IMS™

INTELLIGENT MOTION SYSTEMS, INC.

Excellence in Motion™

HIGH PERFORMANCE MICROSTEPPER DRIVER & INDEXER

SOFTWARE REFERENCE MANUAL

370 NORTH MAIN ST., MARLBOROUGH, CT 06447, PH. (860) 295-6102, FAX (860) 295-6107

Software Reference Manual Revision 051794

CONTENTS

Congratulations on your purchase of the high performance indexer option with Variable Resolution Microstep Control. This unique indexer coupled with our powerful microstepping drivers combine to make a unique cost effective off the shelf solution that can provide designers with a versatile system to get you into the market fast and to provide you with the competitive edge needed in today's fast moving market.

This SOFTWARE REFERENCE MANUAL will assist you in obtaining the best performance from your stepper system. Its purpose is to provide immediate access to information that will facilitate a reliable and trouble-free installation. In addition to the Software Reference Manual, an optional Windows based GUI Indexer Software program is available. This powerful programming tool is an easy to use, menu driven utility file which enables you to quickly take advantage of the advanced programming features and system capabilities inherent in that are standard in our microstepping systems.

Although the SOFTWARE REFERENCE MANUAL and supporting documentation were designed to simplify the installation and on-going operation of your equipment, we recognize that the integration of motion control often requires answers to many complex issues. Please feel free to take advantage of our technical expertise in this area by calling one of our engineers to discuss your application.

THANK YOU!

Traditional designs normally rely on a fixed resolution step motor driver. In order to obtain high shaft speeds, step pulse rates must reach hundreds of thousands of steps per second. This approach requires elaborate and costly circuitry. Now, through the use of advanced techniques, we able to achieve comparable results by varying the step resolution for any required motor shaft velocity by integrating standard hardware with a technology called variable resolution microstep control in the indexer software. At low shaft speeds, the indexer produces high resolution microstep positioning for silent resonance-free operation. As shaft speed increases, the output resolution decreases using our "on motor pole" synchronization. As shaft speed decreases towards the end of a move, the target micro position is trimmed to 1/100 of a step to achieve and maintain precise positioning.

Although the "Command Resolution" is 1/100 of a step, the driver actually allows for a much finer resolution. A built-in 100 position compensation table enables calibration in increments of 1/256 step resolution.

This user programmable "Look-Up" table is located in the non-volatile memory and may be used for specific motor characterization.

Communication to the INDEXER is through the use of high level commands issued over an RS-422 link. This type of communication permits precise motion control and the ability to communicate with multiple units using only one serial port, at a standard speed of 9600 baud.

GENERAL

The INDEXER option comes complete with 2048 bytes of non-volatile memory for the storage of operational parameters and motion sequence programs. Extensive signal buffering, optical isolation and a differential serial interface provide high reliability in harsh industrial environments. Standard features include:

Simple RS-422* serial communications interface
Party Line mode
Assignable series address
Speeds to 20,000 steps per second
Motor shaft speeds to 6,000 RPM
Speed alterable with ramping "on the fly"
Unique variable step mode resolution to 1/256 step
Resonance free motor operation
Bi-directional ramping between speeds
Programmable acceleration/deceleration ramps
Programmable driver run/hold current control 0-100%
Register up to 16.7 million steps per Move command
Absolute or relative position commands
Read position/encoder register counters while moving
Receive/send commands while moving
3 buffered general purpose input ports
3 buffered general purpose output ports
Vector on port instruction
Soft decelerate stop command and input
Moving output signal
Optically isolated limit and home switch inputs
Programmable limit polarity
Self-contained home routine
2 speed bi-directional jog inputs
Over 50 motion command instructions
Programmable trip point
Go on strobe feature
Single step attribute
2048 byte non-volatile memory

**Optional RS-232 to RS-422 converter is available, IMS Part number CV-3222.

COMMUNICATION

PC's or terminals that provide a serial RS-232 communications port may communicate with the indexer/driver models by utilizing INTELLIGENT MOTION SYSTEMS optionally available CV-3222 communications adapter or -2 option board. These small adapter modules provide an "in-line" conversion of single ended RS-232 signals to differentially driven RS-422 protocol.

ENCODER

The INDEXER/DRIVER modules "E" option provides encoder feedback for closed-loop applications that require positional verification and stall detection. A built in line count multiplier extends normal encoder resolution by a factor of "4X." The "E" option is available upon initial order only and not as an upgrade to existing units. A -DE option board is available which allows operation with a differential encoder.

SOFTWARE

QuickSTEP is a GUI (Graphical User Interface) menu driven software program that runs under Windows and provides an easy yet powerful method to control the indexer. This program is designed to get the user up and running with the indexer very quickly.

INTERFACE

The BB-34 breakout board comes with an 18 inch ribbon cable with a 34 position connector that plugs directly onto the INDEXER/DRIVER modules. At the opposite end is a double row 34 position screw type terminal strip to allow easy access to all interface functions.

The INDEXER/DRIVER modules feature full duplex, RS-422 party line communications. The 4 wire interface implements a differential transmission/receiver pair. The differential operation provides a high degree of reliability in industrial environments. Multiple modules may be connected in parallel to the host (terminal or computer) to act as "listeners", awaiting a single character representing its unique "name". Optional RS-232 adapters are also available.

Multiple units may be controlled from a single communications port via Party Line mode. Each controller is initialized by first assigning a unique single character "Name" prior to Party Line operation. Up to 32 controllers may be connected in a multi-drop configuration, requiring only a single host computer or hand held terminal with an RS-422 communications port. This Party Line mode permits full duplex communications with all controllers listening simultaneously for incoming commands.

NOTE: It is recommended that all multi-axis, Party Line configurations be powered-up simultaneously.

Speeds are specified in steps per second. Initial and final velocities are independent variables. Ramp parameters are computed internally depending on specified speeds, ramp slope and selected step resolution. Step motion is controlled at speeds up to 20,000 steps per second.

Bi-directional jog inputs allow manual jog positioning at one of two independent programmable speeds. The speed select input selects between the two independently programmable jog rates. The readable position counter is continually updated, simplifying teach or alignment applications development.

The Trip point is a programmable position that allows pre-defined operations to be triggered when the motor position matches the established Trip point position. A typical application may involve turning on a valve when a desired position is passed. A signal output is available as well as interrogating via commands. Stored programs can execute user defined sequences based on position.

The controller can ramp either up or down to the specified constant velocities or step resolution. The ramp slope may be altered prior to changing speeds while the trip point can be used to trigger velocity changes.

The units hosts 2048 bytes of non-volatile memory as a standard feature. The non-volatile memory provides storage of operational parameters, speeds and user programs. Direct read and write commands allow host use of the memory. Additionally 64 bytes of internal controller RAM is available for "Scratch Pad" (fast Read/Write) applications.

The Go input or "G" command executes user sequences that have been previously loaded and stored in memory. Terminals, hosts, etc. are NOT required, allowing low cost, stand-alone operation.

The INDEXER/DRIVER has a programmable current feature that permits control of motor winding current with 1% resolution. Independent settings for "run" and "hold" currents permit full motor torque when stepping and automatic power down to the hold current value after motion is complete. The "hold" current value is used to minimize motor power dissipation when in the idle mode.

The controller also implements a disable feature that turns off the entire output stage for sensitive applications requiring Quiet mode operation.

Communications are via RS-422 protocol and may operate in either Single or Party Line mode. Each Module receiver monitors the host and responds when receiving a matching Name character. Communication to each individual axis in Single mode is performed to assign the unique Name for each slave axis.

The Single mode provides user friendly one axis communication for set-up and debug functions. Set-up usually involves optimizing operational parameters, writing and storing programs and assigning a unique Name for Party Line operation.

Once the unique Name has been assigned and stored in the non-volatile memory, multiple axis may be connected in parallel for Party Line mode operation. An input on the INDEXER/DRIVER module allows convenient selection between modes.

Party Line protocol requires the correct input character sequence to address an axis. It is recommended that a PC or other computer be used to simplify programming.

Party Line operation provides for parallel communications of up to 32 INDEXER/DRIVER units from a single serial communications port.

To enable Party Line operation, the Party Line enable input is connected to ground. To disable Party line communication, and to put the INDEXER/DRIVER Module in a stand alone mode, the Party enable input can be left floating, or connected to the opto supply.

The 34 pin header provides auxiliary inputs and outputs. Buffered signals use optical isolation or TTL level buffers.

SIGNAL	FUNCTION
OUTPUT 1	General Purpose Output
OUTPUT 2	" " "
OUTPUT 3	" " "
INPUT 1	" " Input
INPUT 2	" " "
INPUT 3	" " "
*LIMIT A	Limit + Input
*LIMIT B	Limit - Input
*HOME	Home Input
JOG 1	Jog + Input
JOG 2	Jog - Input
JOG SPEED	Jog Speed Select Input
GO	Start Program
SOFT STOP	Stop Program
Moving	Motor Moving
Opto Supply	Input Opto Supply Voltage
+5V	+5V from internal regulator
GND	Signal ground for opto reference

*Optically isolated inputs

Spare Outputs are open collector TTL Compatible with 10k pull up resistors. The opto supply input supplies optical isolator power for limits, home and remote party line select.

Refer to the hardware reference manual for interface requirements.

The indexer communication rate is 9600 Baud. The line settings are shown below.

Host/terminal settings:

DataLength	Stop Bits	Parity	Baud
8	1	none	9600

The INDEXER section incorporates a buffered UART input. Because motion control is of the highest priority, processing of received information may be delayed if commands are sent while stepping at very fast rates. This condition may only occur at internal/external step rates exceeding 10,000 steps per second. In serial applications where commands are sent while motion is active, the user should monitor echoed data to avoid UART over-run. Another technique is to insert a small delay between characters.

Automatic reset occurs each time the power is turned on or a "~C" command is issued from serial communications. During reset all inputs and outputs will be at a high state. After hardware reset all parameters are initialized to factory set default values. Checks for Party Line operation and communication baud rate are initialized.

Resident non-volatile memory is accessed, and the parameters most recently stored by "S command are down-loaded, replacing the standard defaults in the working registers of the controller. The following block of parameters are stored and recovered as a set:

PARAMETER	STANDARD DEFAULTS
Initial velocity (I)	400 steps per second
Slew velocity (V)	3004 steps per second
Divide factor (D)	0 (Full Step)
Ramp slope (K)	10/10
Jog speeds (B)	30,200
Trip point (T)	off
Step Resolution (H)	1 (Auto Variable)
Auto power down	yes
Hold/Run current	5/25
Limit polarity	low
Auto position readout (Z)	off
Name (after reset)	undefined

NOTE: Commands that modify these parameters use the working registers inside the controller. Actual non-volatile memory storage is initiated by the "S" (save) command. Once initialization is complete, Jog and Go inputs are active to allow jogging or a low pulse on the Go input to execute a program previously stored in non-volatile memory. A terminal or host is NOT required for these functions and may be initiated from the Auxiliary Input/Output connector.

The words "host" and "terminal" are used interchangeably and refer to any source of commonly available standard RS-422 interface including; terminals, printer ports and UARTs (COM ports). A character is a stream or packet of 8 consecutive bits (1s and 0s) with a defined sequence. Each unique character represents a letter, number, punctuation or unreadable control character as defined by an international set of standards called ASCII. The speed that characters are sent or received is defined by the Baud rate. The serial communications link refers to any two wire, full or half duplex communication link.

If a PC is used for the communication interface, a terminal emulator program is all that is required to "talk" to the INDEXER. The Terminal program that is supplied with Microsoft's Windows is an example of a terminal emulator program.

The INDEXER/DRIVER tests input characters for the proper start sequence. In Single mode a Space (hex 20) initiates the sign-on sequence. If proper communication exists, an acknowledgment code "xxxx xxxx AMS MAX-2000 Vx.xx" * will be echoed back, indicating the Encoder controller date code, INDEXER controller date, and version number. If the Encoder option is not present, the acknowledgment code will not contain the Encoder date code. Motion or program commands may now be initiated. NOTE: Any program loaded at location 1600 in memory will execute on power-up, preempting the sign on message. ESC will abort the program, allowing sign on as described below.

Single Mode Start-up Sequence:

1) Connect a Terminal or PC running a Terminal Emulation program to the Driver.

2) Check all wiring for proper power connections.

3) Apply power to the driver.

4a) Press the key for the Party Line Name. (Optional if only using Single Mode)

4b) Press the SPACE key on the Terminal.

5) The indexer responds with the sign-on message.

6) Press the ENTER key.

