Compumotor

Bulletin OM-8300-372

Operator's Manual

Model 372 Indexer

P/N: 88-006315-03

Parker Hannifin Corporation

Compumotor Division
5500 Business Park Drive

Rohnert Park, CA 94928

TABLE OF CONTENTS

1. INTRODUCTION 1
1.1. OVERVIEW 1
1.2. SPECIFICATIONS 1
1.3. COMPARING THE 372 TO OTHER INDEXERS 3
1.4. HOW TO USE THIS MANUAL 3

2. BASIC SYSTEM CONCEPTS 4
2.1. ACCELERATION/VELOCITY PROFILES 4
2.1.1. ACCELERATION/DECELERATION 5
2.1.2. MOTION PROFILES 5
2.2. DETERMINING MOTION PARAMETERS 8
2.2.1. ANALYTIC APPROACH 8
2.2.2. EXPERIMENTAL APPROACH 9
2.3. CONFIGURATION ALTERNATIVES 11
2.3.1. INTERACTIVE MANUAL CONTROL 11
2.3.2. COMPUTER-CONTROLLED OPERATION 11
2.3.3. PRE-PROGRAMMED OPERATION 12
2.3.4. CONTROL SIGNAL OPTIONS 12
2.3.5. CLOSED LOOP OPERATION 13
2.4. 372 INDEXER CONNECTORS 17

3. OPERATING MODES 20
3.1. OVERVIEW 20
3.1.1. RS-232 COMMUNICATIONS 20
3.1.2. INDICATORS 21
3.1.3. CONTROLS 21
3.2. BASIC MOTION CONTROL 21
3.2.1. MOVE PARAMETERS 22
3.2.2. PRESET MODE 23
3.2.3. ALTERNATING MODE 23
3.2.4. CONTINUOUS MODE 24
3.2.5. HOMING FUNCTION 24
3.2.6. DIRECT VELOCITY MODE 27
3.3. NON-VOLATILE MEMORY FEATURES 29
3.3.1. STORED PARAMETERS 29
3.3.2. STORED SEQUENCES 29
3.4. AUXILIARY CONNECTOR FIXED FUNCTION INPUTS 30
3.4.1. LIMIT SWITCH FUNCTION 30
3.4.2. STOP AND START INPUTS 31
3.4.3. HOMING FUNCTION INPUTS 32
3.4.4. TRIGGER INPUTS 32
3.5. PROGRAMMABLE INPUTS AND OUTPUTS 34
3.5.1. PROGRAMMABLE OUTPUTS 35
3.5.2. DEFAULT INPUTS: INPUT MODE 1 . . . 35
3.5.3. DISCRETE SEQUENCE SELECT: INPUT MODE 2 . . . 37
3.5.4. BCD SEQUENCE SELECT: INPUT MODE 3 . . . 38
3.5.5. OFF LINE LOCKOUT: INPUT MODE 4 . . . 38
3.5.6. APPLICATION OF INPUT MODE FUNCTIONS 39

3.6. POSITION TRACKING 41
3.6.1. ENCODER REFERENCED POSITION 41
3.6.2. ENCODER POSITION SERVO 42
3.6.3. MOTOR STALL DETECTION 43
3.6.4. POSITION LOSS DETECTION 45
3.6.5. MULTIPLE AXIS STOP 45
3.6.6. ABSOLUTE POSITION COORDINATES 46

4. INSTALLATION 48
4.1. INSPECTION 48
4.2. PHYSICAL ASSEMBLY 48
4.3. INSTALLATION 50
4.3.1. QUICK START 50
4.3.2. CONNECTING POWER 50
4.3.3. CONNECTING THE DRIVE AND MOTOR 51
4.4. AUXILIARY CONNECTORS 52
4.5. RS-232C COMMUNICATIONS INTERFACE 56
4.5.1. RS-232C CABLE 56
4.5.2. MATCHING RS-232C INTERFACE CHARACTERISTICS . 60
4.5.3. SETTING THE UNIT NUMBER 60
4.5.4. TESTING THE RS-232C INTERFACE 61
4.5.5. PARAMETER SETUP 62
4.5.6. TROUBLESHOOTING THE RS-232C INTERFACE . . . 64
4.6. ENCODER INSTALLATION 65
4.6.1. CONNECTING AN ENCODER 65
4.6.2. CONFIGURING THE ENCODER OPTIONS 65
4.6.3. SELECTING ENCODER RESOLUTION VALUES 66
4.6.4. TESTING POSITION TRACKING SETUP PARAMETERS . 69
4.7. REPLACING A 372 IN THE FIELD 71

5. INDEXER COMMANDS 72
5.1. COMMAND CONVENTIONS and DEFINITIONS 72
5.1.1. DELIMITER 72
5.1.2. INDIVIDUAL COMMANDS 72
5.1.3. MULTIPLE INDEXERS 75
5.1.4. COMMAND BUFFER CAPACITY 75
5.1.5. FUNCTIONAL GROUPING OF COMMANDS 75
5.1.6. SPECIAL COMMAND SYNTAX 76
5.2. MOTION PARAMETER COMMANDS 77
5.2.1. TYPE OF MOTION COMMANDS 77
5.2.2. MOTION PARAMETERS 79
5.2.3. SET UP PARAMETERS 81
5.2.4. MOTION PARAMETER CHANGES 82
5.2.5. Examples - MOTION PARAMETER COMMANDS 82
5.2.6. POSITION TRACKING COMMANDS 83
5.3. EXECUTION COMMANDS 85
5.3.1. START MOVE COMMANDS 85
5.3.2. LOAD COMMANDS 89
5.3.3. STOP MOVE COMMANDS 90
5.3.4. PAUSE COMMANDS 91

5.3.5. LOOP COMMANDS 93
5.3.6. CONDITIONAL BRANCHING COMMANDS 93
5.3.7. MOTOR SHUTDOWN -- ST Command . 94
5.3.8. EXAMPLES OF EXECUTION COMMANDS 94
5.4. **STATUS REQUEST COMMANDS** 97
5.4.1. IDENTIFICATION OF UNITS (UNIT NUMBERS) . . . 97
5.4.2. FORMAT OF STATUS REQUEST RESPONSES 99
5.4.3. INTERPRETING RESPONSES OF THE FORM *A<CR> . 100
5.4.4. BUFFER STATUS RESPONSE -- uB Command . 101
5.4.5. INDEXER STATUS RESPONSE -- uR Command . 101
5.4.6. LIMIT SWITCH STATUS -- uRA Command . 102
5.4.7. PAUSE STATUS -- uRB Command . 102
5.4.8. CLOSED LOOP STATUS -- uRC Command . 104
5.4.9. TRIGGER INPUT STATUS -- uTS Command . 105
5.4.10. POSITION REPORTING 105
5.4.11. INTERPRETING DECIMAL POSITION REPORTS . . . 106
5.4.12. INTERPRETING HEXADECIMAL POSITION REPORTS . 107
5.4.13. INTERPRETING BINARY POSITION REPORTS . . . 107
5.4.14. COMMAND EXECUTION STATUS -- uCR Command . 109
5.4.15. DEFAULT PARAMETER STATUS -- uDR Command 110
5.4.16. SEQUENCE REPORTS uXUn, uXN, uXV Command . 111
5.4.17. SOFTWARE SWITCH STATUS -- uSR Command . 112
5.4.18. OPTION SWITCH STATUS -- uOR Command . 112
5.4.19. ENCODER FUNCTIONS STATUS -- uFR Command . 113
5.4.20. SIGNAL ENABLE STATUS -- uQR Command . 113
5.4.21. SIGNAL SOURCE STATUS -- uQI Command . 114
5.4.22. SCALE FACTOR STATUS -- uUR Command . 114
5.4.23. SOFTWARE REVISION -- uRV Command . 114
5.5. **GENERAL PURPOSE COMMANDS** 115
5.5.1. COMMUNICATIONS INTERFACE ENABLE/DISABLE . . 116
5.5.2. BACKSPACE 116
5.5.3. PROGRAMMABLE OUTPUTS -- Obb Command . 116
5.5.4. SOFTWARE SWITCH -- SS Command . 117
5.5.5. OPTION SWITCH -- OS Command . 118
5.5.6. SIGNAL ON CONDITION -- QS Command . 119
5.5.7. INPUT MODE -- IM Command . 120
5.5.8. MESSAGES CONTROL -- MS Command 120
5.6. NON-VOLATILE MEMORY CONTROL COMMANDS 120
5.6.1. PARAMETER STORAGE 121
5.6.2. SEQUENCE EDITING 122
5.6.3. SEQUENCE EXECUTION COMMANDS 123
APPENDIX A: 372 CIRCUIT BOARD JUMPERS 124
APPENDIX B: ERROR CODES 125
APPENDIX C: SAMPLE AUXILIARY CONNECTOR WIRING 126
APPENDIX D: Summary of Connectors 129
APPENDIX E: STATUS REQUEST RESPONSE SUMMARY 132
APPENDIX F: COMMAND SUMMARY 135
INDEX 139

LIST OF TABLES

Table 1-1. Performance and Electrical Specifications 2
Table 1-2. Mechanical and Environmental Specifications . . . 3
Table 3-1. Stored Parameters and Configuration data . . . 28
Table 3-2. Programmable Inputs and the Input Modes 36
Table 4-1. Motor/Driver Connection Pinout 52
Table 4-2. Auxiliary Connector Pinouts 54
Table 4-3. Typical DCE Connection Pinouts 58
Table 4-4. Setting the Unit Number 60
Table 4-4. Encoder Connection Pinouts 67
Table 5-1. Shaft Motion Commands 78
Table 5-2. The MR command Motor Resolutions 81
Table 5-3. Execution Commands 87
Table 5-4. Status Request Commands 98
Table 5-5. RA Response 103
Table 5-6. RB Response 103
Table 5-7. RC Response 104
Table 5-8. Binary Position Conversion 107
Table 5-9. General Purpose Commands 115
Table 5-10. Memory Control Commands 121
Table 1-1. Performance and Electrical Specifications 2
Table 1-2. Mechanical and Environmental Specifications . . . 3
Table 3-1. Stored Parameters and Configuration data . . . 28
Table 3-2. Programmable Inputs and the Input Modes 36
Table 4-1. Motor/Driver Connection Pinout 52
Table 4-2. Auxiliary Connector Pinouts 54
Table 4-3. Typical DCE Connection Pinouts 58
Table 4-4. Setting the Unit Number 60
Table 4-4. Encoder Connection Pinouts 67
Table 5-1. Shaft Motion Commands 78
Table 5-2. The MR command Motor Resolutions 81
Table 5-3. Execution Commands 87
Table 5-4. Status Request Commands 98
Table 5-5. RA Response 103
Table 5-6. RB Response 103
Table 5-7. RC Response 104
Table 5-8. Binary Position Conversion 107
Table 5-9. General Purpose Commands 115
Table 5-10. Memory Control Commands 121

LIST OF FIGURES

Figure 2-1. Acceleration Profile 6
Figure 2-2. Motion Profiles 7
Figure 2-3. Basic Elements to be Analyzed 10
Figure 2-4. Calculating Move Time -- Trapezoidal profile . . 14
Figure 2-5. Calculating Move Time -- Triangular profile . . 15
Figure 2-6. Limit Switch Input 16
Figure 2-7. Position Tracking System 18
Figure 2-8. Computer-Controlled 372 19
Figure 3-1. Absolute coordinate system 47

Figure 4-1. Mounting Orientation 49
Figure 4-2. 120/240 VAC Power Connectors 49
Figure 4-3. Determining Direction of Rotation 53
Figure 4-4. Typical AUXILIARY Connection 55
Figure 4-5. RS-232C Cable 57
Figure 4-6. Null DCE Connector 59
Figure 4-7. "Daisy-Chain" Cabling 59
Figure 4-8. Typical Encoder Input Circuit 68
Figure 5-1. Remote Terminal Operation 74
Figure 5-2. W2 and W3 Response Interpretation 108
Figure 5-3. W3 Response Interpretation 109
Figure 2-1. Acceleration Profile 6
Figure 2-2. Motion Profiles 7
Figure 2-3. Basic Elements to be Analyzed 10
Figure 2-4. Calculating Move Time -- Trapezoidal profile . . 14
Figure 2-5. Calculating Move Time -- Triangular profile . . 15
Figure 2-6. Limit Switch Input 16
Figure 2-7. Position Tracking System 18
Figure 2-8. Computer-Controlled 372 19
Figure 3-1. Absolute coordinate system 47
Figure 4-1. Mounting Orientation 49
Figure 4-2. 120/240 VAC Power Connectors 49
Figure 4-3. Determining Direction of Rotation 53
Figure 4-4. Typical AUXILIARY Connection 55
Figure 4-5. RS-232C Cable 57
Figure 4-6. Null DCE Connector 59
Figure 4-7. "Daisy-Chain" Cabling 59
Figure 4-8. Typical Encoder Input Circuit 68
Figure 5-1. Remote Terminal Operation 74
Figure 5-2. W2 and W3 Response Interpretation 108
Figure 5-3. W3 Response Interpretation 109

1. INTRODUCTION

The Compumotor 372 indexer is a microprocessor based programmable pulse generator for the control of pulsed motor and drive systems. It can be controlled remotely, or programmed to operate independently, using the RS-232C interface.

1.1. OVERVIEW

The 372 provides precision control of pulse rates and total pulse count for motor drive amplifiers that convert pulses to motor position. Compumotor's Position Tracking capability is built in; incremental encoder feedback may be connected for verification of motor position.

Flexible input and output capability allows synchronization of motor operations using external sensors and controls, and remote command of preprogrammed units with discrete control signals. Full optical isolation protects the 372 from harsh electrical environments.

The 372 Indexer will drive any size Compumotor motor/drive system, and most translator and stepper motor combinations. Using the 372, you can rotate the motor shaft to a precise position and stop, rotate at a constant velocity, or generate a combination of such moves with varying velocity control parameters.

ASCII character commands are used for control of the 372 from an RS-232 Console device, such as a terminal or computer. (The ASCII character format is standard on many terminals and small computers.) These commands are discussed in detail in Chapter 5.

When the 372 is connected to the RS-232 interface, up to 8 indexers can be controlled from the same Console device. Commands may be directed to any unit or all units on the line. Any 372 is able to provide status information on many aspects of its operations, including position, execution and discrete input status.

The 372 can be controlled "live", or, often used functions can be programmed into non-volatile memory for later execution without a terminal or computer. Programming allows automatic operation or low level remote control using switches and programmable logic controllers. It is simple to switch between the various modes of operation at will.

1.2. SPECIFICATIONS

Performance, electrical, environmental and mechanical specifications are presented in Tables 1-1 and 1-2.

Table 1-1. Performance and Electrical Specifications

Performance:	
Stepping Accuracy	±0 Steps from Preset total
Velocity Accuracy	±0.02% of Set rate, ±0.001 rev/sec
Velocity Repeatability	±0.02% of Set rate above 0.01 rev/sec
Power:	
Input	100 - 130 VAC @ 0.25A, 50/60 Hz 200 - 250 VAC @ 0.125A, 50/60 Hz
Output	+5 VDC .250A
RS-232C Interface:	3-wire DCE configuration (TX,RX,GND) ±5 VDC, 2mA
All other Outputs:	1.1 VDC max low, >3.5 VDC min high +35, -45 mA minimum
Motor Drive Step output Step Pulse Widths:	
4096 to 50,000 steps/rev	1 microsec duration nominal
1000 step/rev	5 microsec duration nominal
200 and 400 steps/rev	15 microsec duration nominal
AUXILIARY Inputs (except Limits)	On current: 10 mA min, 50 mA max. voltage: 3.2 vdc min, 10VDC max.
	Minimum signal duration: 4 msec
Start and Stop:	Minimum signal duration: 100 usec
CW, CCW Limit	10 mA sink to DC COMMON required to move (220 ohms pullup to +5V)
Encoder inputs (All)	
Input Impedance	1Kohm
Maximum Frequency	100 kHz
Maximum Input Voltage	5 VDC
Turn-On Voltage	2.5 VDC
Turn-Off Voltage	2.0 VDC

Table 1-2. Mechanical and Environmental Specifications

Size	Height*	9.5 in.(24.13 cm)	
	Width*	2.61 in.(6.63 cm)	
	Depth*	6.73 in.(17.09 cm)	
Weight in Pounds (KG)		Net	8(3.63)
		Shipping	12(5.44)

Environmental

Operating	32 to 140°F (0 to 60°C) 0 to 95% Humidity, non-condensing
Storage	-22 to 185°F (-30 to 85°C)

1.3. COMPARING THE 372 TO OTHER INDEXERS

The 372 incorporates features found in a number of other Compumotor indexers. The basic command syntax is a superset of that of the Model 2100 (with Position Tracking and IEEE-488 options). The 372 command language adds the additional commands provided with the model PC21 indexer, and the non-volatile memory commands of the AX and CX indexer/drives. Preprogrammed operation using discrete command inputs allows the 372 to function like the model 430 indexer.

1.4. HOW TO USE THIS MANUAL

You do not have to be an expert in any field to use the 372 Indexer. This manual contains all information you will need to install and successfully operate the 372. A terminal or computer is required to control the 372, you may need access to the computer or terminal manuals that define set-up conditions for RS-232C operation of the respective terminal or computer. This manual assumes that you have access to other required manuals.

Chapter 2 presents the concepts and terminology used in the manual and identifies the 372 functional signal inputs and outputs.

Chapter 3 describes modes of operation using a terminal, using control inputs, using encoder feedback, and using memory features.

Chapter 4 describes installation procedures for the 372. Separate sections are included that describe installation of the RS-232C communications interface, limit switches, and optional features.

Chapter 5 describes the 372 interface commands.

2. BASIC SYSTEM CONCEPTS

All applications of the 372 Indexer involve either movement of a motor to a precise position (number of steps) or movement of the motor at a prescribed velocity (steps per second). Output signals for both position and velocity can be controlled with a high degree of precision without any external feedback.

You can choose among Compumotor motor/drive options ranging from 400 to 50,000 steps per revolution (steps/rev) in rotary motors, and 80 to 25,000 steps per inch for linear motors. Each time a motor/drive system receives one step pulse, it moves the motor one increment.

Position or distance parameters are defined in terms of motor steps, but velocity and acceleration parameters ordinarily use revolutions per second, or inches per second. For the sake of brevity, the discussions that follow will refer to velocity units in terms of revolutions for rotary motors. For the most part, the discussions apply to linear motors as well if inches are substituted for revolutions in the units of measure.

When commands are structured for the system, it is necessary to think in terms of the number of steps/rev; e.g., for a 25,000 step motor, 180 degrees = 12,500 steps, 1 revolution = 25,000 steps, etc. You must refer the desired load motion for your application to rotation of the Compumotor motor shaft.

Shaft motion must be specified in terms of acceleration (in rev/sec/sec), angular velocity (in rev/sec) and angular position (either incremental steps from the current motor position or absolute steps from a reference zero position).

Both the size and the resolution of the Compumotor motor must be determined for any given application. Refer to the Compumotor Product Catalog for a discussion of motor selection, calculations, and criteria.

2.1. ACCELERATION/VELOCITY PROFILES

As discussed in the Compumotor Product Catalog, the Compumotor motor moves in incremental steps. In practical applications, load inertia tends to smooth the actual motion of the load. The small size of the motor steps minimizes end-of-move settling time at moderate acceleration values.

2.1.1. ACCELERATION/DECELERATION

Acceleration is defined as the rate of increase of angular velocity in revolutions per second per second (rev/sec/sec). When the 372 drives the motor from rest to a given velocity, a string of pulses of increasing frequency is sent to the motor/drive. The acceleration parameter is also used for deceleration of the motor when changing velocity or for an ordinary stop.

Relatively high pulse rates are used to drive high-resolution Compumotor motors; therefore, the motion can be graphically portrayed by a straight line, as shown in Figure 2-1.

The acceleration/deceleration rate must be selected as a function of the inertia of the load to allow operation within the torque limits of the motor. A margin of 50 to 100 percent of normal load torque requirements is recommended. If the selected acceleration rate requires torque that exceeds the capability of the motor and drive, a stall will occur. When very high acceleration rates are used, there may be a tendency for overshoot and longer settling times.

