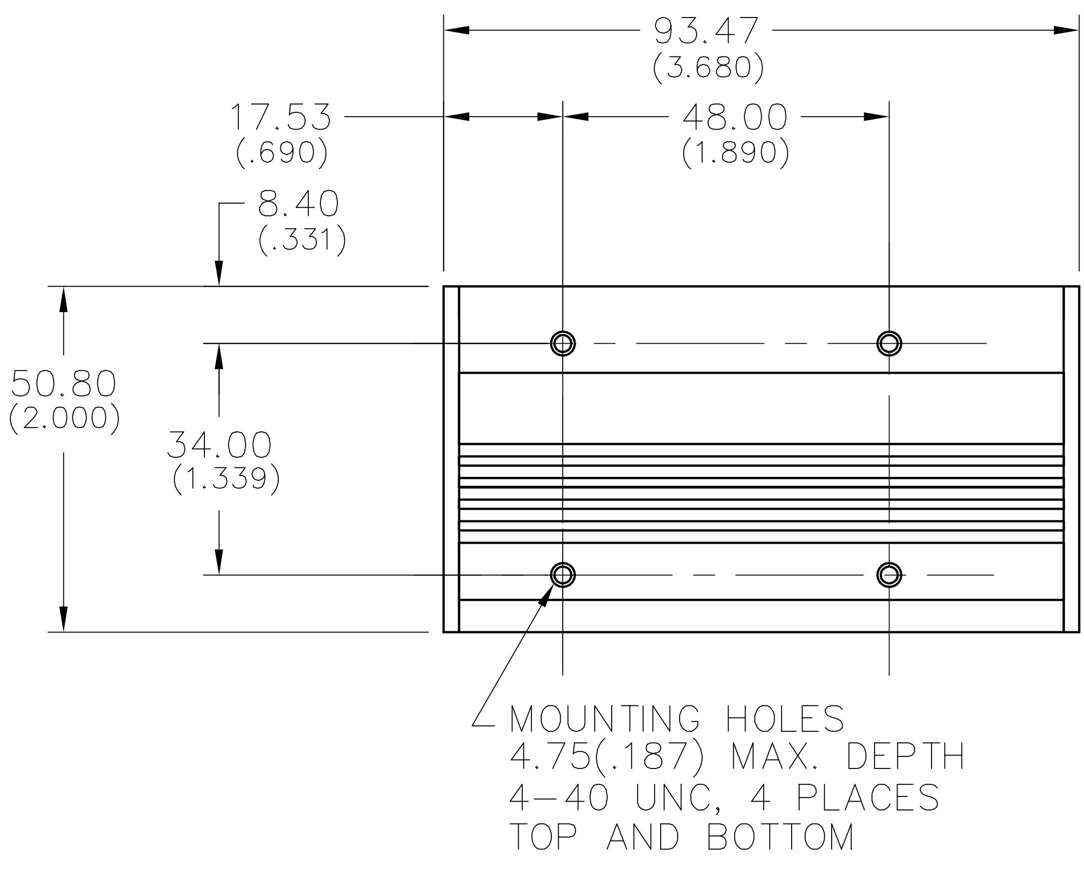
### Environmental

| | | |
|---|---|---|
| Controller Operating Temperature | 0ºC to 70ºC | (32ºF to 158ºF) |
| Ambient Operating Temperature (standard version) | 0ºC to 40ºC | (32ºF to 104ºF) |
| Ambient Storage Temperature (standard version) | -20ºC to 85ºC | (-4ºF to 185ºF) |
| Humidity Tolerance (standard version) | 80% Rh, non-condensing | |