7) Indexer responds with "#" and is ready to accept commands.

8) Press the "S" key to save the name assignment from step 4a.

9) Party Line operation can now be performed.

* Sign-on message for v1.14 and higher is xxxx xxxx ADVANCED MICRO SYSTEMS, INC MAX-2000 vX.XXi

Party Line operation requires each axis to have a unique Name assignment prior to establishing communications AND for the Party input to be properly grounded.

Name assignment is accomplished in Single mode by typing the single character assignment before the Space Bar sign-on and then issuing an "S" (save) command immediately after. (See previous page for sign-on message).

SETTING AXIS NAME:
1) Press Control C (~C) or cycle power to the driver/indexer. (Party input NOT grounded)
2) Type the character that will be the name for this unit. (Name will be echoed)
3) Press the space bar (ASCII 32). The unit will send the sign-on message.
4) Press the S key followed by Enter or Carriage Return to save the assigned name.
5) Ground the Party input for Party Mode operation. (Be sure Opto supply is powered).

When starting in Party Line the controller reads its Name from the non-volatile memory and then skips any sign-on procedure. At this point the INDEXER/DRIVER monitors the start of all host serial communication for a match of its name.

To communicate with the unit, the Name must be issued prior to a command or programming sequence followed by a "~J" or "~Enter" (Hex 0A) to perform a Line Feed (Enter)

DO NOT USE THE ENTER KEY (LF CR) FOR A LINE TERMINATOR IN PARTY MODE.

EXAMPLE: To move axis "X" 1000 steps in + direction: Type X+1000~J

RECOMMENDED NAMES

Valid Axis Name assignments include:
Upper case A through Z
Lower case a through z

ASCII	HEX
[	5B
\	5C
]	5D
^	5E
_	5F
`	60

INVALID NAMES

The following characters cannot be used:

ASCII	HEX
~c	03
CR	0d
LF	0A
@	40
*	2A

During Party Line operation characters will NOT be echoed to the host until the proper "Name" is detected. All Indexer/Drivers monitor concurrently the common TXD line from the host. Once the Name is received, the target axis will wake -up and start echoing as described above. The asterisk character (*) is a global address in indexer versions 1.15 and higher. All axis will respond to commands sent with this address.

Note: The following commands will not echo alpha data Party Line Mode: P,Q,q,X
Although alpha data is not echoed, it is received by the indexer in the case of the P command and the program statements will be stored correctly.

Command lines consist of an ASCII character followed by 0, 1 or 2 decimal ASCII numbers depending on command requirements. The user may edit the line prior to entry by using the BACKSPACE (^H or ASCII 8) key. The command line may be up to 12 characters long, including spaces. Spaces are optional between the command character and first number. Motion commands with 2 data fields require at least one space between the fields. In general, motion command characters are issued as UPPER CASE, while encoder command characters are lower case, with a few exceptions.

COMMAND	NOTE (SINGLE AXIS MODE OPERATION)
+1000 (Carriage Return)	Step 1000 steps in + direction
+ 1000 (Carriage Return)	Same as above
H 0 (Carriage Return)	Set fixed step resolution
H0 (Carriage Return)	Same as above
H (Carriage Return)	Same as above(0 is used by default)
R -1000 (Carriage Return)	Move to position -1000

In single mode all commands are executed upon receipt of a Carriage Return (hex 0D). In Party Line mode all commands are executed upon receipt of a ^J or ^Enter (hex 0A). In Single Mode the controller responds by echoing all characters (EXCEPT the Line Feed character hex 0A) up to and including the Carriage Return. A Line Feed character is automatically output following the Carriage Return. In Party Line mode the Indexer responds by echoing all the characters up to and including the Line Feed. The echoing of characters can be used as a "handshake" with a host system running a program to communicate with the Indexer.

Commands such as Jump and Loop instructions are only valid when used in the Program then Execute mode. The following can only be executed from programs stored in optional non-volatile memory:

J 0 5 (Carriage Return) Jump to location 0, 6 (n + 1) times.
J0 5 (Carriage Return) same as above.

Some commands result in a numerical display. These consist of whole numbers that may have preceding spaces and are followed by a Carriage Return and Line Feed character. Negative numbers are preceded by the minus "-" sign.

For each Motion command there are four cycles:

1. ENTRY
2. EXECUTION
3. RESULT
4. COMPLETION

Other commands have three cycles:

1. ENTRY
2. EXECUTION
3. RESULT

In the idle state the controller continually tests for Jog, Go or Command input. The Following describes each sequence operation that takes place on receipt of a command:

CYCLE 1. ENTRY

The input command and data are loaded via RS-422 interface. Command and data information is placed in a command line buffer as received. Editing is permitted in Single mode. ESCape aborts operation and returns to an idle state. A Carriage Return (^J or ^Enter for Party Line) terminates the Entry cycle and initiates execution.

CYCLE 2. EXECUTION

The command is processed. In the case of two consecutive action commands, execution will be delayed until any previous completion cycle has been completed.

CYCLE 3. RESULT

The result cycle outputs any numerical result required by the command, i.e. the position. The result type is signed numerical data, preceded by space padding and followed by a Carriage Return and Line Feed. If the result does NOT produce numeric data then the Carriage Return, Line Feed output indicates execution is complete.

CYCLE 4. COMPLETION

The completion phase is required for any Action command cycle. The following are Action commands:

ACTION COMMAND	COMPLETION CYCLE
GO	Until last instruction is complete
STEP RESOLUTION	Until previous action complete
CONSTANT SPEED	Until previous ramp is complete
FIND HOME	Until home is found
RELATIVE MOVE	Until full index is complete
+STEP INDEX	Until full index is complete
-STEP INDEX	Until full index is complete

During the Completion cycle (except for Go) any non-action command, such as read position, may be executed. The controller has he capability to queue up another action command during the Completion cycle of a preceding Action command. The Execution and Result cycle of this Pending command is delayed until the Completion phase is finished. This interval is called the Pending Period. During this Pending Period the command accepted is the one character interrupt (abort) command, limit switches, soft stop input, and home switch input.

External indication of Pending Period end, Execution and Result cycle of the pending instruction is the Carriage Return. The Go command is regarded as a command that has a continuous Pending (Instructions Queued) Period.

Interrupt commands are single character commands that will interrupt the operation in process as follows:

Abort: Any Action command may be terminated using the ESCape command.

PROCESS	RESULTING ACTION
Command line input	Clear input buffer
Program Mode	Exit without inserting "End"
Action command	Terminate all motion Hard Stop
Program execution	Terminate execution Hard Stop

NOTE: All process(es) are aborted upon ESCape.

The Soft Stop "@" can be either a command (Immediate mode) or a single character interrupt (Program mode). The Soft Stop operates only when motion resulting from action commands or instructions is taking place.

After velocity deceleration the process is terminated.

PROCESS	RESULTING ACTION
Pending period	Decelerate and cancel pending instruction.
Program execute	Decelerate then terminate execution.

During Pending Periods that are a result of Multiple and Constant Velocity commands (intra-speed ramping) and deceleration will be delayed until the previous ramp-to-speed has been completed.

Jog input and home speed is a special case of the constant velocity command. Intra-speed ramping is used if the programmed jog speeds are above the initial velocity. Homing does NOT employ a deceleration ramp on reaching the home sensor.

NOTE: In any mode jogging and command reception are mutually exclusive. A command can NOT be loaded while jogging and jogging can NOT be performed until the last command is complete. A command starts with the reception of the first command character.

The controller is designed to control step rates with a high degree of accuracy. As a result, step control is given priority over other processes. At high step rates this will manifest itself as a slight latency. The execution time increases when high step rates are active during command cycles. An example might be reading positions while moving at a high speed. Usually this latency has little affect at step rates below 10,000 steps per second. At speeds approaching the maximum step rate the processing latency may have to be taken into account.

When running a program (from the Go command) several "fetches" from the non-volatile memory are required along with the service time. This latency may allow several motor steps to occur before the desired action takes place.

Loop on port may exhibit similar latency effects at high speeds. The port will require a longer "true" condition to be recognized. A faster method is to implement the "wait for port" condition using the "Go-Sub," (branch on port) condition.

The INDEXER/DRIVER is designed utilizing the Xicor type X2416 EEPROM non-volatile memory for programs and parameters. These devices are rated to retain data for 100 years. However, as with all EEPROM devices, the number of times it may be reprogrammed is limited. Each time a cell is written a small number of electrons are trapped in the dielectric. After many write cycles the dielectric becomes less effective and the cell cannot retain its charge. Specifically, the X2416 has an endurance rating of approximately 460,000 cycles.

A special area of "Scratch Pad" memory of 64 bytes is accessed in location 128 to 192 for use in the working registers of the controller.

The EEPROM has a finite life of approximately 400,000 "write" cycles. Care should be used when writing to non-volatile memory to exclude unnecessary write cycles. For example, the Restore command ("^C" from a terminal) will retrieve the parameters from the EEPROM without doing a write. If the Initialize command ("C1") was chosen, the first 256 bytes of EEPROM are written. Should you require a sequence of motions to be done without host attention, you may break-up the motions into sub-groups rather than repeatedly programming the same EEPROM locations. Use the Go from address command to execute the sub-groups in the required sequence.

Use the Save command sparingly. The controller parameters are set so quickly that it is sufficient to just let the host down-load them.

Changing parameters should NOT be done by writing directly to EEPROM locations using the "[" command. The INDEXER will not know that it was changed and may initiate an over-write. Use the commands available to set parameters. Unlike writing, reading is non-taxing on the EEPROM.

The 64 bytes of "Scratch Pad" program memory are configured as fast shadow memory at locations 128 through 192. The data in these locations are down-loaded from the external EEPROM to internal RAM at power up. Instructions executing in this segment run faster than other locations that fetch from relatively slow EEPROM (1Ms. per byte). Programmers should reserve this segment for time critical code or code that changes frequently.

The Scratch Pad RAM area is not actually written to non-volatile memory until the Store ("S") command is issued. Host computers may download subroutines within this area without concern about wearing out the EEPROM.

Locations 256 through 511 are pre-defined if the "G 2048" command is used in an application. The Trip command can jump to any location between 0 and 255.

The nonvolatile memory (NVM) is used to store user programs for future execution via the "G" command or Go input. Any number of programs may co-exist, limited only by the available memory space.

The following memory map defines the address locations, segmented into 8 pages, that are accessible through direct read/write commands:

PAGE	ADDRESS	HEX	STORAGE ALLOCATION TYPE
1	0-255	0-FF	Program, Trip, Go1
	128-192	80-C0	Program, k-Trip, Shadow
2	256-511	100-1FF	Program, Go2
3	512-767	200-2FF	Program
4	768-1023	300-3FF	Program
5	1024-1279	400-FF	Program
6	1280-1535	500-5FF	Program
7	1536-1791	600-6FF	Program
	1600		Program, Power-Up3
8	1792-1893	700-765	Micro-position look-up table
	1894-2047	766-7FF	Operational parameter storage

1. Aux. Input "Go"" (hardware) activated.
2. "Go Sub" Branch on Port Condition.
3. Execute User instruction upon power-up at location 1600.

The values of the following parameters are modified by the command indicated. They are NOT stored in non-volatile memory until a Save command is issued.
Stored default set:

PARAMETER	VALUES AFTER NON-VOLATILE CLEAR (C8, C9 Command)
Initial velocity (I)	400 steps per second
Slew velocity (V)	3004 steps per second
Divide factor (D)	0 (Full Step)
Ramp slope (K)	10/10
Jog speeds (B)	30,200
Trip point (T)	off
Auto power-down	yes
Auto Variable Resolution (H)	1 (Auto variable)
Hold/Run current (Y)	5/25
Limit polarity (I)	low
Auto position readout (Z)	off
Name (after reset)	undefined
User programs	cleared

In the Command mode, commands are normally executed as soon as they are entered. The use of non-volatile memory allows storage of a list of commands. These stored program(s) can be triggered at power-up for automatic or repetitive operations by initiating a "G" command or by strobing the Go input to a logic low. When in the Program mode, the entered commands (now called instructions) are directed into the non-volatile memory. After leaving Program mode, the stored program(s) may be subsequently executed as described above.

The following procedure assumes a standard (RS-422) serial interface using a common terminal:

The Program mode is initiated by entering "P aa" (Carriage Return). The desired start address "aa" is chosen by the user. Generally address location 0 is a good choice for the main program because a program located at that address can be started with a simple "G" command or by strobing the "Go" input low.