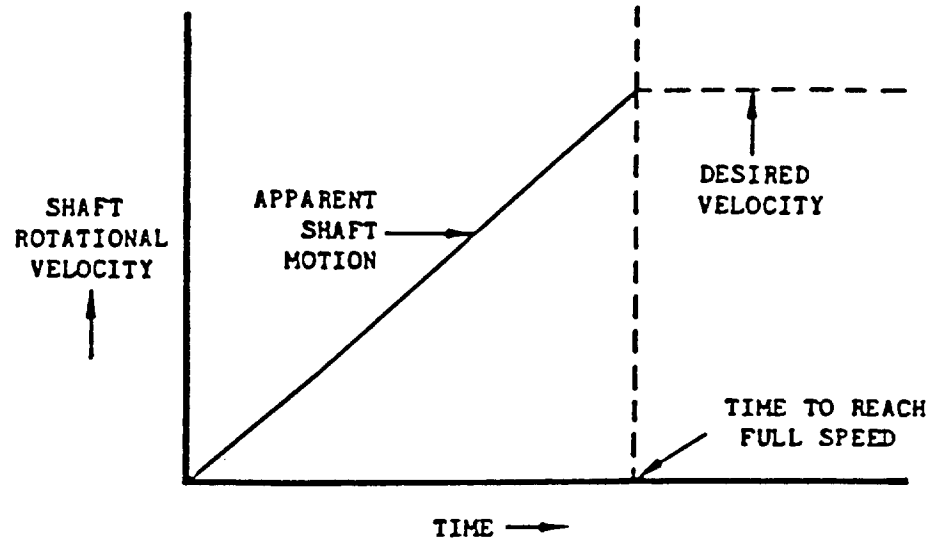
Normally acceleration and deceleration occur at the same rate. In the Continuous mode, different acceleration and deceleration rates may be used.

2.1.2. MOTION PROFILES

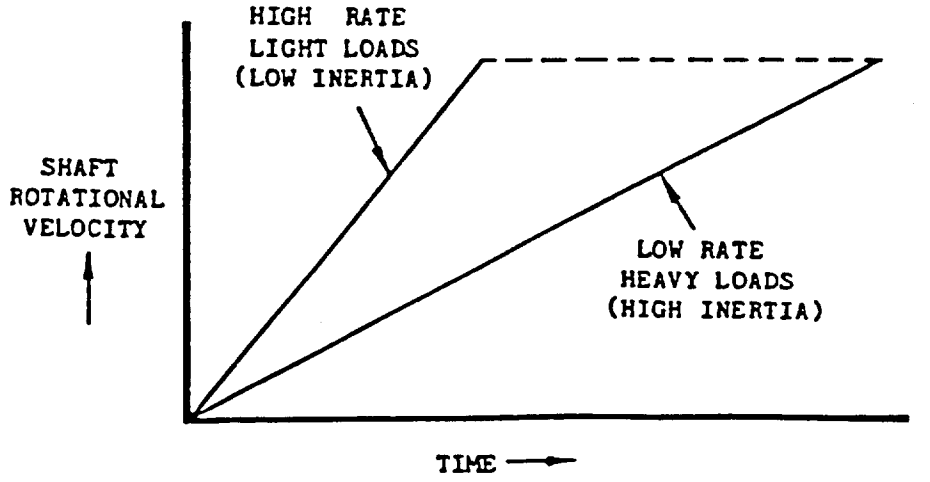
Refer to Figure 2-2. The trapezoidal motion pattern, Figure 2-2A, occurs when the distance to be covered is large compared with the distance required to reach the specified velocity. Part of the motion occurs at a constant specified velocity.

The triangular motion pattern (Figure 2-2B) occurs when the distance required to reach the specified velocity is large compared with the distance to the new position. In this case the motor shaft velocity is continually changing during the move and there is no constant velocity phase. The average velocity would be one-half the peak velocity.

Figure 2-1. Acceleration Profile

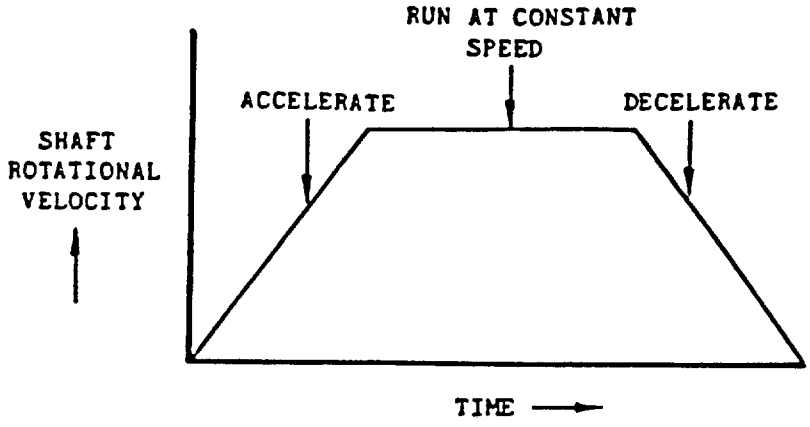


A. ACCELERATION CURVE

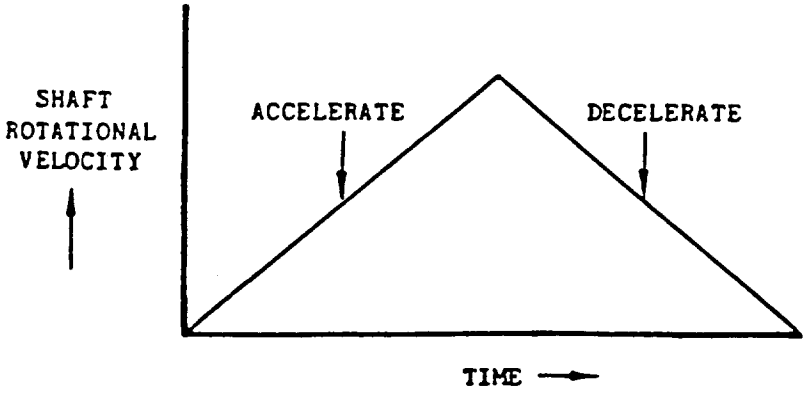


B. CAPABILITY

Figure 2-2. Motion Profiles



A. TRAPEZOIDAL PATTERN



B. TRIANGULAR PATTERN

2.2. DETERMINING MOTION PARAMETERS

You can choose an analytic approach or an experimental approach to establish the acceleration and velocity parameters appropriate to the application. The basic elements that come in to play in most applications are shown in Figure 2-3. The motor/drive must be sized for the application.

Determine the number of motor steps (or encoder steps if appropriate) required to reach a desired angular position of the motor shaft based on the motor resolution and transmission characteristics. The motor/drive must be sized for the application. Refer to the Compumotor Product Catalog for sizing guidelines.

In limited travel situations, end-of-travel limit switches should be installed to protect the load. (See Chapter 4 for installation instructions.)

2.2.1. ANALYTIC APPROACH

The desired load motion pattern must be referenced to the motor shaft. The motor/drive acts as a stiff drive source. The system analysis should take into account any gear play, coupling effects, transmission elasticity, load inertia, load friction, etc. These considerations will impact positioning accuracy, maximum acceleration rate and maximum velocity.

Motion parameters are calculated similarly for positioning applications and speed control applications. Indexing motion parameters are typically defined by how far the load is to be moved in a specified period of time. For Speed control, the question is how much time passes, or what distance is covered before the load gets to the desired velocity. For indexing, distance divided by move time yields average velocity.

For a Triangular motion profile, maximum velocity will be twice the average velocity, and the acceleration time is half the total move time.

A Trapezoidal profile is often desirable to reduce the maximum motor velocity required. If acceleration is increased, velocity may be reduced without increasing move time. A good general purpose Trapezoidal profile is obtained when the maximum velocity selected is 1.33 (four thirds) times the average, and acceleration time is a quarter of total move time.

In all cases, the relationship of the acceleration and velocity settings can be determined by using the equation

$$A = (V_f - V_o)/t_a$$

Where A = acceleration
V_f = final velocity
V_o = initial velocity
t_a = acceleration time

For the acceleration distance,

$$D_a = At_a^2/2$$

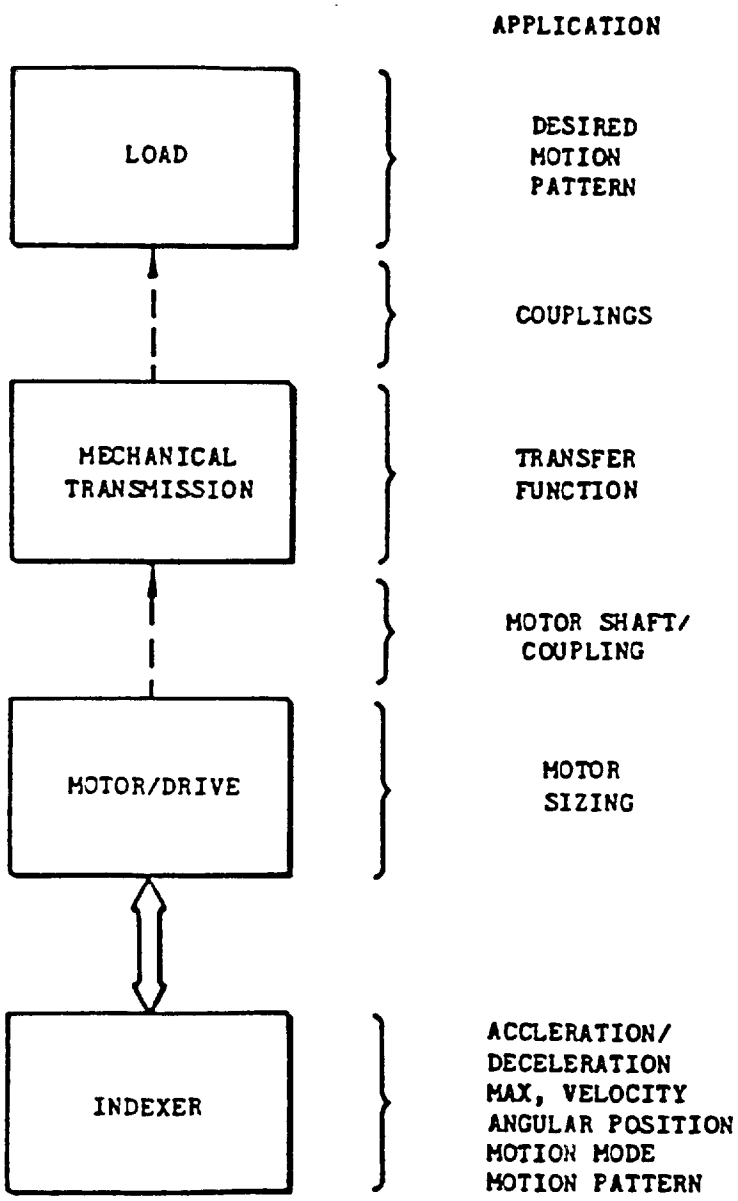
Where D_a = acceleration distance

These equations can be used to calculate any parameter when the others are known. Velocity and acceleration units should be in terms of revolutions and seconds, to match indexer parameters. (See Figures 2-4 and 2-5.) The calculated parameters can be tested and changed quickly once the indexer is "On Line".

2.2.2. EXPERIMENTAL APPROACH

The motor/drive must be sized for the application. Refer to the Compumotor Product Catalog for sizing guidelines. You can then take advantage of the fact that the motor/drive will stall (without any damage) whenever the performance specified by the command exceeds the capability of the motor/drive being used.

Figure 2-3. Basic Elements to be Analyzed



2.3. CONFIGURATION ALTERNATIVES

A large number of options may be employed in the operation of the 372. The requirements of the application will dictate which of the following general modes of operation will be used, and what external connections are required.

Regardless of whether the unit is to interact with a computer or stand alone with no connections except the drive, some kind of Console programming device with an RS-232C interface will be required for initial set up . Refer to Chapter 4.6 for instructions on connecting and operating the interface.

2.3.1. INTERACTIVE MANUAL CONTROL

Once a Console device has been connected and RS-232 communications have been established, the operator has full control of the 372 with access to all its operating modes and parameters. Operator control consists of typing in alphanumeric commands. The command "language" is discussed in detail in Chapter 5. This is ordinarily the mode in which the unit will be configured for operation in other modes.

The indexer may be instructed to index repeatedly, move at a fixed velocity, change speed, report position, etcetera. Once a list of commands has been entered, the programming terminal can be removed without affecting execution. Execution will continue until the command list has been completed. When command sequences are stored in non-volatile memory, these may be executed with a minimum of Console input.

Once sequences and setup parameters are entered, the 372 can be configured to execute stored functions without a Console device connected. The resulting configuration can then be "saved" in non-volatile memory. See section 2.3.4.

The Console device can be located up to 50 feet away from the 372. One such device can control up to 8 372 indexers.

2.3.2. COMPUTER-CONTROLLED OPERATION

The 372 can be controlled interactively by a computer. This is diagrammed in Figure 2-8. Any computer that can generate a string of ASCII characters and has an operational RS-232C serial port can be used. Many programmable controllers also provide the required RS-232C interface.

In a typical computer application, the computer measures or calculates what motor action is required, and tells the 372 to execute "live". The 372 is able to provide machine status information to assist in the process. The advantage of using a computer is that the computer is able to make complicated decisions, calculate parameters, and coordinate multiple axes.

2.3.3. PRE-PROGRAMMED OPERATION

In applications where the tasks required of the 372 are known ahead of time, the required functions can be stored in non-volatile memory and accessed by number.

Stored functions can range from a single motion command to a complex string of repeating and branching instructions with status reports. The 372 can be programmed to remember as many as 48 sequential operations up to 250 characters long, which may be chained, for a total maximum of 1,500 characters.

Any of these stored functions may be displayed, erased, or executed while on line.

Once programmed, the 372 may not need any RS-232 controller. It has two stand alone command modes whereby AUXILIARY inputs can be used as control signals to select and execute stored functions.

2.3.4. CONTROL SIGNAL OPTIONS

The 372 has a pair of inputs on its AUXILIARY 2 connector for End-of-Travel limit switches. For limited travel applications, switches should be installed on the load and fed back to the 372 to protect the load from overtravel. See Figure 2-6.

Note

The Stop input requires the connection of a normally closed switch or hardware jumper before motion is possible.

A Limit switch signal can cause an abrupt deceleration of the motor. Heavy fast moving loads may carry enough energy to fault or even damage a drive if this occurs. To prevent this, the Limit deceleration rate may be reduced. See the "*LA*" command, section 5.2.2.

The 372 has 16 other AUXILIARY inputs that are assigned various functions such as "Start", "Stop", "Jog", and "Go Home" for manual positioning or remote control.

Four inputs function as "Trigger" inputs which can be used to synchronize the execution of 372 sequential operations with external events or signals.

Eight inputs have variable functions according to their "Input Mode" assignment. These inputs can serve as hardware function commands such as Jog or Motor Shutdown, for selection of stored sequences, or for data entry. See section 3.5 for a description of input mode functions, and section 5.5.7 regarding the Input Mode command.

Two programmable outputs are also provided on the AUXILIARY 2 connector. Either output may be turned on and off during the execution of sequential operations, to signal or control external devices. The first of these inputs may be programmed to provide a signal during closed loop operation if feedback related problems occur. The second is programmed at the factory to indicate whether or not the motor is moving. These functions are controlled using the *FS* and *OS* commands. See sections 5.2.6 and 5.5.5.

2.3.5. CLOSED LOOP OPERATION

An incremental encoder may be connected to the 372 for position feedback information if desired. Encoder feedback provides positioning assurance using the 372's Position Tracking features. A Position Tracking system diagram is shown in Figure 2-7. Position Tracking can be used to:

a. Provide closed loop positioning for improved accuracy.

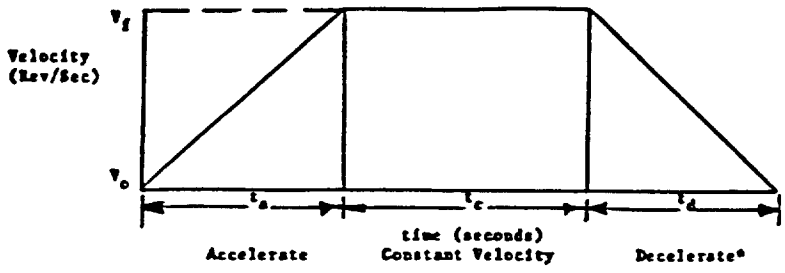
b. Dynamically hold an encoder referenced position.

c. Detect motor stall.

d. Independently verify position.

Any or all of these functions and capabilities can be enabled or disabled at will. Position Tracking functions are described in section 3.6, the commands are described in section 5.2.4.

Figure 2-4. Calculating Move Time -- Trapezoidal profile



*Generally, acceleration rate equals deceleration rate

$t_a = \frac{V_f - V_o}{a}$ $\quad$ a = acceleration (rev/sec/sec)

$t_c = \frac{d}{V_f}$ $\quad$ d = number revolutions

V_f = rev/sec, maximum velocity

t_d (= t_a) = $\frac{V_f - V_o}{a}$ $\quad$ a = Deceleration (rev/sec/sec)

Example (25,000 step motor)

a = Acceleration set to 1.0 rev/sec/sec (deceleration is same value)

V_f = Velocity set to 3.75 rev/sec

d = Distance set to 19.65 revolution

$t_a = \frac{V_f - V_o}{a} = \frac{3.75-0}{1.0} = 3.75$ seconds

V_{avg} = Average velocity in period t_a = 3.75/2 = 1.875

$t_a \times V_{avg} = 3.75 \times 1.875 = 7.03125$ revolution
= acceleration distance

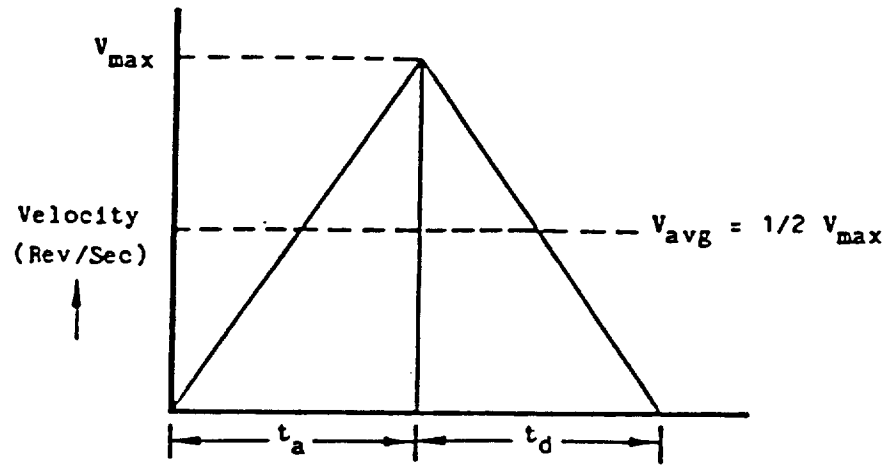
$t_d = t_a$, therefore distance moved during deceleration equals 7.03125 revolution

d = Distance to be moved at constant velocity (V_f) equals 19.65-2(7.03.25) = 5.5875 revolutions

$t_c = d/V_f = 5.5875/3.75 = 1.49$ seconds

Time to move = 2 x 3.75 + 1.49 = 8.99 seconds

Figure 2-5. Calculating Move Time -- Triangular profile



$t = t_a + t_d$ $\quad (t_a = t_d)$

$V_{max} = a\ t_a$ $\quad$ (a = acceleration rev/sec/sec)

