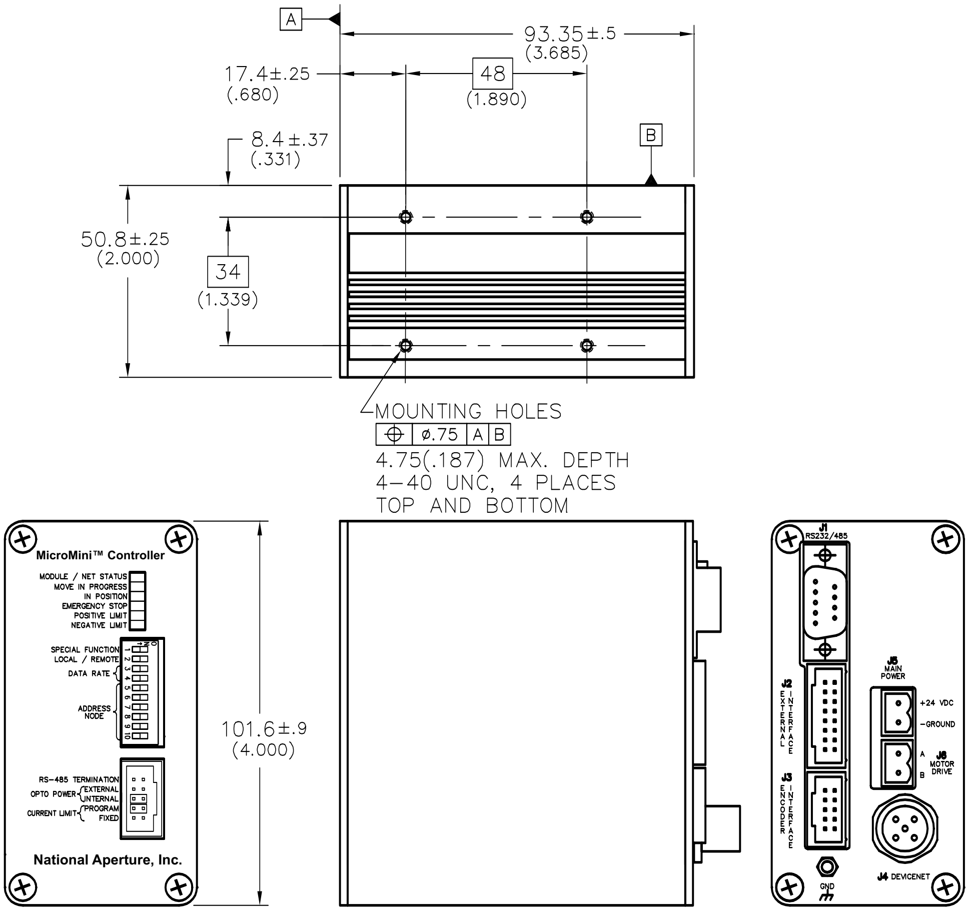
Manual #003.3

USER MANUAL

MICROMINI™ CONTROLLER



ALL MODELS
FRONT VIEW

LINEAR OUTPUT
REAR VIEW

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

Table of Contents

MicroMini™ Controller Commands

Appendices

Index

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 and 0-100% and 50/50 PWM 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, or PC Compatible Operation
- Interface Software Demo and Example Code
- Very Compact - only 2 x 4 x 4 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 4 MHz max. frequency, optically isolated
Analog:	One analog input (0-5 VDC, 10 bit)
General Purpose:	Two hard limits, one emergency stop, 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

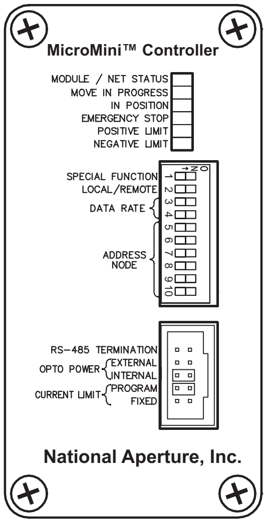
Environmental (standard version):

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

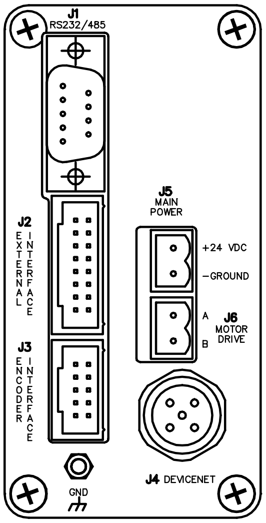
Unpacking

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

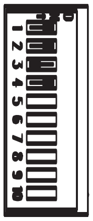


LINEAR OUTPUT REAR VIEW

Establishing Communications with your MicroMini™ Controller.

(WIN 3.x, 95, 98, NT, 2000, XP)

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.

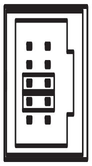


1. Leave power off

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

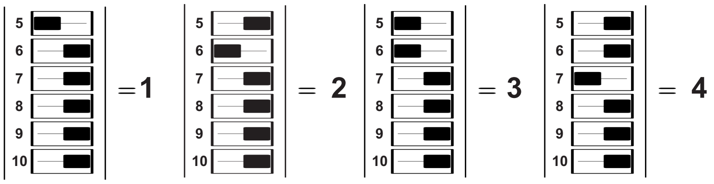
2. Check front panel.

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 page34 and proceed as follows:

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

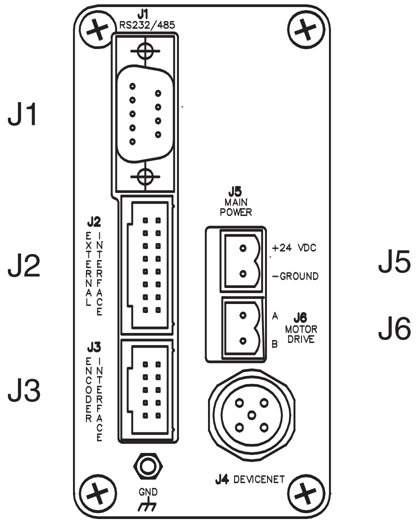
   (Don’t plug MicroMini™ Controller supply in wall yet.)

   (b).Using main (3 to 1) motor cable, CA-MVP-1(See also Appendix C page 35 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 ” page 8.

7. **If computer is not already on, turn on now.**
   (Do not plug in MicroMini™ Controller wall supply yet.).

8. Download and install a copy of “MVPDEMO” software from our website: www.naimotion.com . Note, the software is in a “.ZIP” form and must be unzipped first. Once unzipped, double click on the “Setup” application file and follow the installation steps. Once the software has been installed on your computer, power up the **MicroMini™ Controller.** The controller will go through some diagnostics which will be indicated by its LED’s blinking one at a time from top to bottom.When the controller has completed its power up cycle the following is the LED pattern:

   MODULE/NET STATUS - On Blinking Green
   MOVE IN PROGRESS - Off
   IN POSITION - On Solid Green
   EMERGENCY STOP - Off
   POSITIVE LIMIT - Off (if stage limit active will be On solid Red)
   NEGATIVE LIMIT - Off (if stage limit active will be On solid Red)

9. Start MVPDEMO program by clicking on the “MVPDEMO” icon. the top level Graphical User Interface (GUI) should look like the interface on page 9. With the **MicroMini™ Controller** powered and connected to one of the computer’s serial ports, select the “setup” option on the top of the MVPDEMO software GUI. This will display another GUI (see page 10) which is useful for establishing communications with the controller. Select the “Auto Configure” option within this GUI and the software will attempt to automatically configure the appropriate communication port. If this is

successful, the software will display the following message:

“Serial Port Configured Successfully!”

If the automatic configuration is not successful, no message will be displayed. Should this occur, check the cabling and DIP switch settings. Often times the controllers are configured for a particular communication interface and the user’s cabling is not appropriate (i.e., RS-485 interface and the user attempts to communicate with an RS-232 interface or vice versa). Note, if the controller was configured for RS-485, a serial port “straight thru” cable and RS-485 converter is required.

10. Choose either <u>Rotary</u> or <u>Linear</u> translation stage configuration.

– If you have a **<u>Linear</u>** 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 **<u>Rotary</u>** 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 “Manual Command Entry Screen” page 11).

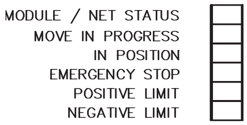
b.) Clip or unplug the jumper on the J-2 connector (eliminate pin 5).
This method will be necessary if you plan to use external limit protection:
pins 5 & 6 will be used for positive and negative limits respectively with pin 16 as the common ground. Pin 7 (event 2) will be reserved exclusively for Homing.
The correct way to apply hard limit protection to a Rotary stage is <u>externally</u>.

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

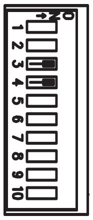
Node Address (switches 5-10)

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

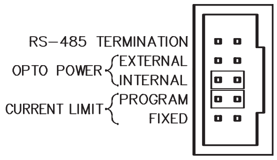


Local/Remote Mode

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. (See Appendix C-1, page 35)
Remember that the configuration switches must be set unless the EEBoot function is active before powering up the unit. If the unit is operated in Local Mode, switches 3-10 may be changed "0n the fly" (i.e. without resetting the power to the module).



RS-485 Network Termination
The termination jumper pins are below the front mounted switches (see drawing at right). These switches provide end of line termination for the 485 network.

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.

Selecting an Operating Mode:.

There are two operating modes available for the MicroMini™ Controller ; Remote and Local.
Remote Mode(Position and/or velocity control) is appropriate when an external digital device such as a computer or host controller will be running your motion profile programs.
Local Mode is appropriate when you want to control velocity or position with an analog input such as a potentiometer or external encoder.

Stand-alone:
This is the **Local Mode** option which is designed to operate as a free-standing module. Consequently, it needs no host connections or software installation. It is controlled solely by an analog input signal such as a potentiometer. In **Local Mode**, the MicroMini™ Controller can operate in **Velocity Control** (to command a desired motor velocity) or in **Position Control** (to track an external encoder signal). In **Velocity Control** the unit can be run in either **Absolute Mode** (for coarse speed control) or **Relative Mode** (for fine tuning the velocity). A controller configured for Local Mode Velocity Control can run on power-up at the same speed every time. Optional position tracking mode allows MicroMini™ Controller to perform electronic gearing or use step/direction signals from a stepping motor controller.

In **Absolute Velocity Control (Switch 4 = Right)**, the motor shaft velocity is the value read from the analog input or potentiometer multiplied by the range selected with switches 7 through 10. **Switching between ranges is allowed at any time.** The new range will become effective during the next filter sample. This mode allows you to select a coarse velocity value before fine tuning.

In **Relative Velocity Control (Switch 4 = Left)**, the motor shaft velocity is calculated in the same manner as in Absolute Velocity Control above. The difference in this operating mode is that once a velocity close to the desired velocity is achieved, in absolute mode, switching to relative mode and selecting a lower range on the fly will allow you to “tune” the final velocity to precisely your desired value.

Local Mode Control Loop (Gain) Configuration: Switches 5 and 6 allow you to select four different proportional gain values. The gain selected is determined by the binary encoding of these switches. Switch 5 is the least significant bit. The lowest gain is enabled with both switches (5 and 6) = **Right**. The highest gain is selected with both switches (5 and 6) = **Left**. The operational mode as well as the gain configuration is determined at power-up.

It is important to note that RS-232C or RS-485 configuration is set internally at the factory. You must specify a specific serial operating mode when you order . Also keep in mind that RS-232C cables and RS-485 cables cannot be interchanged even though both may fit your serial port.

Remote Mode:

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. The format is always 8 data bits, no parity, and 1 stop bit. 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, Inc.

Event Input Overview

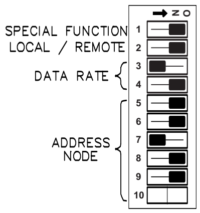
Inputs for analog control or monitoring, overtravel, emergency stop, and HOME or event sensing are also accessible, some of which are used by the CA-MVP-1 interface cable. An amplifier enable output can also be used to automatically select an external amplifier when a control mode is activated. The overtravel (Limit switch) inputs are intended to protect the drive from a hard stop and require manual intervention to recover. 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 page35-36.

Software Installation

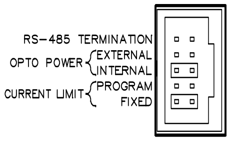
Operation in Remote Mode allows you to install the Demo Software available for the MicroMini™ Controller. The Demo software is designed to highlight the capabilities of the Remote Mode Operation. the Demo program will function on Windows 3.x, 95, 98, NT, 2000 and XP operating systems, utilizing a 16550 UART or equivalent on a 486/66 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. The ”MVPDEMO” program may be obtained from the National Aperture website www.naimotion.com. Note, the software is in a “.ZIP” form and must be unzipped first. Once unzipped, double click on the “Setup” application file and follow the installation steps.

Remote Mode: Position Control

To set the unit up for Remote Position Control, set the configuration switches as shown before you power up the module.