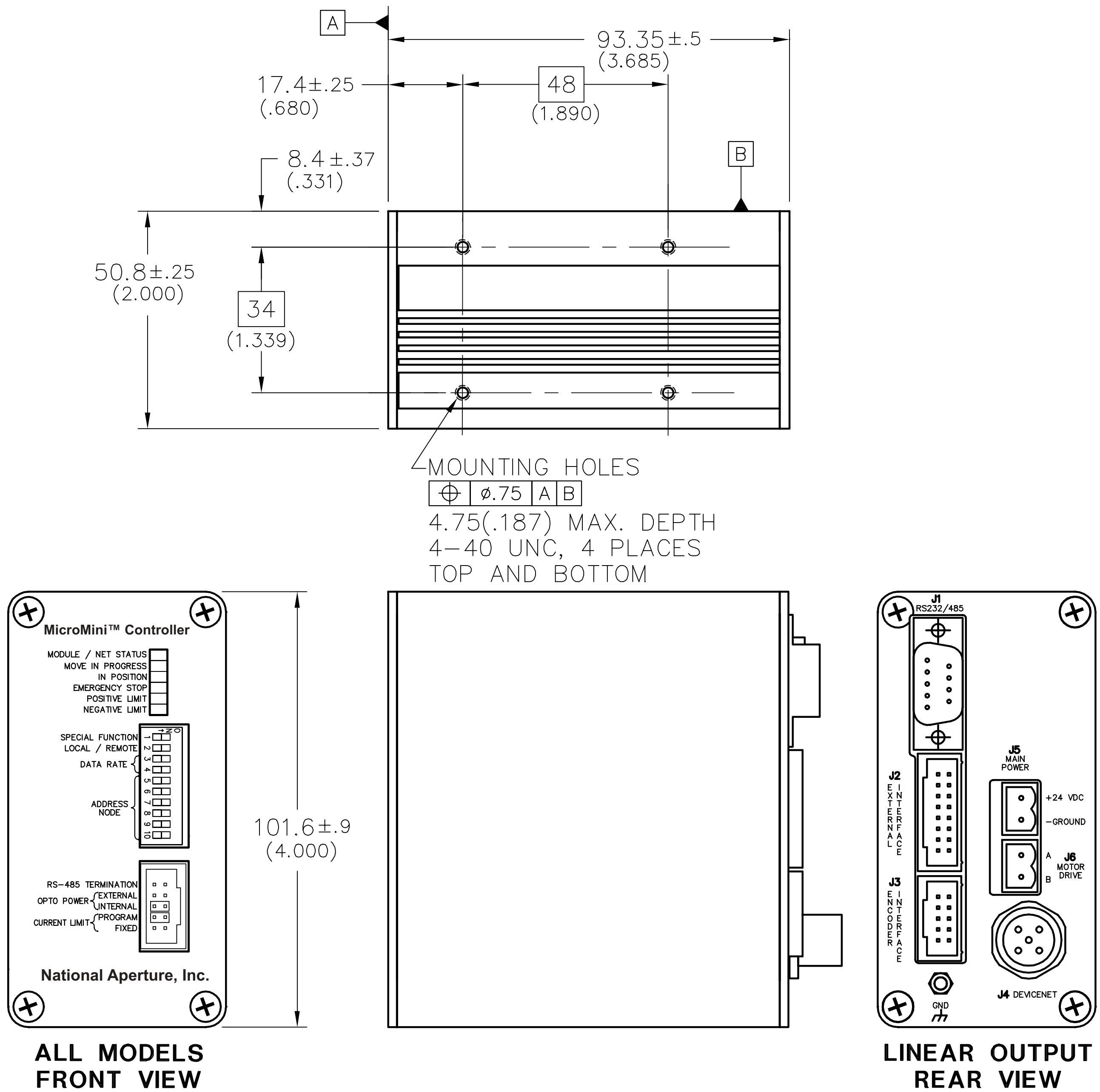
# Mounting

All MicroMini™ Controller modules can be mounted to machinery, in racks, or in cabinets with an optional DIN rail mount or with mounting screws. Regardless of which mounting system you use, **do not penetrate screw holes deeper than 4.75mm (3/16 inch)** from the surface of the controller module, or irreparable damage to the circuit boards may result. The mounting scheme for DIN rails is shown in Figure 1. General mounting guidelines and mechanical references are shown.