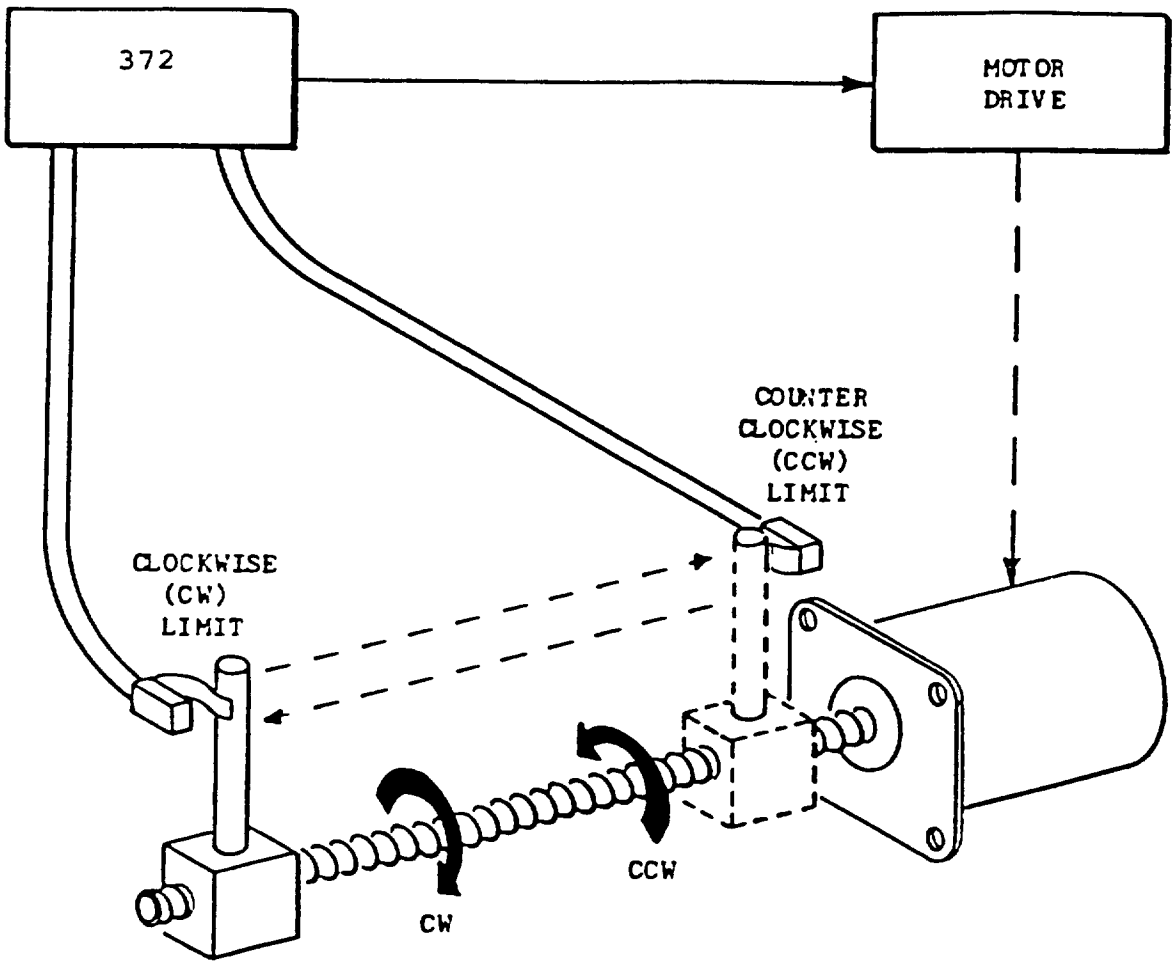
$t = \frac{d}{1/2\ V_{max}}$ (d = distance in revolutions)

$t^2 = \frac{2d}{a}$ $(t_a = 1/2\ t)$

$t = \sqrt{\frac{2d}{a}}$

Example: Move 5 revolutions in 4 seconds
Average velocity = 5/4 = 1.25 rev/sec
Max velocity = 2.5 rev/sec
Acceleration/Deceleration = 2.5/2 = 1.25 rev/sec/sec

Figure 2-6. Limit Switch Input



2.4. 372 INDEXER CONNECTORS

All 372 connectors are female. The 372 provides RS-232C, Driver, and Auxiliary, and Encoder connectors for inputs and outputs

These indexer connectors are described below.

A. AC POWER

Provides power for Indexer alone. Motor/drive is separately powered.

B. RS-232C SERIAL DATA PORT

25-pin "D" connector. Three-wire cable connection (transmit, signal ground, receive).

C. DRIVER CONNECTOR

25-pin "D" connector. Connects Indexer to motor/drive.

D. ENCODER

25-pin "D" connector. Accepts two-phase TTL-compatible inputs from and provides power for an incremental encoder.

E. AUXILIARY 1 CONNECTOR

25-pin "D" connector. Provides isolated input connections for motor Jog, 4 Trigger inputs, and other programmable command input functions. Isolated power is provided suitable for driving the inputs and powering remote switching devices.

F. AUXILIARY 2 CONNECTOR

25-pin "D" connector. Provides connections for isolated Limit switches; Remote Start; Remote Stop; Go Home; Home Enable; and Motor Shutdown inputs, and two user Programmable Outputs with isolated power and ground suitable for driving the inputs and powering remote switching devices.

Figure 2-7. Position Tracking System

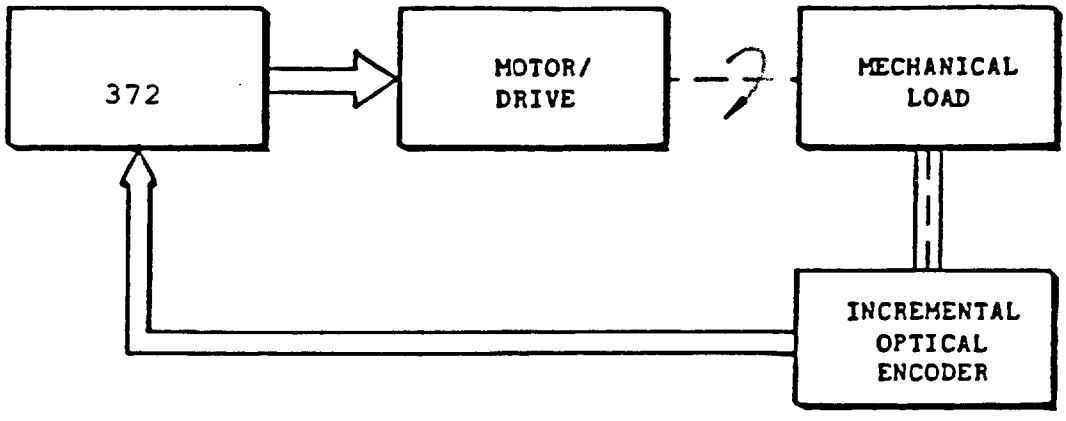
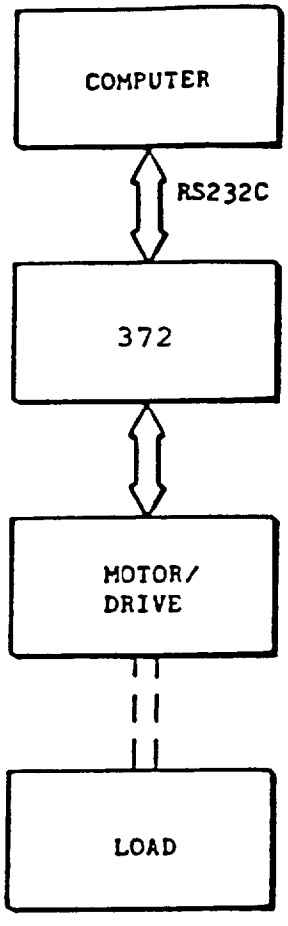


Figure 2-8. Computer-Controlled 372



3. OPERATING MODES

This chapter describes the various modes of operation for the 372 indexer. Specifically, the following features are described:

Basic Motion Control
Non-volatile Memory features
Use of AUXILIARY inputs
Position Tracking

3.1. OVERVIEW

Ultimately, the 372 is designed for operation and control over the RS-232C interface. It is possible for the 372 to move the motor without ever using the interface, by asserting the START input, but to change any of the default modes or parameters, such as the speed or distance, it will be necessary to go "On Line". Refer to Chapter 4 for instructions on how to implement RS-232 communications.

Once interface communications have been established, you can move the motor a preset distance (Preset mode), run at a constant velocity (Continuous mode), or move back and forth between a start position and a specified position (Alternating mode). Additional operations can be performed using a combination or sequence of the basic instructions used to move the motor.

Note
A Stop switch must be connected before movement of the motor is possible. (see Fig. 4-4, pg 55).

3.1.1. RS-232 COMMUNICATIONS

The 372 supports a standard three wire RS-232C interface which allows rapid transmission of configuration and execution instructions. Up to eight units may be connected to a single communication channel for multiple axis applications. A certain amount of operator interaction is available through the indicators and control inputs described below, but the great majority of setup and control operations are implemented using the RS-232 serial interface. For this purpose, the 372 has a straightforward command language which is detailed in Chapter 5.

Initial activation of the interface is covered in Chapter 4.

3.1.2. INDICATORS

Their are three LED indicators on the front of the 372. These are the "MOVING", "ON LINE", and "STATUS" indicators.

MOVING -- this indicator is illuminated while the motor is in motion or the system faults.

ON LINE -- this indicator is illuminated once communications have been enabled. The indexer will ordinarily echo the characters it receives, but will not respond unless enabled. See the "*E*" and "*F*" commands in section 5.5.1. This LED also comes on when the system faults.

STATUS -- this indicator is normally green, and may serve as a power on indicator. If the 372's microprocessor encounter's processing difficulties, an automatic "watchdog" timer will halt indexer operations, and turn this indicator red.

3.1.3. CONTROLS

The 372 has no front panel controls other than four configuration switches. These switches are located just below the status indicator LED's. They are used to establish communications Baud rate, and a unique identification number (unit number or "device address") for each 372 in a multiaxis system. These switches are described in Section 4.5.

Control switches may be connected to various inputs on the AUXILIARY connectors as described in Section 3.4, 3.5, and 4.4.

3.2. BASIC MOTION CONTROL

Once 372 RS-232 communications have been enabled, only a GO command is needed to initiate motor motion. There are two principal indexing modes, the Preset (or "Normal") and the Continuous, for positioning and speed control respectively. The Alternating mode is a third mode, a special case of the Preset mode.

Both the Preset modes require a distance parameter. All three modes require acceleration and velocity parameters. If no parameter is supplied before a GO command is issued, prior parameters, or the 372's default parameters will be used. The required parameter and mode commands are detailed in section 5.2.

3.2.1. MOVE PARAMETERS

For rotary motors, velocity and acceleration parameters are expressed in units of motor revolutions. For linear motors, parameter units involve inches. For accurate speed control, the 372 needs to know what resolution the motor is. Use the *"MR"* command to change motor resolution from the default (25,000 steps/rev) as required. See section 5.2.

The default interpretation of the *"D"* parameter is as a distance to be moved from the current position in motor steps. The *"MPA"* command may be used to switch from this "Incremental" indexing mode to the "Absolute" positioning mode where the *D* parameter represents a position, that is, a distance from the Home position or zero position. In this case, the sign of the parameter indicates which side of the Home position the target location is.

The *"MPI"* command switches back to Incremental positioning. This function may also be implemented using the *"FS"* command. Refer to section 3.6.6 and 3.2.5 for further discussion of the Home position, the Homing function, and Absolute Positioning.

It is desirable in many applications for the distance parameter to be specified in units other than motor steps, such as hundredths of a degree or thousandths of an inch. If the desired units correspond to an integer number of motor steps, the 372's position multiplier capability can be used to convert the *D* command position parameter to those units.

Example: A 25,000 step motor drives a 5 turn per inch leadscrew. The system resolution is 125,000 steps per inch. To have the distance parameter expressed in thousandths of an inch, the operator uses the command *"US125"* to set a distance multiplier of 125. If he then gives a distance command *"D1000"*, the 372 will index 1,000 units of 125 steps each (one inch) on the next GO command. See the *"US"* command, section 5.3.3.

> Note
>
> When scale factors greater than 1 are used, and an indexed move is interrupted by a STOP or by activation of a limit switch, the final position may not be a multiple of the scale factor. Repeatability is affected.

3.2.2. PRESET MODE

This is the default indexing mode. It is selected using the "*MN*" command. The Indexer drives the motor to a specified position at a specified velocity and accelerates and decelerates at a specified rate. No other speed changes are possible until the motor stops. Proceed as follows:

a. Determine the distance to be moved. If the distance is stated in revolutions or degrees, convert the data to number of steps required for the move. This distance parameter is set with the "*D*" command. A "-" sign may precede the specified distance to set the direction of motion (+ = clockwise motion, - = counterclockwise motion, referenced to the motor mounting face.

b. Select an acceleration rate and set this value with the "*A*" command. An easy way to determine the maximum value is to start with a low value, establish the velocity to be used, and then keep increasing the acceleration value until the motor stalls or does not start.

c. Select the velocity to be used and set up this value with the "*V*" command. An easy way to determine the maximum value is to start low and then increase the speed until the motor stalls or does not start.

d. The "*G*" command is used to initiate each move whether or not new parameters are entered. Following this, the motor shaft will rotate to the commanded position. The Indexer will then return to the ready state. This cycle will be repeated each time the "*G*" command is used.

3.2.3. ALTERNATING MODE

Set-up for this mode is identical to that for MODE NORMAL, and the procedures of section 3.2.2 apply to this mode. The difference will be that when the GO command is given, the motor shaft will rotate to the commanded position corresponding to the value set by the *D* command, and then retrace its path back to the start position. The shaft will continually move between the start position and the command position until a Stop command ("*S*") is received. Then the motor will complete the cycle and stop at the start position. The indexer can be configured to stop immediately on command using the "*SS*" command if desired.

Another GO command will repeat the same motion pattern.

This indexing mode is not functional when the Absolute positioning mode is selected (*MPA*). The "Load and Go" function is not operative in the Alternating mode.

3.2.4. CONTINUOUS MODE

This mode requires only acceleration and velocity values to move the motor. Remember that the actual angular velocity depends upon the motor resolution setting of the indexer. In addition, it is possible to switch the 372 to low velocity range with the "*SS*" command. This range increases low speed velocity resolution, while reducing the available top speed, both by a factor of ten.

Once the GO command is given, the motor will rotate at a constant velocity until a STOP command is received, or until a new velocity (and a new acceleration if desired) are commanded.

3.2.5. HOMING FUNCTION

This function may be used to establish a home reference position. The Indexer may be instructed to move the motor to a Home position using either an interface command or the GO HOME input on the AUXILIARY 2 connector. The target Home position is typically established by mounting a load-activated switch so that the switch opens when the load is in a desired reference position. This switch must be connected to the HOME ENABLE input on the AUXILIARY 2 connector.

When an encoder is used, the 372 assumes that the encoder Z Channel, will provide the reference position signal instead. In this case, the Z Channel and HOME ENABLE inputs must both be active to mark the Home position. This is to accommodate rotary encoders, where an index signal occurs every revolution, the HOME ENABLE input may be used in conjunction with the Z Channel input to allow selecting one of many index signals generated by multiple revolutions of the motor and encoder. A switch connected to the HOME ENABLE input is used to locate the general HOME position area, and the Z Channel signal from the encoder is used for final Home positioning.

Note

In "Encoder Step" mode, the Z Channel and HOME ENABLE inputs must both be in the proper state simultaneously to mark the Home position.

When no HOME ENABLE switch is needed, this input should be left unconnected. All references to "Home Limit" below refer to either HOME ENABLE, Z Channel, or both where appropriate.

While seeking the Home position, the indexer will note the motor position when a Home Limit transition (from "false" to "true") is detected. The motor is then decelerated to a stop. The indexer moves back in the opposite direction to a position which it calculates to be .002 rev counterclockwise from the point where the signal was detected.

At this point, if the Home Limit input is false, the indexer will creep in the clockwise direction at 0.1 rev/sec until the signal goes true.

If the input is already true, the indexer stops where it is.

If the Home Limit signal is true when the Go Home instruction is received, the indexer moves the motor .002 rev counterclockwise in an effort to get off the Home position.

If the signal goes false as a result, the indexer will creep in the clockwise direction at 0.1 rev/sec until the signal goes true again.

If the signal remains true in spite of the initial move, the indexer stops.

Note

Because the final approach direction is always clockwise, it is necessary to insure that the final approach starts from the counterclockwise side of the signal.

If there is significant backlash and friction in the system, and the indexer is instructed to Go Home in the clockwise direction, the motor can easily end up on the wrong side of the signal, and execute its final approach in the wrong direction. This problem can also occur if the Go Home speed is high, and the Home limit signal is delayed, by a relay or Programmable Controller for example. In such a situation, it is advisable to initiate homing operations from the clockwise side of home, in the counterclockwise (-) direction. In the case where the Home signal is wide, it will be necessary to initiate the homing operation in the (+) direction from the (-) side of the signal to insure stopping on the signal edge.

When the Homing operation has concluded, the indexer will reset its internal position counter as with the "*PZ*" interface instruction. Indexer position referenced to Home is available using the "*PR*" command.

The GO HOME command has the form GH<direction><velocity>, indicating which direction to move, and at what velocity. For example,

the command *GH-1* will send the motor in the counterclockwise direction at 2 rev/sec in search of the Home signal. The acceleration parameter used for this move is set with the "*GA*" command. The default value is 0.2 rev/sec/sec.

If an end-of-travel limit is activated before Home is found, the controller will reverse direction and attempt to find the Home position again. If the other limit is activated before Home is found, the controller will stop trying to go Home. The controller can indicate whether or not the homing process was successful by means of the "*R*" and "*RC*" status request commands.

The homing operation may be initiated with hardware by asserting the GO HOME input of the AUXILIARY 2 connector when the 372 is Off Line.

Note:

> If the Go Home command input is used to initiate Homing, and the Home Limit signal is true, the .002 rev move described above will be executed repeatedly while both signals are asserted.

Upon completion of a homing operation, both the encoder-referenced position and the motor-step position are automatically set to zero, as with the "*PZ*" instruction.

Figure 3-1 depicts an Absolute coordinate system with Home.

3.2.6. DIRECT VELOCITY MODE

The 372 has an operating mode that allows the user direct access to the pulse-generating (rate-multiplier) circuit that controls the motor step rate. In this mode the operator controls all speed changes and must take care that changes are not too abrupt for the motor to follow, or the motor will stall.

Once this operating mode is entered, the 372 expects four-digit hexadecimal numbers to set the motor speed. The range of values for these numbers is as follows:

Direction:	Clockwise		Counterclockwise	
Speed:	decimal	hex	decimal	hex
Zero	0	0000	32,768	8000
Maximum	32766	7FFD	65,533	FFFD

In the high velocity range, a rate multiplier value of 0001 hex results in an output step rate of 15.625 steps/sec. Larger numbers will produce multiples of this frequency. The low range is ten times slower than the high range. The velocity range is selected by the *SSF*x command (see section 5.4.6).

Note

To change direction, it is necessary to send a zero speed value. This is determined by current direction of travel.