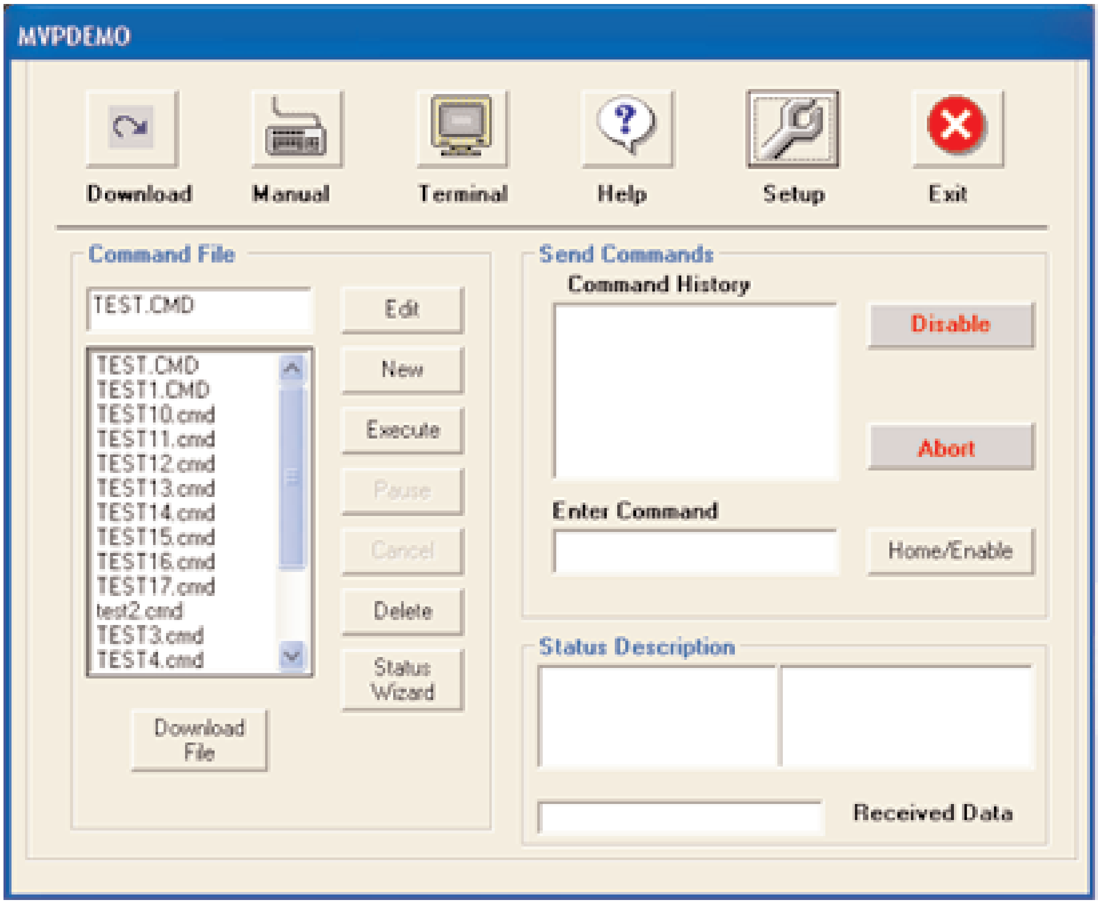


Next power up the module to load the configuration settings.



Configuration

Click on the MVPDEMO 2000 icon to begin setting up your system..

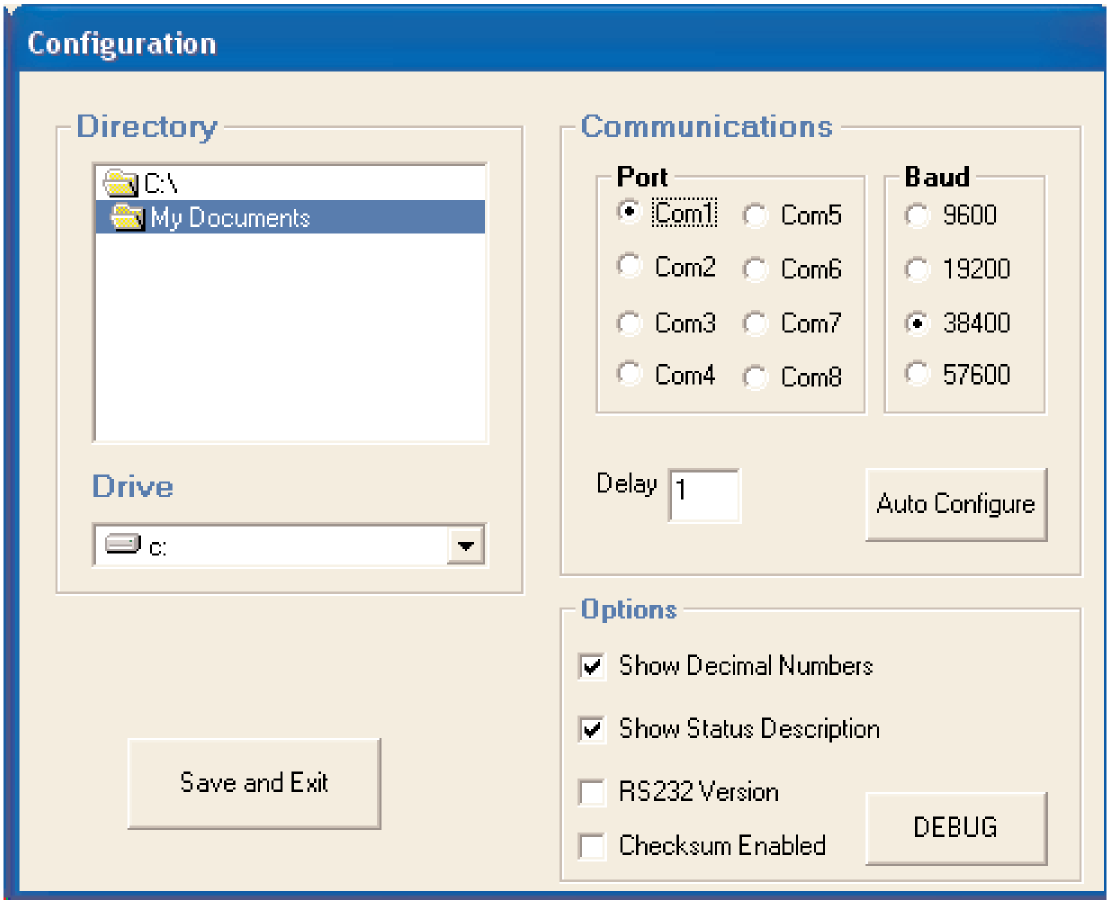


TEST.CMD

TEST.CMD
TEST1.CMD
TEST10.cmd
TEST11.cmd
TEST12.cmd
TEST13.cmd
TEST14.cmd
TEST15.cmd
TEST16.cmd
TEST17.cmd
test2.cmd
TEST3.cmd
TEST4.cmd

The default screen which you will open looks like this

The “Setup” button will bring up the screen on the next page.



Configuration

Directory

C:\
My Documents

Drive

c:

Communications

Port

● Com1 ○ Com5
○ Com2 ○ Com6
○ Com3 ○ Com7
○ Com4 ○ Com8

Baud

○ 9600
○ 19200
● 38400
○ 57600

Delay 1

Auto Configure

Options

☑ Show Decimal Numbers
☑ Show Status Description
☐ RS232 Version
☐ Checksum Enabled

Save and Exit

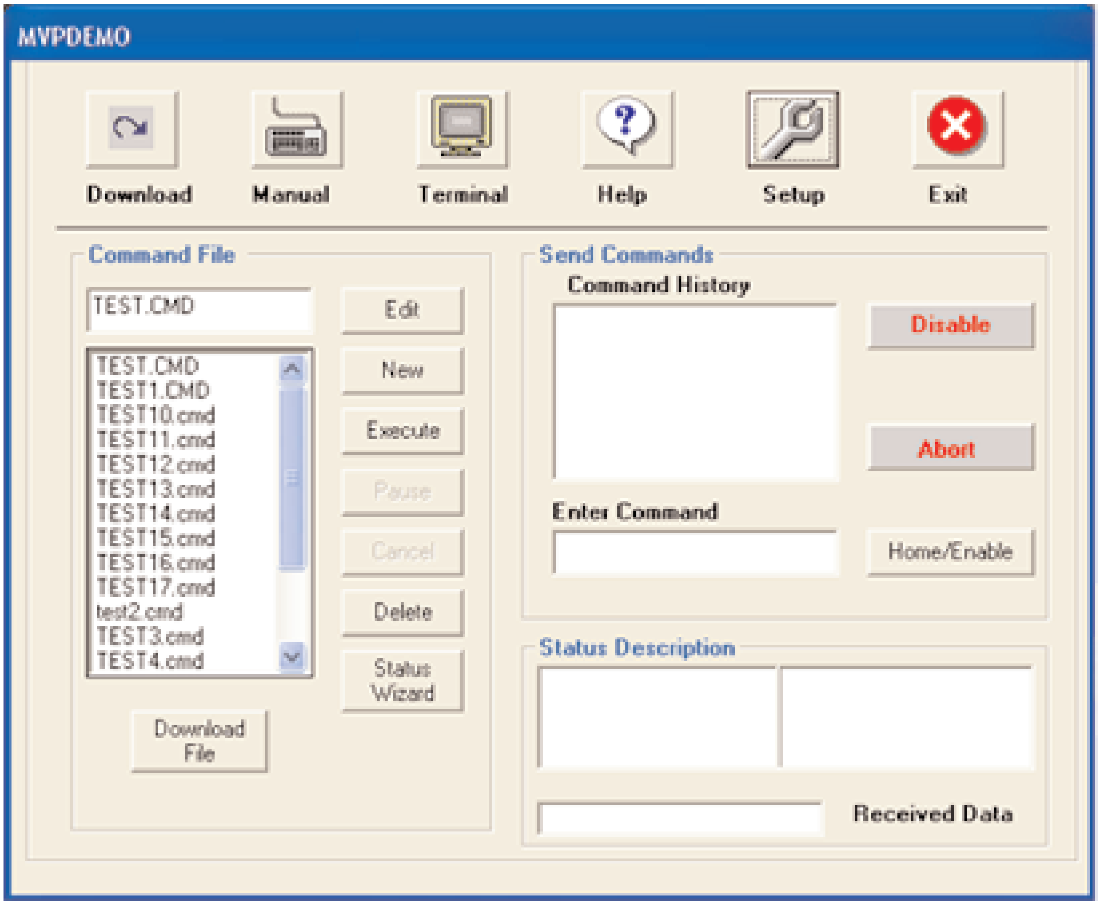
DEBUG

This screen allows you to choose the baud rate at which your system will operate and the Comm 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.Options for configuring the status description and response rate are also available.

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

Manual Command Entry Screen



TEST.CMD

TEST.CMD
TEST1.CMD
TEST10.cmd
TEST11.cmd
TEST12.cmd
TEST13.cmd
TEST14.cmd
TEST15.cmd
TEST16.cmd
TEST17.cmd
test2.cmd
TEST3.cmd
TEST4.cmd

The sample code you will put into the "**Enter Command**" screen is only for exercises you will run. It can't be saved and 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's position and velocity. But first you will need to know a few generic motion command formalities.

1. A motor moves from a known location. This location from which it starts its initial move is called "home". When the motor knows where its "home" is, you can make it go anywhere within the limits of the system, and then on to new locations or back to its original position. "Home" is identified by the abbreviated Command **HO,** or **home**--both will work.

2. Before a motor can move, it must be "enabled." Enabling allows the MicroMini Controller™ to source current by turning on its amplifier. To "enable" your drive, you will use the Command **EN**.

3. 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.

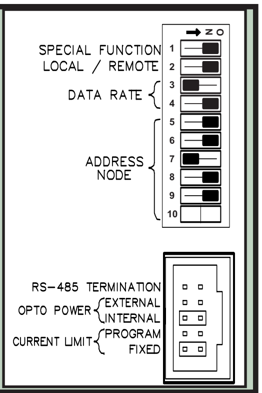
Examples and Explanations:

Practicing a Simple Sequence:
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. In the following examples we assume that your MicroMini™ Controller module has an address of 4. Try the following simple sequence to make the motor move, **after** first setting the MicroMini™ Controller to Remote Mode, setting Baud rate to 38,400 baud, selecting an available comm port, and making sure that the address of your unit is 4. Whether the MicroMini™ Controller operates in Velocity Control or Position Control is command dependent (versus being set by changing the DIP switches). Your DIP switches should look like the box to the right:

Node addresses are selected using switches 5-10 on the front of the unit. The addresses are encoded in binary format with switch 5 as the least significant bit. In this scheme, each switch has a numeric value as indicated.

Switch Number	Value
5	1
6	2
7	4
8	8
9	16
10	32

To select an address, turn the switches to the left who's value adds up to the desired address.



1. Home: A motor's home location will be at a point you define as zero (0). To tell your module number 4 that it is home, enter "4 home" (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 "4 en" and return.

3. Normally, you would next Set Proportional and Integral Gain, but for this exercise you can leave the gain settings 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." page 57.

4. Prepare to Make Your Move! To do this you will need to know the following :
   a. 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 "load [the] absolute [destination position]" which you can do by giving the module(s) the command **LA**. Putting "4 la" followed by a space and a number less than +/- 2 billion (the range limit) will allow the shaft to turn up to the selected distance in either direction. If you want to go to a position a given distance from the point where you are presently located (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 "4 LR 50000", followed by "M" to initiate the motion, should give you an idea if your motor is moving smoothly.
   Don't expect to achieve ranges of performance beyond the capabilities of the range limits (+ and-), of the system.
   b. 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 next page). If you are using a 500 cpr encoder, each unit = 1.0 rpm. If you are using a 10 cpr encoder, each unit = 50 rpm. Typing "4 sp 1000" (then enter) should be adequate for demonstration purposes.

Encoder Resolution	Quadrature	rpm/unit	Example
1000	4000	0.50	2000/4000 = 0.50 rpm/step
500	2000	1.0	2000/2000 = 1.0 rpm/step
200	800	2.5	2000/800 = 2.5 rpm/step
100	400	5.0	2000/400 = 5.0 rpm/step
16	64	31.25	2000/64 = 31.2 rpm/step
15	60	33.33	2000/60 = 33.3rpm/step
10	40	50.0	2000/40 = 50.0 rpm/step
Note: Using a 500μs sampling period			

c. The acceleration (command AC) which is the number of quadrature counts per sample period.The sample period for the MicroMini™ Controller is 500 μs. Try “4 ac 100” (enter) for starters.

5. Insert a “status” command. The status of your operations can be determined at any point in time by invoking the “status” command (represented as **ST**). You can enter any command you wish, without an argument, to verify the current setting. [No argument] is treated as a request to send the stored value back to the Host or Master. The **ST** command, for example, serves to verify that the move is happening or has happened. This is a very useful feature which avoids starting a move before the previous move has been completed. Try typing “4 st 10” to confirm your move. 10 (binary 1010) according to the status bit code, signified “command recognized,” “motor in position,” and “no other errors or exceptions.”

6. Make it move by using the command “4 M” (MicroMini™ Controller box at address node #4 MOVE!). If your motor doesn’t move, recheck all connections and try reentering the command string as follows:

```
4 home
4 en
4 la 500000
4 sp 4000
4 ac 100
4 st 10 [“command accepted” , motor in position]
4 M
```

If your motor doesn’t move after reentering this string, try running the stored command files in the “Command File Processing” screen. If that does not work, please consult the “Troubleshooting” guidelines in Appendix “E” page 38.

Building a Command File:

In order to construct , store, edit and execute a simulated program, you must go into the editing screen by selecting the “Edit” button from the Mode Selection box.