Once in the Program mode the current memory location is displayed on the terminal and instructions have the same format as in the Command mode.

Terminating the Program mode is done by entering a "P". This will cause the end of program flag to be inserted and the controller will echo the pound (#) character. The INDEXER will return to the Command mode.

Several programs may co-exist in memory. Each program may be executed independently by issuing a "G aaa" command, where "aaa" is the appropriate address. The length and quantity of programs may occupy the full 2K memory space.

NOTE: The end of program indicator occupies one additional byte. A program sequence that will be "called" when a trip point is passed may be located at an address defined for the trip point.

Existing program(s) may be modified at anytime. The user can review the existing instructions by entering the "q" command. This command produces a list of instructions along with their memory addresses. The "Q" command lists the program instructions one at a time.

To edit an existing program enter "P aaa" where "aaa" is the desired address. Proceed to enter the new instruction(s) as in the Program mode. The edit session may be terminated in two ways. If the edit results in a program that is longer than the previous program or if the user wishes to discard the old instructions (shorten program), enter "P" to terminate edit and cause an end of program marker to be inserted before the end of the old program. If only one or several successive new instructions are to be altered, entering ESCape will terminate the edit. Any instructions outside of the edit area will NOT be altered.

NOTE: If any instructions are of different byte lengths than existed previously, the program could wind up with invalid instructions in the middle of the program. Keeping track of the byte count will avoid this condition. The user may insert redundant or "dummy" one byte instructions to fill the gap. If in doubt re-enter the remaining portion of the program. Use of the "Q" or "q" command to review the stored program is highly recommended.

Command	Function				Type	NV Bytes
	Mnemonic	ASCII	Data 1	Data 2	Example	

PROGRAM COMMAND DEFINITIONS

Command: Keystroke.

Function: Functional description of command.

Type: Immediate: direct execution.
Program: executable in stored program.
Global: all axis present.
Default: initial parameter setting.
Hardware: auxiliary I/O

Non-volatile Bytes: Storage requirements in program.

Mnemonic: Single character prefix used in multi-axis protocol: (prefixed by axis "name" assignment in Party-Line mode).

ASCII: ASCII code in decimal for the command character.

Data 1: Affected parameter(s) and the range of accepted data.

Data 2: (as required).

Example: Example of the command in use.

Command	Function			Type		NV Bytes
ESC	Abort			Immediate, Global		n/a
	Mnemonic	ASCII	Data 1	Data 2	Example	
	Esc	27	none	none	ESC	

Global Abort terminates any active operation and forces the controller to revert to the idle state. Output drivers or ports are not effected. Stepping and position counter update will cease immediately without deceleration. Any program "running" will be terminated. Encoder auto-hunt functions are also aborted. Any axis in the program mode will exit the program mode without creating the "end of program marker," therefore the escape character is useful in editing non-volatile program segments. In single mode a pound (#) sign is returned.

NOTE: Because the deceleration is immediate (without ramping) mechanical overshoot may result, especially with high speeds and/or inertia loads.

Command	Function			Type		NV Bytes
@	Soft Stop			Immediate, Global,Progm		1
	Mnemonic	ASCII	Data 1	Data 2	Example	
	@	64	none,0,1	none	@ <cr>	

The Soft Stop command is useful as a gentle stop. It behaves differently depending on how it is used. If the axis is moving it causes an immediate deceleration to a stop, based on the established deceleration K value.

If the command is issued with no data or a 0 following the command, the Indexer will decelerate to a stop and will not terminate a running program.

If the command is issued with a 1 following the command, the Indexer will decelerate to a stop and any running program(s) will terminate.

NOTE: DO NOT ALLOW THE SOFT STOP INPUT TO BE LOW DURING POWER-UP.

Command	Function		Type	NV Bytes
^C	Software Reset		Immediate, Global	n/a
	Mnemonic	**ASCII**	**Data 1**	**Data 2**	**Example**
	^C	03	none	none	ESC

Software Reset is a global RESET command. All axis ABORT immediately and a reset, equivalent to the power-up condition, is executed:

A. Down load default values from the non-volatile memory.
B. Determine encoder presence.
C. Calibrate power driver to on-phase position.
D. Set origin(s) to zero.
E. Calibrate motor current to hold value.
F. Test for and execute any user power-up program starting at location 1600.
G. Assume an idle state waiting for GO pulse input, Jog input serial command input.

Encoder position maintenance will not be active until a motion function happens.

Command	Function		Type	NV Bytes
A	Port Read/Write		Immediate, Program	2
	Mnemonic	**ASCII**	**Data 1**	**Data 2**	**Example**
	(name)A	65	0-128	none	To Read: A 129<cr> To Write: A 48<cr>

The Port command provides access to the general purpose inputs and outputs. Six (6) general purpose ports are available for user applications. The pre-configured hardware defines 3 inputs (IN 1, 2 & 3) and 3 outputs (Out 1, 2 & 3). All ports are read or written as a parallel 6 bit binary value. To set the output ports, the following table is used:

"A" DATA	OUT 1	OUT 2	OUT 3 (TRIP)
0	low	low	low
8	high	low	low
16	low	high	low
24	high	high	low
32	low	low	high
40	high	low	high
48	low	high	high
56	high	high	high

Other values beside those listed may cause improper operation of the input ports.
Reading of the inputs is accomplished by sending an "A129". A decimal number is returned indicating the state of all of the I/O bits. Convert this number to binary to determine the state of the 6 I/O bits. The bit weighting for the I/O status is shown below.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
n/a	n/a	out3	out2	out1	in3	in2	in1
bit weights		32	16	8	4	2	1

Inputs and outputs are the TTL compatible levels with internal 10K pull-up resistors.

Command	Function		Type	NV Bytes
B	Set Jog Speeds		Default,Immed,Program	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)B	66	Slow Speed 0-255	High Speed 0-255	B 50 250<cr>

Data range 1 and data range 2 represent the speeds to use when the jog inputs are utilized. The first is usually a slower speed. The second number is used when the high speed jog is held low. The values are multiplied by 30 to determine the actual step rate in steps per second. Setting values of 0 will disable the jog inputs. Jog resolution is determined by the microstep mode "H" and "D" values. Ramped acceleration using the "K" value is implemented. Ramped deceleration is used if the high speed input is released while maintaining one of the two jog inputs on (low). Both + and - jog inputs operate at the same speeds.

The Jog inputs are active:

A.. After power-up.
B. When not executing a motion command.
C. When not running a program. (A GO command will take priority over Jog inputs)

Jogging is inhibited during actual command line entry.

NOTE: DO NOT ALLOW ANY JOG INPUTS TO BE LOW DURING POWER-UP.

Command	Function		Type	NV Bytes
C	Clear and Restore		Immediate	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)C	67	Page 0-9	none	C2 <cr>

The controller incorporates a total of 8 pages of program memory. The data values refer to the page of non-volatile memory to clear. Page 8 (1792-2047) contains the parameter default storage, pages 1 thru 7 (0-1791) may be used for user programs.

Clear and Restore re-initializes or clears the system non-volatile memory. All page numbers except a zero cause a 256 byte write cycle, taking approximately 200 Ms Frequent use of this command should be avoided as memory longevity may be effected.

"C 0" Reloads the last saved parameters from non-volatile memory. No write operations occur and operation is similar to what happens during a power-on reset.

"C 1" thru "C 7" Erase the corresponding page of non-volatile memory, 256 bytes at a time.

NOTE: Programming overwrites non-volatile memory as it proceeds. It is not necessary to use this complete erase, except to "clean up" a segment.

"C 8" Forces complete non-volatile initialization to factory default values. This should be done only when new non-volatile memory is installed or existing memory is corrupted.

"C 9" Forces initialization of the "micro look up table." This list of 100 eight bit numbers starts at location 1792 in page 8. Each value corresponds to a micro position with a resolution of 1/256 micro step. The default values assume an ideal linear relationship between phase currents and actual armature positions. Designers may insert their custom tables here. Any new list must be monotonic in nature.

NOTE: The INDEXER/DRIVER must be completely reset (i.e. Power on reset) after executing a C8 or C9 command.

Command	Function		Type		NV Bytes
D	Divide Resolution		Default,Immed,Program		2
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)D	68	Resolution 0-8	none	D 0<cr>

The Divide Resolution factor is used to set the microstep resolution when operating in the fixed resolution mode. Indexes are made using the specified resolution.

RESOLUTION	STEP SIZE	MAX RPM
0	full	6,000
1	1/2	3,000
2	1/4	1,500
3	1/8	750
4	1/16	375
5	1/32	187
6	1/64	93
7	1/128	46
8	1/256	23

This command may be used (with caution) when operating in the Variable Resolution mode. Its effect is to pre-scale the motor shaft speed to operate with higher resolution. The programmer should observe that the maximum velocity specified does not exceed the maximum shaft RPM available at a given resolution.

Command	Function		Type		NV Bytes
E	Setting Time Delay		Default,Immediate		2
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)E	69	0.01 sec (0-255)	none	E 100<cr>

The Controller features automatic motor reduction capability. After an index, the winding current will be reduced or shut off as determined with the "Y" parameter. This specifies the delay in 10's of milliseconds before activation of the auto-power-down feature. Successive indexes will activate the run current (as required) prior to the index. Auto-power-down will not occur if the encoder auto-hunt-mode is on.

Command	Function		Type	NV Bytes
F	Find Home		Immed,Program	3
Mnemonic	ASCII	Data 1	Data 2	Example
(name)F	70	Seek Speed 0-20000	Direction 0 or 1	F 500 1<cr>

The Find Home command eliminates hysteresis inherent with encoders, switches and system mechanical backlash. The Controller implements an intelligent homing algorithm that is designed in such a way that home is always approached from the same direction on receipt of a home command:

A. Initial direction is determined by the state of the home switch and direction parameter.
   If the dir = 0 and Home switch is low, motor starts in the negative direction.
   If the dir = 0 and Home switch is high, motor starts in the positive direction.
   If the dir = 1 and Home switch is low, motor starts in the positive direction.
   If the dir = 1 and Home switch is high, motor starts in the negative direction.

B. The motor steps until the home switch changes state.
C. If home switch was low and changes to a high, homing is finished.
D. If home switch was high and changes to a low, direction is reversed and motor steps until home switch goes high, then stops.

The homing speed is specified in the speed parameter. The direction parameter establishes the initial seek direction to use. If the specified speed is above the initial velocity a start ramp will be used. When the Home switch changes state, motor is stopped without ramp. Then, if needed, the last phase of home proceeds at the initial velocity.

Seek speed is a word value (20-20,000) allowing a range to 20,000 steps per second.

Command	Function		Type	NV Bytes
G	Go		Immed,Program,Hardware	3
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)G	71	Address (0-1791,2048)	Trace 0 or 1	G 1001<cr>

The Go command executes a user programmed sequence starting at a predefined address location. Although most programs will start at "0", the user can start at another address. That address, however, MUST begin at a stored instruction address.

The Trace option is useful in debugging single axis programs. If trace is a one, the Trace mode is turned on. A display of the current step being executed is produced while the program is running. The list format is the same as that of the "Q" command. The Trace mode will be in effect until the program execution terminates or until an embedded Go without the trace attribute is encountered. Address locations between 1894 and 2047 are reserved for parameter storage and may not be used in programs.

The Controller features a special case for the Go instruction. If the address is specified as 2048 (above the last non-volatile address), the control signal will branch to an address based on the state of input ports 1 through 3. The target address starts at the second page of program memory at address 256 with 16 character (byte) intervals. This instruction is analogous to "on PORT go to."

Input port state			Address of go to	The remainder of this page of NVM must contain the following instructions:	
IN1	IN2	IN3			
0	0	0	256	384	G256
1	0	0	272	400	G272
0	1	0	288	416	G288
1	1	0	304	432	G304
0	0	1	320	448	G320
1	0	1	336	464	G336
0	1	1	352	480	G352
1	1	1	368	496	G368

Strobing the hardware Go input to a momentary low pulse will have the same effect as the "G" command.