Note that the hex values 7FFE, 7FFF, FFFE, and FFFF are not listed above. These are special function values that do not produce a motor speed value. Instead, they have been supplied as a programmer's tool to provide signals in the middle of the Velocity Profiling operation. Either 7FFF or FFFF will set Programmable Output 1 high, and 7FFE or FFFE will set it low, regardless of direction. These values may be embedded in the velocity stream to generate a signal when the Indexer gets to critical speed values. The 372 commands used to operate in this mode are as follows:

a. Q1 -- Enter SPECIAL INPUT mode

b. Q0 -- Exit SPECIAL INPUT mode

c. RMhhhh, where (h) is a hexadecimal digit; for example,

RM09A3

Table 3-1. Stored Parameters and Configuration data

FUNCTION	OPTIONS	COMMAND (Section)
Indexing Parameters:		
Indexing Mode	Preset, Continuous, Alternate	Ma (5.2.1)
Velocity	.0001 to 20 rev/sec (.2)	V (5.2.2)
Acceleration	.001 to 999 rev/sec/sec (100)	A (5.2.2)
Direction	+ or -, CW or CCW	H (5.2.2)
Distance	0 to ±99,999,999 units (25,000)	D (5.2.2)
Positioning Mode	Incremental or Absolute	MPa (5.2.1)
Limit Acceleration	1 to 999 rev/sec (999)	LA (5.2.2)
Homing Parameters:		
Direction	+ or -, CW or CCW	GH (5.3.1)
Velocity	.0001 to 20 rev/sec (.2)	GA (5.2.1)
Acceleration	.001 to 999 rev/sec/sec (100)	GH (5.3.1)
Jog Parameters:		
Velocity	.0001 to 20 rev/sec (.1)	JA (5.2.1.))
Acceleration	.001 to 999 rev/sec/sec (100)	JV (5.2.1.))
Setup Parameters:		
Position multiplier	1 to 255 (1)	US (5.3.3)
Motor resolution	200 to 50,000 (MR10-25,000)	MR (5.2.3)
Input Mode	1 to 4	IM (5.5.7)
Interface Echo	On or Off	SSA (5.5.4)
Alternate Mode Stop	Immediate or End of Cycle	SSB (5.5.4)
LOAD and GO mode	Enabled or Disabled	SSC (5.5.4)
START during Pause	Ignore or Clear Pause	SSD (5.5.4)
Off Line Direction	Enabled or Disabled	SSE (5.5.4)
Velocity Range	Normal or Low	SSF (5.5.4)
Buffer on Limit	Save or Clear	SSG (5.5.4)
Buffer on Stop	Save or Clear	SSH (5.5.4)
Power up Off Line	Enabled or Disabled	OSA (5.5.5)
Output while Moving	Enabled or Disabled	OSC (5.5.5)
Start Input Repeat	Enabled or Disabled	OSD (5.5.5)
Editor Line width	70 Characters or 40	OSE (5.5.5)
Interface Messages	On or Off	MS (5.5.8)
Encoder Parameters:		
Encoder Resolution	1 to 99,999,999 (4,000)	ER (5.2.3)
Backlash	1 to 99,999,999 (250)	DW (5.2.3)
Step Mode	Encoder or Motor	FSB (5.2.4)
Position Maintenance	Enabled or Disabled	FSC (5.2.4)
Stop on Stall	Enabled or Disabled	FSD (5.2.4)
Output on Stall	Enabled or Disabled	FSE (5.2.4)
Stop on Trigger	Enabled or Disabled	FSF (5.2.4)
Position Loss	Enabled or Disabled	FSG (5.2.4)
Output on Pos. Loss	Enabled or Disabled	FSH (5.2.4)
Correction Gain	15 to 1900 (240)	CG (5.2.4)
Max. Correction Speed	250 to 32,000	MV (5.2.4)
Correction Mode	linear, square, root	CM (5.2.4)

3.3. NON-VOLATILE MEMORY FEATURES

When one or more 372 operations are frequently repeated, it can become tedious transmitting the same instructions to it. The 372 has solid state non-volatile read/write memory and the capability to retain motion parameters and sequences of instructions when the power is off.

3.3.1. STORED PARAMETERS

The term "stored parameters" refers to a set of motion parameter and indexer configuration commands to be used as default values when none have been specified by the operator. If, for example the operator gives the 372 a "GO" command or START input signal, without any velocity, acceleration, distance, or mode commands, the 372 will move the motor according to default values for these motion parameters.

The indexer functions for which parameters are stored are listed in Table 3-1 above along with their associated commands and factory default values. For a discussion of each listed parameter, refer to the discussion of the associated command in Chapter 5. Initial setup is discussed in section 4.5.4.

When the indexer is powered up, or comes up from a reset, these parameters are loaded into working memory. When a command is encountered which alters a parameter, the value in working memory is changed. The stored default value is not affected unless a Save Parameter command is issued with the "*SA*" command, or the special "=" command syntax. This syntax is described in section 5.1.6. The Save Parameter command will store the current value of each of the parameters listed in Table 3-1.

Factory default values may be restored with the "*ZD*" command.

The 372 will transmit all stored parameters in response to the "*DR*" command. See Section 5.4.15 for a discussion of the *DR* command response format.

3.3.2. STORED SEQUENCES

A Sequence consists of one or more interface commands stored in non-volatile memory. Such a stored sequence is assigned an identification number, and may be accessed by instructing the indexer to "execute sequence number n". A total of 48 sequences may be defined.

The longest possible sequence is 245 characters. Longer execution strings are possible if one sequence contains a command to execute another. Sequence memory is allocated in blocks of 32 characters.

In truly dedicated applications where all indexer operations are predetermined, the 372 can be programmed to execute stored sequences automatically when the power is turned on, without any RS-232 input at all. See the "*XP*" command in section 5.6.3 Alternatively, the 372 can be configured to execute selected command sequences in response to AUXILIARY connector inputs as discussed in section 3.4 below.

Saving, reading, erasing, and executing sequences are implemented with other various "*X*" commands discussed in section 5.6.

3.4. AUXILIARY CONNECTOR FIXED FUNCTION INPUTS

Many applications do not need to make unpredictable motion parameter changes or do not otherwise require interactive control of operations. In the case where the indexer only needs to run at a constant speed or to repeat a single motion, no RS-232 control device is required to implement these operations. In fact, the 372 is capable of a complex range of functions which can be executed using AUXILIARY connector inputs alone.

Indexer operations ranging from a single move to a number of complex motion sequences can be selected and executed with no Console device connected. The RS-232 Console device is required only to initially enter desired set-up parameters or sequences. Then sequences may be stored and parameters may be saved in non-volatile memory as defaults for later use Off Line.

Simple move functions can be executed with the dedicated command inputs described below. More complicated functions are possible using programmable inputs and outputs discussed in section 3.5. Refer to section 3.3 for a discussion of non-volatile memory.

3.4.1. LIMIT SWITCH FUNCTION

The Clockwise and Counterclockwise Limit Switch inputs are provided to protect limited travel mechanisms from over-travel. The factory default configuration has these inputs defeated. See the "LD" command, section 5.2.3 on how to enable this function.

Switches are typically mounted close to the end of travel of a limited travel mechanism to provide an emergency stop signal. When the indexer receives a Limit signal corresponding to the direction it is moving the motor, it stops the motor.

The Limit function is not recommended for use in establishing an initial reference position. The Homing function is provided for this.

If the stop is too abrupt for the drive or the load to handle safely, the default deceleration rate can be reduced. See section 5.2.2, the "*LA*" command.

3.4.2. STOP AND START INPUTS

The Stop input always works, whether the 372 is on or off line. This input will generate a controlled stop of any motor motion and will terminate any execution loop or sequence of pending commands. Programmed automatic sequence repeat modes will be aborted.

The Stop input is a negative true input. That means the input signal must be asserted to allow motion. The 372 interprets the absence of a signal as a Stop command. A normally closed Stop switch or jumpers wired from "+5VDC" to "STOP+" and from "STOP-" to "DC Common" are required to allow any execution at all. See Figure 4-4 for an example connection scheme.

If the indexer is executing buffered commands or a sequence when the Stop input is asserted, all pending commands are purged from the command buffer and must be reloaded to resume execution. (This purging action can be defeated with the "*SS*" command.)

When the 372 is off line, the START input is active. In Input Mode 1, the default, 372 will begin a move if it has valid default move parameters. The motion mode will be that which was last saved, or the default incremental Preset mode. Once turned on, the input must be shut off again before a new Start command can be given. The "*SS*" command (see section 5.5.4) can be used to eliminate this requirement, such that the current move will repeat as long as Start is asserted. This command also allows the Start input to clear a Pause condition, as a fifth Trigger input. The Trigger input function is discussed in section 3.4.4 below.

See sections 3.5.3 and 3.5.4 on using the Start input to execute selected sequences in Input Modes 2 amd 3.

3.4.3. HOMING FUNCTION INPUTS

The 372 AUXILIARY 2 connector has two inputs that affect the Homing function. Briefly, the Homing function has the indexer move the motor until the load encounters a switch at the Home position. This switch opens and signals the indexer that the load is at the Home position.

- The HOME ENABLE input is critical, and the Homing function will not work if this input is connected improperly. See Figure 4-4 for sample connections.

The Homing function operates in two different ways depending on whether the 372 is configured to use an encoder or not. If no encoder is used, the HOME ENABLE signal becomes the Home Limit, identifying the Home position. A load activated switch should be mounted and connected so as to drive the input until the load is in position.

If the Encoder Step mode is invoked (with the "*FS*" command), the 372 will use both the encoder Z Channel signal and the HOME ENABLE signal to establish the Home position. When both signals are true at once, the indexer considers this to be the Home Limit.

If no signal other than the encoder Z CHANNEL is required, HOME ENABLE should be left unconnected.

- The GO HOME input is active only when the indexer is off line (in Input Mode 1). The function of this input is otherwise identical to that of the on line "*GH*" command. When this input is asserted, the indexer will seek the Home Limit signal in the direction indicated by the Direction input.

Acceleration and velocity parameters used will be those saved in non-volatile memory, or the factory default parameters. Refer to section 3.2.5 for a discussion of the Homing function. Default parameters are listed in Table 3-1.

3.4.4. TRIGGER INPUTS

The four Trigger inputs differ from the inputs above in that they are passive data type inputs rather than command inputs. These inputs are typically connected to discrete sensors or other signal sources to provide the 372 with information on external conditions. The only time the Trigger inputs have any function is when the 372 is executing buffered instructions.

There are two types of 372 commands that act on Trigger information. The 372 can be programmed to wait until the state of one

3.4.3. HOMING FUNCTION INPUTS

The 372 AUXILIARY 2 connector has two inputs that affect the Homing function. Briefly, the Homing function has the indexer move the motor until the load encounters a switch at the Home position. This switch opens and signals the indexer that the load is at the Home position.

- The HOME ENABLE input is critical, and the Homing function will not work if this input is connected improperly. See Figure 4-4 for sample connections.

The Homing function operates in two different ways depending on whether the 372 is configured to use an encoder or not. If no encoder is used, the HOME ENABLE signal becomes the Home Limit, identifying the Home position. A load activated switch should be mounted and connected so as to drive the input until the load is in position.

If the Encoder Step mode is invoked (with the "*FS*" command), the 372 will use both the encoder Z Channel signal and the HOME ENABLE signal to establish the Home position. When both signals are true at once, the indexer considers this to be the Home Limit.

If no signal other than the encoder Z CHANNEL is required, HOME ENABLE should be left unconnected.

- The GO HOME input is active only when the indexer is off line (in Input Mode 1). The function of this input is otherwise identical to that of the on line "*GH*" command. When this input is asserted, the indexer will seek the Home Limit signal in the direction indicated by the Direction input.

Acceleration and velocity parameters used will be those saved in non-volatile memory, or the factory default parameters. Refer to section 3.2.5 for a discussion of the Homing function. Default parameters are listed in Table 3-1.

3.4.4. TRIGGER INPUTS

The four Trigger inputs differ from the inputs above in that they are passive data type inputs rather than command inputs. These inputs are typically connected to discrete sensors or other signal sources to provide the 372 with information on external conditions. The only time the Trigger inputs have any function is when the 372 is executing buffered instructions.

There are two types of 372 commands that act on Trigger information. The 372 can be programmed to wait until the state of one

or more inputs matches a specified pattern. Also it can skip the command that follows if one or more inputs does not match a specified pattern. Input patterns are specified as high or low, with "1" or "0" in the command syntax.

The "*TR*" command causes a pause in the execution of buffered commands if one or more Trigger inputs are not in the state specified by the command.

Example:

A process uses the 372 to move a grinding wheel over a surface after a hole has been drilled in it.

1. A switch verifies that the material is in place.
2. A photoeye detects the drill bit in case it has broken off in the hole.
3. A thermostat determines whether material temperature is too high.

These sensors are connected to Trigger inputs 1, 2, and 3. This command syntax: " *TR100X G* " translates into English as: "Wait until the material is in place (switch on), the drill bit is gone (photoeye off), and temperature is below threshold (thermostat off). Then go." The "*X*" instructs the 372 to ignore Trigger 4. See section 5.3.4 for more on the *TR* command.

The "*SKE*" and "*SKN*" commands cause the 372 to either ignore or execute the command directly following Skip command if one or more Trigger inputs are not in the state specified by the command. *SKE* (skip if equal) causes a skip when the Trigger inputs match the specified state, *SKN* causes a skip if they don't.

Example:

In the situation above, this expanded command syntax: " *SKE10XX O11 TR100X G* " adds the capability to signal if trouble occurs. This command set translates into English as: "If the material is in place and the drill bit is unbroken, then don't turn on both programmable outputs (skip)"

See section 5.3.6 for more on the *SK* commands.

The state of all four Trigger inputs may be determined any time using the "*uTS*" command. See section 5.4.9.

The Trigger inputs have a parallel function when the Remote Data command syntax is used. In this mode, the four Trigger inputs may be used as the third of three BCD digits of data. This mode is described in section 3.5.5 below.

3.5. PROGRAMMABLE INPUTS AND OUTPUTS

Programmable I/O makes the 372 very flexible in its ability to interact with other pieces of equipment. Outputs can be used to control external devices, or signal other controls. The 372 can wait or decide what to do next based on the state of its inputs.

There are 12 programmable inputs which assume different functions depending which of four different Input Modes is active. In brief, the four different Input Modes provide the following:

Input Mode 1: Discrete command inputs for Manual Jog, Remote Direction, Motor Shutdown, and START for the "default move", off line.

Input Mode 2: Command inputs for Manual Jog and six sequence select inputs which allow off line execution of stored sequences (numbers one through six).

Input Mode 3: Command inputs for Manual Jog and six BCD sequence select inputs which allow off line execution of up to thirty-nine stored sequences by number.

Input Mode 4: Use of the inputs exclusively for entry of up to three digits of BCD data to be read into buffered instructions during execution using the special "()" command syntax for Remote Data entry.

Eleven of the twelve 12 inputs are located on the AUXILIARY 1 connector, the twelfth on AUXILIARY 2. The function of each input in each Input Mode is listed in Table 3-2. Underlined entries refer to input assignments that may change when the Input Mode command is used. Input Mode may be selected while on line with the "*IM*" command. The default Input Mode can be saved with the "*SA*" command or the "*=IM*" syntax.

Input Mode four is a special case. This mode comes into play during the execution of buffered instructions that use the Remote Data command syntax. This syntax allows command data to be read from BCD inputs at execution time. Remote Data syntax is described in section 5.1.6.

The "*IM4*" command instructs the 372 NOT to interpret the programmable inputs as anything other than passive BCD data. This is to prevent BCD thumbwheel switches from inadvertantly causing the motor to Jog or Shutdown, for example. Operation of the Trigger inputs is not affected.

The four Input Modes are described further in the sections below.

3.5.1. PROGRAMMABLE OUTPUTS

Two outputs are provided on the AUXILIARY 2 connector. They are both programmable. They are controlled using the "*O*" command. These outputs are typically programmed to switch on and off during sequential operations, to control process related devices or signal external controllers. They can drive a solid state relay for energizing solenoids and contactors, or they can directly drive optically isolated inputs and TTL inputs.

Programmable Output 1 may be assigned to reflect the state of certain Position Tracking functions via the "*FS*" command. This function overrides the *O* command. See sections 3.6 and 5.3.6.

The factory default condition for Programmable Output 2 is to function as a "Moving/Not Moving" indicator for motor activity like the MOVING indicator on the front of the unit. This function can be switched on and off via the "*OS*" command. If the function is enabled, the output will source current at 3 to 5 volts (high), when the motor is stopped, and sink current at 0 to 1 volt (low) when the motor is moving. This function co-exists with the *O* command; the output will change in response to either condition.

Note

It should be noted that outputs go on when unit faults or times out.

See Table 1-1 for output electrical specifications.

3.5.2. DEFAULT INPUTS: INPUT MODE 1

This is the default input mode. The input functions in the first column of Table 3-2 below will be active when the 372 is Off Line. Note that the ON LINE indicator LED on the front of the unit will be off to reflect this condition.

JOG CW, CCW

These two inputs allow manual movement of the motor without the need to establish interface communications. This function is useful for initial positioning or setup. An active JOG signal will cause the indexer to move the motor using the current Jog Velocity and Acceleration parameters which are set using the "*JV*" and "*JA*" commands. These parameters may be saved in non-volatile memory. The direction of motion is determined by which input is used. Asserting the "JOG CW" input yields clockwise rotation.

Table 3-2. Programmable Inputs and the Input Modes

AUXILIARY 1

pins	MODE 1	MODE 2	MODE 3	MODE 4
3+,15-	SHUTDOWN	SEQUENCE 6	BCD-2 2	BCD-2 2
4-,16+	JOG CCW	JOG CCW	JOG CCW	BCD-2 4
5+,17-	JOG CW	JOG CW	JOG CW	BCD-2 8
6-,18+	DIRECTION	SEQUENCE 1	BCD-1 1	BCD-1 1
7+,19-	RESERVED 1	SEQUENCE 2	BCD-1 2	BCD-1 2
8-,20+	RESERVED 2	SEQUENCE 3	BCD-1 4	BCD-1 4
9+,21-	RESERVED 3	SEQUENCE 4	BCD-1 8	BCD-1 8
10-,22+	TRIGGER 1	TRIGGER 1	TRIGGER 1	BCD-3 1 *
11+,23-	TRIGGER 2	TRIGGER 2	TRIGGER 2	BCD-3 2 *
12-,24+	TRIGGER 3	TRIGGER 3	TRIGGER 3	BCD-3 4 *
13+,25-	TRIGGER 4	TRIGGER 4	TRIGGER 4	BCD-3 8 *

AUXILIARY 2

pins	MODE 1	MODE 2	MODE 3	MODE 4
1+,14-	START	START	START	START
2	CCW LIMIT	CCW LIMIT	CCW LIMIT	CCW LIMIT
3	CW LIMIT	CW LIMIT	CW LIMIT	CW LIMIT
4+,17-	STOP	STOP	STOP	STOP
5-,18+	HOME ENABLE	HOME ENABLE	HOME ENABLE	HOME ENABLE
6-,19+	GO HOME	GO HOME	GO HOME	GO HOME
7	OUTPUT #1	OUTPUT #1	OUTPUT #1	OUTPUT #1
8	OUTPUT #2	OUTPUT #2	OUTPUT #2	OUTPUT #2
9+,22-	RESERVED 4	SEQUENCE 5	BCD-2 1	BCD-2 1

NOTE: Except for Limit inputs, all inputs require current flow from positive to negative terminals to turn on.

Underlined entries are programmable inputs which may change function in different Input Modes.

* Operation of Trigger input commands is unaffected by the selection of Input Mode 4. Use of these inputs as the third BCD digit for Remote Data is possible in any Input Mode.

DIRECTION

This input allows remote control of motor direction for the Off-Line motion. Direction control applies to the Homing function and execution of the default move, when the GO HOME or START input is asserted. These two direction parameters are ordinarily internal as are velocity and acceleration. The "*SS*" command is used to allow external direction control in place of internal parameters. When this input is enabled and ON, the direction will be "+" (CW), otherwise "-" (CCW).

SHUTDOWN

This input can be used to remove power (remove torque) from the motor. It has the same function as the On Line "*ST*" command. While this input is asserted, and the motor is shut down, the 372 will not respond to Start or Go Home input commands.

Note:
This function can be used only when the motor is stopped. When the motor is shut down, it does not produce holding torque and the precise position of the shaft may be lost.

The "*RC*" command indicates whether or not the Shutdown output is on. See section 5.4.8.

3.5.3. DISCRETE SEQUENCE SELECT: INPUT MODE 2

If Input Mode 2 is selected with the "*IM2*" command, the operator may use six of the programmable inputs to select any of six stored functions (sequences). The JOG input function is retained, but three other Input Mode 1 functions are replaced by the sequence select function. If the operator has programmed from one to six sequences, (and assigned them numbers one through six) any of the six can be executed while Off Line.

This input mode uses START as the command input. When the 372 receives a START signal Off Line, it checks the designated inputs for an active select input.

If the numbered input corresponds to a stored sequence, the sequence will be executed.

If the select inputs are off, the default move is executed.

If neither input condition is true, the 372 takes no action.

Programmable inputs are listed in Table 3-2 above. Sequence programming is discussed in section 5.6.

3.5.4. BCD SEQUENCE SELECT: INPUT MODE 3

In the case where six sequences (Input Mode 2) are not enough to meet the application requirement, the same six inputs may be reconfigured to function as BCD inputs for numerical selection of any of up to 39 stored functions. Table 3-2 shows the assignment of each input for this interpretation. The notation "BCD-1 4" implies that the input is one of the four inputs for the first (least significant) BCD digit, and has the BCD code value 4.

This input mode uses START as the command input. When the 372 receives a START signal (Off Line), it checks the designated BCD inputs for a valid sequence number.

If the BCD number present corresponds to a stored sequence, it will be executed.

If the BCD input is zero, the default move will be executed.

If neither input condition is true, the 372 takes no action.

Refer to section 5.6 for details on sequence programming, refer to Appendix C for a sample BCD wiring diagram.