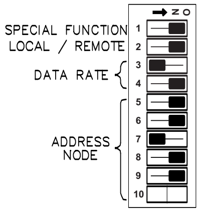
Write a stored executable file by clicking the edit button. An example is provided for you in the file called “demo.cmd.” The example is provided only to show you how the motor moves. Execute the sample program by highlighting the “demo.cmd” file and instruct it to run continuously. This will provide a working demo which will loop continuously until you “Cancel.”

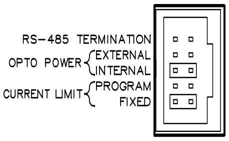
Remote Velocity Control:

Remote Velocity Control affords you the opportunity to control a motor’s velocity. The DIP switch settings are the same as for the Remote Position Control Velocity instructions can be interspersed with position commands via the instruction set. Velocity commands can be set on the fly using the form:

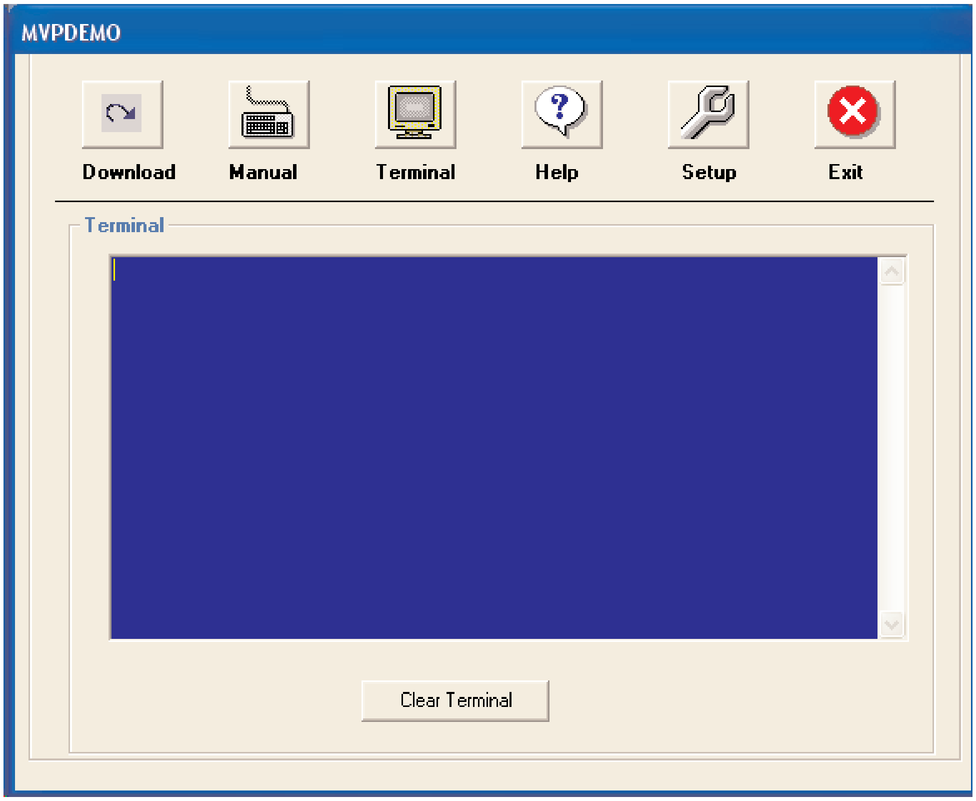
[node#] v [number]

As an example: “4 v 100” Mixed mode moves are common in motion control applications. An example of mixed mode operation is included in Applications Examples , Appendix ”G-1“ page43.





Terminal Emulation Screen



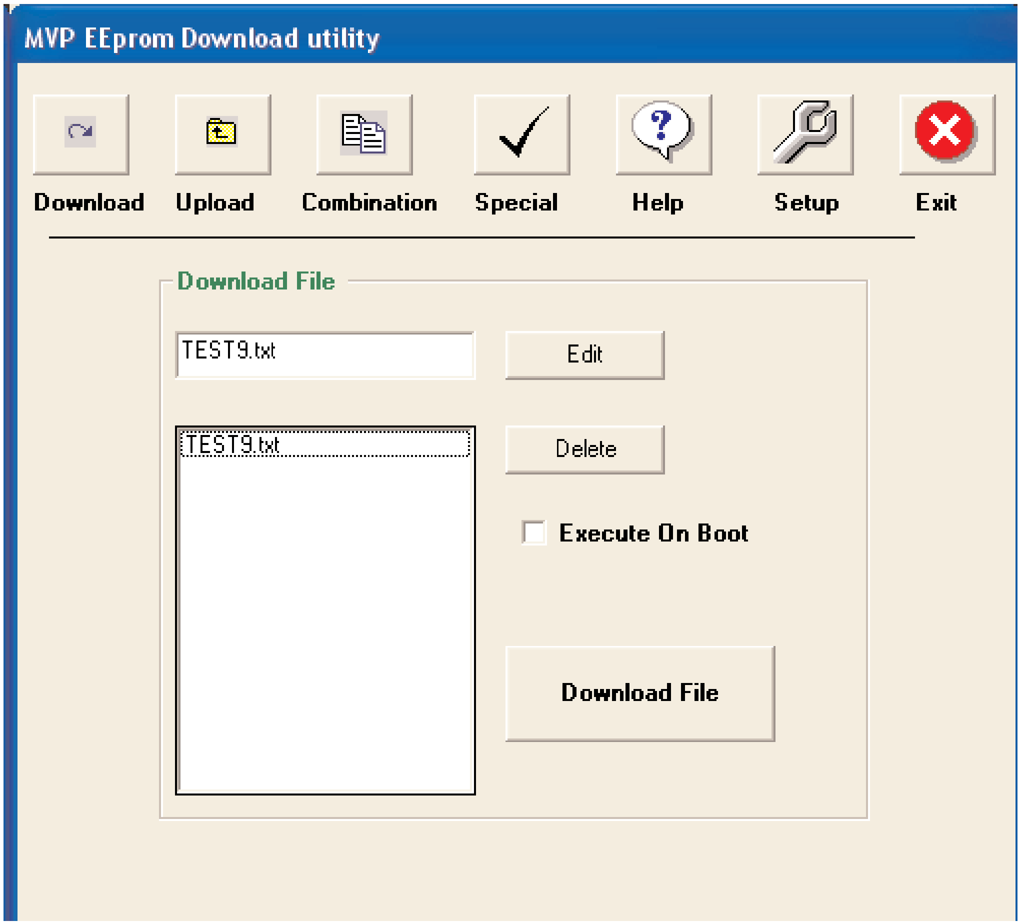
Terminal Emulation mode allows direct interaction with the MicroMini™ Controller. Command strings entered in terminal mode are sent directly to the target controller and response data is displayed in the terminal window. Configuration for the terminal mode is the same as for all other modes. This data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

Macro Operation:

The macro programming functionality provides the stand-alone capabilities of the MicroMini™ Controller. In this configuration, programs developed using the Demo Software can be downloaded into non-volatile memory contained within a MicroMini™ Controller Servo Drive. A single Macro Module is capable of controlling up to 63 axis of MicroMini™ Controller 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,or serially. Macro routines provide a mechanism to implement configuration sequences, indexing operations, or variable motion profiles with minimal host interaction. Macro's are usually developed and tested as command files and then downloaded once proper operation has been verified.

Download Operation:

Once a command file has been developed and tested, press the download button located on the bottom of the main demo window. The download menu on the following page will appear and includes a list of available command files to select from.



TEST9.txt

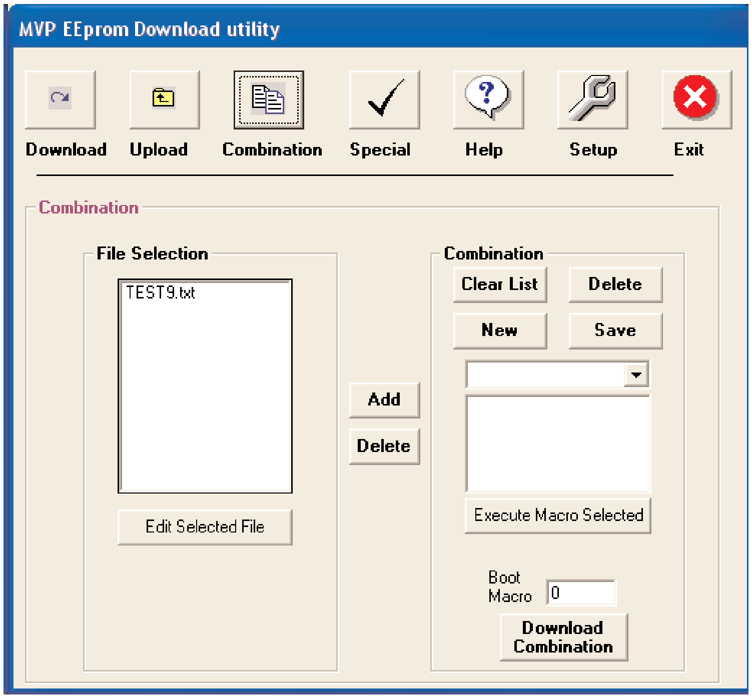
TEST9.txt

□ Execute On Boot

To download a single file, simply select the appropriate file and press the download button. If multiple macro routines are required, press the combination button to create a custom configuration.

Select files to be included in the combination and “add” them to the combination list. While no specific order is required for macro operation, it is a good practice to list the macros in order of execution. Once the list is complete, select the macro to be executed at power-up in the “boot macro” window located in the lower right hand corner. If no power-up macro is desired, enter 0 in this window. Press the “Download” button to return to the previous screen and press “Download”.

After the download process is complete, the macro routines are available for use.



TEST9.txt

Boot Macro 0

Notes on Actual (real life) Application

The MicroMini™ Controllers are designed to be slave devices which are controlled by a Host or Master device. In the Remote Operating Mode, the controller will transmit on the communications interface except on power-up, unless it is instructed to do so by the Master device. 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.

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 G-1 page 43. 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.

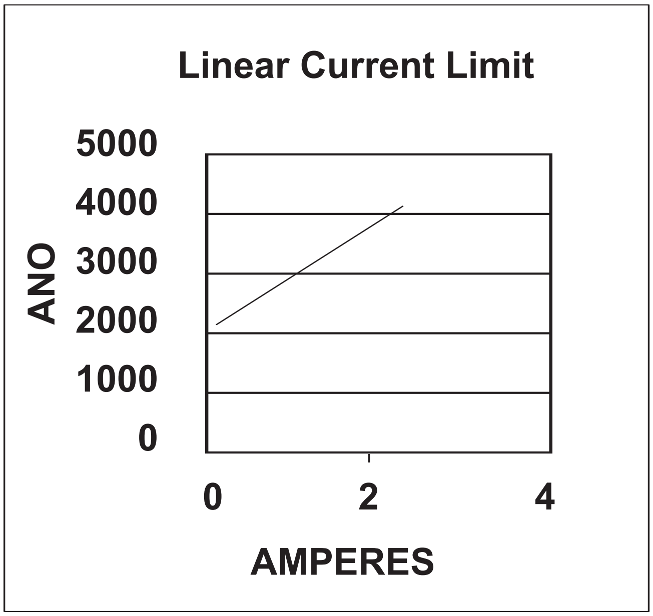
The system status responses provide a wealth of information about the condition of each node in the system. It is useful for controlling operational flow as well as detecting system level errors. When implementing a position move for example, it is normally good practice to ensure that a commanded profile 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.*

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 zero current (ANO=2047 decimal).



Amps	ANO	
0.0	2048	
0.1	2190	
0.2	2277	
	2350	Recommended for 10 mm motors*
0.3	2364	
0.4	2450	
0.5	2537	
	2600	
0.6	2623	
	2700	Recommended for 13mm motors *
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	

***Consult with National Aperture, Inc. before using values other than recommended**

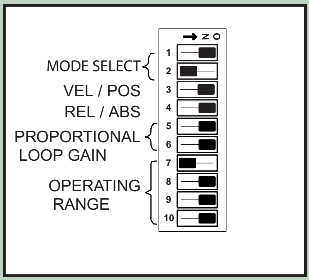
When the programmable output current capability is used in conjunction with the programmable following error limit feature, torque control mode operation can be implemented. In this instance the servo loop should be configured for following error recovery **(FA=2)**. The default dynamic following error value **(FD)** is set to 1500 quadrature counts. When the following error exceeds the programmed amount (1500 counts) the servo loop stops producing trajectory data until this error is recovered. This contains the accumulated following error within the programmed limit. Since the motor current is limited, so is the available torque. If the current is insufficient to allow the motor to produce enough torque to recover the following error, the MicroMini™ Controller will supply this "stall torque" indefinitely. If the motor is able to recover the following error, the trajectory generator will resume operation.on the next filter sample 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 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.

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

If the programmable current limiting feature is not required by the application, the MicroMini™ Controller can be configured to allow the drive to supply maximum motor current on demand, freeing the Digital to Analog converter to set the current for general purpose use. Operation in this manner does not compromise the short circuit protection of the drive components.

Local Mode:

Prior to entering local mode, **remove power to the unit** as some configuration options are determined at power up. Local operating mode provides closed loop speed control or position control using a potentiometer or other analog signal source as an input command. The configuration switches are designed to provide you with a wide range of operating options. Once configured, the controller will become fully functional immediately upon power-up. **It is important to verify all connections prior to power-up to avoid unintended consequences**. The analog input command **must** be maintained between **0-5 VDC**. The resolution of the port is +/-512 steps or 9 bits. This allows bi-directional position and velocity control with 2.5 VDT being the “zero” input command. Although some internal hardware and software filtering is provided, you should make sure during installation that no extraneous noise is induced into the control loop. To set the controller for **Local Mode** move **Switch 1 to the right** and **Switch 2 to the left:** the balance of the switches will be set depending on the subsequent Modes.



NOTE THAT THE DIP SWITCH LABELS ON THIS DIAGRAM ARE FUNCTIONAL LABELS AND DO NOT CORRESPOND TO THE LABELS PRINTED ON THE FRONT OF THE MODULE.