NOTE: DO NOT ALLOW THE GO INPUT TO BE LOW DURING POWER-UP

Command	Function		Type	NV Bytes
H	Resolution Mode		Immed,Default	2
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)H	72	Fr=0 Vr=1	none	H 1 <cr>

This command selects one of two primary step modes.

mode 0 – Fixed Resolution (FR MODE)
Stepping occurs at a fixed resolution (full to 1/128) determined by the "D" command. The index distance for a given step count is also determined by the specified resolution, i.e. a + 200 step index will result in a 1 revolution at full step, 1/2 revolutions at 1/2 step, 1/32 revolution at 1/32 step, etc. The maximum shaft speed is limited to 20,000 steps/second at a given resolution. Fractional positioning is not available. Ramping is used.

mode 1 – Auto Variable Resolution (VR MODE)
This is the most powerful index capability of the Controller. The Controller determines the optimal step resolution for the specified initial and slew velocities. During acceleration or deceleration, step resolution is coordinated with step rate and changed at predetermined speeds to maintain smooth acceleration. This "gear shifting" always occurs on a motor full step pole position, where torque is highest.

The following events take place in response to an index command. This example assumes an index in excess of 1 full step:
***See "D" Command; The user must send D0 to get max velocity.

A. Any prior motion from a previous command is completed.
B. The next "TARGET" position is calculated in "N" whole steps and a fractional "F" remainder.
C. The motor is energized to the run current, as required.
D. A course index of "N" whole steps is executed.
E. The motor is "Trimmed" to the final position "F" with 1/256 step resolution.

Note: Do not use Variable Resolution with an encoder.

Command	Function		Type	NV Bytes
I	Initial Velocity		Immed,Default,Program	3
	Mnemonic (name)I	ASCII 73	Data 1 Speed 20-20000	Data 2 none	Example I 400 <cr>

The initial velocity command sets the parameters to be used in subsequent motion commands. In VR Mode the speed is in equivalent full steps per second. In FR Mode the speed is in pulses per second applied at the specified resolution. i.e., the shaft speed is halved for each smaller microstep setting (see the ""D" command).

The initial velocity applies to:

A. All index commands (+, -, R)
B. Start speed used in constant velocity (M).
C. Decelerate to 0 in constant velocity or soft stop.
D. Final phase of home routine.

Command	Function		Type	NV Bytes
J,j	Primary and Secondary Jump		Program	4
	Mnemonic (name)J	ASCII 74,106	Data 1 Address 0-1791	Data 2 n+1 times 0-255	Example J 10 5<cr>

Jump to address. count + 1 times. This loop command allows repetition of a sequence up to 255 times. The address specified MUST be a valid instruction address and may be used only within a program. Only one jump counter is available for use at any given time. However, the Primary "J" command and Secondary "j" command can be nested. Use of the "G" command is recommended for single jump commands.

Note that if a jump loop does not run to completion (due to branching out of the loop) the jump counter will have the remaining counts left to complete the jump loop that was not completed. Use a dummy jump loop to run the count to zero if a "fresh" jump loop counter is needed.

Command	Function		Type	NV Bytes
K	Ramp Slope		Immed,Default,Program	3
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)K	75	Acceleration 0-255	Deceleration 0-255	K 20 40<cr>

The ramp slope is used to adjust the dv/dt during acceleration and deceleration. The profile or shape of the acceleration/deceleration curve is defined by an internal look-up table. Depending on the initial and final velocities, 0 to 500 discrete velocities may be required to accelerate or decelerate. The K value determines how many steps are made at each step rate point along the acceleration curve during ramping. High values increase the time of the acceleration ramp. Lower values decrease the time of the ramp. A value of 0 eliminates any ramping.

In practical applications, it is easier to decelerate a system rather than accelerate a system. The separate acceleration and deceleration ramp values allow for greater flexibility over systems with fixed acceleration and deceleration ramps.

Command	Function		Type	NV Bytes
L	Loop on port		Immed,Program	4
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)L	76	Address 0-1791,2048	Port Condition	L 20 1<cr>

The Loop command will test the specified input port for the required condition. If the port is NOT at the required level, the program will jump to the specified address. If the address is to a previous instruction, the program will loop until it becomes the specified level. The program will then continue to the next step. Input ports are tested as follows:

CONDITION	WAIT FOR
0	IN1 High
1	IN1 Low
2	IN2 High
3	IN2 Low
4	IN3 High
5	IN3 Low

The Controller has an additional feature for implementing a "wait until" function. The standard loop tests the condition every 2-3 mS. If the unique address is 2048, the Controller executes a tight loop at this instruction while monitoring the specified condition. When the condition is met, program execution continues. This feature is helpful in situations where the condition may be of short duration. This command is usable only in non-volatile program execution.

NOTE: Reference the special case of the Go command for branching on inputs.

Command	Function	Move at fixed velocity	Type	Immed,Program	NV Bytes	3
M	Mnemonic	ASCII	Data 1	Data 2	Example	
	(name)M	77	Dir/ Speed 0,±20-20000	none	M+2000<cr>	

The "+" or "-" sign determines the direction during the move at constant velocity function. The motor will ramp up or down to a constant step velocity and motion will continue at the given speed until a new velocity is entered. The specified slew speed is in steps per second.
Ramp parameters may be modified prior to each velocity command, allowing different ramp slopes. The direction is specified by the sign preceding the velocity. The controller has the capability of decelerating from full speed in one direction to full acceleration speed in the opposite direction with this single command. Motion may be terminated by:

A. The "M 0" command
B. Soft stop "@" command or interrupt.
C. ABORT (ESC) interrupt (without deceleration).

The default initial velocity is used at the first invocation of the command. The following commands modify effective speeds and resolutions:

D. Resolution
K. Ramp factor
H. Mode

Position Trip points may be used. If the encoder feedback option is implemented, stall supervision is available.

Command	Function	Set Origin	Type	Immediate,Program	NV Bytes	1
O	Mnemonic	ASCII	Data 1	Data 2	Example	
	(name)O	79	none	none	O<cr>	

The Set origin command resets the internal position counter to Zero. Units with encoder feedback are also reset. This command is not allowed while moving. The following commands will also reset the origin:

A. Power on reset
B. (e) enabling the encoder.

The micro trim position is re-calibrated to the "on pole" position as required (Non-Encoder mode).

Command	Function		Type	NV Bytes
P	Program mode		Immediate	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)P	80	Address 0-1791	none	P0<cr>

Program Mode allows entering commands for future execution by use of the "G" command or external go signal. Existing programs are overwritten as new instructions are stored. Entering a second "P" command will terminate the Program mode, and insert an end of program marker in the stored program before returning to the Immediate Command mode. While in Program mode, commands and data are directed into the non-volatile memory. The address specifies the start point in non-volatile memory where the application program will reside. As instructions are entered, the address counter is updated and displayed. Any number of independent program segments can co-exist. These can be accessed via Jump, Loop, Go or other special instructions.

Special locations:
"Shadow Memory": 64 bytes of the program are configured as FAST memory. Locations 128 through 192 are down-loaded from the external EEPROM to internal RAM at power-up. Instructions executing in this segment run faster than other locations that fetch from relatively slow EEPROM (1 Ms. per byte). Programmers should reserve this segment for time critical code. The shadow RAM is not actually written to non-volatile memory until the Store command is issued. Host computers may down load subroutines without concern about wearing out the EEPROM.

Locations 256 through 511 are pre-defined if the "G 2048" command is used in an application. The Trip command can jump to any locations between 0 and 255.

Editing of programs should be done WITH CAUTION as follows:

A. Start programming at desired address.
B. Enter new instructions.
C. Terminate programming with the ESC command. This will cause a return to the Command mode without inserting the end of program marker.

NOTE: Do not attempt to use locations above 1792 for programs (default storage).

Command	Function		Type		NV Bytes
Q	Query Stored Program		Immediate		n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)Q	81	Address 0-1791	none	Q 100<cr>

This command will produce a list (disassembled) of the instructions stored in non-volatile memory using the format:

ADDRESS INSTRUCTION DATA1 DATA2

A. The values will be displayed only if applicable to the particular instruction type.
B. One instruction will be listed at a time.
C. The space key will advance the address and list more instructions.
D. Listing is terminated when an "end of program" marker is found or the ESC character is received.

A lower case "q" command may be used to produce a listing of up to twenty consecutive instructions.

Note: This command does not function in party line mode.

Command	Function		Type		NV Bytes
R	Relative Index		Immediate,Program		n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)R	82	Position±8,388,607.9	none	R-200.25<cr>

Move with ramping relative to the origin. The distance of the index is determined by subtracting the last (or current) position from the target position. If an index is under way when the command is issued, the controller will wait until that index is complete. The units used for target position can assume three meanings based on the mode of operation:

No-encoder, Variable Resolution mode Range +/- 8,388,607.99 Full Steps
The target position is specified in terms of whole motor steps (typically 200 steps per revolution) and a fractional portion with .01 step resolution. Indexing will proceed at various resolutions, based on velocities, until the target "whole step" position is reached. A fine adjustment is then made to position the rotor to .01 to .99 step.

Non-encoder, Fixed Resolution mode Range +/-8,388,607 Steps
The target position is specified in terms of steps at the specified resolution (see the "D" command). This mode permits step translations of 1/200 thru 1/51,200 motor steps per revolution. Shaft speed is limited to 20,000 steps per second at a given resolution.

Encoder, Fixed Resolution mode Range +/- 8,388,607 Encoder position
The target position is specified in terms of encoder position. The shaft resolution is 4 times the encoder line count. Indexing will proceed at various resolutions, based on velocities until the target is reached. A fine adjustment is then made to position the encoder at the specified target. The resolution used for this phase is up to 1/256 motor/step. Automatic stall detect (with "n" retries) may be enabled. Automatic position maintenance after index may also be enabled.
Related commands D, E, H, I, K, O, V.

Command	Function		Type	NV Bytes
S	Store Parameters		Immediate	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)S	83	none	none	S<cr>

Store Parameters to non-volatile memory. Many parameters are automatically read from non-volatile memory each time power-up occurs, eliminating the necessity of inserting redundant initializing commands in programs. The following are user defaults that are stored in the non-volatile memory:

COMMAND	INITIALIZING VALUES (RESET VIA THE 'C 8' COMMAND ONLY)
B Jog speeds	3/20 (30/200 steps per second)
D Divide	0 (full step)
E Settling delay	50 (500 Ms.)
H Resolution mode	1 (Variable)
I Initial velocity	400
K Ramp slopes	10/10
T Trip point	Off
V Slew speed	3004
Y Currents	5/25
Encoder option	
d Dead zone	30
e Encoder setup	off
h Hunt resolution	4
r Stall retries	5
s Lag factor	20
t Stall delta	5
v Hunt speed	700
Assigned Name	

The Store command also saves the contents of the high speed "Shadow RAM." Locations between 128 and 192 are stored only when this command is issued, extending the life of the EEPROM. These 64 locations are automatically downloaded on power-up reset.

OBSERVE THE PRECAUTIONS OF WRITING TOO FREQUENTLY TO NON-VOLATILE MEMORY.

Command	Function		Type		NV Bytes
T	Trip point		Default,Program		5
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)T	84	Position ±8,388,607	Address 0-255	T 233 100<cr>

The controller has a programmable "Trip Point" feature. During motion operations, the position counter is continuously updated. If the trip point function is enabled, the position is continuously compared to the programmed trip position. When an equality is detected, a trip event will be triggered. If a program is running, a call or "Go Sub" will be made to the specified address between 1 and 255.
Programs located at the specified address can perform almost any function, including turning on/off ports and setting new trip points. A trip point cannot be "re-entered" i.e., when executing a trip subroutine and a new trip is set as part of the subroutine, the new trip cannot be triggered until the end of the first trip subroutine. Routines located between 128 and 192 will execute faster because of the "Shadow RAM" feature. Trip service routines should not contain index, wait or time consuming instructions. The P command is used at the end of the trip subroutine in program memory to cause a return to the point where the trip occured in the main program.

DISABLE
To turn off the trip function use 0 (zero) as the address parameter.

NOTE: This command is only functional in fixed resolution (non encoder) mode.

Command	Function		Type		NV Bytes
V	Slew Velocity		Immediate,Program,Default		3
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)T	86	Speed 20-20000	none	V 1000<cr>

Set final Slew Velocity after ramping up to speed.

The following commands use this parameter.

R Relative index
\+ Plus index
\- Minus index

Note: For entered values of slew velocity, the indexer calculates the actual velocity used and displays this number in () in the X command output. This actual velocity may be different than the entered value since the indexer uses an internal counter to generate the clock pulses and the actual divide ratio for this counter may not correspond to the exact user specified velocity. In all cases, the indexer will always calculate and use the next highest value for velocity if it can't achieve the exact user entered value.
Example: Type V3000. The X command displays 3004 indicating 3004 sps will be used for slew velocity.

Command	Function		Type	NV Bytes
W	Wait Time		Immediate,Program	3
	Mnemonic (name)W	ASCII 87	Data 1 Time in .01sec 0-255	Data 2 none	Example W 100<cr>

Wait n x 10 milliseconds, where n is the Data 1 field.