3.5.5. OFF LINE LOCKOUT: INPUT MODE 4

This mode is intended for use when the 372 is using the Remote Data command syntax. The Remote Data feature allows entry of motion control parameter data by external devices, such as thumbwheels or programmable logic controllers. See section 5.1.6.

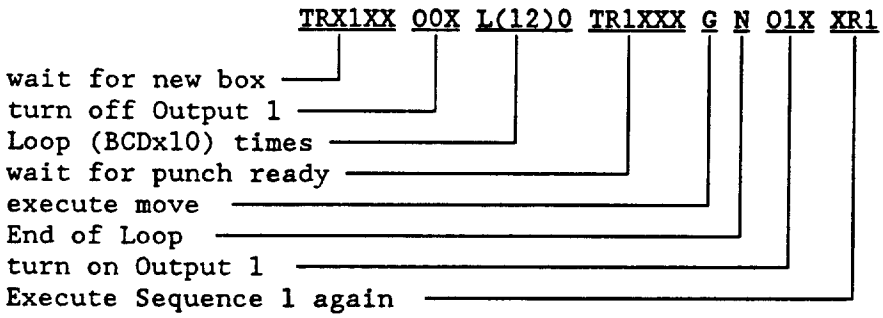
In Input Mode 4, all twelve programmable inputs serve as BCD inputs and have no other Off Line function. Specifically, the command input functions like Jog and Shutdown are disabled.
Input Mode 4 is provided to prevent inputs which have command functions from conflicting with the Remote Data function when the 372 goes Off Line. On Line, the Trigger commands may be used without any conflict.

Note that the *XD* and *XT* commands in the above example are programming commands used to load the sequence. They are only used once, and are not part of the sequence itself.

As the manufacturer's business expands, he must provide widgets in larger boxes. He programs Sequence 2 and 3 with a different loop instruction for 250 and 500 piece boxes. New connections are made to the SEQUENCE 2 and SEQUENCE 3 inputs.

Soon widgets are so popular that the manufacturer must supply them in both retail and bulk quantities. He has so many sequences for the different sizes of widget containers that he must replace his six Sequence select input lines with a pair of BCD thumbwheel switches to select box size. The 372 is programmed for Input Mode 3. He is now able to select 39 different size boxes, but partial shipments are still a problem.

Ultimately the manufacturer wants to be able to dial in the number of widgets per box. This calls for using Input Mode 4 which allows direct BCD data input. Two thumbwheels are used to replace the two upper digits of the Loop command data. This allows from 10 to 990 widgets per box. Once the thumbwheel switches are wired up, he has no inputs left to start the sequence. He uses the automatic run feature, with an "*XP1*" command, to run Sequence 1 on power up. The "new box" sensor is now connected to the TRIGGER 2 input. The 372 is changed to Input Mode 4 to avoid input confusion. The sequence is as follows:



In the Loop command above, the number of loop repeats will be read directly from the BCD inputs for digits 1 and 2 each time the sequence is executed. The zero after the parentheses will cause the BCD number to be multiplied by ten.

3.6. POSITION TRACKING

Position Tracking is Compumotor's closed-loop positioning system. Several parameters must be defined to enable closed loop operation. These are itemized in Encoder Installation section 4.8. All commands dealing with the encoder interface are detailed in section 5.2.6.

The Position Tracking function provides any or all of the following capabilities:

a. Increased resistance to external forces.
b. Compensation for friction and gear play.
c. A means to independently determine load position.
d. Stall detection.

Once the system is installed and operating, no further operator action is required.

The encoder may be used as part of a closed loop system or as an independent means to verify position. The encoder may also be used to detect whether a stall has occurred during a move.
The encoder functions include:

a. Encoder referenced position
b. Encoder position servo
c. Motor stall detection
d. Position loss detection
e. Multiple axis stop
f. Absolute position coordinates
g. Homing function

Implementation of the encoder functions listed in a through e above requires connection of the encoder and issuing the appropriate configuration commands.

3.6.1. ENCODER REFERENCED POSITION

In the factory configuration, the 372 operates using position parameters in units of motor steps. With the Position Tracking option the parameters may be in units of encoder steps. In this mode, when moving a specified distance, the Indexer effectively sends out steps to the motor until the correct number of encoder steps is returned.

Position information provided by the Indexer may be in units of either encoder steps or motor steps. The default power-on configuration is motor steps, but can be programmed to encoder steps. The ("*FS*") command allows switching between modes. The

absolute position is always maintained both in encoder steps and motor steps in either mode. The user may switch from one mode to the other between moves but not during moves. Position data will be reported in the same format.

Refer to section 5.2.6 regarding the functions of the *FS* command.

3.6.2. ENCODER POSITION SERVO

Although Encoder Step Mode provides that a Preset move ends up in the right position, when the move is complete, the encoder is no longer monitored. External forces can move the load, and the indexer will not take any corrective action.

The Indexer may be configured to resist any external change in its position when at rest. Then, if external forces cause the encoder position to change, the Indexer will attempt to correct.

This Position Maintenance function is also selected via the "*FS*" command. When selected, the system will attempt to maintain the current encoder step reference position. The reference position is established at the end of a move or whenever this Servo function is turned ON.

Position Maintenance is effective only while in ENCODER STEP mode. After a move, if it is determined that the encoder position is not changing in spite of steps sent to the motor, this function is automatically disabled.

Position Maintenance Parameters

The "*FS*" command is used to invoke the POSITION MAINTENANCE mode. In this mode, the Indexer repeatedly compares the actual encoder position with the desired position while the motor is stopped. If an error is detected, the magnitude of the error determines the magnitude of the correction response.
The correction response calculation yields a correction velocity, and the nature of the servo is to drive until the error (and correction velocity) is zero.
There are three options for how the error is used to calculate the correction velocity:

Linear:	f(e) = e
Square root:	f(e) = square root of e
Square:	f(e) = e squared

These are selected using the "*CM*" command. The linear correction mode is the default operating mode. Linear correction calculation is suitable for the majority of applications. See section 5.2.6.

The formula for generating the correction velocity is as follows:

V=K(f(e)), V less than or equal to Vmax

V = correction velocity in motor steps/sec
K = correction gain constant in steps/sec/encoder step
f = correction mode formula
e = error in encoder steps
Vmax = maximum correction velocity

The Correction Gain parameter determines the magnitude of response to a given position error. The larger the Gain value, the higher the correction velocity used. The numerical value of the Gain ranges from 0 to 255. The default value is 16. The "*CG*" command is used to set the Gain parameter. See section 5.2.6.

The Maximum Correction Velocity parameter puts an upper limit on servo response speed. This limit ranges from 250 to 32,000 steps per second. The "*MV*" command is used to establish this parameter. See section 5.2.6.

Lower encoder resolution requires higher gain. Load inertia also affects gain selection. The higher the load inertia, the lower the gain must be.

As with any servo, if the gain and maximum correction velocity are set too high, the system will overshoot, and oscillation about the desired point will result. When the gain is too low, the system responds slowly. The ratio of motor pulses to encoder pulses has a significant impact on the performance of the system.

An empirical approach to optimizing these parameters calls for first estimating an appropriate maximum correction velocity. This value depends on the motor and transmission resolution. A much lower value is required for a 200 step motor than for a 25,000 step/rev motor.

Next, the gain is established by setting it at a low value, then increasing it to the point where the system begins to show signs of overshoot. Then reduce it a small amount.

3.6.3. MOTOR STALL DETECTION

When the motor is being moved, the Indexer repeatedly compares the number of encoder steps coming in against the number of motor steps being sent out to verify that no gross discrepancy exists as in the case of a motor stall. If the encoder position deviates

excessively from the desired position in the course of a move, the Indexer assumes the motor has stalled.

Several configuration options pertain to this feature. These include setting a "window" of allowable deviation to account for mechanical backlash, terminating the move if a stall is detected and sending an output signal (see section 5.2.6).

> Note
> Each comparison is independent of the prior one, so that if an allowable error exists every time the comparison is made, no stall is detected. Such a condition might occur, for example, if there is minor slipping in the motor or encoder coupling.

a. The Deadband Window parameter is set by command. The *DW* command allows precise deadband specification in motor pulses. The deadband compensation feature allows mechanical systems with backlash in the geartrain to use the stall detect and position loss detection features.

   When a nonzero deadband is selected, the position-loss detect and stall detect will not be active until the error exceeds the deadband (plus half a motor pole; .01 revolutions).

b. The Stop on Stall function is selected via the "*FS*" command. If selected, this function will cause termination of a move when a stall is detected. The move will stop as soon as the stall is detected, without any deceleration. This function works either in MOTOR STEP or ENCODER STEP mode.

c. The Output on Stall function is also selected via the "*FS*" command. If selected, the 372 programmable Output 1 will go low when a stall is detected and remain low until a new move begins.

   If this function is selected, the normal function of Programmable Output 1 is inhibited.

3.6.4. POSITION LOSS DETECTION

The Position Loss Detection function is also invoked by the "*FS*" command. If selected, the system will compare the motor step position and encoder step position at the end of each move. The comparison will also take place following execution of the "*K*" command. The Indexer will not be "ready" until this comparison is complete. The comparison, which may take 20 to 30 milliseconds, takes into account the selected encoder resolution, motor resolution, and Deadband Window.

The comparison will indicate a position loss if the mechanical and electrical synchronization have been lost. The result may be requested via a Status Request command. This function supplements the Stall Detection function by indicating any slippage that may have occurred short of a full stall. The Position Loss Detection function works in both Encoder or Motor Step modes.

Two configuration options pertain to this feature: setting the Deadband Window, as discussed above, and sending an output signal.

The Output on Position Loss function is selected with the "*FS*" command. If selected, the 372 Programmable Output 1 will go low when a stall or positioning failure is detected and remain low until a new move begins. If this function is selected, the normal function of the programmable output is inhibited.

3.6.5. MULTIPLE AXIS STOP

In a multi-axis 372 system, the operator may wish to have all axes stop motion if a stall or position loss is detected on any one axis. This is possible using the KILL on TRIGGER 3 function. This function is also selected with the "*FS*" command.

If selected, a low signal on the TRIGGER 3 input will cause immediate termination of the move. By connecting Programmable Output 1 from one 372 to TRIGGER 3 on the next, this function and the output signal functions above will allow all axes in a system to stop at once. If Output on Stall is enabled, the Trigger 3 input signal will result not only in a cessation of motion, but also in an output signal to the next indexer.

Multiple units should be connected in a "circle", with Output 1 from each unit going to Trigger 3 on the next.

3.6.6. ABSOLUTE POSITION COORDINATES

In the factory configuration, 372 indexing is incremental, where the operator specifies the distance to be travelled from the current position. The 372 also has the capability for absolute indexing, where the operator specifies a destination position, referenced to a zero or Home position.

The ABSOLUTE POSITIONING mode may be selected with either the "*FSA*" or "*MPA*" commands. The Home position is used as a zero reference for absolute positioning. The homing function is normally used to establish the absolute zero position. Alternatively, the "*PZ*" command may be used to establish the current position as the zero position.

In absolute mode, the "*PR*" instruction may be used to report absolute position in either Motor or Encoder Step Mode. The "*X1*" and "*X2*" commands also report absolute position, but these reports may be referenced to a different zero position.
If separate coordinate systems are not desirable, the "*X0*" instruction must be issued at the same time as the "*PZ*" instruction, or following a Homing operation where the "*PZ*" function is automatic. Otherwise, only the position information yielded by the "*PR*" instruction will correspond to the absolute position commanded with the "*D*" instruction.

Figure 3-1 depicts an Absolute coordinate system with Home.

Figure 3-1. Absolute coordinate system

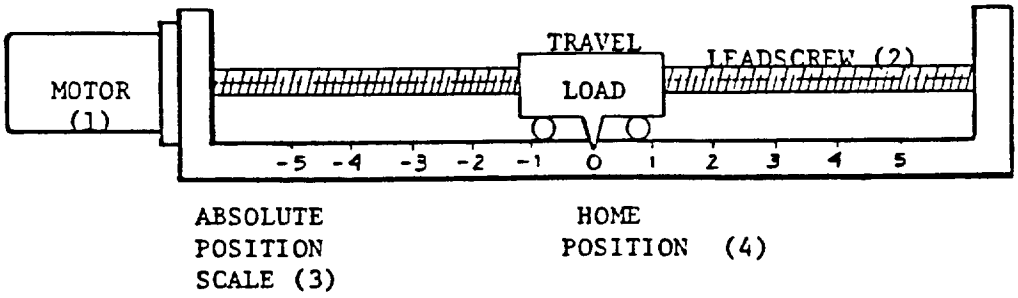
(1) The motor turns the leadscrew, 25,000 steps per revolution

(2) The leadscrew pitch is 4 revolutions per inch

(3) Absolute Positions are; then in inches from HOME, 100,000 motor steps per inch

(4) The Indexer must be notified as to the location of the Home position on the system. If a switch is located there, the Indexer homing function can establish this zero position.

TYPICAL LIMITED TRAVEL APPLICATION



4. INSTALLATION

This chapter describes the installation procedures for 372 Indexers including the RS-232C interface.

4.1. INSPECTION

Carefully inspect the shipping carton for any evidence of physical abuse or damage. Report any such findings immediately to your Receiving Department and to the carrier. Compumotor Corporation cannot be responsible for in-transit damage.

Unpack the shipping carton and inspect the Model 372 Indexer for any damage, cracks, broken parts, or damaged cables. Save the packing materials until functional checks have been completed.

The following items should be included in the packing case with the 372:

1. Operator's Manual	p/n 88-006315-0x
2. Power cord	p/n 71-006269-01
3. Driver cable	p/n 71-004832-10
4. Mounting bracket (2)	p/n 51-006007-01
5. No.6 Mounting screws (4)	p/n 51-006037-01
6. 25 pin "D" connector (3)	p/n 43-001989-01
7. Connector shell (3)	p/n 43-001990-01

4.2. PHYSICAL ASSEMBLY

The 372 is delivered as complete, self-contained unit. No assembly of the these units other than connections and mounting is required.

Mounting may call for attachment of the supplied mounting ears. These screw on with the supplied screws in either of two physical orientations. The unit should be mounted vertically for best convection cooling. See figure 4-1.

Setting the unit identification number and Baud rate switches are the only external adjustments. See section 4.5.

The internal adjustments include configuration jumpers, notably the one for configuration of the unit to accept 240 VAC Power instead of the standard 120 VAC. See section 4.3.2 for voltage selection. Internal jumpers on the main circuit board allow step and direction output configuration, and Z Channel polarity switching. See Appendix A.

Figure 4-1. Mounting Orientation

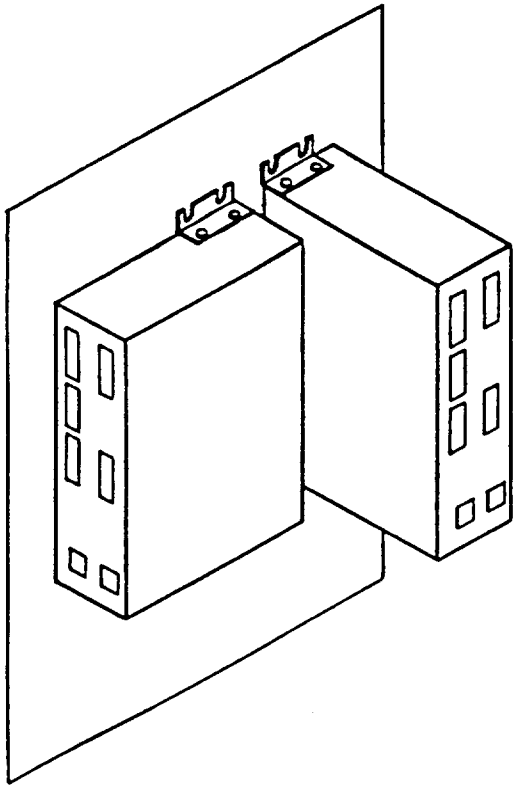
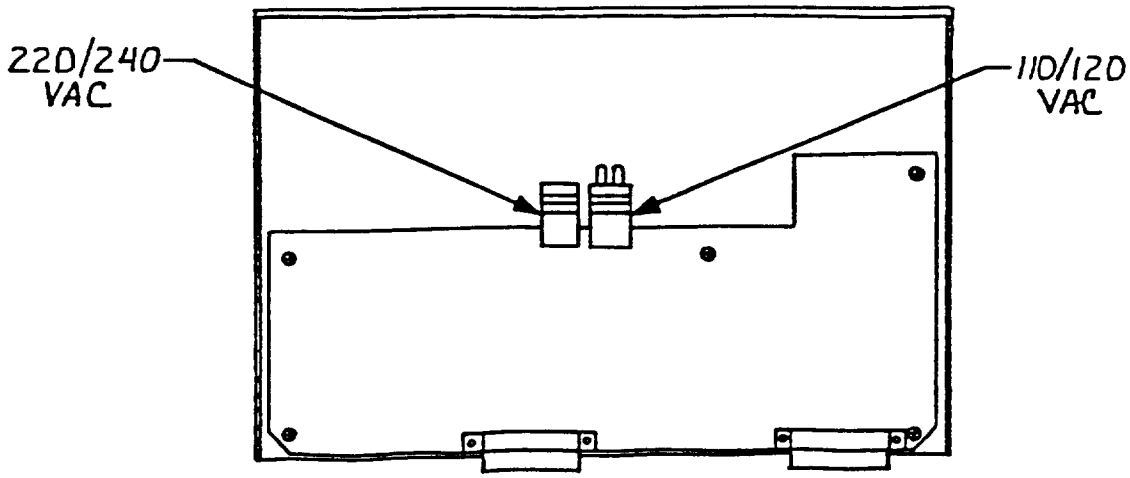


Figure 4-2: 120/240 VAC Power Connectors



4.3. INSTALLATION

All 372 connections except the Power use standard 25-Pin "D" connectors (Canon DB25P or equivalent). The Power connector is a screw terminal connector which can be unplugged.

To operate the 372 for the first time, it is necessary to:

1. Provide power
2. Connect a motor and drive
3. Connect the RS-232 Console device

4.3.1. QUICK START

If the available power is 115 to 120 VAC, and a 25,000 step Compumotor motor/drive is on hand, the system can be rapidly connected with supplied cables for initial testing.

The required items not specifically provided by Compumotor are a 9600 Baud Full duplex Console device, and a standard RS-232 cable. If these items are available, connections can be made as follows:

a. Connect the 10-foot cable supplied with the 372 from the MOTOR/DRIVER Connector to the input connector of the Drive. The motor must also be connected.

b. Connect the Console device to the 372 RS-232 connector.

c. Connect the AC power cables for all three devices to a properly rated AC outlet.

Now it is possible to proceed directly to section 4.6.3. If any problems are encountered, the following step by step approach will be required.

4.3.2. CONNECTING POWER

The AC Power connector is a detachable screw terminal connector designed to simplify installation and replacement. The supplied power cord is suitable for 120 Volt installations where outlets are available.

If 240 VAC power is to be used, or power is to be directly wired to the unit, the supplied cord may be removed from the screw terminals and replaced with facility wiring.

Standard color power wires should be connected to 372 terminals as follows:

Power terminal	Wire color	(European)
Line	BLACK	(BROWN)
Neutral	WHITE	(BLUE)
Gnd	GREEN	(GREEN with YELLOW)

For 240 VAC installations, it is necessary to remove the side cover of the 372 to change the fuse, and relocate the voltage select plug from the 120 Volt to the 240 Volt connector as shown in Figure 4-2 above. Fuse requirements are as follows:

Line voltage	Fuse rating
110-120 VAC	1/4 A.
220-240 VAC	1/8 A.

4.3.3. CONNECTING THE DRIVE AND MOTOR

Compumotor supplies a cable with the 372 which will mate with any Compumotor drive. Connecting the indexer to the drive merely calls for plugging the cable into appropriately marked connectors on each device.

> CAUTION: Indexers, Drives, and Console devices all have multiple connectors of the same type. Be sure that each cable connects to the appropriately marked connector at both ends. If installed cables will be hidden, LABEL BOTH ENDS before installing.