Local Velocity Control (Switch 3 is to the Left):

Local Velocity Control has two functional modes: Absolute and Relative. Regardless of which of these modes is selected, the actual change in the motor shaft velocity for an incremental change in the output command voltage is determined by the resolution of the encoder feedback device and the system sampling frequency. For the MicroMini™ controller the control loop expects to receive 33.3 encoder quadrature feedback pulses per second for each velocity unit specified. The table below illustrates the incremental velocity delta for some of the more commonly available encoder resolutions. [33.3 quadrature pulses/sampling period * 60 = 2000 pulses/minutes.] The default sampling period in local mode is 500μs.

Encoder Resolution	Quadrature	rpm/unit	Example
1000	4000	0.50	2000/4000 = 0.50 rpm/step
500	2000	1.0	2000/2000 = 1.0 rpm/step
200	800	2.5	2000/800 = 2.5 rpm/step
100	400	5.0	2000/400 = 5.0 rpm/step
16	64	31.25	2000/64 = 31.2 rpm/step
15	60	33.33	2000/60 = 33.3rpm/step
10	40	50.0	2000/40 = 50.0 rpm/step
Note: Using a 500μs sampling period			

For example the resolution of the input A/D converter is +/- 512, 5 VDC/512 = 0.0097 Volts/bit, with 2.5 VDC being the “zero” input command. Therefore, with the lowest operational range selected (through manipulation of switches 7-10), the shaft velocity of a 500 cpr encoder with an input voltage command of 2.12 Volts produces an output command value of 77 velocity units. Multiplying this value by 1.0 from the table on the preceding page for 500 line encoders provides a command velocity of 77 rpm. For increased functionality, switches 7 through 10 allow you to select five range multiplies from 1 to 80 which can be applied to the initial base velocity. The multiplies for the available ranges are as follows:

Switch	7	8	9	10	Multiplier
	Right	Right	Right	Right	1
	Left	Right	Right	Right	4
	Right	Left	Right	Right	20
	Right	Right	Left	Right	40
	Right	Right	Right	Left	80

In **Absolute Velocity Control (Switch 4 is Right),** the motor shaft velocity is the value read from the analog input or potentiometer multiplied by the range selected with switches 7 through 10. **Switching between ranges is allowed at any time.** The new range will become effective within 500μs. This mode allows you to select a coarse velocity value before fine tuning.

In **Relative Velocity Control (Switch 4 is Left),** the motor shaft velocity is calculated in the same manner as in Absolute Velocity Control above. The difference in this operating mode is that once a velocity close to the desired velocity is achieved, in absolute mode, switching to relative mode and selecting a lower range on the fly will allow you to “tune” the final velocity to precisely your desired value.

Local Mode Control Loop (Gain) Configuration: Switches 5 and 6 allow you to select four different proportional gain values. The gain selected is determined by the binary encoding of these switches . Switch 5 is the least significant bit. The lowest gain is enabled with **both** switches (5 and 6) **to the right.** The highest gain is selected with **both** switches (5 and 6) **to the left. Switching from Velocity Control to Position Control is not possible without shutting down and repowering the module.** The operational mode as well as the gain configuration is determined at power up.

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 0=not commanded to move	
2	Bit 1:	1=motor is in position 0=motor is not in position	
4	Bit 2:	1=MicroMini™ Controller is in Velocity Mode 0=MicroMini™ Controller is in Position Mode	
8	Bit 3:	This bit indicates the logic state of the analog input when **ANM** mode 1 is selected. If **ANM** mode =0, this bit will always be set to 1.	
16	Bit 4:	1=Indicates trajectory percentage defined by the "T" command is complete 0=Trajectory complete percentage not yet achieved	
32	Bit 5:	1=module is in DeviceNet™ Mode 0=module is not in DeviceNet™ Mode	
64	Bit 6:	1=a DeviceNet™ message error has occurred in one or more packets 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) 0=current move is going o.k.	
256	Bit 8:	1=motor is not enabled or has been disabled by some other error 0=motor is enabled	
512	Bit 9:	1=you have reached the program range limit (set by the **LL** command) 0=move is within the range limits	
1024	Bit 10:	1=Local Mode is active 0=Remote Mode is active	
2,048	Bit 11:	Emergency stop flag (1=active)	} These flags signal the status of event inputs
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)‡	

*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.

Positive and Negative hard limits 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

In serial communication mode, the controller is controlled via a series of ASCII instructions issued by the Master computer or PLC. Multiple device nodes can be communicated with simultaneously by inserting a comma between the node addresses to be accessed. All nodes will also respond to a command issued to address 0. This function, when combined with the X parameter allows the digital filters of multiple nodes to be synchronized.

See Examples and Explanations (page 11) before attempting to use commands.

Motion Related Commands:

M **(Move) Initiate Motion** **4 M**

Receipt of the "M" Move command initiates motion using the values presently loaded to determine the profile characteristics.

LA **Load Absolute Target Position** **4 LA 10000**

Allows programming of the desired target position relative to the present zero or "home" position

LR **Load Relative Target Position** **4 LR 10000**

Allows programming of the desired target position relative to the present position.

S P **Load Maximum Commanded Velocity** **4 SP 3000**

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**.** Velocity is specified as quadrature counts/sample period

AC **Load Profile Acceleration** **4 AC 100**

This parameter determines the acceleration and deceleration rate to be calculated by the profiler during execution. Acceleration (AC) = (SP * 16) / (T/1/SR)) T=Time to Accelerate, SP=Max Velocity, AC=Acceleration, and SR=Sample Rate.

AS **Report Actual Speed** **4 AS**

This command queries the MicroMini™ Controller for actual speed.

DC **Load Profile Deceleration** **4 DC 100**

This parameter determines the deceleration rate to be calculated by the profiler during execution.

V **Constant Velocity Command** **4 V 2200**

Execution of this command causes the controller to switch into the Velocity control mode. The motor will ramp the selected velocity at the rate defined by the **AC** parameter. Velocity mode can be used interactively with Position mode.

DI **Disable Drive** **4 DI**

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**

This instruction enables the drive electronics and allows the servo controller to become active, providing motor current as required.

LL **Set Position Range Limits**

4 LL 2000000 or 4 LL -130000

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.

SA **Select Range Limit Action**

4 SA 2 **(Code = 0 - 2)**

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.

T **Set Percentage Trajectory Parameter**

4 T 45

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.

AB **Abort Motion Command**

4 AB

Execution of this command terminates the present motion and the motor will ramp to a stop at the rate determined by the **AD** parameter.

AD **Abort Deceleration Parameter**

4 AD 200

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.)

AA **Abort Action Codes**

4 AA 2

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.

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".]

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. Setting the **HA** parameter argument to "0" will disable the function. The input is also disarmed once it has been triggered. (Note: HA 2 configures J2-8 as the home input.) T*he Status Query in this particular command is achieved with the* **HS** *command*.

HP **Define Home Arming Polarity**

4 HP 0 (0 = Negative Logic, normally open)
The HP command allows the user to select positive or negative logic input signals to trigger the HOME. 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*

LP **Define Limit and Estop Polarity**

4 LP 0 (0 = Negative Logic)
The LP command allows the user to select positive or negative logic input signals to trigger the Error inputs. This flexibility eases installation into existing applications where the state of these signals is predetermined. The Error operation is performed upon the appropriate change of state of the selected input.

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 only the event inputs, as well as the overtravel and emergency stop inputs. *This an* ***exception*** *to the general rule of Status Query*

HF **Homing Sequence Action Code**

4 HF 2
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.

FD **Set Max Dynamic Following Error**

4 FD 1000
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.

FDT **Set Following Error Delay**

4 FDT 1000
This parameter allows the user to set a delay before following error flag given that following error has occurred.

FA **Configure Following Error Action**

4 FA 2 (0-2)
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.

LS **Limit Sequence Enable**

4 LS 1
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 disable the 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.)

Parameters for Analog (Local)Control:

O **Local/Remote Mode Flag** **4 O 0**

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.

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 cpr encoder.

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

JM **Select Joystick Mid Point** **4 JM 512**

Setting this parameter permits the user to determine the MID point of the joystick analog input in order to mitigate any offset in the input signal. (Default = 512)

JW **Set Joystick Center Window** **4 JW 10**

Setting this parameter adjusts the size of the center zero position of the analog input command signal. (Default = 10)

JH **Set Joystick Hysteresis Window** **4 JH 1**

Setting this parameter permits the user to define the amount of hysteresis is applied to the analog control input value to eliminate the effects of noise on the input. (Default = 1)

K **Select Continuous Integration** **4 K 1**

Setting this parameter to 0 permits the user to disable the integral term of the digital filter.

N **Define the "In Position" Range** **4 N 1 (0 not allowed)**

The value of the N parameter determines the size of the position window that the MicroMini™ Controller considers to be "In Position".

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.

Example Status Response 0004 000A

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.

POS **Query Present Position**

4 POS
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
The selected MicroMini™ Controller node will report the present node status immediately upon receipt of this instruction.

Bit 15	**N-Lim**	**Bit 7**	**FD Error**
Bit 14	**Event2**	**Bit 6**	**DNErr**
Bit 13	**P-Lim**	**Bit 5**	**DNet**
Bit 12	**Event1**	**Bit 4**	**%Done**
Bit 11	**Estop**	**Bit 3**	**ANM Mode**
Bit 10	**Local**	**Bit 2**	**Mode**
Bit 9	**R-Lim**	**Bit 1**	**In-Pos**
Bit 8	**Disable**	**Bit 0**	**Moving**

ANI **Query Analog Input Value**

4 ANI
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
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.

ANM **Analog Input Mode**

4 ANM 1
This command configures the MicroMini™ Controller analog input pin (J2-1) as either a digital input or analog input. Setting ANM to 1 configures the analog input as a digital input whose signal level is reported in the MicroMini™ Controller's status mask bit 3.

VLIM **Control Output Voltage Limit**

4 VLIM 650 (6.5 volts)
This command sets the Maximum output voltage supplied to a DC motor. **This command should only be used with units that have integrated Linear Amplifiers.**

ERR **Query for Actual Positional Error**

4 ERR
The selected MicroMini™ Controller node will respond with the present servo loop error value.

SD **Set Serial Response Delay** **4 SD 2**

The selected MicroMini™ Controller node will delay serial responses 500μs for each **SD** unit specified.

OK **Set Serial “OK” Response** **4 OK 1**

Activating this function causes the MicroMini™ Controller to respond with “ok” upon successful receipt of a command. Setting this parameter to zero disables this function. (Default =0)

CK **Set CHECKSUM calculation Mode** **4 CK 1**

Activating this function causes the MicroMini™ Controller to calculate checksums for all serial messages received and transmitted. Setting this parameter to zero disables this function. (Default=0) All messages transmitted by the MicroMini™ Controller will have appended a “#” and checksum byte.

Example: “0004 000A#8” All messages transmitted to the MicroMini™ Controller while this mode is active should include the standard syntax plus the appended “#” and calculated checksum byte terminated by a carriage return. The checksum is calculated by adding the absolute value of each character without a carry. The resulting eight-bit value is the checksum.

Configuration Related Commands:

POR **Set Proportional Loop Gain** **4 POR 20000 (4K-32K)**

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.

I **Set Integral Loop Gain** **4 I 50 (1 - 32K)**

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.

DER **Set the Derivative Loop Gain** **4 DER -25000 (1K-32K)**

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.

AE **Auto Enable on boot** **4 AE 1**

This command causes the MicroMini™ Controller Enable/Home next time it boots (AE=1).

AV **Auto Velocity on boot** **4 AV 500**

This command causes the MicroMini™ Controller Enable next time it boots and run its motor at the given velocity .

RN **Reset Node** **4 RN**

This command causes the MicroMini™ Controller to perform a power on reset operation.

RE **Reverse Encoder Phasing** **4 RE 1**

This parameter allows the encoder phasing to be reversed to facilitate closure of the control loop.

RD **Reverse Operational Direction**

4 RD 1

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.

SR **Set MicroMini™ Controller Loop Sample Period**

4 SR 1000

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

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.

Flash EEprom Related Commands:

EEPSAV **Save all parameters in EEprom**

4 EEPSAV

This parameter allows saving of all current configuration parameters to EEprom. Please note that for the MicroMini™ Controller to use the saves parameters, the EEBOOT command must be configured.

EEBOOT **Use configuration from EEprom on boot**

4 EEBOOT 1 (0/1)

This command when active (EEBOOT 1) tells the MicroMini™ Controller to read all configuration parameters from EEprom next boot cycle.

EEADDR **Write/Read EEprom memory address**

4 EEADDR 1024 (read)
4 EEADDR 1024,C (write)

This command performs write/read access to EEprom memory addresses. The first command shows the syntax for reading location 1024 in the EEprom. The second command shows the syntax for writing a "C" to location1024 in EEprom. Please note that this command performs the Read/Write command automatically and performs an automatic loading of EEdata as well.

EEDAT **Data for EEprom**

4 EEDAT

This command holds the data from a previous read or current write command.

EEWR **Write data to EEprom**

4 EEWR

This command executes a write to EEprom. EEdata and EEaddr should hold pertinent information.

EERD **Read data from EEprom**

4 EERD

This command executes read from EEprom. EEdata and EEaddr should hold pertinent information.

Macro Related Commands:

ME **Macro Execute** **4 ME 1**

This command is used to execute a preciously downloaded macro sequence.

MS **Macro Status** **4 MS**

This command is used to query the MicroMini™ Controller for which Macro it is currently executing. If the response is Zero, then it signifies that no Macro is executing.

Indexing Commands:

ID **Index Destination** **4 ID 2000**

This parameter permits the user to configure the destination position for an indexing operation.

IS **Indexing Velocity** **4 IS 2000**

This parameter permits configuration of the indexing profile velocity.

IA **Indexing Acceleration** **4 IA 100**

This parameter permits configuration of the indexing profile acceleration.

IDC **Indexing Deceleration** **4 IDC 100**

This parameter permits configuration of the indexing profile deceleration.

IC **Index Counts** **4 IC 50 (0-65000)**

This parameter determines the the number of limited cycles to be executed.

IO **Index Destination (Final)** **4 IO 100000**

This is the final move destination following index cycle completion.

IE **Indexing Enable** **4 IE 1**

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/Limited/Continuous Indexing** **4 IM (X)**

Single Operation **4 IM 0**

Indexing operation in single step absolute mode in response to an external pulse applied to J2, pin 8, or the invocation of the ITR serial index trigger command.

Continuous Operation **4 IM 1**

Indexing operation in continuous absolute mode in response to an external pulse applied to J2, pin 8, or the invocation of the ITR serial index trigger command.