The controller will remain in an idle state for the specified time. This time is derived from the crystal clock and is very repeatable and accurate. It is independent of motion in progress and times functions while moving. One timer is available.

NOTE: SPECIAL CASE--wait for motion completion:

Using a 0 (Zero) time value will delay the next command/instruction until any index (+,- or R command) motion has been completed or a Limit switch input in the direction of the move is activated. In the Immediate Command mode the echoed line feed is delayed providing an alternative means for the host to determine motion complete (only practical in single axis designs). The W0 must follow the motion command to be waited on and precede the next command to be executed following the motion command. The W0 must be used for every instance when this action is desired.

Command	Function		Type	NV Bytes
X	Examine Parameters		Immediate Single Mode	n/a
	Mnemonic (name)X	ASCII 88	Data 1 none	Data 2 none	Example X<cr>

The Examine command displays settings of many parameters and provides useful information. Two lines of operational parameter data are displayed. Up to five lines of data are displayed if the Encoder and Auto position functions are enabled. In Single mode the display is as follows. In Party Line Mode the X command returns "32. Refer to the Appendix for information on obtaining parameters using the Read NVM command to view parameters in Party Line Mode.

Line #1: Y=hold/run,E=n, K=up/dn, H=vr(fr), na=name, {T=ppp/aaa}
Where: run=run current, hold=hold current, n=10x milliseconds
up=acceleration slope, dn=deceleration slope vr=Variable Resolution mode,
fr=Fixed Resolution mode name=most recent name assignment for party line
ppp=trip position (if enabled)aaa=trip branch address

Line #2 I=iv (av/res), V=sv (av/res), (rl=nnn)
Where: iv=initial velocity, sv=slew velocity av=actual velocity used, res=actual resolution used,
rl=ramp length (multiply times K for actual)
NOTE: The divide factor (D) can be determined from the res. values

Line #3 (if encoder is enabled) e=ll, (ratio=rr)
Where: ll=lines specified, rr=computed ratio (lines/full-step)

Line #4 (if auto position is enabled) d = dz, v=vh/hres
Where: dz=dead zone size, vh=hunt steps per second, hres=hunt resolution

Line #5 (if auto stall is enabled) s=ss, t=tt, r=rr, (lag=lll)
Where: ss=stall percentage, tt=tt=test delta, rr=retry attempts lll=maximum allowable encoder lag distance (based on s and t values)

Items displayed in parentheses () are computed internally, based on parameter settings.

Command	Function		Type	NV Bytes
Y	Hold/Run Current		Immediate,Default	n/a
	Mnemonic (name)Y	ASCII 89	Data 1: Hold current 0-100 (%)	Data 2: Run Current 0-100 (%)
	Example: Y 1 75<cr>			

This command allows specifying the hold and run values of motor current (per phase) between 0 and 100% with a resolution of 1%.
The switching between Hold and Run values is automatic whenever a motion function is executed. Current reduction to the "hold" value is automatic and occurs when stationary. A programmable settling time is inserted after each index (see "E" command)
NOTE: In units that output STEP CLOCK and DIRECTION, false pulses will occur when switching to a non-zero hold current value. This "phenomenon" can be eliminated by using either a value of 0 for the hold current or setting the hold and run current to the same value.

Command	Function		Type	NV Bytes
Z	Read Position(non encoder)		Immediate	n/a
	Mnemonic (name)Z	ASCII 90	Data 1: Readout mode 0 or 1	Data 2: none
	Example: Z 1<cr>			

Read and display the current position. During motor move commands the value will change depending on the direction of travel

The "Z1" enables continuous readout via the serial interface. Any change in position causes the position data to be sent to the serial output. The readout is terminated by a Carriage Return only.
The Readout mode will be defaulted "on" if a Save command is issued. This mode is only practical using single axis protocol. The decimal portion will be non-zero only when fractional positioning is enabled (non-encoder, Variable Resolution mode).

NOTE: When an Encoder is being used, the lowercase z will display encoder position information.

NOTE: Issuing a Z1 in Party Line Mode may produce undesired results.

Command	Function		Type	NV Bytes
[	Read NVM Address		Immediate	n/a
	Mnemonic (name)[	ASCII 91	Data 1: Address 0-2048	Data 2: count 0-255
	Example: [120<cr>			

The Read Non-Volatile Memory command allows the user to display any byte of the 2047 byte external non-volatile memory. The address specifies the desired location to access.

The optional count allows display up to 255 locations, 10 bytes per line.

For listing memory as program instructions, refer to the q and Q commands.

Command	Function	Read Limits/Hardware	Type Immediate		NV Bytes n/a
]	Mnemonic (name)]	ASCII 93	Data 1 limits=0 hardware=1	Data 2 none	Example] 0<cr>

"] 0" QUERY LIMIT STATUS
A Zero is returned if no limits are on.
1=Limit A, 2=limit B, 3=Both limits

"] 1" QUERY HARDWARE
Permits the user to examine the status of the various signal inputs. The result will contain the sum of the decimal values as follows:

Decimal value
1. Home input
2. Encoder error direction (internal)
4. Encoder error (internal)
8. Go input (not usually readable when running)
16. Soft stop input
32. Jog + input
64. Jog - input
128 Jog speed input

Command	Function	+/- Index	Type Immediate,Program		NV Bytes 5
+ −	Mnemonic (name)+,-	ASCII 43,45	Data 1 Steps 0.01-8,388,607.99	Data 2 none	Example -200.5<cr> or +200.5<cr>

Step in a positive or negative direction for the specified distance. The motor will ramp up, slew, then ramp down per the previously set parameters. If an index is under way when the command is issued, the controller will wait until the index is complete. The unit used for target position can assume three meanings based on the mode of operation:

Non-encoder, Variable Resolution mode Range +/- 8,388,607.99 Full Steps
The target position is specified in terms of whole motor steps (typically 200 steps per revolution) and a fractional portion with .01 step resolution. Indexing will proceed at various resolutions, based on velocities until the target "whole step" position is reached. A fine adjustment is then made to position the rotor to .01 to .99 step.

Non-encoder, Fixed Resolution Range +/- 8,388,607 Steps
The target position is specified in terms of steps at the specified resolution (see the "D" command). This mode permits step translations of 1/200 through 1/51,200 motor steps per revolution. Shaft speed is limited to 20,000 steps per second (at a given resolution).

Encoder, Fixed Resolution mode Range +/- 8,388,607 Encoder Position
The target position is specified in terms of encoder position. The shaft resolution is typically 4 times the encoder line count. Indexing will proceed at various resolutions, based on velocities until the target is reached. A fine adjustment is then made to position the encoder at the specified target. The resolution used for this phase is up to 1/256 motor/step. Automatic stall detect (with "n" retries) may be enabled. Automatic position maintenance after index may be enabled.

Related commands D, E, H, I, K, O, V.

Command	Function		Type		NV Bytes
^	Read Moving Status		Immediate		n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)^	94	none	none	^<cr>

The Read Moving Status command is used to determine the current moving and mode status of the controller. These status bits are converted to a decimal number (0-255).

The status byte result contains the decimal sum of status bits as follows:

Bit	Decimal	
0	1	moving - indicates axis moving
1	2	constant-high in constant velocity
3	8	homing-homing routine is active
4	16	hunting-encoder correction
5	32	ramping up or down.

Other bits in this result should be ignored.

EX. If the returned value is a 9 this would indicate moving (1) + homing (8).

Command	Function		Type		NV Bytes
\	NVM Direct Write		Immediate		n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)\	92	address 2048	data 0-255	\ 100 20<cr>

The Write to Non-Volatile Memory command allows the programmer to directly modify any byte in the memory. The life expectancy of the non-volatile memory may be effected by excessive use of this command.

NOTE: Non-volatile memory has a finite life of approximately 10 years for data retention and 460,000 write cycles.

Command	Function		Type	NV Bytes
d	Deadband Enable		Immediate,Default	3
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)d	100	steps 0-255	none	d 10<cr>

This command specifies the differential position distance, in encoder counts, that the motor shaft is permitted to differ from the encoder position register before automatic position correction is executed.

When a correction is required the position is re-homed to the desired position. The total dead zone value is double the specified distance, thus a value of 10 will maintain the position within +/- 10 encoder steps. After completing a move using automatic position correction, further position corrections will be automatic. Full motor power is maintained, and the moving output signal is asserted "on". The minimum practical value is affected by encoder resolution, backlash, and hunt step rate/resolutions. A value that is too low will result in constant hunting or shaft oscillation.

A value of 0 disables this function. A non-zero value activates position maintenance immediately. An abort command (ESC) will halt tracking.

Command	Function		Type	NV Bytes
e	Encoder Resolution		Immediate,Default	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)e	101	Line Count 0,50-2000	none	e 500<cr>

This command specifies the encoder resolution, in lines, for one revolution of the motor. The controller multiplies this value times 4 for encoder position readings. A 500 line encoder will produce 2000 steps per motor revolution. The line count values are based on a standard 200 step/rev motor, resulting in allowable settings between 50 and 2000 in increments of 50 (i.e. 100, 150, 200, etc...). Non standard applications must be scaled appropriately.

A value of 0 disables all encoder functions, and subsequent indexing is in motor steps rather than encoder steps.

NOTE: Always execute this command last in the sequence when configuring the encoder settings since this command immediately enables encoder actions.

NOTE: Be sure the HOLD current setting (Y command) is not set to 0 when using position maintenance.

Command	Function		Type	NV Bytes
f	Find Encoder Index Mark		Immediate,Program	2
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)f	102	direction 0 or 1	none	f 1<cr>

This command causes the motor to rotate in the specified direction until the optional encoder marker output (option on encoder) is detected. Searching is accomplished using the hunt speed ("v") and resolution ("h"). To avoid missing the index signal, the microstep resolution should be high. Values between 1/32 and 1/256 are recommended. The mark is only usable in the fixed resolution mode. The motor pole to mark relationship requirements preclude practical use in variable resolution mode.

Command	Function		Type	NV Bytes
h	Hunt Resolution		Immediate,Default	2
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)h	104	microstep res. 0-8	none	h 8<cr>

The Hunt Resolution defines the motor micro step resolution as defined by the H command, used during deadband recalibration operations. A value of 0 corresponds to full step and 8 equals 1/256 step. See the "H" command.

For moderate to high resolution applications, resolutions of 1/16 and above are recommended.

Very high resolutions will tend to increase the time required for large position error corrections.
See:

"v" Hunt Velocity
"d" Deadband Enable

Command	Function		Type	NV Bytes
k	Special Trip		Immediate,Program	5
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)k	107	position ±8,388,607.99	Port value 0–56	k 300.6 56<cr>

The SPECIAL TRIP command is a REAL TIME TRIP sequence, stored in "Shadow" RAM. It permits programming port output sequences to occur based on position. This powerful command can be used to generate port sequences in real time, while moving. For example, "turn on and off 3 valves at various positions during a cycle".

Description of operation

A. A trip position is set using the T command, address field points to k command in RAM.
B. When the trip position match occurs:
   a. Program vectors to trip address and the port data is output (see the "A" command).
   b. The trip position register is loaded with the NEXT trip position as specified in the k command.
   c. The trip vector address is automatically advanced by 5 to point to the next k command.
C. A new trip position is now set (k command just executed) and the process repeats from step B.

Rules:

1. The k instructions must exist in high speed RAM (BETWEEN 128 AND 192).
2. A trip point must be active with a vector address pointing to the k commands in RAM.
3. Sequences must occupy contiguous RAM address space.
4. To disable the last k vector address, use a T 0 0 after sequence executes.

The use of this powerful command is best understood by way of example. Two valves V1 on port #4, & V2 on port #5 need to be sequenced as follows:

```
   POSITION      VALVES      PORT VALUE (DATA 2 of A command)
start  0       V1 off, V2 off        0
       100     V1 on                 8
       200     V1 on, V2 on         24
       300     V2 off, V1 on         5
       400     V2 off, V1 off        0
```

a. Enter the "STATES":

```
        P 130          'start programing in shadow ram area
   130  k 200 8        'V1 on, next trip set to position 200
   135  k 300 24       'both on, next trip set to position 300
   140  k 400 5        'V2 off, next trip set to position 400
   145  k 500 0        'both off (position 500 is set but might never be hit)
   150  P              'exit program mode
```

b. The commands now exist in RAM. Issue a Save "S" command to store in NV memory.

c. Load the main program:

```
        P 0
         0  0          'set origin to 0
         1  T 100 130  'set trip position and vector address
         6  A 0        'set outputs off
         8  R+1000     'start move
        13  W0         'wait for move to complete
        16  T 0 0      'clear trip at 500 set by last k command
```

Command	Function		Type	NV Bytes
l	Limit Polarity		Default	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)l	108	0,1,2,3	none	l 1<cr>

The lower case L controls the polarity of the limit switch inputs as recognized by the controller. The data value determines how the limit switch input is acted upon when closed.