> **THE 372 MOTOR/DRIVER CABLE IS NOT SUITABLE FOR USE AS AN RS-232 CABLE.**

The 372 MOTOR/DRIVER connector has multiple outputs suitable for controlling other manufacturers equipment. The outputs used by Compumotor drives are TTL level STEP and DIRECTION signals with ample current capability to drive the optical isolation circuits common to motor drive electronics. These outputs can be internally switched for differential or single-ended configuration. Refer to Appendix A.

The alternative outputs are open collector clockwise (+) step and counterclockwise (-) step. These outputs are active low, and will sink 250 mA.

Refer to Table 4-1 for the MOTOR/DRIVER connector pin assignment.

Table 4-1. Motor/Driver Connection Pinout

1. Step +	14. Step Return
2. Direction +	15. Direction Return
3. CW Step (open collector)	16. Drive Shutdown +
4. CCW Step (open collector)	17. Drive Shutdown Return
5. Shield	18. Reserved output +
6. Reserved output Return	19. Fault Reset Return
7. Fault Reset +	20. DC Common
8. + 5 VDC	21. Drive Fault common
9. Drive Fault input +	22. Slip Fault common
10. Slip Fault input +	23. Boost Return
11. Boost output +	24. DC Common
12. DC Common	25. NC
13. NC	

4.4. AUXILIARY CONNECTORS

Each AUXILIARY connector input (except the CW and CCW Limit inputs) have both "+" and "-" terminals. These optically isolated inputs require that current be driven from the + to the - terminal to assert the input signal. A typical circuit has the "+5VDC" power terminal connected to the + terminal of the desired input, and the - terminal is wired to a remote switching device. The remote switch returns to the "DC Common" terminal. A switch in this configuration, capable of conducting 15 mA, is adequate to drive any input. See Figure 4-3 for a typical AUXILIARY control signal wiring diagram.

External sources of DC current may be used to drive the inputs as long as current flow is limited to 50 mA. Series resistance must be added for voltages above 10VDC. Twenty-five to fifty ohms per volt above 10 volts is adequate; for 12VDC, add 50 to 100 ohms in series, for 24VDC add 600 to 1200 ohms.

DO NOT CONNECT VOLTAGES ABOVE 48 VDC.

The connector pinouts are shown in Table 4-1 and Appendix D. The limit switches are discussed in section 4.5 below. The Trigger inputs are discussed in section 5.3.4. The Programmable Outputs are described in section 5.5.5.

Refer to section 3.3 for a discussion of the following functions:

REMOTE START and STOP inputs; GO HOME input; HOME ENABLE input; and MOTOR SHUTDOWN input;

4.5 LIMIT SWITCHES

Limit switches are connected to the AUXILIARY 2 connector. Pinouts for this connector are shown in Table 4-2.

The *LD3* command is issued at the factory to defeat the limits for testing. If initial systems tests are to be performed with the motor connected to a limited-travel mechanism, it is advisable that the limits be enabled (*LD0*), and proper limit switches be installed to protect the mechanism. The motor itself can be stalled indefinitely without damage.

Typical switches include mechanical contact closures, optical switches, or Hall-effect devices. The easiest to mount and wire are typically the mechanical-type switches.

Disconnect AC power to the motor/drive and rotate the motor by hand to determine in which direction the mechanism moves for clockwise rotation of the motor shaft. The sense of CW and CCW motor rotation is determined as if the observer were looking at the motor from the mounting flange end. (See Figure 4-3.) Install the CW Limit switch at the far end of CW travel.

Install the limit switches so that there is room to stop. Actual limit stopping distance depends on motor speed and the Limit Acceleration parameter (actually deceleration) set with the "*LA*" command. The Limit deceleration rate should be decreased if heavy loads and high speeds are anticipated, to guarantee that the load does not overrun the motor. See section 5.2.2.

Limit switch inputs are optically isolated and internally powered and require only a contact closure.

Wire the Common (C) terminal to Pin 16 of the AUXILIARY 2 connector (25 Pin "D" Canon DB25P or equivalent) and the normally closed (NC) terminal to Pin 4 (CW Limit).

Place and mount the CCW Limit Switch in a similar fashion, wiring the Common terminal to Pin 15 (GND) and the NC terminal to Pin 2 (CCW Limit).

Figure 4-3. Determining Direction of Rotation

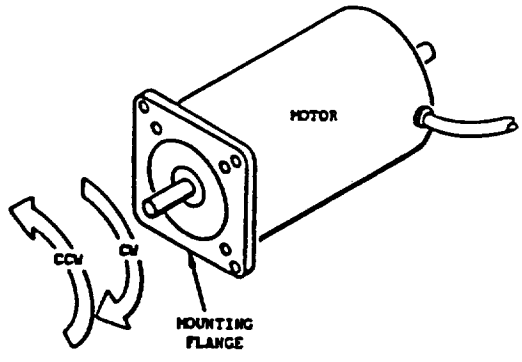


Table 4-2. Auxiliary Connector Pinouts

AUXILIARY 2:

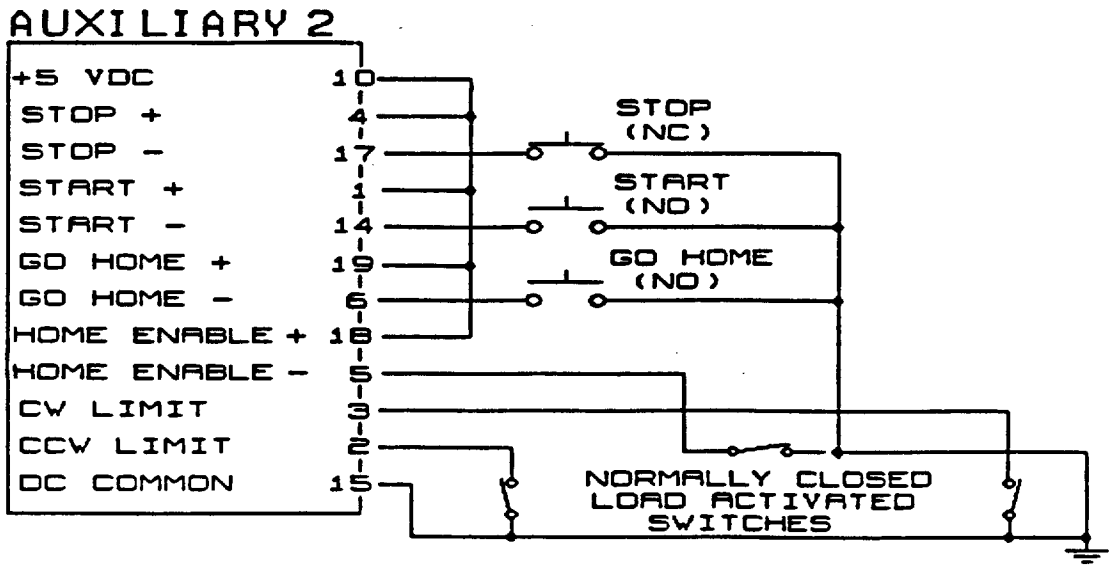
1. START +
2. CCW Limit input -
3. CW Limit input -
4. STOP +
5. Home Enable -
6. Go Home -
7. Programmable output #2
8. Programmable output #1
9. Reserved input 4 + *
10. +5 VDC
11. Reserved input
12. Reserved input
13. DC common
14. START -
15. DC common
16. DC common
17. STOP -
18. Home Enable +
19. Go Home +
20. DC common
21. DC common
22. Reserved input 4 - *
23. NC
24. Reserved input
25. Reserved input

AUXILIARY 1:

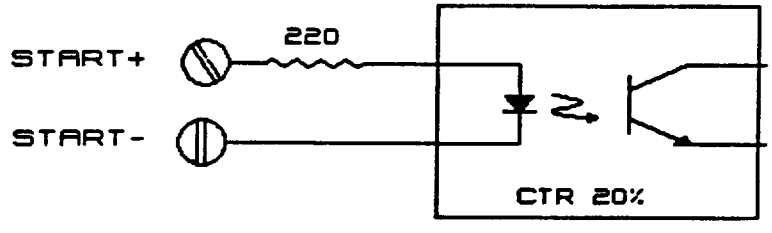
1. +5 VDC
2. DC common
3. Shutdown + *
4. Jog CCW - *
5. Jog CW + *
6. Direction - *
7. Reserved input 1 + *
8. Reserved input 2 - *
9. Reserved input 3 + *
10. Trigger 1 - *
11. Trigger 2 + *
12. Trigger 3 - *
13. Trigger 4 + *
14. DC common
15. Shutdown - *
16. Jog CCW + *
17. Jog CW - *
18. Direction + *
19. Reserved input 1 *
20. Reserved input 2 + *
21. Reserved input 3 - *
22. Trigger 1 + *
23. Trigger 2 - *
24. Trigger 3 + *
25. Trigger 4 - *

* Marked inputs have alternate functions in different Input Modes. Refer to Table 3-2, Section 3.5, or Appendix D.

Figure 4-4. Typical AUXILIARY Connection



TYPICAL INPUT CIRCUIT



4.5. RS-232C COMMUNICATIONS INTERFACE

The RS-232 interface is a widely used technique to transmit information with a minimum of wiring. Alphanumeric characters are transmitted and received one at a time in the form of a binary character code. Transmitter and receiver must use the same signal timing ("Baud rate"), and coding scheme (ASCII code) for communications to be successful.

4.5.1. RS-232C CABLE

A three-wire cable is required for the 372 to transmit and receive signals. The use of a shielded cable is recommended. The connection pinouts are shown in Figure 4-5.

Some terminals and computers may require additional "handshaking" signals not supplied by the 372. A typical DCE connection is shown in Table 4-3. A simple null DCE connector is shown in Figure 4-6. Most terminals and personal computers are in the DTE configuration (data terminal equipment). The 372 is configured as DCE to mate with these.

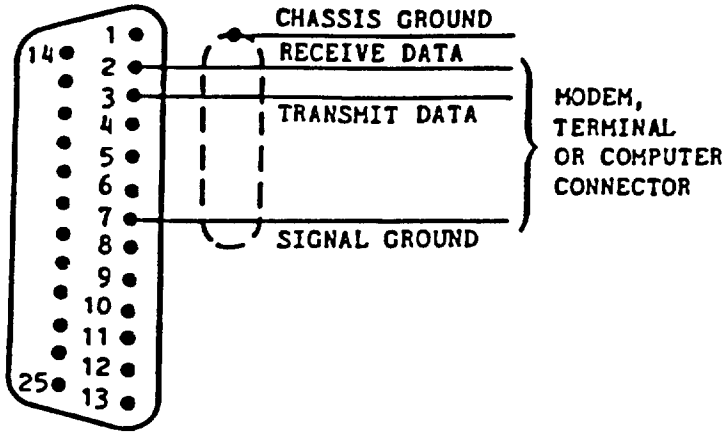
If the control device is also DCE, such as Compumotor's Model 52, it will be necessary to exchange the transmit and receive connections in the cable. Verify that you have the proper signals.

Each unit's transmit output must be connected to the other's receive input. These signals may be identified with a voltmeter if the configuration is unknown. The unconnected Transmit output will ordinarily sit at -5 to -12 Volts from Ground, the unconnected Receive input will be close to zero Volts.

You may have to interconnect some of the 25 pins of the RS-232C "D" connector at the terminal end of the cable to establish the proper signals. The 372 does not provide signals on any pins other than 2, 3, and 7 of the RS-232C interface connector.

For multiple axis connections, "daisy-chained" cabling is required: refer to Figure 4-7. In this configuration, every character sent by the Console device passes through every indexer in the chain. It is necessary to give each unit its own Unit Number for identification purposes. See section 4.5.3 below.

Figure 4-2. RS-232C Cable



NOTE: 1. CONNECT SHIELD TO PIN 1 ON INDEXER SIDE OF CABLE. CONNECT OTHER END OF SHIELD TO TERMINAL, MODEM OR COMPUTER EARTH GROUND. NO INTERNAL INDEXER CONNECTION TO PIN 1.

2. MAXIMUM LENGTH OF CABLE PER THE EIA RS-232C STANDARD IS 50 FT.

3. VOLTAGE LEVELS PER THE EIA STANDARD ARE 5 TO 15 VOLTS IN MAGNITUDE. EXPECTED LEVELS FOR THE INDEXER ARE 12 VOLTS.

4. INDEXER USES 8251A UART.

5. CONNECTOR IS 25 PIN "D" CONNECTOR (DB25P)

6. VIEW LOOKING AT CABLE CONNECTOR AT INDEXER END

Table 4-3. Typical DCE Connection Pinouts

25 Pin Connector Pin No.	EIA Designation	Function	Signal Originated by DTE or DCE
1	AA	Chassis Ground	---
2	BA	Send Data (SD)	DTE
3	BB	Receive Data (RD)	DCE
4	CA	Request To Send (RTS)	DTE
5	CB	Clear to Send (CS)	DCE
6	CC	Data Set Ready (MR)	DCE
7	AB	Signal Ground (SG)	---
8	CF	Received Line Signal Detector (CO)	DCE
20	CD	Data Terminals Ready (TR)	DTE

* DTE -- Data terminal equipment
DCE -- Data communication equipment

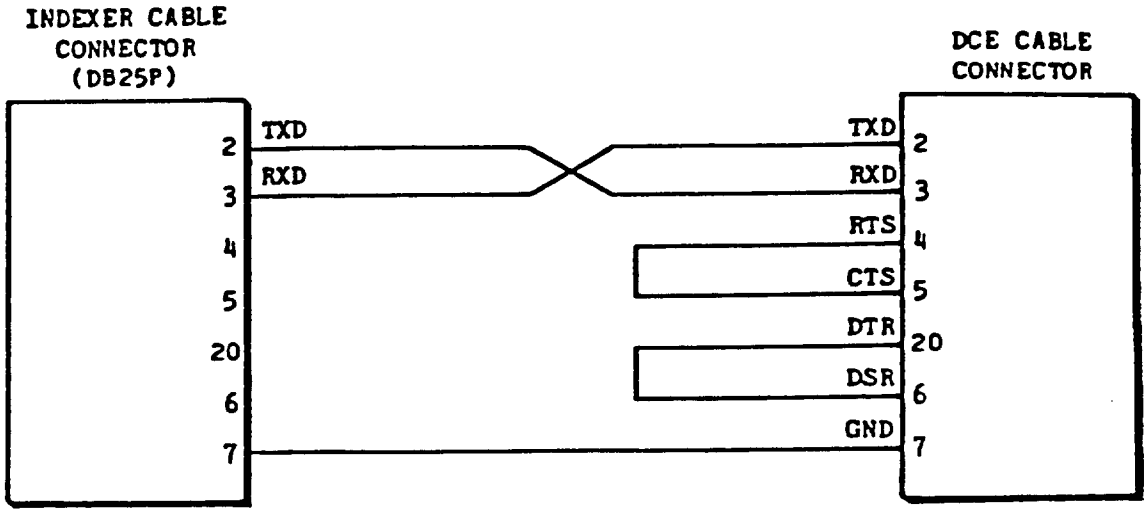


Figure 4-6. Null DCE Connector

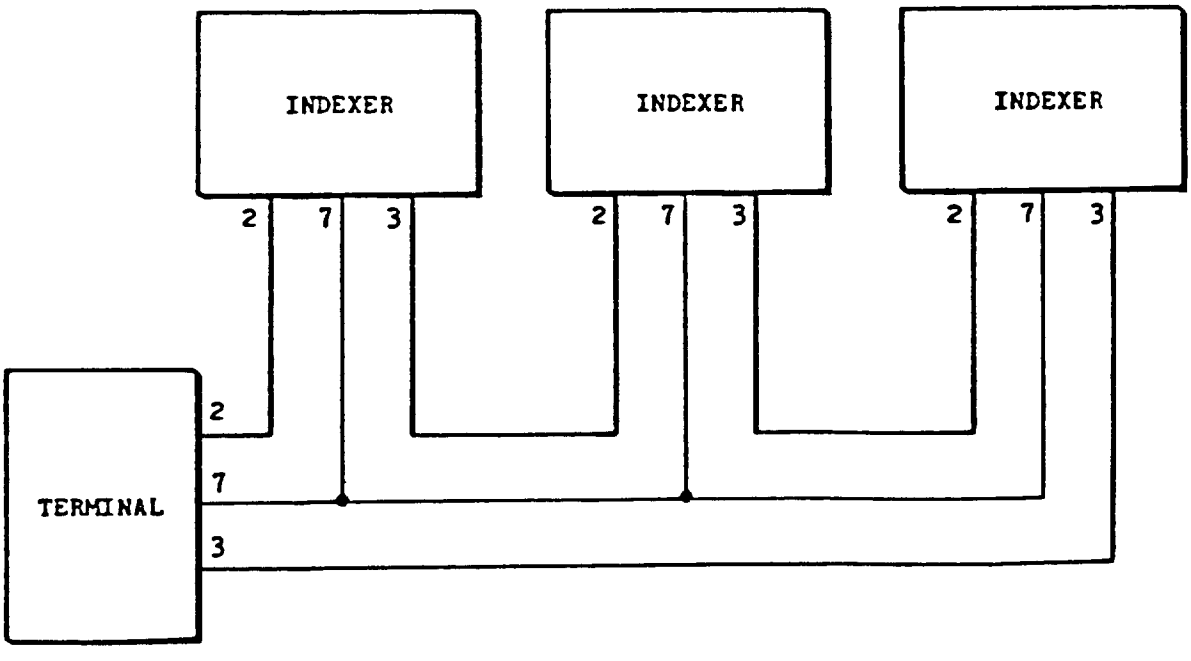


Figure 4-7. "Daisy-Chain" Cabling

4.5.2. MATCHING RS-232C INTERFACE CHARACTERISTICS

If the wiring requirements of section 4.5.1 are met, the user must synchronize the 372 with his Console device. Two RS-232 Baud rates are available for communications, 1200 and 9600 Baud. The 372 is factory set at 9600 Baud. At this rate, data and commands are transmitted at a rate of about a millisecond per character.

The externally accessible configuration switches are located just below the front panel Status Indicator LED's. They are protected by a cover which doubles as a retainer for the Power connector. Four small switches are located under the cover, three Unit Number (Device Address) switches, and the BAUD switch. Set the BAUD switch as required:

OFF: 9600 Baud ON: 1200 Baud

Either the 372 must be switched to match the Console device, or the Console must be switched to match the 372.

Other protocol parameters are:

8 Data bits, 1 Stop bit, and No Parity.

4.5.3. SETTING THE UNIT NUMBER

The other three switches described in 4.4.3 are labelled ADDR 0 through ADDR 2. These are used to set the Unit Number which identifies a single 372 in a multiple axis situation. A Unit Numbers permits any command to be addressed to a single unit on a multiple axis communication line. The factory setting is Unit Number 1.

If more than one axis is being connected, each indexer should be given a different Unit Number. Table 4-4 indicates the various switch combinations and the corresponding Unit Numbers. A change in the switch setting is not effective until reset or Power on. Refer to section 5.1.3 for a discussion on the use of Unit Numbers.

Table 4-4. Setting the Unit Number

Switch	Unit Number 1	2	3	4	5	6	7	8
ADDR 0	OFF	ON	OFF	ON	OFF	ON	OFF	ON
ADDR 1	OFF	OFF	ON	ON	OFF	OFF	ON	ON
ADDR 2	OFF	OFF	OFF	OFF	ON	ON	ON	ON

4.5.4. TESTING THE RS-232C INTERFACE

The following test assumes that a dumb terminal is used for testing. Once the terminal device and all motor and drive connections are complete, all the devices can be powered up, and system testing may begin.

The 372 echoes all the characters it receives. The Console terminal device should be in the "Full Duplex" configuration where all keystrokes are transmitted, but only echoes are displayed.

Press the terminal space bar and verify that the cursor moves on the terminal screen.

> If no cursor movement is apparent, Transmit and Receive wires may be crossed. Terminal handshaking signal requirements may not have been defeated.
>
> If the screen exhibits double characters, switch the Console from Half Duplex to Full Duplex.
>
> If unexpected characters are displayed, RS-232 protocol is not correct.

A quick functional test is to type "*G*" followed by a space. This instructs the 372 to move the motor according to whatever motion parameter values are saved in memory.

> Note:
> The motor will not move if the STOP INPUT of the AUXILIARY 2 connector is not wired properly. (See Fig 4-4, pg 55).