Limited Operation **4 IM 2**

Indexing operation for a specified number of absolute cycles in response to an external pulse applied to J2, pin 8, or the invocation of the ITR serial index trigger command.

Single Operation

4 IM 3

Indexing operation in single step relative mode in response to an external pulse applied to J2, pin 8, or the invocation of the ITR serial index trigger command.

Continuous Operation

4 IM 4

Indexing operation in continuous relative mode in response to an external pulse applied to J2, pin 8, or the invocation of the ITR serial index trigger command.

Limited Operation

4 IM 5

Indexing operation for a specified number of relative cycles in response to an external pulse applied to J2, pin 8, or the invocation of the ITR serial index trigger command.

ITR **Index Serial Trigger**

4 ITR

This command triggers an indexing operation via the serial interface.

ITD **Index Destination Delay**

4 ITD 2000

This parameter allows the user to determine the dwell time at the index destination position. Each unit is equal to a 500μs-time delay.

ITZ **Index Zero Delay**

4 ITZ 2000

This parameter allows the user to determine the dwell time at the index zero position. Each unit is equal to a 500μs-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. This function is automatically cleared when the selected capture input is triggered.

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.

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. This function is automatically cleared when the selected capture input is 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>label

4 ST 10>mainloop 9>loop2

Adding a label name to the mask parameter (separated with a >) provides a means of implementing a logical goto operation for structured control. If the MicroMini™ Controller system status matches a mask, a jump to the appropriate label will be performed.

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.

Repeat **Repeat Command**

Repeat 4 repeat 4 times

The command allows a portion of the command file instructions to be repeated a number of times. The command **REND** must be used when the **REPEAT** command is invoked. Please note also that the repeat count can be used inside the loop. The arithmetic operators possible are:
add(+), subtract(-), multiply(*), and divide(%).

Rend Repeat End **Rend**

This command marks where the Repeat code stops.

Repeat Examples

```
Example #1
4 HO
4 EN
REPEAT 4                /*the code below will be executed 4 times*/
4LA 10000
4 M
4 ST 10
4 LA 0
4 M
4 ST 10
REND
```

```
Example #2
4 HO
4 EN
REPEAT 4                /*the code below will be executed 4 times*/
4 LA 1000*REPEAT        /*additionally, the LA destination will change from 1000 4 M
                        /*to 2000, 3000 and finally 4000 since it uses the REPEAT
                        /*count as a multiplier*/
4 ST 10
4 LA 0
4 M
4 ST 10
REND
```

/* */ **Comment Fields**

4 HO /*Main Loop*/

The slash asterisk allows comments to be inserted into command files.

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)
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)
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)
MicroMini™ Controller User Manual

4 Axis Configuration

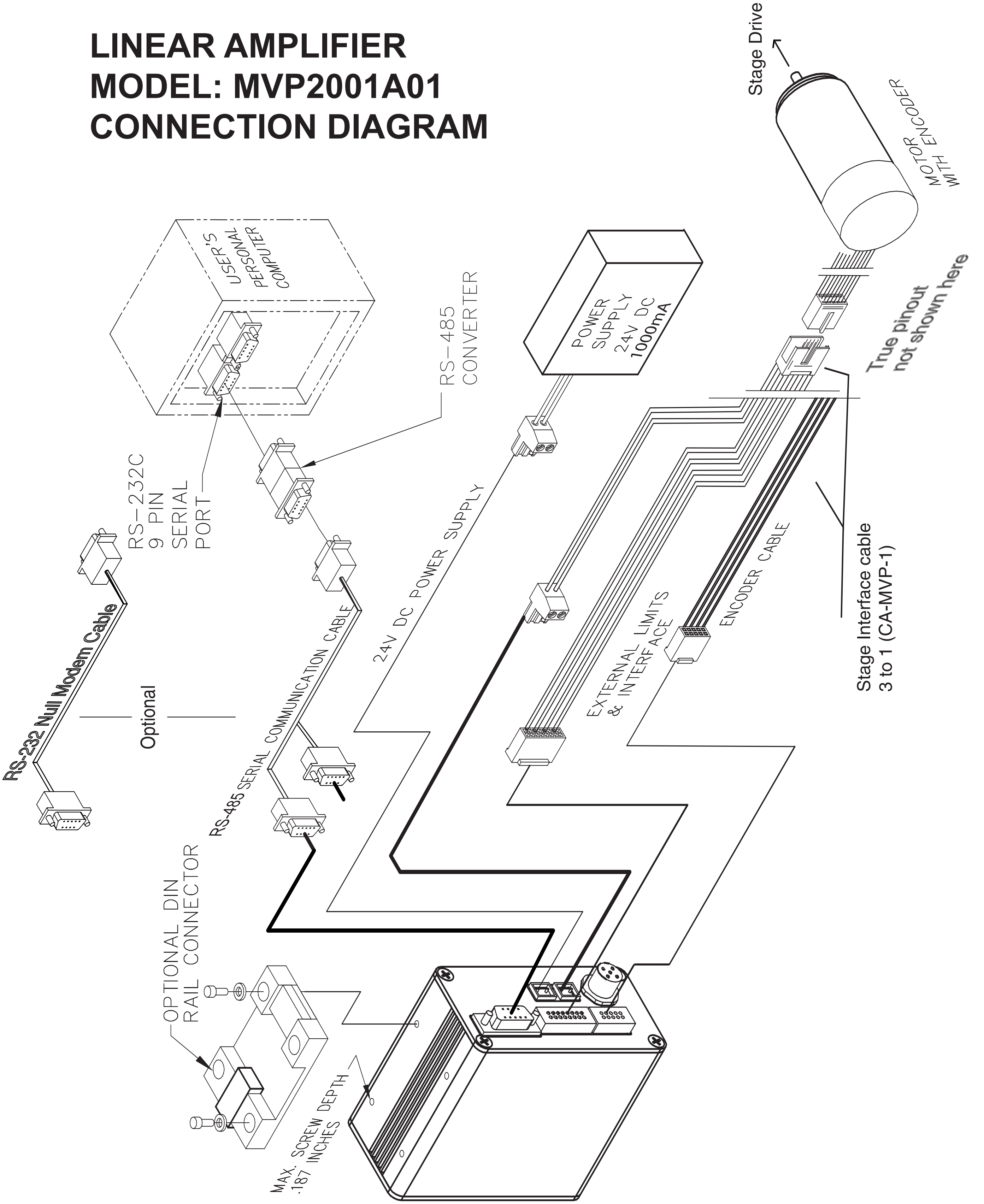
4 RS-485 Motion Controller
1 RS-485 Converter Module
1 4-axis Communication Cable
4 24VDC Power Supply
4 Stage Interface Cable (single axis)
MicroMini™ Controller User Manual

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

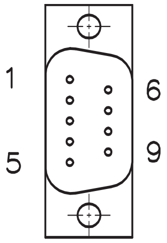
- Electronic Gearing
- Step-Direction Control
- LabView™ Virtual Instruments
- Example Source Code

LINEAR AMPLIFIER
MODEL: MVP2001A01
CONNECTION DIAGRAM


Stage Drive
MOTOR WITH ENCODER
USER'S PERSONAL COMPUTER
POWER SUPPLY 24V DC 1000mA
True pinout not shown here
RS-485 CONVERTER
RS-232C 9 PIN SERIAL PORT
Stage Interface cable 3 to 1 (CA-MVP-1)
ENCODER CABLE
EXTERNAL LIMITS & INTERFACE
24V DC POWER SUPPLY
RS-232 Null Modem Cable
Optional
RS-485 SERIAL COMMUNICATION CABLE
OPTIONAL DIN RAIL CONNECTOR
MAX. SCREW DEPTH .187 INCHES

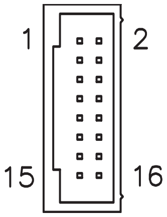
External Connections

J1 RS232C/485 Remote Communications Interface



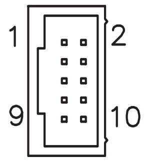
Pin	1	Not Connected
	2	RxD/RS485-
	3	TxD/RS485+
	4	Not Connected
	5	Ground
	6	Not Connected
	7	Not Connected
	8	Not Connected
	9	Not Connected

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



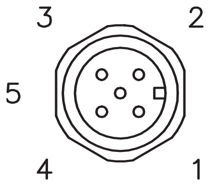
Pin	1	Analog Input Signal (0 to 5 VDC, 10 bits)		
	2	Analog Reference Ground		
	3	External Opto Power (+5 VDC)		
	4	Emergency Stop Input		
	5	Positive Hard Limit Input	Status bit 13	
	6	Negative Hard Limit Input	Status bit 15	
	7	External Event #2, Homing	Status bit 14	also: position capture input, Status mask input, index input
	8	External Event #1,	Status bit 12,	also: position capture input, index trigger, etc.
	9	External Drive Enable (not applicable) 0 = disable 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		

J3 Encoder Interface (HEDS 5000 Series Compatible)



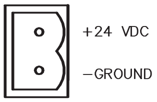
Pin	1	Encoder Channel A Input
	2	Encoder Power (+5 VDC)
	3	Encoder Return (Ground)
	4	Not Connected
	5	Encoder Return (Ground)
	6	Encoder Return (Ground)
	7	Encoder Power (+5VDC)
	8	Encoder Channel B Input
	9	Encoder Power (+5 VDC)
	10	Encoder Channel Z (index) Input

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



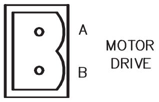
Pin	1	Drain (Case)
	2	V+ (Power Input)
	3	V- (Power Return)
	4	CAN_H (Communications Interface)
	5	CAN_L (Communications Interface)

J5 Main Power



Pin	1	+24 VDC
	2	Ground

J6 Motor Drive

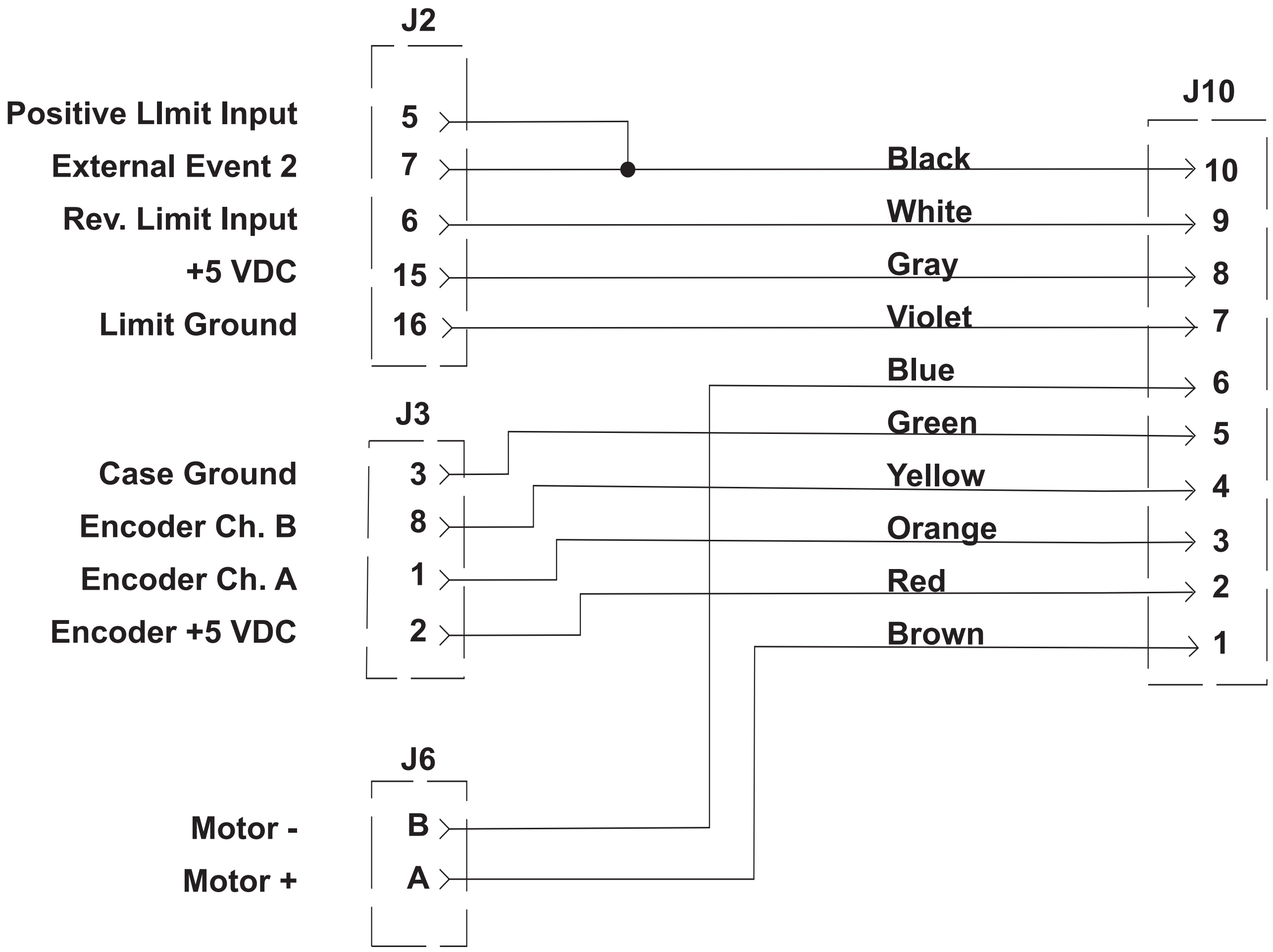


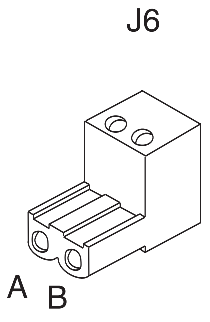
A

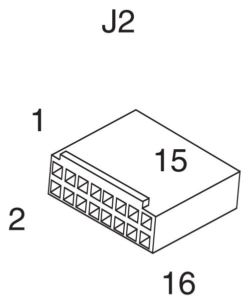
B

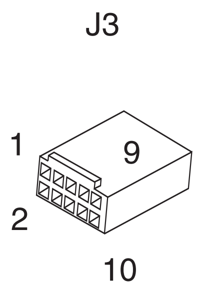
Appendix D

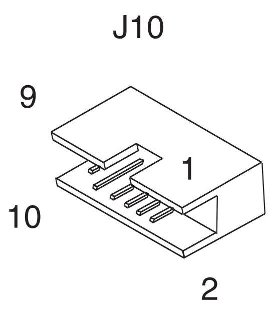
MicroMini™ Stage Controller Interface Cable







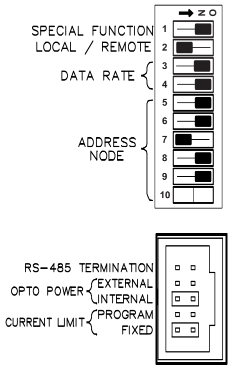




Troubleshooting

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.