0 - Normal (Active Low) limit inputs
1 - Invert (Active High) limit inputs
2 - Enable Soft Stop on limit input low
3 - Enable Soft Stop on limit input high

NOTE: Limit A is the limit for moves in the "+" direction.
Limit B is the limit for moves in the "-" direction.

Command	Function		Type	NV Bytes
o	Set Origin		Default	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)o	111	none	none	o <cr>

This command sets the position counter to 0. This command is identical to the upper case version.

Command	Function		Type	NV Bytes
q	Query Program as List		Immediate	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)q	113	address 0-1791	none	q <cr>

List application programs from non-volatile address, 20 lines at a time. See the "Q" Command.

Command	Function		Type	NV Bytes
r	Stall Retry Count		Default,Immediate,Program	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)r	114	number of retries 0-255	none	r 5 <cr>

Set Stall Retry Count will automatically re-attempt to execute a new index if, during the course of an index, a stall condition is detected.

A. Stop motion.
B. Read encoder position.
C. Automatically execute a new index based on the new computation.

Upon exhaustion of the retry count the controller will still attempt to acquire the desired position if the Hunt (Deadband) feature is enabled. Limit inputs or Abort commands will terminate retries.

Command	Function		Type	NV Bytes
s	Stall Factor		Immediate,Default	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)s	115	stall factor 0-255	none	s 100<cr>

This command sets the encoder counts lag distance as a percentage of the sample rate (t). A value of 255 represents 100%. The formula for determining the number of encoder lag steps is:

Lag Distance = Stall Test Value(t) x Encoder Steps per Motor Step x Stall Factor(s) / 255

If s = 224 the maximum lag is: 10 x 10 x 224/255 = 87.8 encoder steps. (see t command below)

Every 10 motor steps the encoder position is sampled. If the motor distance has not moved at least 87 encoder steps a stall condition is set and the controller takes corrective action.

Command	Function		Type	NV Bytes
t	Stall Test Delta		Immediate,Default	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)t	116	steps 0-255	none	t 10<cr>

This defines the sample rate or distance increment (in full motor steps) between stall tests.

By example:

A 200 step/rev motor is coupled to a 500 line encoder, which yields 2000 encoder steps per revolution, or 10 encoder steps per motor step (2000/200).
The sample distance (t) is 10 full steps.
The encoder change each "t" step is 10 x 10 = 100 encoder steps.
The actual sampling should allow for some backlash as defined by the stall factor.

Command	Function		Type	NV Bytes
v	Hunt Velocity		Immediate,Default	n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)v	118	speed 20-8000	none	v 500<cr>

Hunt Velocity specifies the hunt step rate to be used during deadband repositioning. Motion is at a fixed resolution, specified by the "h" parameter.

NOTE: The controller may get "stuck" in a hunt mode if v is set too high or too low and ESC will not work to abort the hunt function. Turning the motor shaft may allow the position to be found and the hunt function to terminate If the controller does get "stuck", cycle power and change the v parameter before attempting to move again.

Command	Function		Type		NV Bytes
y	Output CR LF		Immediate,Program		1
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)y	121	none	none	y<cr>

This command causes a carriage return line feed to be sent to the terminal. This command is useful when displaying position as part of a program. Without this command following the Z to display position, the position reports would be send on the same line, causing the display to wrap around at the end of a display line. When the y command follows a Z command, the next position report would appear on the next display line, providing a format which is easier to read.

Command	Function		Type		NV Bytes
z	Read Encoder Position		Immediate		n/a
	Mnemonic	ASCII	Data 1	Data 2	Example
	(name)z	122	Readout mode 0 or 1	none	z 1<cr>

Read and display the current encoder position. During move commands the value will change depending on the direction of travel and the position of the encoder.

The "z1" enables continuous readout via the serial interface. The readout is terminated by a Carriage Return only.

The Readout mode will be defaulted "on" if a Save command is issued. This mode is only practical using single axis protocol.

NOTE: The use of the uppercase Z command to display position when using an encoder may produce unpredictable results in the position information.

The following example illustrates the method of generating, editing and executing programs through the serial port, in Single mode operation. (CR) indicates the Carriage Return character (hex 0A).

ENTER	REMARK
P 0 (CR)	Place in Program mode. insert instructions at location 0.

Start inserting instructions. The address is displayed during entry.

ADDRESS	INSTRUCTION	REMARK
0	F1000 0(CR)	Find home at 1,000 SPS in the "+" direction
3	W 0(CR)	Wait until move completes
6	0 (CR)	Set origin to zero
7	R+10000 (CR)	Index "+" 10,000 steps relative to origin.
12	W 0(CR)	Wait until complete
15	R-10000 (CR)	Index "-" 10,000 steps relative to origin
20	W 0(CR)	Wait until complete
23	J 7 3 (CR)	Jump to address 7 (4) times
27	R+1000 (CR)	Index "+" 1,000 steps relative to origin
32	W 0(CR)	Wait until complete
35	R-1000 (CR)	Index "-" 1,000 steps relative to origin
40	W 0(CR)	Wait until complete
43	G 7 (CR)	Go to address 7 (endless loop)
46	P(CR)	End of program flag

The program now resides in the non-volatile memory.

The indexer responds with a "#".

Enter: "q" 0 (Carriage Return) or "q" (Carriage Return)

The terminal will display the instructions and addresses as follows (Note that the P at the end is not displayed):

```
0    F    1000 0
3    W
6    0    0
7    R    10000
12   W
15   R    -10000
20   W
23   J    7 3
27   R    1000
32   W
35   R    -1000
40   W
43   G    7
46
```

Starting at location 0; Enter: "G" 0 1 (Carriage Return) Note: The trace is turned on.

The indexer will start executing the preprogrammed instructions starting at location 0 and loop forever. Because the Trace option is in effect the display will list each instruction prior to execution. The user may terminate program execution at any time by entering ESCape.

Alter one or more instructions:

Example: It is desired to change the instruction #30 to 5000 steps. as follows:

Enter:
P 30 (Carriage Return)
+5000 (Carriage Return)
"ESC"
The edit is complete.

Enter:
"Q" 0 (Carriage Return)
To list each subsequent program line enter a Carriage Return:

```
0    F    1000 0
3    W
6    O    0
7    R    10000
12   W
15   R    -10000
20   W
23   J    7 3
30   R    5000
35   W
38   R    -1000
43   W
46   G    7
49   P
```

The indexer has the ability to control the motor operational "Run" and "Hold" drive currents. The following procedure is used to access the independently programmable "Run" and "Hold" current feature. Values from 0 to 100% may be entered; Ref. "Y" command.

1. Issue the "Y" command to program the desired current values. Entering Y 10 80 yields a 10% Hold current and 80% Run current.
2. Issue an "S" Save command. The values and mode are stored to non-volatile memory.

On receipt of an index or other motion command, the control circuits are incremented to the 80% boost, while moving is in process. On completion of motion (and a settling delay time) the current is automatically reduced to the 10% Hold current level.

Accessing the Current Disable feature will require using the following current set-up procedure:

To establish a "Current Disable" and Running current of 80% of maximum:

TYPE	RESULT
Y0 80	sets hold at 0% (disabled) and run at 80%.

NOTE: If the number "0" is entered as the first data value during current control programming, the output chopper driver is disabled during "Hold" modes of operation.

The indexer/driver modules are extremely flexible in the manner in which they may be utilized. This is especially true for multiple axis applications. This section will outline a basic trouble shooting process. If you are still experiencing difficulties after following these procedures, please call IMS 8:30 AM to 5:00 PM EST for applications assistance.

NOTE: Unauthorized service of your module may void warranty.

Most common causes of operational problems arise from improper configuration/wiring of connections to the modules. The following notes should help to isolate these problems.

NOTE: BE SURE THE FOLLOWING HARWARE INPUTS ARE HIGH DURING POWER-UP:
SOFTSTOP, JOG +, JOG -, GO, JOG SPEED

EQUIPMENT

1. A general purpose VOM or DVM, preferably including a 10 Amp current scale (0 to 2 Amps will also work).
2. A suitable short, small gauge wire with small jumper clips or female .025" contacts. (test jumper).

NOTE: DO NOT REMOVE, OR CAUSE MOTOR LEADS TO BECOME DISCONNECTED WHILE POWER IS ON.

OPTIONAL EQUIPMENT

3. An Oscilloscope may be useful for observing signal wave forms.
4. A Logic Probe gives a quick indication of logic states at a glance.

PROCEDURE

1. As a basic precaution, disconnect all mechanical coupling and electrical connections to the stepper motor (s). These tests will eventually require an electrical load. however until indexer operation is achieved it is best to inhibit mechanical movement.
2. Verify proper DC power to the module by measuring the DC voltage output on the DC power supply.
3. Verify motor winding continuity and phase polarity.
4. Attach the current meter in series with one of the Phase B motor connections.
5. Plug in the DC power supply.

The following test will verify basic motor operation:

6. With power and a motor/ammeter connected to the module the motor should exhibit "Holding Torque." The current value observed will correspond to the last Saved or Default value of holding current.
7. Place a test jumper connecting Logic Ground and Jog +. The motor should begin to rotate at the (default) rate of 70 steps per second.
8. Remove the connection to Jog + and place the jumper Jog -. The motor should begin to rotate in the opposite direction.

The above tests have verified basic controller, driver and motor operation. Assuming that the initial trouble shooting tests 1-8 passed successfully the next step in the debug process is to verify system communications. The following procedure will test the serial communication.

Primary communication to the module is through the RS-422 serial port. A PC/AT compatible personal computer or a dumb terminal is required.

DUME TERMINAL TESTING:

9. Attach a terminal to the module.
10. Set the communication settings to 9600 baud, 1 stop, no parity, 8 data bits.
11. With power applied to the indexer/driver module, press the Space Bar.
    The module should return with a xxxx yyyy AMS MAX-2000 vX.XX Response.
    xxxx = Indexer version number
    yyyy = Encoder (optional) version number
    X.XX = Operational code revision level.

CONDITION: NO RESPONSE

A. If there is no response, check the wire continuity and pin assignments from the terminal and confirm the terminal settings.
B. Use an oscilloscope to observe transmitted data output from pin 2 of the (25 pin) RS-232 terminal output, while repeatedly pressing the Space Bar on the keyboard.
C. Seek an alternate terminal for communication if there is no output observed as a result of step B.

CONDITION: RESPONSE

D. Type X (Examine Command) followed by a Carriage Return.

The following parameters should be echoed back:

E. If the response does not match these default parameters Type "C 8" with a Carriage Return. This is an initialization procedure. Type a ^C followed by a space, followed by an X followed by a Carriage Return and observe the following:

xxxx yyyy AMS MAX-2000 vX.XX

X Y = 20/75 E = 100 K = 10/10 H = vr na =
I = 400 (12000/32) V = 3004 (12047/4) (rl = 15)

CONDITION: RESPONSE APPEARS DISTORTED

G. Type X and a Carriage Return and observe the parameters indicated in procedure D. If the echoed parameters do not match, appear garbled, broken, spaced or partially missing check the following:

1. Verify correct terminal settings.
2. Check the routing of the serial communications cable. Verify that the cables are not bundled with motor connection cables or in close proximity to other sources of high speed switching currents or transient voltage inducing devices (Solenoids, Relays, etc.).

If you are still experiencing difficulties after following the above procedure, please call and ask for one of our applications engineers.

PLEASE READ THE FOLLOWING NOTES BEFORE USING YOUR INDEXER/DRIVER.

1) DO NOT ALLOW THE MOTOR LEADS OR POWER LEADS TO BECOME LOOSE OR DISCONNECTED WHILE POWER IS APPLIED TO THE UNIT.

2) DO NOT ALLOW THE FOLLOWING INPUTS TO BE LOW DURING POWER-UP:
   SOFTSTOP
   GO
   JOG +
   JOG -

3) IN PARTY LINE MODE, THE LF CHARACTER IS THE LINE TERMINATOR. IN SINGLE MODE, THE CR IS THE LINE TERMINATOR. IN EITHER MODE, THE INDEXER WILL ECHO ALL CHARACTERS UP TO AND INCLUDING THE LINE TERMINATOR.