Factory settings call for the 372 to send the drive 25,000 steps at a maximum rate of 5,000 steps per second. A 25,000 step motor will rotate 1 revolution in the clockwise direction. The acceleration and deceleration will appear to be instantaneous, the motion will take about 5 seconds.

Lower resolution motors will move proportionally farther and faster if the step pulses from the 372 are not too narrow.

Type "*1RA*" followed by a space to verify Limit input wiring. If the 372 has been set for some Unit Number other than 1, use that number in place of the "1" before the *RA* command. The 372 will respond with "*@" if the Limit inputs are properly connected.

For multiple axes, make sure that each unit in the chain responds to a command preceded by its Unit Number. If so, the internal setup parameters may now be adjusted to match the application.

4.5.5. PARAMETER SETUP

The 372 retains many internal operating parameters that pertain to motor control. Some obvious examples are the velocity, acceleration, and distance parameters to be used when a GO command is received. The stored parameters are listed in Table 3-1.

At this point, if your motor is not a 25,000 step/rev motor, it would be a good idea to tell the 372 what your motor resolution is. Motor resolution is entered using the "*MR*" command. This command uses a code number to indicate the number of steps per revolution, or steps per inch for linear motors. Refer to Table 5-2 for the available resolutions.

Example: "*MR12*" (set the 372 for a 36,000 step motor)

If the 372 is to be used in an On Line application where it is controlled by a host, it may not be necessary to change defaults because the host will be sending parameter commands anyhow.

The same can be true if the 372 will be running stored sequences. Even so, the number of commands that must be sent for initial configuration of the indexer can be reduced by fixing appropriate parameters in non-volatile memory for automatic loading on power up. An example would be the Jog velocity. It is likely that the operator that uses the Jog function would want to program in the optimum Jog speed and never change it.

To inspect the default parameters type in the command "*1DR*" followed by a space. Use the correct Unit Number in place of the "1" as required. The 372 will respond as follows:

```
MN   H+   GH+  IM1   CM0  MV7  CG4  MS1
FS00000000    QS00000000    SS00000000
OS00000000    MR10          ER4000
D25000        US1           DW250
V.2           LA999         VS0
A100          GH.2          GA100
JV.1          JA100         XP0
```

NOTE:
The *DR* parameter report is in the form of the commands that are used to set those parameters. If another 372 is on the communication line after the one that reports, it will interpret the parameter report as a series of commands. This is intentional and may be used to duplicate setup from unit to unit. The "*F*" command may be addressed to any units that should be prevented from adopting the reported parameters.

The parameter report indicates how the 372 will respond when the next GO command is received, or when command inputs START, GO HOME, or JOG are activated while Off Line.

If commands are given to change any of these parameters, the change will be reflected in the *DR* report, but to make the changes permanent, use the Save Defaults command: "*SA*"

Many of the *DR* parameters pertain to closed loop operation. If no encoder feedback is used, then the following parameters from the above report may be ignored:

CM0, MV7, CG4, FS00000000, ER4000, DW250

Otherwise, refer to section 4.6 after this setup is complete.

The remaining parameter commands from the *DR* report are described briefly below. Refer to the Chapter 5 description for each command that may be useful in initialization for the 372's intended application.

MN - motion mode, positioning vs speed control
H+ - motion direction
GH+ - Go Home direction (for Go Home input only)
IM1 - Input Mode (commands, sequences, or remote data)
MS1 - Messages on or off
QS00000000 - Transmit a signal on various conditions
SS00000000 - Turn various configuration options on or off
OS00000000 - Turn various other configuration options on or off
MR10 - Match the motor resolution
D25000 - Move distance
US1 - Distance units
V.2 - Move velocity
LA999 - Limit stop deceleration rate
VS0 - Stop/Start rate for low resolution motors
A100 - Move acceleration rate
GH.2 - Go Home velocity (for Go Home input only)
GA100 - Go Home acceleration rate
JV.1 - Jog velocity
JA100 - Jog acceleration
XP0 - Sequence number for automatic execution

4.5.6. TROUBLESHOOTING THE RS-232C INTERFACE

Check the following items if you have any difficulty in operating via the RS-232C Communications Interface:

a. Garbage characters are displayed -- Are the Console device and the 372 set for the same Baud rate?

b. Connections -- Does the interface cable have the required connections specified above? Is the cable properly seated? Is the right connector being used on the terminal?

c. DCE vs DTE -- Do you have the transmit and receive wires crossed? Transmit outputs will measure -5 to -12 volts to ground. Inputs will be close to zero volts. Connect outputs to inputs.

d. Other Signals on RS-232C -- Does your terminal require other signals on its RS-232C port? Have jumpers been installed in the RS- 232C cable to provide the extra signals?

e. Is the cable too long? (Fifty feet is the EIA recommended maximum length.

4.6. ENCODER INSTALLATION

The ENCODER connector is provided to allow the connection of an incremental optical encoder to the 372. Dual-channel incremental encoders with quadrature and with single-ended or differential TTL outputs may be used. The encoder may be used as part of a closed loop system or as an independent means to verify position. The encoder may also be used to detect whether a stall has occurred during a move. The encoder functions include:

a. Encoder referenced position
b. Encoder position servo
c. Motor stall detection
d. Position loss detection
e. Multiple axis stop
f. Absolute position coordinates
g. Homing function

Implementation of the encoder functions listed in a through e above requires connection of the encoder and issuing the appropriate configuration commands. Functions f and g may be implemented without an encoder. Section 3.6 describes the function of these options, section 4.8.12 describes the procedure for setting them.

4.6.1. CONNECTING AN ENCODER

To implement the Position Tracking functions, it is necessary to connect an incremental optical encoder to the Indexer. The Indexer will supply power (up to 150 mA) to drive the encoder. Encoder outputs must be 3-5 VDC, square wave, and TTL compatible. When encoders with ended outputs are used, the unused CHANNEL A-, B-, and Z- should be left unconnected.

The pinouts for the encoder connector (25 pin "D") are shown in Table 4-4. Signal characteristics are summarized in Table 1-1. A typical input circuit for channels A, B, and Z is shown in Figure 4-8.

4.6.2. CONFIGURING THE ENCODER OPTIONS

Most of the various encoder functions are controlled with the "*FS*" command. Several functions controlled by this command are critical to the proper operation of the equipment.

To match the 372 to a given closed loop system, it is necessary to issue the proper setup instructions. Priority for setting

parameters follows, with the required command and the default settings indicated in parentheses.

To operate with encoder:

1. Set the motor resolution: *MR* command, (25000)
2. Set the encoder resolution: *ER* command, (4000)
3. Enable the desired encoder functions: *FS* command, (off)

To optimize performance:

5. Set the Go Home acceleration: *GA* command, (10)
6. Set the Deadband Width: *DW* command, (1 rev)
7. Set the Position Maintenance parameters: *CG*, *MV*

Encoder resolution is discussed below, other encoder functions and the Homing function are discussed in Chapter 5. All of the above commands are detailed in Chapter 5.

4.6.3. SELECTING ENCODER RESOLUTION VALUES

The number of encoder steps seen by the Indexer is equal to four times the number of encoder "lines". For example, a 1000-line encoder mounted directly on the motor will generate 4000 encoder steps for each revolution of the motor shaft. This is due to the 372 encoder quadrature detection circuitry. The resolution of the system can become confused when an encoder manufacturer's resolution specification includes this times four multiplication.

A minimum of two motor steps per encoder step is required for successful operation of the Position Maintenance function. With the above encoder, a motor resolution of better than 8000 steps/rev would be required. Ratios above four motor steps per encoder step insure stability of the Position Maintenance servo function.

The 372 must know the resolution of both the feedback encoder and the motor for the Position Tracking functions to work properly. These parameters are entered using the *MR* and *ER* commands. The *ER* command parameter must reflect the number of feedback counts received for one revolution of the motor.

The Factory default values for these parameters accommodate 25,000 step motors and directly mounted 1,000 line encoders. Other values may be entered and stored permanently. The *ER* parameter must reflect the quadrature detection. For the directly driven 1,000 line encoder, the correct *ER* instruction would be *ER4000*.

If a reducer is introduced between the motor shaft and the encoder, the number of encoder steps must be divided by the equivalent gear ratio to define the number of encoder steps received by the Indexer.

For example, using a 25,000-step motor, a 1000-line encoder, and a 10:1 reducer, the ratio of motor revolutions to encoder steps would be changed as shown below:

Parameter	Encoder Direct Mounted to Motor	10:1 Reducer between Motor and Encoder
Number Encoder steps seen by Indexer per motor revolution	4,000	400

Table 4-4. Encoder Connection Pinouts

A standard 25-pin male "D" connector (DB25P) is required to connect the feedback encoder to the 372 "POSITION SENSOR" connector.

Pin	Signal	Pin	Signal
1.	Quadrature Channel A+	14.	Ground
2.	Quadrature Channel A-	15.	Ground
3.	Quadrature Channel B+	16.	Ground
4.	Quadrature Channel B-	17.	Ground
5.	Z Channel+	18.	Ground
6.	Z Channel-	19.	Ground
7.	NC	20.	Ground
8.	Shield (case ground)	21.	NC
9.	NC	22.	NC
10.	NC	23.	+5 VDC
11.	NC	24.	+5 VDC
12.	NC	25.	+5 VDC
13.	NC		

4.6.4. TESTING POSITION TRACKING SETUP PARAMETERS

a. Testing encoder resolution:

The encoder feedback functions of the 372 will inevitably foul up if encoder resolution or signal polarity are wrong. The effects of an erroneous *ER* instruction are not always obvious at all. In the process of setting up a system, it is wise to verify that the system functions as expected without active encoder feedback, before the encoder functions are enabled.

The best tests for proper function involve the use of the status requests. It is advisable that the operator be familiar with the way open loop commands and status requests are used before attempting to execute the tests below.

The encoder resolution setting may be tested using interface commands. The idea is to move the motor open loop and verify that the number of feedback encoder counts meets with expectation.

Step 1 - Assuming that the indexer under test is "On Line" and is Unit Number 1, use the "*1FR*" status request to verify that all encoder functions are off. The response should be "00000000".

Set the internal encoder position to zero: "*PZ*". Have the motor move one revolution: "*MN A1 V2 D25000 G* " (for a 25,000 motor). Report encoder position: "*FSB1 1PR FSB0* ".

Step 2 - Repeat the test to insure that backlash or motor windup are not interfering with the test. The resulting encoder position report should be very close to the number set with the *ER* instruction. If the position report is negative, encoder channels A and B are reversed.

Once the encoder resolution is set, it should be safe to enable Encoder Step Mode and Position Maintenance, and all other encoder functions except Stop on Stall. Verify proper operation of the Go Home function before enabling Stop on Stall.

b. Testing the Go Home function:

Limited travel systems need End-of-Travel limit switches to prevent overtravel during this test. It is advisable that the limit switches be tested first to verify that (1) they will allow the motor to move and (2) they will stop the motor as intended.

If the HOME ENABLE input is connected, force it to the true (off) state to avoid interference.

Use the interface "*GH-1*" command or ground the Go Home command input to initiate the Home search.

If the motor encounters both limit switches, or just runs forever, the Home limit signal (Z Channel) was not asserted.

If the motor makes a very short move in the "-" direction, the Home signal is active. This indicates that either the signal is wide, or its polarity is reversed. If the motor is not even at the Home position, reverse Home Limit signal polarity. If the signal is wide, it will be necessary to approach it in the "+" direction. The repeatability of the Homing operation may be tested using the "*X1*" or "*X2*" commands.

c. Testing the Stop on Stall function:

The Stop on Stall function is dependent on the setting for Backlash to give optimum operation, as is the Position Loss Detection function. The factory default setting for this parameter is 2 "full steps", 250 microsteps for a 25,000 step motor. This default setting is optimum if the encoder is mounted on the motor, but should be increased to accomodate backlash or windup if there is a transmission between the motor and encoder.

The actual backlash may be measured using interface commands. The idea is to move in one direction, stop, and make a series of one step moves in the opposite direction. No change in encoder position will occur while the Backlash is being taken up. The number of motor steps counted before any encoder counts are received is the measure of the backlash.

Move the motor in one direction, clear position counters:

"MN A10 V1 D-1000 G X0 PZ "

Execute a series of one step moves and report both motor and encoder position each time:

"D1 PS L G 1X1 FSB1 1PR FSB0 T.5 N C "

This command sequence will move one step, report both motor and encoder position, wait a half second and repeat. The "*PS*" command allows entering the entire sequence before any motion occurs. The "*U*" command may be used to pause the execution, "*S*" will stop it.

Watch the position reports until the encoder step position begins to change. The accumulated motor step position at that time will be the measure of the backlash.

4.7. REPLACING A 372 IN THE FIELD

The degree of difficulty of replacing a 372 in a working installation depends on how lengthy the internal programming is.

> A RECORD OF PROGRAMMED FUNCTIONS SHOULD BE KEPT TO ALLOW FIELD REPLACEMENT.

The replacement of the installed unit calls for the following steps:

1. Set the BAUD switch and jumpers on the new unit to match those on the old. (See 4.3.2, and 4.5.2)

2. Set the stored internal operating parameters on the new unit to match those on the old.

3. Set the stored internal sequence functions on the new unit to match those on the old.

4. Set the Unit Number on the new unit to match the old. (section 4.5.3)

5. Move all the connections one at a time from the old unit to the new.

If the unit being replaced is still able to communicate using its RS-232 interface, it can be instructed to display stored information. The "*DR*", "*XN*", and "*XU*" commands are used to display stored commands. See sections 5.4.15 and 5.4.16. This information must be entered into the new unit by hand.

The report given by the *DR* command may be entered verbatim into the new unit. At present, the 372 is not fast enough to receive the *DR* report sent by another 372.

Sequences reported by the *XU* command must be entered using the *XD* command. See section 5.6.2.

If the old unit is dead or the Status light is red, this stored information will have to be entered into the new unit from some other source.

5. INDEXER COMMANDS

This chapter describes the Indexer commands that are used with a remote terminal or computer to control the Indexer via a communications interface.

Using a remote terminal or computer, you can instruct the Indexer to execute a series of commands, including a sequential change of the basic motion parameters, time delays, pauses, and repeating (looping) operations.

5.1. COMMAND CONVENTIONS and DEFINITIONS

Indexer commands are one or two letter mnemonic "codes", some with numerical data attached. For the most part they are simple and obvious; "*G*" for "go" and "*S*" for "stop". When a common "dumb terminal" or a computer with terminal emulation software is used, these commands can be simply typed in. The 372 executes them immediately.

5.1.1. DELIMITER

All commands must end in a delimiter. The delimiter signifies that the command is complete. A delimiter serves the same function as the space between words in a sentence. The delimiter, which is part of the command, is a space character (entered with the keyboard space bar) or a carriage return (<CR>). The "Return" key is sometimes called the "Enter" key. It produces the carriage return character.

5.1.2. INDIVIDUAL COMMANDS

An individual Indexer command controls a single parameter, function, or action such as acceleration, velocity, position, time delay, pause, loop, go, etc. There are two classes of individual commands, Immediate and Buffered.

BUFFERED COMMANDS: Commands that are identified as "Buffered" are received by the indexer and stored (in the command buffer) if the indexer is not free to execute them. Stored commands are executed as soon as the indexer can get to them, in the order they were received. Only buffered commands may be included in a stored sequence.

IMMEDIATE COMMANDS: Commands that are identified as "Immediate" are executed immediately on receipt and will take priority over whatever operation is in progress. These commands include various Stop commands that clear the command buffer, and various Status Request commands that have no affect on the command buffer.

Individual commands are variable in length. They can consist of one or more letters with a delimiter, and one or more letters and numbers with a delimiter. Each command is entered as a character/delimiter combination. Some commands include a sign (±) to denote direction of motion. The number of characters used depends on the type of command entered.

Typical commands have the form:

S A10 V3 D46000

When two or more individual commands are entered on the same line, they are separated by spaces, and multiple command entries will be displayed on a single line of your terminal screen. The following example shows a set of individual command entries with space delimiters on the same line:

MN A10 V2 D25000 L10 G N

If spaces are used as the delimiter and a large number of multiple command entries are made, you could exceed the ability of your terminal to display characters as a single line (80 characters per line is a typical value). When the carriage return is used as a delimiter, the cursor returns to the beginning of the line.

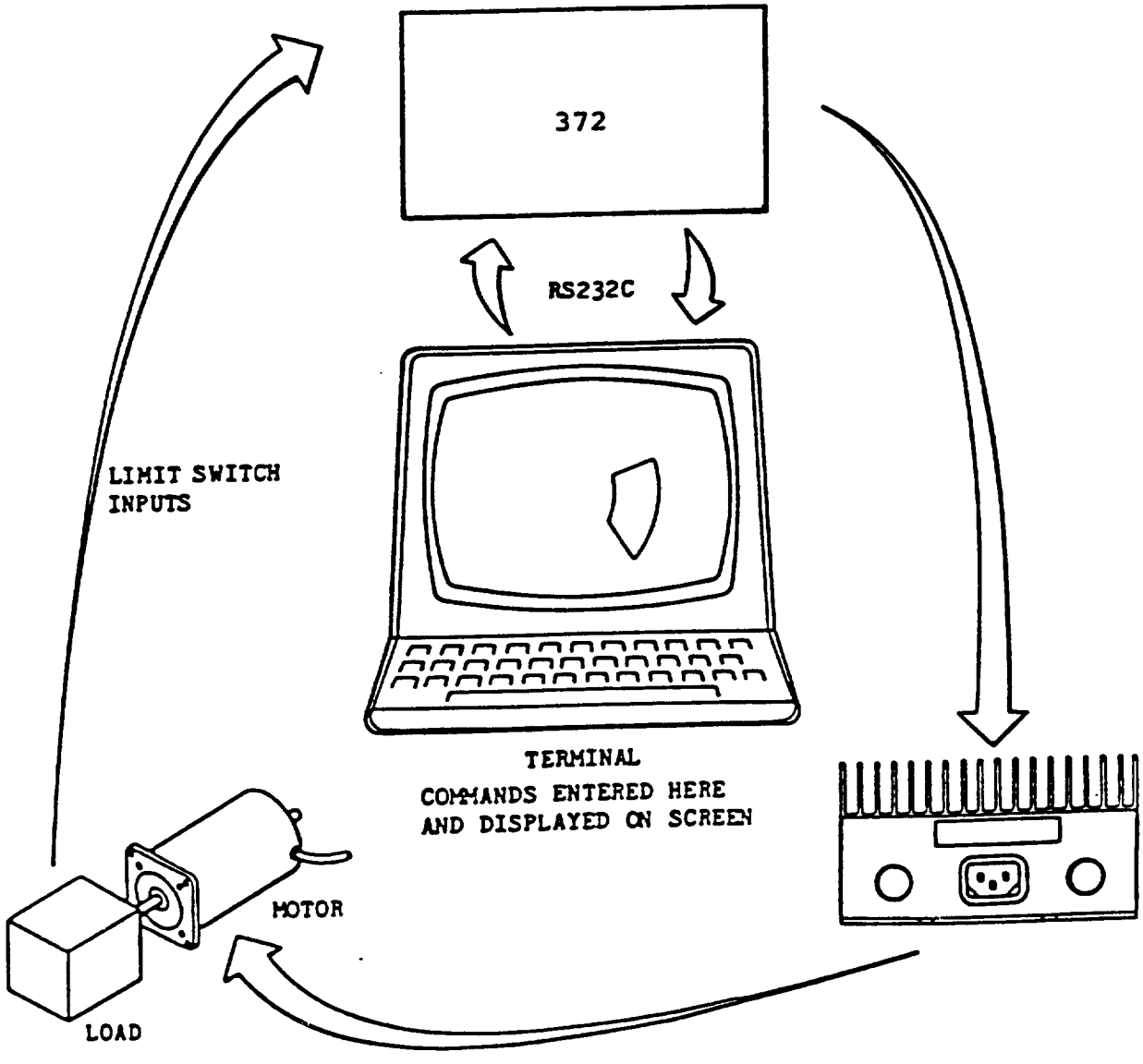
Note

Pressing carriage return on some terminals will return the cursor to the start of the line. The line-feed key (or Control J) may also have to be pressed to move the cursor to the next line. This will not affect the delimiter entered using carriage return.

The example that follows shows the effect of using carriage return/line feed as the delimiter.