To make sure you are operating in the Local Mode, configure the front panel switches as follows:



Apply power and observe the LED display and verify that every LED flashes as the unit performs the start-up diagnostics. The **Module/Net Status** LED and the **Move in Progress** LED's should remain illuminated when the diagnostics are completed. The motor should be running at a low to moderate speed and should maintain that velocity as it is loaded. If these conditions do not exist, contact NAI at 1-800-360-4598 (USA and Canada) for MicroMini™ Controller technical assistance. Connecting the controller to a data terminal or computer in remote mode will aid in diagnosis of your problem.

In the Remote Mode (Switch 2 to the right), after power is applied, the **Module/Net Status** LED will be flashing and the **Motor In Position** LED will be illuminated. The start-up message **"MVP2001 Ready"** should appear on your communication terminal. At this point you should be able to issue serial commands to control the module. By issuing a **"Home"** command followed by an "**Enable**" command the motor should begin to servo in position. This can be verified by attempting to turn the motor shaft. The motor should resist the efforts to move it from the **Home** position manually. If these conditions do not exist, verify you have established communications. Recheck all connections and configuration settings. If necessary, contact NAI for assistance. You should be able to access the computer and controller while talking to the support engineer.

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. Applications have been done using MicroMini™ Controller Series controllers that provide 0.5μm accuracy and 0.25μm repeatability using low resolution magnetic encoders with 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, Inc. 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).

Appendix F-2

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, Inc. 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:

Switch 7	**Switch 8**	**Switch 9**	**Switch 10**	
1000	5000	10000	20000	(RPM)

*ALL SWITCHES **RIGHT** IS 250 RPM

examples

To set a range of 5000 rpm, only Switch 8 would be turned LEFT.
To set a range of 20,000 rpm, only Switch 10 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 which is a position other than “home”.

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. 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.

What baud rates are available with the MicroMini™ Controller's serial communications interface?

The MicroMini™ Controller uses standard serial communications baud rates.
These include the following:

9600 baud N,8,1
19200 baud N,8,1
38400 baud N,8,1
57600 baud N,8,1

How do I send commands to the MicroMini™ Controller?

Typically, each MicroMini™ Controller has a unique address or ID. Every command must begin with this address and all parameters are separated by spaces. All commands *must* be terminated with a carriage return. For example: to home unit #4 the following instruction is sent "4 HO" terminated by a carriage return.[ENTER key or ASCII character 13].

How is the MicroMini™ Controller' serial response structures?

The MicroMini™ Controller responds with its address header (typically 4 digits - address and leading zeros) a space and data response which can be 4 or 8 digits long depending on the type of data requested.

For example:

A status response from unit #4 will be;
"0004 000A"
A position response might be;
"0004 00002710"
Please note that these packets are preceded by a Carriage Return
and Line Feed and are terminated with a Carriage Return and Line Feed.
The responses are also 16 bit and must be converted to 32 bit before evaluating

Can I convert an RS232 version of the MicroMini™ Controller to RS485 or vice versa?

Yes, all MicroMini™ Controllers can operate in either RS232 or RS485. Please contact NAI at 603-893-7393 or 800-360-4598 and ask for MicroMini™ Controller technical support.

I tell the MicroMini™ Controller to execute a Macro and nothing happens. What is wrong?

The MicroMini™ Controller must be programmed properly before it can execute a Macro. This programming includes "downloading" a command file previously created using the MVPDEMO program. Once this takes place the "ME" instruction can command the MicroMini™ Controller to execute the download macro.
Note: The MicroMini™ Controller must be configured for RS485 operation for Macro execution to take place.

Appendix G-1

Application 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.

Appendix G-2

The following example routines are intended for use with the MVPDEMO software available @ www.naimotion.com. They are designed to demonstrate the type of logical operational processes typically found in automation applications.

Initialization Examples:

This command file initializes and MVP servodrive, Node address = 4

```
4 sr 500        /*Use the “SR” command to set filter sample rate to 500us*/
4 ano 2000      /*The “ANO” command is used to set the programmable current limit*/
4 por 15000     /*Set the proportional gain to control motor stiffness*/
4 i 100         /*Use the integral gain to adjust position acquisition time*/
4 der 25000     /*The derivative gain affects the control loop damping*/
4 ho            /*The “HOme” command defines the present position as position zero*/
4 en            /*The “ENable” command activates the drive amplifier to allow servoing*/
```

At this point the motor should be servoing in a stopped position. If this is not the case, encoder phasing may be the problem. This condition can be verified by inoking the “RE” (reverse encoder) command.

This command file initializes multiple MVP servodrive modules, Node address = 0

```
0 sr 500        /*Use the “SR” command to set filter sample rate to 500us*/
0 ano 2000      /*The “ANO” command is used to set the programmable current limit*/
0 por 15000     /*Set the proportional gain to control motor stiffness*/
0 i 100         /*Use the integral gain to adjust position acquisition time*/
0 der 25000     /*The derivative gain affects the control loop damping*/
0 ho            /*The “HOme” command defines the present position as position zero*/
0 en            /*The “ENable” command activates the drive amplifier to allow servoing*/
```

At this point the motor should be servoing in a stopped position. If this is not the case, encoder phasing may be the problem. This condition can be verified by inoking the “RE” (reverse encoder) command.

Using Multiple Addresses:

Using address “0” via the RS-485 interface causes any MVP module receiving the message to respond to the command. This proves to be useful during initialization as well as for adjusting other global variables within a system. This address should not be used with commands where a response is expected. An alternative addressing scheme is detailed below. In this example, addresses (1) and (2) are used.

```
1,2 sr 500      /*Use the “SR” command to set filter sample rate to 500us*/
1,2 ano 2000    /*The “ANO” command is used to set the programmable current limit*/
1,2 por 15000   /*Set the proportional gain to control motor stiffness*/
1,2 i 100       /*Use the integral gain to adjust position acquisition time*/
1,2 der 25000   /*The derivative gain affects the control loop damping*/
1,2 ho          /*The “HOme” command defines the present position as position zero*/
1,2 en          /*The “ENable” command activates the drive amplifier to allow servoing*/
```

Homing Examples:

In most position control applications, the system operates between defined mechanical limits. In order to establish a relationship between these limits and the servo controller’s position domain once the power-up initialization is complete, some type of “Homing” sequence usually performed. The motor is normally driven toward some type of sensor or hard stop having a known location. Once this position has been located, the servo controller’s location is redefined to permit the operation of the control program to function with the desired mechanical limits. The HP and LP commands are used to select the proper polarity of the HOME and LIMIT inputs as required.

This example illustrates a simple homing sequence using external event input #1 (J2-8).

```
4 ho            /*Designate initial HOME location*/
4 en            /*Enable drive amplifier*/
4 v 20          /*Activate velocity mode to seek “Home” sensor*/
4 st 4109       /*Check status for sensor active*/
4 ho            /*Execute “Home” function on detection of sensor*/
```

To use external event input #2 (J2-7), change the status mask to 16397.

Appendix G-4

This example illustrates a simple homing sequence using an end of travel hard stop. Homing by this means usually doesn't provide the homing accuracy that can be achieved using switches, optical, magnetic, or inductive proximity sensors. In addition, utilizing this type of homing mechanism can potentially cause damage in geared mechanical systems.

```
4 v 20          /*Activate velocity mode to seek "Home" location*/
4 st 141        /*Check status for the following error flag*/
4 ho            /*Execute "Home" function on detection of the limit*/
4 la -100       /*Program a move to a location a safe distance from the limit*/
                /*to prevent the motor from overheating due to integration error*/
4 m             /*and move there.*/
```

This example illustrates a more complex homing sequence using external event input #1 (J2-8). This routine demonstrates how to detect the activation of an over travel limit switch during the homing process. When the over travel limit is detected, the motor direction is reversed in order to continue the search in the opposite direction.

```
4 ho            /*Designate initial HOME location*/
4 en            /*Enable drive amplifier*/
4 ls 1          /*Set limit sequence behavior*/
4 ha 1          /*Arm the "HOME" input*/
4 v 20          /*Activate velocity mode to seek "Home" sensor*/
                /*Check for positive limit or "HOME" complete*/
4 st 8206 st 16394>done
4 v -20         /*Reverse the direction of travel*/
                /*Check for positive limit or "HOME" complete*/
4 st -32754 st 16394>done
done:
```

At this point the home sensor has been located and the home position has been determined. Additional homing can be implemented to enhance precision if necessary. It is also possible to use the over travel limit sensor to detect the home position.

This example illustrates a more complex homing sequence using external event input #1 (J2-8). In this case, the sensor is detected a second time to allow the axis to be homed on a particular edge of the sensors signal. For added precision, the motor velocity is reduced for the second approach.

```
4 ho           /*Designate initial HOME location*/
4 en           /*Enable drive amplifier*/
4 v 20         /*Activate velocity mode to seek "Home" sensor*/
4 ha 1         /*Arm the "HOME" input*/
4 st 4106      /*Check status for sensor active*/
4 v -2         /*Reverse the direction of travel*/
4 st 13        /* until the sensor is cleared.*/
4 v 1          /*Move toward sensor again*/
4 ha 1         /*Arm the "HOME" input*/
4 st 4106      /*and recheck status*/
4 ho           /*Execute "Home" function on detection of sensor*/
```

To use external event input #2 (J2-7), change the status mask to 16397. The secondary input can also be used to locate the motor on the encoder index channel if one is present after the initial homing routine using a sensor as described above is completed. Simply connect the index channel to the second input and use the same procedure as before, with the new status mask value.

Indexing Examples:

This example routine demonstrates mode 5 indexing embedded within normal positioning tasks. Additionally, the state of event input #2 determines the velocity and acceleration of the indexing operation by redirecting program flow. In this example, the index trigger source is via command from the host device.

```
4 itd 10                 /*Set destination delay*/
4 is 10000               /*Set indexing velocity*/
4 ia 500                 /*Set indexing acceleration*/
loop:                    /*Flow control label*/
4 la 100000              /*Load first destination*/
4 sp 3000                /*Load initial move velocity*/
4 ac 300                 /*Load initial move acceleration*/
4 m                      /*Perform initial move*/
4 st 10 16394            /*Wait until complete, event input 0 or 1, ok */
4 id 500                 /*Configure indexing destination*/
4 im 5                   /*Select mode 5, relative/counts*/
4 itz 10                           /*Set zero location delay time*/
4 io 124000              /*Select index final location*/
4 ic 20                  /*Set number of indexing counts*/
4 ie 1                   /*Enable indexing mode*/
4 itr                    /*Provide serial indexing trigger*/
4 st 10 16394            /*Make sure indexing operation is complete*/
4 st 10 16394            /*Event input 0 or 1, ok */
4 st 10 16394
```

Appendix G-6

```
4 la 0                        /*Load next conventional destination*/
4 sp 1000                     /*Set desired velocity*/
4 ac 20                       /*and acceleration*/
4 m                           /*Perform next conventional move*/
4 st 10>exit st 16394>slow    /*Wait until complete, event input 0 or 1, ok */
slow:                         /*Flow control label, slow motion operation*/
4 ia 20                       /*Set indexing acceleration to slow*/
4 is 1000                     /*Set indexing velocity to slow*/
4 itd 10                      /*Increase index destination delay*/
4 st 10>loop st 16394>loop    /*Execute next loop*/
exit:
```

This routine demonstrates indexing mode 0. This mode provides absolute indexing for a single index cycle. The trigger source can be either an external command or event input #1 (J2-8).

```
4 ho          /*Designate HOME location*/
4 en          /*Enable drive amplifier*/
4 por 12000   /*Tighten loop gain and*/
4 i 30        /*integral term*/
4 itz 10              /*Set zero delay in milliseconds*/
4 itd 10      /*Set destination delay in milliseconds*/
4 is 10000    /*Set indexing velocity*/
4 ia 100      /*Set index acceleration*/
4 id 500      /*Set index destination*/
4 io 0        /*Select final destination*/
4 im 0        /*Select index mode- 0 = absolute/single step operation*/
4 ie 1        /*Enable indexing mode*/
```

The system is now configured and is waiting for the index strobe on event input#1
Interface connector J2, pin 8. To trigger serially, use the ITR command.

This routine demonstrates indexing mode 1. This mode provides absolute indexing for an unlimited number of index cycles. The trigger source can be either an external command or event input #1 (J2-8). A second trigger will terminate execution

```
4 ho          /*Designate HOME location*/
4 en          /*Enable drive amplifier*/
4 por 12000   /*Tighten loop gain and*/
4 i 30        /*integral term*/
4 itz 10              /*Set zero delay in milliseconds*/
4 itd 10      /*Set destination delay in milliseconds*/
4 is 10000    /*Set indexing velocity*/
4 ia 100      /*Set index acceleration*/
4 id 500      /*Set index destination*/
4 io 0        /*Select final destination*/
4 im 1        /*Select index mode- 1 = absolute/repeat operation*/
4 ie 1        /*Enable indexing mode*/
```

The system is now configured and is waiting for the index strobe on event input#1. Interface connector J2, pin 8. To trigger serially, use the ITR command.

This routine demonstrates indexing mode 2. This mode provides absolute indexing for a predetermined number of index counts. The trigger source can be either an external command or event input #1 (J2-8).

```
4 ho          /*Designate HOME location*/
4 en          /*Enable drive amplifier*/
4 por 12000   /*Tighten loop gain and*/
4 i 30        /*integral term*/
4 itz 1       /*Set zero delay in milliseconds*/
4 itd 1       /*Set destination delay in milliseconds*/
4 is 10000    /*Set indexing velocity*/
4 ia 100      /*Set index acceleration*/
4 id 500      /*Set index destination*/
4 io 0        /*Select final destination*/
4 im 2        /*Select index mode- 2 = absolute/counts*/
4 ic 20       /*Select number of index counts*/
4 ie 1        /*Enable indexing mode*/
```

The system is now configured and is waiting for the index strobe on event input#1. Interface connector J2, pin 8. To trigger serially, use the ITR command.