4) IN VERSION 1.09 OF THE INDEXER SETTING THE CURRENT TO "0" WILL CAUSE UNPREDICTABLE OPERATION WHEN THE NEXT INDEX IS MADE. USE A SETTING OF "1" INSTEAD. THIS PROBLEM IS RESOLVED IN LATER VERSIONS OF THE INDEXER.

COMMAND SUMMARY

	COMMAND	DATA 1	RANGE	DATA 2	RANGE
SET-UP COMMANDS					
B	SET JOG SPEEDS	SLOW SPEEDx30	0-255	HIGH SPEEDx30	0-255
C	CLEAR AND RESTORE	NVM PAGE	0-9	NONE	NONE
D	MICROSTEP RESOLUTION	RESOLUTION	0-8	NONE	NONE
E	SETTLING TIME DELAY	0.01 SEC	0-255	NONE	NONE
H	RESOLUTION MODE	FR/VR	0-1	NONE	NONE
I	INITIAL VELOCITY	STEPS/SEC	20-20,000	NONE	NONE
K	RAMP SLOPE	ACCEL	0-255	DECEL	0-255
S	STORE PARAMETERS	NONE	NONE	NONE	NONE
T	TRIP POINT	POSITION	±8,388,607.99	VECTOR ADDRESS	0-255
V	SLEW VELOCITY	STEPS/SEC	20-20,000	NONE	NONE
Y	HOLD/RUN CURRENT %	HOLD %	0-100	RUN %	0-100
MOTION COMMANDS					
F	FIND HOME	STEPS/SEC	20-20,000	DIRECTION	0-1
M	MOVE AT CONSTANT VELOCITY	STEPS/SEC	0,±20-20000	NONE	NONE
R	INDEX TO POSITION	POSITION	±8,388,607.99	NONE	NONE
+	INDEX IN "+" DIRECTION	STEPS	0.01-8,388,607.99	NONE	NONE
-	INDEX IN "-" DIRECTION	STEPS	0.01-8,388,607.99	NONE	NONE
PROGRAM COMMANDS					
G	GO	ADDRESS	0-1791,2048	TRACE OFF/ON	0-1
J,j	JUMP	ADDRESS	0-1791	N+1 TIMES	0-255
k	SPECIAL TRIP	POSITION	±8,388,607.99	PORT VALUE	0-56
L	LOOP ON PORT	ADDRESS	0-1791,2048	CONDITION	0-5
P	PROGRAM MODE	ADDRESS	0-1791	NONE	NONE
Q	QUERY (LIST) PROGRAM	ADDRESS	0-1791	NONE	NONE
q	QUERY (PAGE) PROGRAMS	ADDRESS	0-1791	NONE	NONE
UTILITY COMMANDS					
ESC	ABORT/TERMINATE	NONE	NONE	NONE	NONE
@	SOFT STOP	STOP MODE	NONE,01	NONE	NONE
^C	SOFTWARE RESET	NONE	NONE	NONE	NONE
A	PORT	WRITE/READ	0-128/129	NONE	NONE
l	LIMIT POLARITY	INVERT,SOFTSTOP	0-3	NONE	NONE
O,o	SET ORIGIN	NONE	NONE	NONE	NONE
W	WAIT DELAY	0.01 SECONDS	0-65,535	NONE	NONE
X	EXAMINE PARAMETERS	NONE	NONE	NONE	NONE
Z,z	DISPLAY POSITION	READOUT MODE	0-1	NONE	NONE
[	READ NV MEMORY	ADDRESS	0-2047	COUNT	0-255
\	WRITE TO NV MEMORY	ADDRESS	0-2047	DATA	0-255
]	READ LIMITS/HARDWARE	LIMITS/HDW	0-1	NONE	NONE
~	READ MOVING STATUS	NONE	NONE	NONE	NONE
y	OUTPUT CR LF	NONE	NONE	NONE	NONE
ENCODER COMMANDS					
d	DEADBAND ENABLE	STEPS	0-255	NONE	NONE
e	ENCODER RESOLUTION	LINE COUNT	0,50-12750	NONE	NONE
f	FIND ENCODER MARK	DIRECTION	0-1	NONE	NONE
h	HUNT RESOLUTION	MICROSTEP RES	0-8	NONE	NONE
r	SET STALL RETRY COUNT	RETRIES	0-255	NONE	NONE
s	STALL FACTOR	FACTOR	0-255	NONE	NONE
t	STALL TEST DELTA	STEPS	0-255	NONE	NONE
v	HUNT VELOCITY	STEPS/SEC	20-8,000	NONE	NONE

When the indexer is operating in Party line mode, the X command does not provide the operational parameters as described in the manual for Single line mode. This application note shows a method for reading the stored parameters directly from NVM by reading the locations where the parameters are stored. Use the "{" command to read the desired location.

Note that this table is only valid following an "S" store command and immediately following a power on reset.
reset.

The following paramters are used for the example in the following table. (Display shown after an 'X' command).

```
X Y= 0/25  E= 50  K= 22/45  H=vr  na=1
I= 400 ( 12800/32)  V= 2048 ( 8192 /4) (rl= 8 )
e = 500  (ratio = 10)
d= 30  v= 700 /16
s= 20  t= 5  r= 5 (lag = 3)
```
(Jog speed set to 11/33. Resolution set to Full Step)

This following map defines the internal locations where the defalts are stored. These values resided in the NV memory starting at address 1984 and continue through NVM address 2019.

Example Dump of 1984-1993:

32	15	5	0	11	33	22	45	0	0

Location	Parameter	Value	Notes
1984	statbits	32	used internally
1985	resmode	15	15=vr, 14= fr
1986	retry_count	5	r set to 5
1987	divide factor	0	D command set to 0
1988	jog slow speed	11	First parameter in B command
1989	jog hi speed	33	Second parameter in B command
1990	up k fact	22	First parameter in K command
1991	down k fact	45	Second parameter in K command
1992	trip_lo	0	No Trip Point set
1993	trip_mid	0	

Example Dump of 1994-2003:

0	0	0	25	50	244	1	20	30	0

1994	trip_hi	0	
1995	trip_vector	0	
1996	i hold	0	First parameter in Y command
1997	i run	25	Second parameter in Y command
1998	settle time	50	E command
1999	encoder lines lo	244	Remainder of Encoder Lines/ 256 (e command)
2000	encoder lines hi	1	Whole portion of Encoder Lines/256 Ex. e=500 => 256+244
2001	stall factor	20	s command
2002	dead zone lo	30	d command
2003	dead zone hi	0	

Example Dump of 2004-2013:

5	10	255	243	129	26	192	167	253	137

Location	Parameter	Value	Notes
2004	stall interval	5	t command
2005	encoder ratio	10	internally calculated
2006	initial velocity lo	255	I command
2007	initial velocity hi	243	
2008	init velocity ptrl	129	used internally
2009	init velocity ptrh	26	used internally
2010	initial velocity res	192	used internally
2011	slew velocity lo	167	V command
2012	slew velocity hi	253	
2013	slew velocity ptrl	137	used internally

Example Dump of 2004-2019:

26	192	36	249	196	49

2014	slew velocity ptrh	26	used internally	
2015	slew velovity res	192	used internally	
2016	hunt velocity lo	36	h command	
2017	hunt velocity hi	249		
2018	hunt reslolution	196	used internally	
2019	name	49	ASCII "1"	(decimal)

The following formula is used to calculate velocity values from the data stored in NVM:

$$\frac{14.7 \times 10^{-6}}{(65535 - ((\text{NVM Hi} * 256) + \text{NVM Lo})) * 12}$$

To communicate with the indexer, a terminal emulation program running on a PC is commonly used. An example of a terminal emulator is the TERMINAL program supplied with all versions of Microsoft Windows. To use this program to communicate with the indexer, the COM port must be properly configured.

Start the TERMINAL program running by double clicking on the TERMINAL icon in the Accessories Screen in Windows. Refer to figure 1 for the menu to select the Communication settings screen in TERMINAL. Once in the settings screen, configure the COM settings as shown in figure 2. Be sure to set the COM port to the appropriate selection for the PC being used. Once these settings are established, click on the OK button. Press the space key to initiate Single Mode communication with the connected indexer. Refer to the Communication section of this manual for more information.

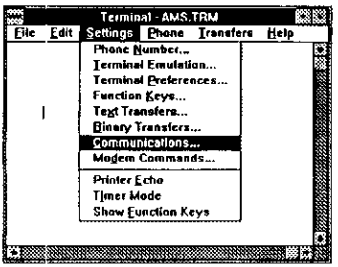


Figure 1.

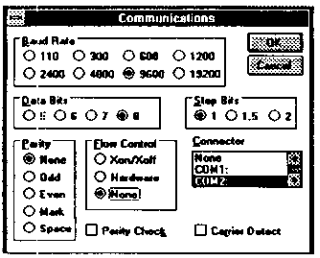


Figure 2.

The TERMINAL program can also be used for downloading pre-written programs to the indexer. An ASCII editor is used to create the program using the instructions shown in this manual. (For information on an ASCII editor, refer to the NOTEPAD program supplied with Microsoft Windows).

To download a program using TERMINAL, some default TERMINAL settings must be changed as shown in figures 3-6. Start TERMINAL and select the Text Transfers menu under the Settings menu as shown in figure 3. Once in the Text Transfers screen, configure the settings as shown in figure 4. Note that the delay between characters is set to 2/10 seconds and the send option is single character at a time. The delay value may need to be increased depending on the structure of the program being downloaded. This value should be set to 5/10 seconds if the "P" command is embedded in the program to be downloaded.

To download the program, select the Text Transfer menu as shown in figure 5. Select the Send File setting. Select the file to download and select OK as shown in figure 6. The file will be automatically sent by TERMINAL . All characters sent will be echoed by the indexer.

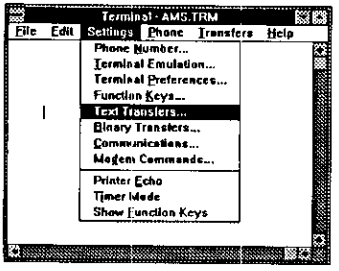


Figure 3.

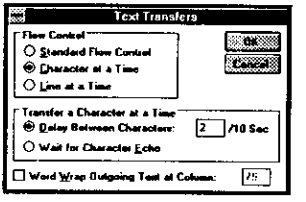


Figure 4.

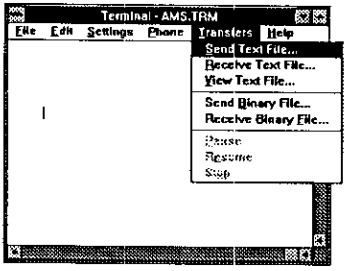


Figure 5.

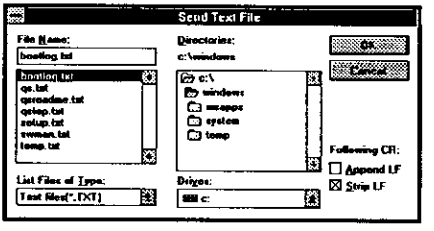


Figure 6.

This application note illustrates the use of the LIMIT input under program control to reverse direction of a move when the LIMIT is hit. Note that the system is configured such that index move +9000 will cause the LIMIT switch to be hit before position 9000 is reached. The W0 following the +9000 causes the program to wait until the move of 9000 steps is complete OR a LIMIT input is hit. Once the LIMIT input is hit, program control continues with the next statement after the W0 instruction.

```
P0
0 0       set position counter to 0
1 +9000   move in + direction toward limit, (assume limit will be hit before 9000)
6 W0      wait for move complete
9 W500    wait 5 seconds at limit position
12 R0     return to 0 when limit is hit
17 P      end of program
```

This application program illustrates the programming of a rotary motion using indexer commands. The desired motion is to move the motor 370 degrees at a rate of 200 rpm, wait 5 seconds, move back 10 degrees, wait 5 seconds then return to the starting point. During the first part of the move, an output must be turned on when the position is 270 degrees. Note that a 1.8 degree/step motor is used, giving 200 full steps per revolution.