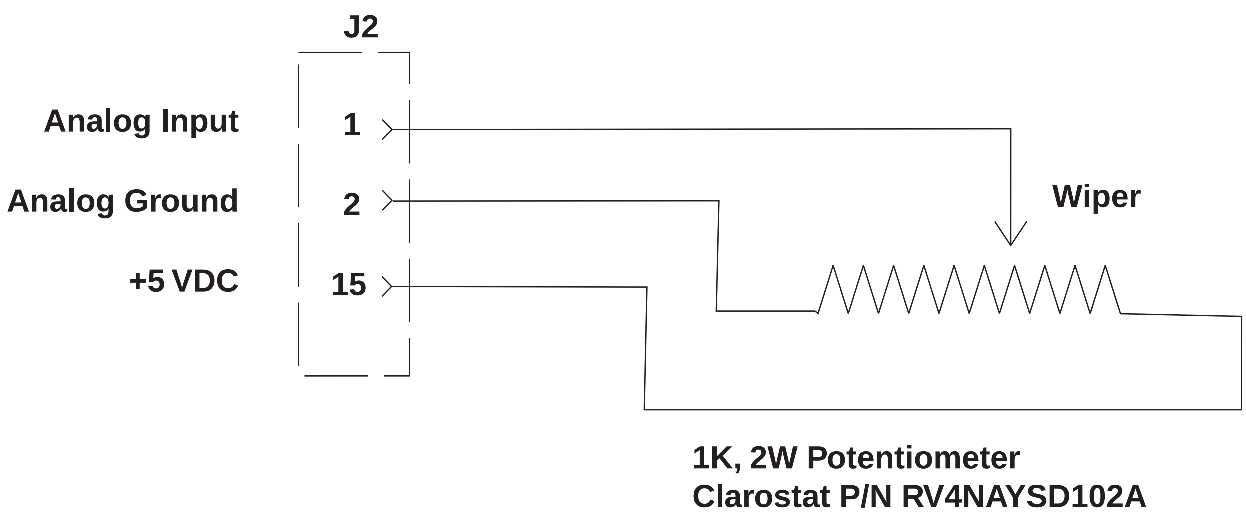


# Dimensional Outlines



# Typical Potentiometer Circuit



# MicroMini™ Controller Warranty

All MicroMini™ Controller Series controller modules (excluding motors, gearheads, encoders, applications notes, software, cabling, and auxiliary devices) are warranted against defects in workmanship and materials for 1 year after date of shipment to the original purchaser. In the event of defects, National Aperture will, at its sole option, repair or replace the defective controller covered by this warranty without charge. To avail themselves of this warranty, purchasers must obtain an RMA number from National Aperture's customer service office, describe the alleged defect in writing, and return the properly packaged defective product within 30 days of discovering an alleged defect, with transportation and insurance prepaid. Replacement or repaired units will be reshipped at our expense to North American destinations only. This warranty shall also apply to controllers which have been repaired or replaced with respect to the original warranty commencement date. In no event will National Aperture be liable or held responsible under this warranty if the controller has been improperly stored, installed, used, or maintained, or if the purchaser has performed or permitted any unauthorized modifications, adjustments, or repairs to the product.

National Aperture has no control over the use of this product or the associated applications notes and demonstration software. The applications notes and software are provided free of charge as is, for illustrative purposes, without warranty of any kind, either expressed or implied, including, but not limited to quality, performance, merchantability, or fitness for any particular use. Neither National Aperture, its affiliates, its employees, or its distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to have been caused directly or indirectly by the product, the applications notes, engineering, application device, or associated software.

No person or entity, including any employee, agent, distributor, or representative of National Aperture is authorized to make any representation or warranty on behalf of National Aperture concerning any products, guidelines, or software except as set forth herein. No other warranties, either expressed or implied are made, and National Aperture shall not be held liable or responsible for any incidental or consequential damages or claims, including, but not limited to breach of contract, negligence, strict liability, tort warranty, or patent or copyright infringement, and in no case shall be responsible or liable under this warranty or otherwise, even if National Aperture has been advised of the possibility of such damages as inability to use the product, increased operating costs from failure to be able to use the product, loss of anticipated profits, or other special, incidental, or consequential damages, whether similar or dissimilar, of any nature arising or resulting from the purchase, installation, removal, repair, operation, use or breakdown of the product or any other cause whatsoever.

# INDEX

## A



## B



## C



## D



## E



## F



## G

## Q



## R



## S



## T



## V



## W