Appendix G-8

This routine demonstrates indexing mode 3. This mode provides relative indexing for a single index cycle. The trigger source can be either an external command or event input #1 (J2-8).

```
4 ho          /*Designate HOME location*/
4 en          /*Enable drive amplifier*/
4 por 12000   /*Tighten loop gain and*/
4 i 30        /*integral term*/
4 itz 10              /*Set zero delay in milliseconds*/
4 itd 10      /*Set destination delay in milliseconds*/
4 is 10000    /*Set indexing velocity*/
4 ia 100      /*Set index acceleration*/
4 id 500      /*Set index destination*/
4 io 0        /*Select final destination*/
4 im 3        /*Select index mode- 3 = relative/single step operation*/
4 ie 1        /*Enable indexing mode*/
```

The system is now configured and is waiting for the index strobe on event input#1*/
Interface connector J2, pin 8. To trigger serially, use the ITR command */

This routine demonstrates indexing mode 4. This mode provides relative indexing for an unlimited number of index cycles. The trigger source can be either an external command or event input #1 (J2-8).

```
4 ho          /*Designate HOME location*/
4 en          /*Enable drive amplifier*/
4 por 12000   /*Tighten loop gain and*/
4 i 30        /*integral term*/
4 itz 10              /*Set zero delay in milliseconds*/
4 itd 10      /*Set destination delay in milliseconds*/
4 is 10000    /*Set indexing velocity*/
4 ia 100      /*Set index acceleration*/
4 id 500      /*Set index destination*/
4 io 0        /*Select final destination*/
4 im 4        /*Select index mode- 4 = relative/repeat operation*/
4 ie 1        /*Enable indexing mode*/
```

The system is now configured and is waiting for the index strobe on event input#1.
Interface connector J2, pin 8. To trigger serially, use the ITR command.

Appendix G-9

This routine demonstrates indexing mode 5. This mode provides relative indexing for a predetermined number of index counts. The trigger source can be either an external command or event input #1 (J2-8).

```
4 ho          /*Designate HOME location*/
4 en          /*Enable drive amplifier*/
4 por 12000   /*Tighten loop gain and*/
4 i 30        /*integral term*/
4 itz 1       /*Set zero delay in milliseconds*/
4 itd 1       /*Set destination delay in milliseconds*/
4 is 10000    /*Set indexing velocity*/
4 ia 100      /*Set index acceleration*/
4 id 500      /*Set index destination*/
4 io 0        /*Select final destination*/
4 im 5        /*Select index mode- 5 = relative/counts*/
4 ic 20       /*Select number of index counts*/
4 ie 1        /*Enable indexing mode*/
```

The system is now configured and is waiting for the index strobe on event input#1,
Interface connector J2, pin 8.

This routine demonstrates indexing mode 5. This mode provides relative indexing for a predetermined number of index counts. The trigger source can be either an external command or event input #1 (J2-8).

```
4 ho          /*Designate HOME location*/
4 en          /*Enable drive amplifier*/
4 por 12000   /*Tighten loop gain and*/
4 i 30        /*integral term*/
4 itz 1       /*Set zero delay in milliseconds*/
4 itd 1       /*Set destination delay in milliseconds*/
4 is 10000    /*Set indexing velocity*/
4 ia 100      /*Set index acceleration*/
4 id 500      /*Set index destination*/
4 io 124000   /*Select final destination*/
4 im 5        /*Select index mode- 5 = relative/counts*/
4 ic 20       /*Select number of index counts*/
4 ie 1        /*Enable indexing mode*/
```

The system is now configured and is waiting for the index strobe on event input#1
Interface connector J2, pin 8.

Appendix G-10

Limit Handling Examples:

This example demonstrates a procedure for implementing over travel limits. These limits are useful for protecting the motor and drive system from damage that can occur when the system encounters an end of travel condition. Without protection, the servo controller will try to recover the following error that accumulates as the motor is stalled, producing high currents and high torque to be generated. The MVP2001 is equipped with 2 over travel limit inputs designed to provide real protection for drive system components. The positive limit input is located on the J2 connector, pin5, and the negative limit input is on J2, pin 6.

```
message Main Loop
start:
4 ho
4 en
4 por 12000
4 i 10
4 st 10 st 14
main:
message Executing primary routine
4 la 200000
4 sp 2000
4 ac 100
4 m
4 st 10 st 8202>pos_limit st -32758>neg_limit
message Returning to Origin
4 la 0
4 m
4 st 10>main st 8202>pos_limit st -32758>neg_limit
pos_limit:
message Moving to clear positive limit sensor
4 v -5
4 st 13 */Wait for limit input to become inactive/*
message Limit Clear....restarting routine
4 v 0
4 st 14>start st 10>start
gotostart

neg_limit:
message Moving to clear negative limit sensor
4 v 5
4 st 13 */Wait for limit input to become inactive/*
message Limit Clear....restarting routine
4 v 0
4 st 14>start st 10>start
gotostart
```

Program Flow Control Examples:

This routine demonstrates how to use event#, (J2-8) input to control program flow. This provides the ability to vary the operational sequence of an application or respond to exceptions such as limits or other error conditions.

```
loop:message Executing main loop
4 la 10000                  /*Load primary move destination*/
4 sp 1000                   /*Set initial velocity*/
4 ac 100                    /*Set initial acceleration*/
4 m                         /*Move to destination position*/
4 st 10 st 4106>loop2       /*Status test- Event input#1 active, goto loop2,*/
                            /*else continue*/
4 la 0                      /*Load move back to origin location*/
4 m                         /*Return move*/
4 st 10>loop st 4106>loop2  /*Status test-Event input#1 active, goto loop2,*/
                            /*else continue*/
loop2:message Executing secondary loop
4 la 100000                 /*Load secondary destination*/
4 sp 7000                   /*Set secondary velocity*/
4 ac 300                    /*and acceleration*/
4 m                         /*Initiate move to new destination*/
4 st 4106 10>loop           /*Status test- Event input#1 active, continue*/
                            /*else return to primary operation*/
4 la 80000                  /*Load next process position*/
4 m                         /*Initiate move*/
4 st 4106 10>loop           /*Status test- Event input#1 active, continue*/
                            /*else return to primary operation*/
4 la 40000                  /*Load next process position*/
4 ac 200                    /*and adjust acceleration, this step*/
4 m                         /*Move to new location*/
4 st 4106 10>loop           /*Status test- Event input#1 active, continue*/
                            /*else return to primary operation*/
4 sp 8000                   /*Adjust velocity and acceleration for*/
4 ac 300                    /*final secondary move*/
4 la 0                      /*Load new destination position and move*/
4 m
4 st 10>loop st 4106>loop2  /*Continue with selected routine*/
```

Appendix G-12

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 spa)
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: spa[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:
Blue indicates user comments actually allowable in code in the MVP Demo .cmd files.
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.

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:

Blue indicates user comments allowable in the MVP Demo .cmd files.

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

Green indicates messages. They will work in command files. These commands can be eliminated, along with the delays that allow time to read them.

Appendix G-14

(cont'd) Page 3 of 3 Rotary Stage Homing

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
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

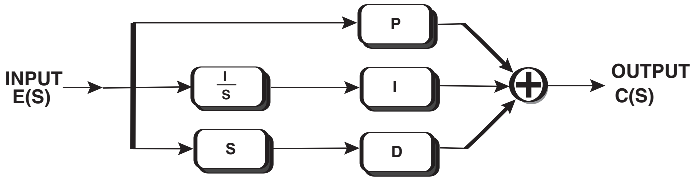
```
1 st 16394>pullaway st 16392>pullaway st 16648>pullaway st 14>home st 10>home
pullaway:
1 v-100
1 st 13
delay 25
1 v 0
home:
1 ha 1
1 fd 100
1 v 700
1 st 16394>section2 st 16392>section2
section2:
1 fd 130
1 la -102000
1 m
1 st 10
1 ho
1 la 0
```

NOTES:
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:



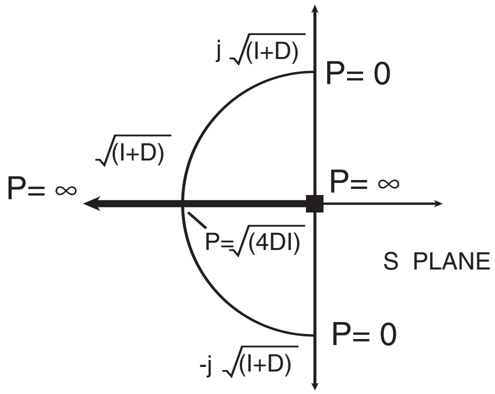
$$\frac{C(s)}{E(s)} = \frac{Ps + I + Ds^2}{s}$$

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:

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

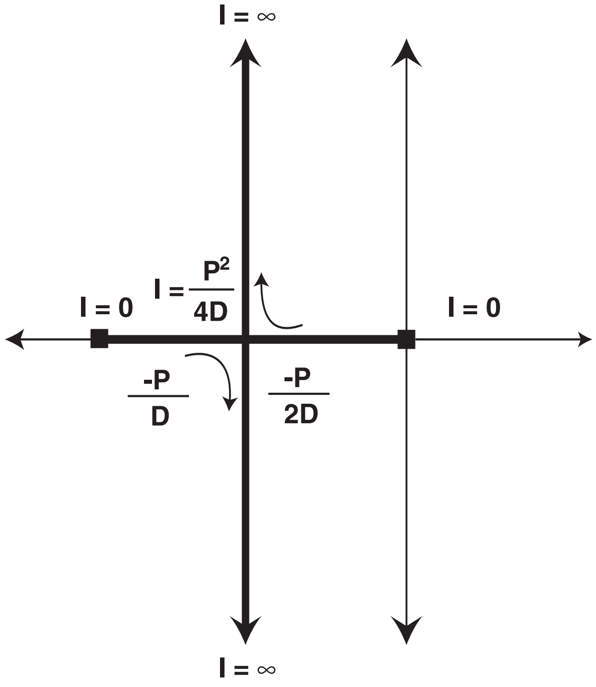
The two zeros are real-valued when 4**DI** ≥**P**^{2}. A Bode plot of the PID transfer function with real valued zeros reveals that one of the zeros is used to brake the 20 dB/decade descent associated with the integrator, and the other one is used to provide a 20dB/decade rise and positive phase lead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



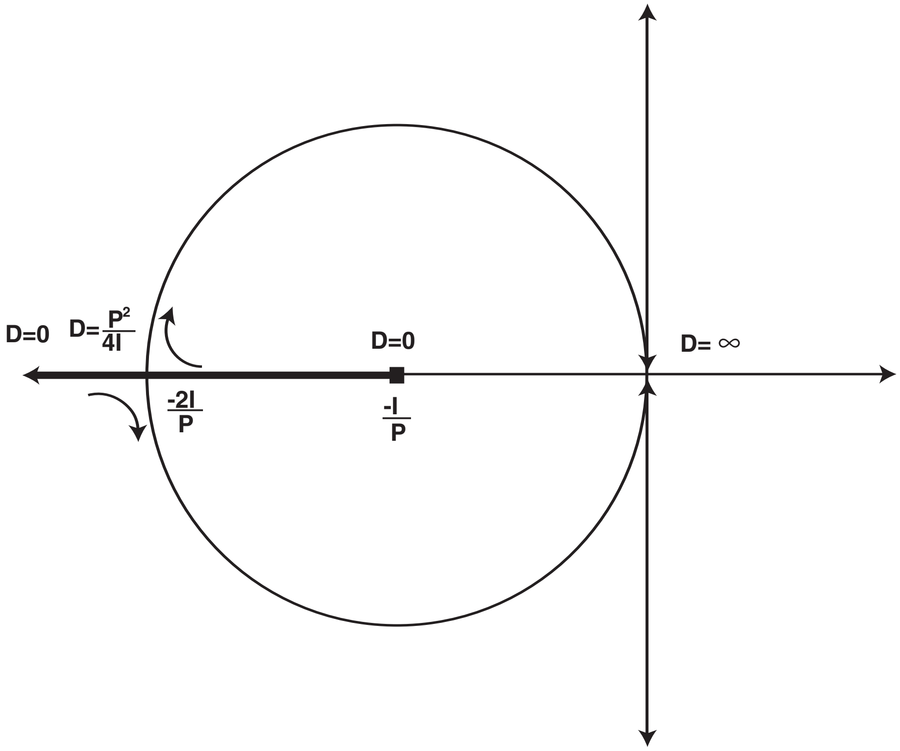
The **Proportional** term amplifies the error signal by a constant amount. However, the **P** term is not in series, but in parallel with **I** and **D**, which implies that **P** cannot be used to scale the transfer function amplitude at all frequencies. Instead, the **P** term interacts with the **I** 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:

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

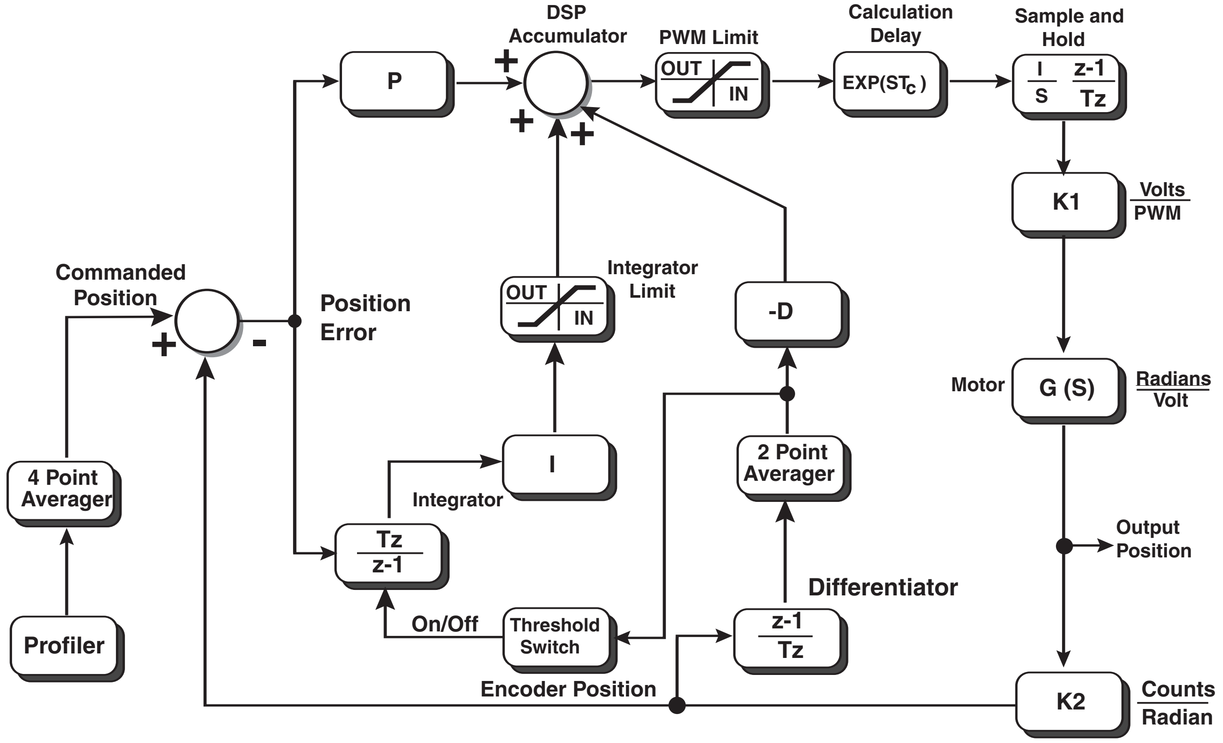
Although these are not the only **z**-domain representations of these functions, they are widely used in control applications.