First, the conversion of degrees to steps is made for each movement:

370 deg. x 1 step/1.8 deg. = 205.56 full steps
270 deg. x 1 step/1.8 deg. = 150 full steps
10 deg. x 1 step/1.8 deg. = 5.56 full steps

Next, the speed in steps/second is determined:

200 rev/min. x 1 min./60 sec x 200 steps/rev = 667 steps/sec

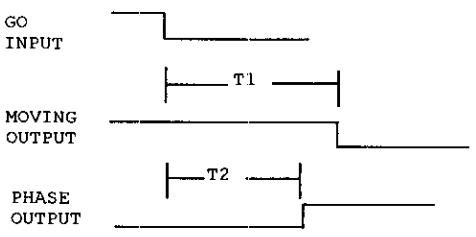
Next, program the indexer: (Memory addresses are not shown in this example)

```
P0          start at location 0
V 667       set velocity
H1          set variable resolution mode
D0          set full step resolution maximum
0           set origin
T 200 150   set trip point at position 150, vector to address 200
A 0         set output off
R +205.56   index to first position (trip routine will be executed at position 150)
W0          wait for move complete
W 500       wait 5 seconds
-5.56       move back 10 degrees
W0          wait for move complete
W500        wait 5 seconds
R0          return to starting position
W0          wait for move complete
P           end of program

P 200       program trip routine at address 200
A8          set output 1 on
T0 0        turn off trip
P           end of trip routine
```

Here are the results of the timing measurements for the IM483i for starting a move from program mode and from immediate mode. Note that the testing was done using a IM483i version 1.09. Timing for program mode was done using the command +1000 executed from both EEPROM (location 0) and RAM (location 130 with G130 at location 0).

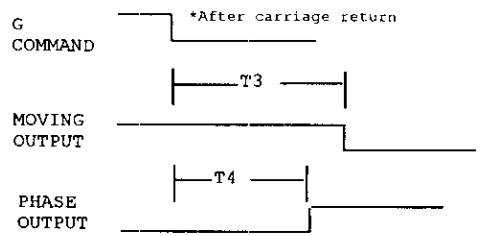
Program Mode using GO switch input



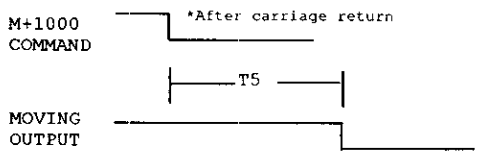
	EEPROM	RAM
T1	11.5	8
T2	8	8
T3	12	10
T4	10	10

all values in mS

Program Mode using G command



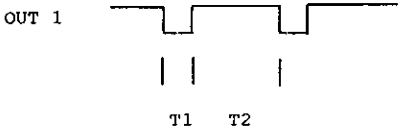
The following timing was done by sending M+1000 to the IM483i.



T5 = 5 mS

Here are the results of measurements taken on the timing of instruction execution in the indexer. The program used in each case is as follows:

```
A0
A8
G (to 1st location)
```



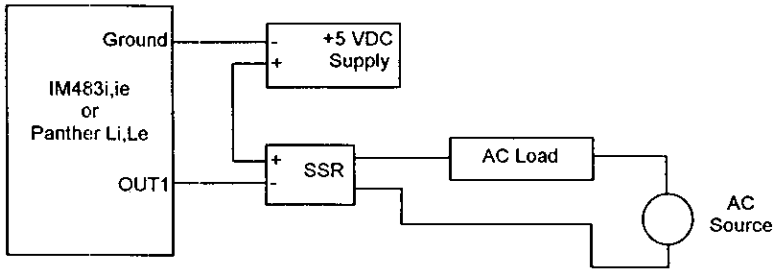
	EEPROM	RAM
T1	3mS	.18mS
T2	7mS	.34mS

Note that the internal fetch instruction cycle time varies depending on the length and complexity of the instruction. In the above case, the A command uses 2 bytes of memory while the G command uses 3 bytes. The total program is 7 bytes long and takes an average of 10mS from EEPROM and .52mS from RAM. This translates to an average instruction time of 1.4 mS and .07 mS for EEPROM and RAM respectively.

This application illustrates a method for controlling an AC powered device using the indexers OUT1 output. Examples of applications requiring AC control are counters, lamps, solenoids ...etc. Each of the three general purpose outputs of the indexer are capable of SINKING up to 20mA of current when low. These outputs can be used to control a device called a Solid State Relay, or SSR, using an external power supply. The type of SSR used in this example has a DC input and an AC output and can be used to control the types of AC devices mentioned above. Connection of the input of the relay to the OUT1 signal is shown below. The output of the relay controls the AC load.

The programming of the output port requires the use of the "A" command. For this example OUT1 is used and the commands A8 and A0 are used to toggle the state of bit 3 in the "A" command data which in turn controls the state of OUT1 through the indexer. The following program segment shows a method to toggle the relay on and off at a 5 second interval. Note that the indexer can be used as a logic controller, not requiring the use of a motor if motion is not required.

```
P0
A8      set out1 high turning off relay
W500    delay 5 seconds
A0      set out1 low turning on relay
W500    wait 5 seconds
G0      loop forever
P       end of program
```



*Note that the current into the drive's OUT1 pin must be limited to 20mA max.

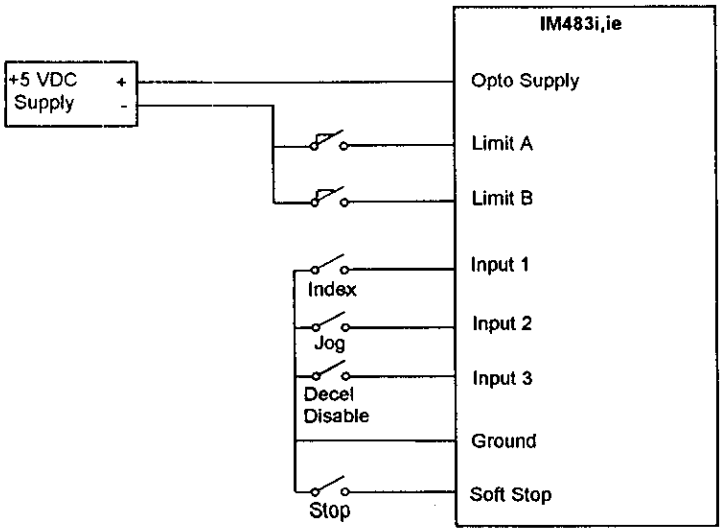
This application note demonstrates how to program the indexer to simulate a simple oscillator board using the three general purpose inputs. A cut to length machine is a typical application of this program. The inputs are defined as follows:

Input 1 - Active low input that initiates an indexed move.
Input 2 - Active low Jog input.
Input 3 - Enables motor deceleration when high and disables deceleration when low.
Soft stop - dedicated input that decelerates the motor if it is moving and aborts the program. To restart the program, power must be cycled.

LimitA,B - Optically isolated limit inputs to restrict travel in specified direction. (Hardware controlled, no software programming necessary.)

The hardware connections are shown below.

Program Description:
On power up, the indexer will begin executing instructions at address 1600. The G 0 instruction at this location branches the program to address 0, where the main section of the program is located. Addresses 0 through 10 and address 17 contain operating parameters that can be tailored for the application. Note that to change acceleration, the K parameters in both address 10 and address 17 must be changed. Address 10 through 28 form a loop that will continuously poll the three input switches for a closure. If input 3 is high, the acceleration parameters at address 10 will be used by any subsequent moves. If input 3 is low, the acceleration parameters at address 17 will be used. (At address 17 the deceleration slope is set to zero). Input 1 and input 2 each have their own service routines that will be executed when a closure is detected. The last instruction of each service routine must be a G10. This causes the program to return to the beginning of the loop that polls for switch closures.



Main Program:

Add.	Instruction	Comments
0	D 0	Divide resolution
2	H 0	Resolution mode
4	I 255	Initial velocity
7	V 2000 +	Slew Velocity
10	K 20 100	Ramp Slope (normal)
13	L 20 5	Branch to add. 20 if input 3 NOT Low
17	K 20 0	Ramp Slope (no deceleration)
20	L 100 0	Branch to add. 100 if input 1 NOT High
24	L 200 2	Branch to add. 200 if input 2 NOT High
28	G 10	Always jump to add. 10

Index (Input 1) service routine:

The instructions shown here will drive the motor 4000 steps in the positive direction, delay 500 milliseconds and move 4000 steps in the negative direction, returning to the original location. Change address 100 and address 111 for different length moves. Change address 108 for a different delay.

Add.	Instruction	Comments
100	+4000	Any sequence of instructions can be programmed here
105	W 0	Wait for motion to stop.
108	W 50	Delay 500 milliseconds.
111	-4000	
116	W 0	
119	G 10	Return to address 10. Must be the last line of service routine.

Jog (Input 2) service routine:

Add.	Instruction	Comments
200	M 2000	Change Jog speed or direction here
203	L 203 2	Wait for input 2 to go high
207	M 0 +	Stop moving
210	W 0	
213	G 10	Return to address 10. Must be the last line of service routine.

Boot instruction:

Add.	Instruction	Comments
1600	G 0	Automatically start program at address 0 on power up

WARRANTY

TWENTY-FOUR (24) MONTH LIMITED WARRANTY

Intelligent Motion Systems, Inc. ("IMS"), warrants only to the purchaser of the Product from IMS (the "Customer") that the product purchased from IMS (the "Product") will be free from defects in materials and workmanship under the normal use and service for which the Product was designed for a period of 24 months from the date of purchase of the Product by the Customer. Customer's exclusive remedy under this Limited Warranty shall be the repair or replacement, at Company's sole option, of the Product, or any part of the Product, determined by IMS to be defective. In order to exercise its warranty rights, Customer must notify Company in accordance with the instructions described under the heading "Obtaining Warranty Service."

This Limited Warranty does not extend to any Product damaged by reason of alteration, accident, abuse, neglect or misuse or improper or inadequate handling; improper or inadequate wiring utilized or installed in connection with the Product; installation, operation or use of the Product not made in strict accordance with the specifications and written instructions provided by IMS; use of the Product for any purpose other than those for which it was designed; ordinary wear and tear; disasters or Acts of God; unauthorized attachments, alterations or modifications to the Product; the misuse or failure of any item or equipment connected to the Product not supplied by IMS; improper maintenance or repair of the Product; or any other reason or event not caused by IMS.

IMS HEREBY DISCLAIMS ALL OTHER WARRANTIES, WHETHER WRITTEN OR ORAL, EXPRESS OR IMPLIED BY LAW OR OTHERWISE, INCLUDING WITHOUT LIMITATION, **ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE**. CUSTOMER'S SOLE REMEDY FOR ANY DEFECTIVE PRODUCT WILL BE AS STATED ABOVE, AND IN NO EVENT WILL THE IMS BE LIABLE FOR INCIDENTAL, CONSEQUENTIAL, SPECIAL OR INDIRECT DAMAGES IN CONNECTION WITH THE PRODUCT.

This Limited Warranty shall be void if the Customer fails to comply with all of the terms set forth in this Limited Warranty. This Limited Warranty is the sole warranty offered by IMS with respect to the Product. IMS does not assume any other liability in connection with the sale of the Product. No representative of IMS is authorized to extend this Limited Warranty or to change it in any manner whatsoever. No warranty applies to any party other than the original Customer.

IMS and its directors, officers, employees, subsidiaries and affiliates shall not be liable for any damages arising from any loss of equipment, loss or distortion of data, loss of time, loss or destruction of software or other property, loss of production or profits, overhead costs, claims of third parties, labor or materials, penalties or liquidated damages or punitive damages, whatsoever, whether based upon breach of warranty, breach of contract, negligence, strict liability or any other legal theory, or other losses or expenses incurred by the Customer or any third party.

OBTAINING WARRANTY SERVICE

Warranty service may obtained by a distributor, if the Product was purchased from IMS by a distributor, or by the Customer directly from IMS, if the Product was purchased directly from IMS. Prior to returning the Product for service, a Returned Material Authorization (RMA) number must be obtained. Complete the form at http://www.imshome.com/rma.html after which an RMA Authorization Form with RMA number will then be faxed to you. Any questions, contact IMS Customer Service (860) 295-6102.

Include a copy of the RMA Authorization Form, contact name and address, and any additional notes regarding the Product failure with shipment. Return Product in its original packaging, or packaged so it is protected against electrostatic discharge or physical damage in transit. The RMA number MUST appear on the box or packing slip. Send Product to: Intelligent Motion Systems, Inc., 370 N. Main Street, Marlborough, CT 06447.

Customer shall prepay shipping changes for Products returned to IMS for warranty service and IMS shall pay for return of Products to Customer by ground transportation. However, Customer shall pay all shipping charges, duties and taxes for Products returned to IMS from outside the United States.