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.

- RS-232, RS-485 Multidrop or DeviceNet™ (CAN) compatible interfaces
- Linear drive capable of 10 Watts continuous at 22°C ambient
- Accepts Step/Direction Input (optional)
- Torque Limiting Capability
- Programmable Current Limit/Short Circuit Protection
- Flash Based Parameter Storage/EDS
- Indexing and Registration Support (Single, Continuous, Count, Absolute & Rel)
- Position Capture (2)
- Macro Programming Support
- Custom firmware support for embedded OEM applications.
- No external amplifier required
- Local mode speed range up to 20,000:1

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	±2,147,483,647 quadrature counts
Velocity Range	up to 1,966,020 quadrature counts/sec
Velocity Accuracy	±0.001% of max. rate
Servo Loop Update Time	Programmable (200μs Min)
Servo Tuning	Programmable PID filter
Acceleration Range	32,767 quadrature counts/sample period (programmable)

Power Requirement

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

Input voltage range	Linear 18-40 VDC

Communication

RS-232, RS-485 Multidrop, DeviceNet™

Controller Inputs

Encoder	Two channel, Single-ended +5VDC TTL compatible, All digital inputs isolated and interrupt driven
Analog	One analog input (0-5 VDC, 10 bit)
General Purpose	Two external event inputs, Two hard limits, One emergency stop. 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) (Also used for programmable current limit)

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

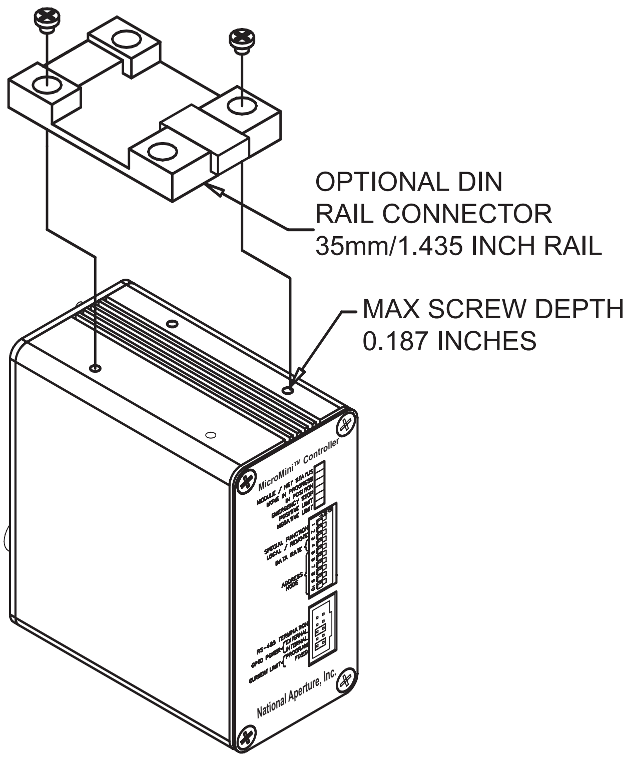
Environmental

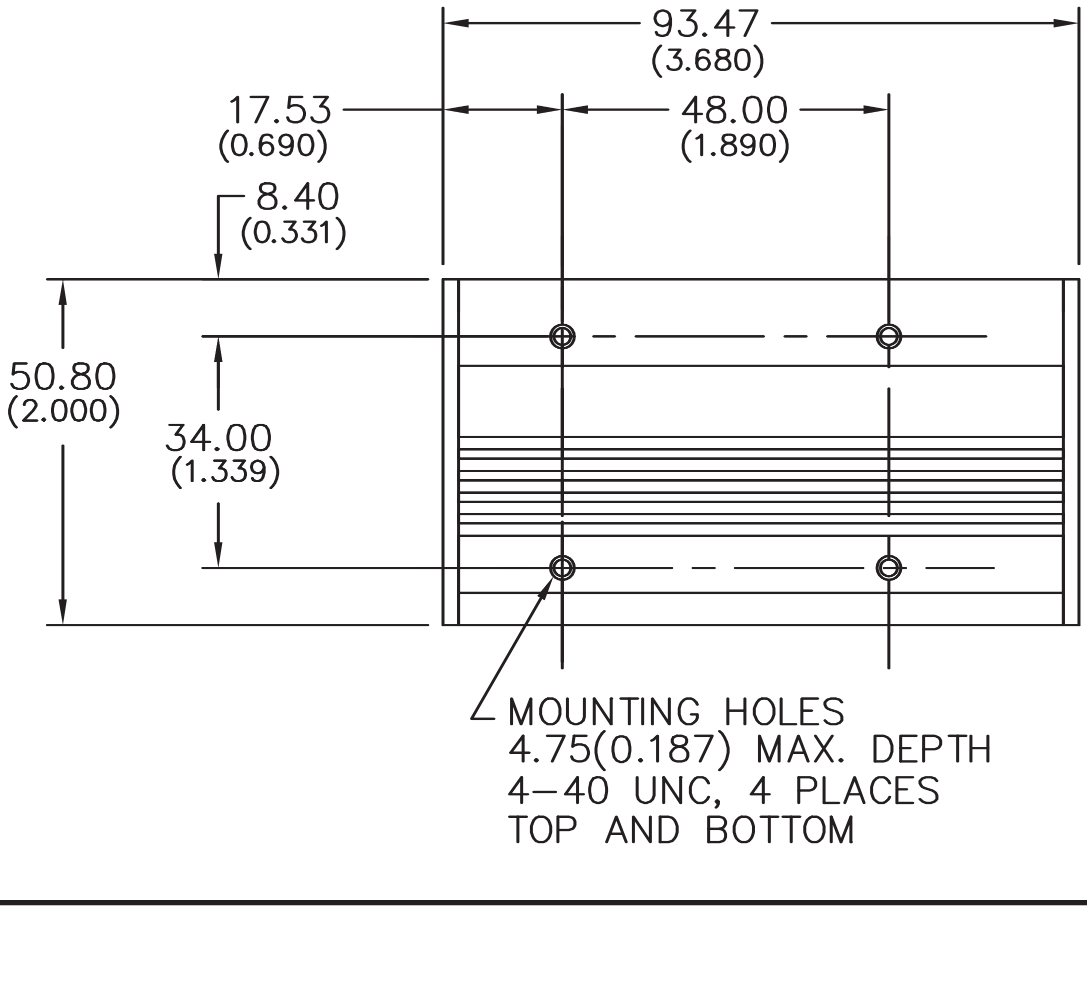
Controller Operating Temperature	0ºC to 70ºC	(32ºF to 158ºF)
Ambient Operating Temperature (standard version)	0ºC to 40ºC	(32ºF to 104ºF)
Ambient Storage Temperature (standard version)	-20ºC to 85ºC	(-4ºF to 185ºF)
Humidity Tolerance (standard version)	80% Rh, non-condensing	

Appendix J

Mounting

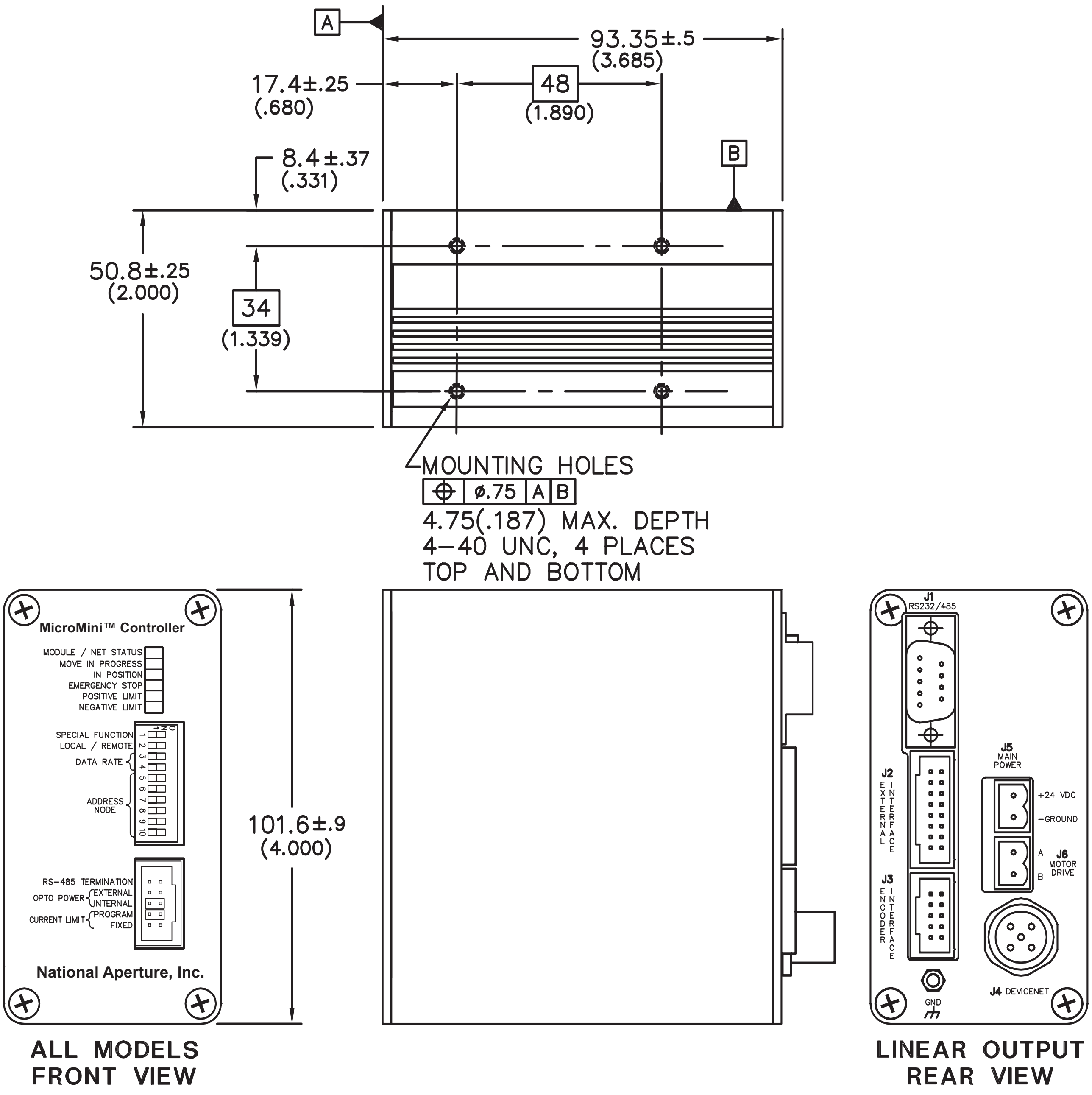
All MicroMini™ Controller modules can be mounted to machinery, in racks, or in cabinets with an optional DIN rail mount or with mounting screws. Regardless of which mounting system you use, **do not penetrate screw holes deeper than 4.75mm (3/16 inch)** from the surface of the controller module, or irreparable 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.



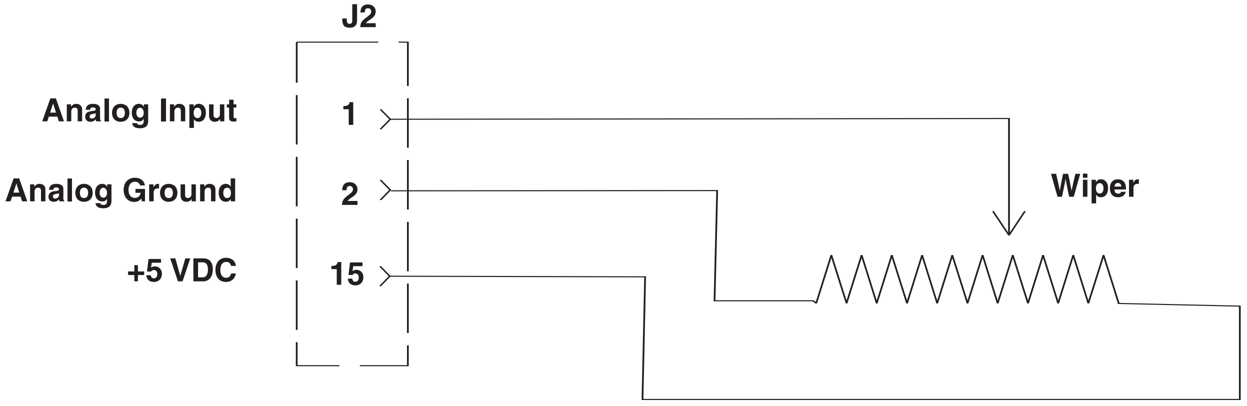


Appendix K

Dimensional Outlines



Typical Potentiometer Circuit



1K, 2W Potentiometer
Clarostat P/N RV4NAYSD102A

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 90 days 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