MN
A10
V2
D25000
G

Figure 5-1. Remote Terminal Operation



5.1.3. MULTIPLE INDEXERS

When multiple indexers are on the communications line, commands like the previous example are executed by all indexers on the line. To send commands to a single unit, the Unit Number of that indexer should be put in front of the instruction.
Example: Three indexers are on an RS-232 "daisy chain". They are sent the following commands:

E MN A10 V10 1D25000 2D50000 3T1 3D100000 G

Unit 1 moves 25,000 steps, unit 2 moves 50,000 steps, and unit 3 waits 1 second and moves 100,000 steps. All three use the same rates.

The Unit Number is set on external switches. See section 4.5.3.

5.1.4. COMMAND BUFFER CAPACITY

Any combination of individual commands and command groups can be entered in the buffer until the total number of characters currently stored (including the delimiters) equals 1200. The Indexer uses a first-in-first-out serial buffer. As the commands are read from the buffer, additional commands can be entered to replace them. Therefore, the possibility exists that a command series could actually consist of more than 1200 characters. Examples of command sets that might be used for different applications are presented later in this chapter.

5.1.5. FUNCTIONAL GROUPING OF COMMANDS

The basic Indexer individual commands can be subdivided into five categories: Motion Parameter commands; Execution commands; Status Request commands; General Purpose commands; and Memory Control commands. These commands are discussed in sections 5.2 through 5.6. A summary table is included in each section that indicates the character combination used to enter the commands, the command classification (Buffered or Immediate), and for some, the section of the chapter that describes them. A summary of all commands appears at the end of this manual.

5.1.6. SPECIAL COMMAND SYNTAX

In addition to its repertory of alphanumeric commands, the 372 uses some non-alphanumeric characters in special functions. The "=", "(", and ")" characters are valid in commands.

"=" - Save Default Parameter

The "=" character is used as a short form command to store a parameter in non-volatile memory. This syntax allows a single parameter to be added to the semi-permanent set of defaults without changing other defaults. Otherwise, the *SA* command would be used to save all default parameters. The "=" is used with the desired command letters (no parameters) to save the current parameter value.

Example: *JV1.5 =JV*
(change the Jog velocity to 1.5 rev/sec, save that as a default)

"()" - Remote Data Entry

Any command having a numerical parameter, such as the velocity command, can substitute a number read from AUXILIARY inputs for the one that normally follows the command. In this fashion, velocity data may be entered at execution time by the operator.

The parentheses are inserted in the command in place of digits of data that would normally be there. Inside the parentheses, the numbers 1, 2, and 3 may be used to instruct the 372 as to which of the three available BCD digit inputs to use as a source for a digit of remote data.

Example: *A10 V(1) G*

In this example, the 372 is instructed to set its acceleration and velocity parameters, and then run at the designated speed. To interpret the velocity command, it reads the AUXILIARY connector BCD inputs designated for digit number 1. This number is used as velocity data, and the 372 accelerates the motor to that speed. If a BCD thumbwheel switch is connected to the inputs designated "BCD-1" in Table 3-2, an operator can change the motor speed every time this instruction sequence is issued.

This feature allows connection of up to three BCD digits. Each digit is identified by a number which indicates which pins of the AUXILIARY connector the BCD data is read from. In a command, the parentheses may contain a decimal point. They may be preceded and followed by other digits.

Refer to figure 4-4 for a sample BCD wiring diagram.

5.2. MOTION PARAMETER COMMANDS

These commands, which set up the way the motor will move, are summarized in Table 5-1. All of these commands can be stored in the Indexer buffer and as part of sequences stored in non-volatile memory. Most of these parameters have default values which may also be stored, and loaded automatically on power up.

Initially, the 372 assumes default values for motion parameters. The Indexer will continue to use these values until you enter a new value for one or more of the motion parameters. If you do not enter the type of motion command when the interface is enabled, the Indexer will operate in the MN (Mode Normal) mode.

Once entered, a parameter will be remembered. It is not necessary to enter the same parameter value twice unless the indexer is reset.

If the acceleration or velocity command values entered exceed the capability of the motor/drive, it will stall without damage.

5.2.1. TYPE OF MOTION COMMANDS

The commands for the three modes of motion are:

MN NORMAL Mode -- Move to a preset position and stop. Accelerate at last rate, "*A*" and velocity, "*V*" specified. The motor will move a distance corresponding to the "*D*" value entered.

MC CONTINUOUS Mode -- Accelerate at the last rate specified ("*A*") to the velocity ("*V*") and continue to run at the specified velocity until stopped.

MA ALTERNATE Mode -- Alternately move between an initial start position to a distance position "*D*". Backtrack from the specified position to return to the start point.

MP*a* Positioning Mode -- Select either Incremental or Absolute interpretation for the "*D*" command position parameter.
(a = A or I, to select Absolute or Incremental positioning)

Q*b* Direct Velocity Mode -- Hexadecimal numbers representing motor velocity (pulse frequency) are input directly into 372 pulse generation circuitry using the "*RM*" command.
(b = 0 or 1, to enable or disable this mode)

Table 5-1. Shaft Motion Commands

Command Controls	Format of Entry (Note 1)	Description of Commands	Class (Note 2)
Type of Motion	MN	PRESET MODE (Go to Position)	B
	MC	CONTINUOUS MODE (Rotate at constant velocity)	B
	MA	ALTERNATE MODE (Continuous back and forth)	B
	MP*a*	Positioning Mode (a=A/I, Absolute/Incremental)	B
	Q*b*	Direct Velocity Mode (b=1/0, enable/disable)	B
Motion Parameters	D±*nnnnnnnn*	Set Direction (optional) and Distance (position) Value	B
	H	Reverse Motor Direction	B
	H±	Set Direction of Motion + = Clockwise - = counterclockwise	B
	V*nn.nnn*	Set Velocity	B
	VS*nn.nnn*	Set STOP/START Velocity	B
	JV*nn.nnn*	Set Jog Velocity	B
	A*nnn.nn*	Set Acceleration/Deceleration	B
	GA*nnn.nn*	Set Go Home Acceleration	B
	JA*nnn.nn*	Set Jog Acceleration	B
	LA*nnn.nn*	Set Limit Acceleration	B
	RM*hhhh*	Direct Velocity Input (Q1 mode)	B
Set Up Parameters	MR*nn*	Match Indexer to Motor Resolution	B
	LD*n*	Define Limit Function	B
	US*nnn*	Set Position Scale Factor	B

NOTE 1. *nnn* indicates numeric value to be entered.
2. B = Buffered command that is stored and executed in sequence.
"*a*" indicates a single alpha character response
"*b*" indicates variable parameter, b=1 or 0

Table 5-1. Shaft Motion Commands (continued)

Command Controls	Format of Entry	Description of Commands	Class
Position Tracking	**ER***nnnnnn*	Set Encoder Resolution	B
	FS*ab*	Set Encoder feedback function	B
	DW*nnnnnn*	Set Backlash window	B
	CM*n*	Set Position Maintenance Correction Mode	B
	CG*n*	Set Position Maintenance Correction Gain	B
	MV*n*	Set Position Maintenance Correction Velocity	B

NOTE 1. *nnn* indicates numeric value to be entered.
2. B = Buffered command that is stored and executed in sequence.
"*a*" indicates a single alpha character response
"*b*" indicates variable parameter, b=1 or 0

5.2.2. MOTION PARAMETERS

The commands for shaft motion parameters are:

D$\pm$*nnnnnnnn* Distance -- May be combined with direction command. Choices are a function of the scale factor selected (1 to 255) with the **US** command for the distance units. The sign indicates direction. (default: +25,000)

Typical commands: (scale factor 1)

D25000 (move clockwise 25000 steps)
D-43570 (rotate counterclockwise 43570 steps)

H Change motor direction -- This command will reverse the current motor direction, effective at the beginning of the next move.

H$\pm$ Direction of Motion -- The choices are clockwise rotation (+) or counterclockwise rotation (-). (default: CW)

V*nn.nnn* Velocity -- The range of choices are a function of the specific motor/drive. For the default 25,000 step motor, the range is .001 to 20 rev/sec. (default: .2)

Typical Commands:

V3.49 (set velocity to 3.49 rev/sec)
V10 (set velocity to 10 rev/sec)

JV*nn.nnn* Set Jog Velocity -- This command sets the Jog velocity to be used when an Off Line Jog input signal is received. The velocity is specified in rev/sec. (default: 0.1)

VS*nn.nnn* Set Start/Stop Velocity -- This command sets a start/stop velocity to be used on all future moves. The velocity is specified in rev/sec.
(default: 0)

A*nnn.nn* Acceleration -- The range of choices is from 0.01 to 990.00 rev/sec/sec (default: 100 rev/sec/sec)

Typical commands:

A.1 (set acceleration to 0.1 rev/sec/sec)
A10 (set acceleration to 10 rev/sec/sec)
A89 (set acceleration to 89 rev/sec/sec)

GA*nn.nnn* Go Home Acceleration -- The rate used for acceleration during Homing. The range of choices is from 0.01 to 990.00 rev/sec/sec. (default: 10 rev/sec/sec)

JA*nn.nnn* Jog Acceleration -- The rate used for acceleration for the Jog function. The range of choices is from 0.01 to 990.00 rev/sec/sec. (default: 10 rev/sec/sec)

LA*nn.nnn* Limit Acceleration -- The rate used for deceleration when an End-of-Travel limit is encountered. The range is from 1 to 990.00 rev/sec/sec. (default: 999 rev/sec/sec)

RM*hhhh* Direct Velocity parameter -- When the indexer is in Direct Velocity Input Mode (see the "Q" command above), range is from 0 to 32,765 (0000 to 7FFF hex) for clockwise rotation, and 32,768 to 65,533 (8000 to FFFD hex) for counterclockwise rotation. Special values 7FFFh, and 8FFFh will set Programmable Output 1 high, 7FFEh, and 8FFEh set it low.

5.2.3. SET UP PARAMETERS

MR*nn* Motor Resolution Select -- Use this instruction to match the indexer to the motor. The 372 retains a number of different motor resolutions. Refer to Table 5-2 for the numerical code representing the desired motor resolution.

Example: *MR12* (match the indexer to a 36,000 step/rev motor)

Table 5-2. The *MR* command Motor Resolutions

MR	Resolution	MR	Resolution
0	200	10	25,000
1	400	11	25,400
2	800	12	36,000
3	1000	13	50,000
4	1600	14	51,200
5	3200	15	4,096
6	5000	16	12,800
7	6400	17	25,600
8	10,000	18	12,500
9	21,600	19	16,284

US*nnn* Position Scale Factor -- The Scale Factor (distance multiplier) is set with this command. The multiplication feature allows the *D* distance parameter to reflect engineering units (degrees, inches) in place of motor steps. The default value for the Scale Factor is 1. This value may be changed to any number between 1 and 255. The current value can be determined using the status request command "u*UR*".

Example: US25 D1000 G (motor moves 25,000 steps)

LD*n* Limit Definition -- This command allows the operator to disable the "failsafe" limit switch inputs for Clockwise (CW) and Counterclockwise (CCW) end of travel limits.

When travel need not be limited, or for the purpose of testing, the limit function can be defeated using the a single digit parameter that follows the command:

LD0 - Enable the Limit function for both CW and CCW limits
LD1 - Disable CCW Limit function
LD2 - Disable CW Limit function
LD3 - Disable both the CW and CCW Limit functions

LD3 is the factory default condition. This command is Buffered.

5.2.4. MOTION PARAMETER CHANGES

Any of the basic motion parameters may be changed without affecting the others. These include Mode (Preset, etc.), Acceleration, Velocity, Distance, and Direction parameters. The following example will accelerate quickly, run the motor for two seconds at one speed, decelerate slowly, reverse, and turn two revolutions at another speed. All of the above parameters except distance are changed.

A50 V5 D-50000 MC G T2 A1 V0 G V.5 MN H+ G<CR>

To implement this string from a terminal,

>> Insert a pause command *PS* "in front of the command sequence
>> Complete entry of the entire command set

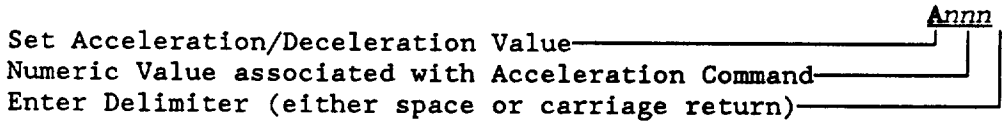
>> Type: *C*<CR> to clear the pause and initiate action.

This procedure allows you to enter the entire command set before any move is initiated.

5.2.5. Examples - MOTION PARAMETER COMMANDS

Example One: Individual Parameter Command for Shaft Motion

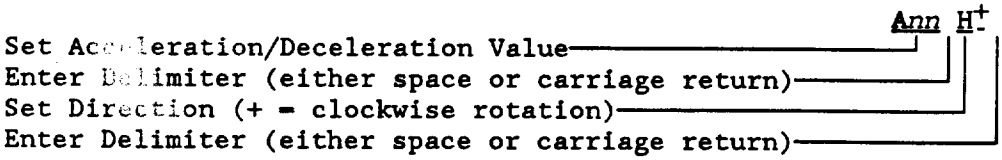
Function: Set acceleration/deceleration value



Sample Command: A109

Example Two: Two Individual Parameter Commands for Shaft Motion.

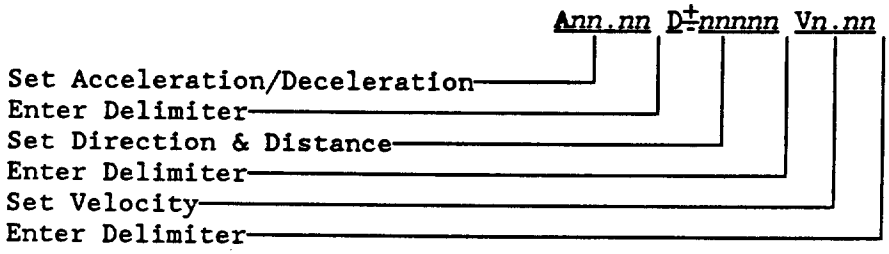
Function: Set acceleration/deceleration; set direction of shaft rotation.



Sample Command: A14 H+

Example Three: Three Individual Parameter Commands for Motion.

Function: Set acceleration/deceleration value; set direction and distance; and set velocity for motion.



Sample Command: A23.66 D-29763 V3.71

5.2.6. POSITION TRACKING COMMANDS

ER*nnnnn* Encoder Resolution Select -- Set the number of encoder steps to be returned for one revolution of the motor. This number reflects the fact that the 372 quadrature detector multiplies the encoder resolution by four. Use this instruction to match the indexer to the encoder.

The 372 factory default setting is for a 1000-line optical encoder. This is equivalent to the command:

ER4000

DW*nnnnn* Backlash Deadband width -- Set the number of motor steps expected after a direction change before feedback resumes.

The backlash deadband allows systems with backlash in the geartrain to use the stall-detect and position-loss detection features. If a nonzero deadband is selected, a position loss condition and stall condition will not be assumed until the feedback position error exceeds the deadband width. See section 4.6.4 for a discussion of how to establish this parameter. (default: 25,000)

FSab Encoder Function set -- Enable or disable absolute positioning or any of seven encoder feedback functions. The "*a*" indicates one of the letters from the list below which is to be entered. Enter b = 0 or 1 as indicated below to obtain the desired result. The valid letters for "*a*" are A through H. The number "b" sets the function OFF = (0), or ON = (1). See section 3.6 for detailed discussion of these various modes of operation.

A:	Incremental/Absolute positioning	(0/1)
B:	MOTOR STEP mode/ENCODER STEP mode	(0/1)
C:	Encoder Position Maintenance OFF/ON	(0/1)
D:	Stop on Stall OFF/ON	(0/1)
E:	Output On Stall OFF/ON	(0/1)
F:	Kill Motion On Trigger 3 OFF/ON	(0/1)
G:	Post Move Position Loss Detection OFF/ON	(0/1)
H:	Output On Position Loss OFF/ON	(0/1)

(default: all off)

Notes:

A Defines "*D*" parameter as a distance (incremental from current position), or as a position (absolute, referenced to Home) Identical to the *MPI/MPA* command function

B Defines "*D*" parameter units as motor steps or encoder steps.

C Enables the position servo, where the indexer will step the motor in an effort to maintain encoder position. (Encoder step only)

D Instructs the indexer to abort any move if a Stall is detected.

E Instructs the indexer to set Output 1 low if a Stall is detected.

F Instructs the indexer to abort any move if a signal is received on the Trigger 3 input.

G Instructs the indexer to check for an encoder position error at the end of every move.

H Instructs the indexer to set Output 1 low if a position error at is detected at the end of a move.

Example: " *FSB1* " (enable ENCODER STEP Mode)

CM*n* -- Correction mode for position maintenance. When the 372 senses that it is off-position a distance equal to "x", the CORRECTION MODE command CM tells it whether to move back at a speed proportional to "x" (Linear), move back at a speed proportional to the square root of the value of "x", or move back at a speed proportional to the square of the value of "x". Allowable entries for "n" are 0, 1, or 2 where 0 = Linear; 1 = Square Root; 2 = Square.

The 372 is set to Linear correction mode on power-up or board reset.

CG*n* -- Correction gain factor where n can be 0 through 7 such that correction gain factor is 2 to the power n. The default value for n is 4 on power-up or board reset. The correction gain factor determines the velocity that the 372 will command the motor to use in attempting to correct its position. A value for 0 will yield minimum velocity; a value of 7 will yield the maximum for a given position error.

MV*n* -- Maximum correction velocity, where n can be 0 through 7 such that the maximum correction velocity will be 250 times 2 to the power n. The correction velocity is the maximum velocity at which position correction will be made. The factory default value for n 7, following power-up or board reset.

5.3. EXECUTION COMMANDS

The Motion parameter commands described in section 5.2 will not cause any motion unless they are combined with an Execution command such as GO.

The Execution commands are summarized in Table 5-3. These commands instruct the indexer to start moving, to pause or delay, to recall and execute a stored sequence of commands, or to repeat (loop) a string of commands. The Execution commands also include motor shutdown, and "software" reset sequences.

5.3.1. START MOVE COMMANDS

There are two commands that result in motor motion, The "*G*" (GO) command and the "*GH* " (Go Home) command.

G -- THE GO COMMAND

The "*G*" (GO) command is the principal buffered command that initiates shaft motion. When the buffer reads a GO command, the motor will then move in accordance with the motion mode and move parameters that reside in Indexer memory at the time the GO command is read. Any motion parameter change entered before the GO command will be reflected in the motion.

If only "*G*" is entered, the Indexer will repeat the previous motion pattern. For example, a string of GO commands

G G G G

would instruct the Indexer to move the same way four times.

You do not have to re-enter all the Motion parameter commands to change one of the variables. If one or more of the motion parameters is

changed by a command entry and the GO command is then entered, the prior pattern will be repeated using the new motion parameter(s) that were just entered. For example,

A14 G V2.6 G D-27634 G

would change acceleration to 14 and move, then change velocity to 2.6 and move with 14 as the acceleration value, then change position and move again using 14 and 2.6 as the acceleration and velocity values.

Table 5-3. Execution Commands

Command Controls	Format of Entry (Note 1)	Description of Command	Class. (Note 2)
Start Move	G	GO (start executing move)	B
	GH[+]*nn.nnn*	Go Home (seek Home signal)	B
Recall and execute	XR*n*	Execute stored command sequence	B
Load Data (LOAD AND GO)	I	Load Move Parameters, Compute Move Profile	B
	uIC	Load Move Parameters, Compute Move Profile, Transmit carriage return when done	B,D*
Stop	Y	Exit from Loop after the current cycle	I
	Q	Clear buffer following execution of the current command	I
	S	Stop (now) at deceleration rate equal to last used acceleration rate	I
	K	Kill (stop pulse generation now)	I
	Z	Software Reset (stop pulse generation now and reset 372)	I

NOTE 1. All entries must be upper case, with a delimiter.
"*nnn*" indicates variable numeric parameter
"*xxx*" indicates variable parameter, x=1 or 0 (or X)
(the maximum number of parameter digits is shown)

2. B = Buffered command that is stored and executed in sequence.
I = Immediate command that will be executed as soon as a delimiter is entered.

* The "IC" command is a "Device Specific" command, see section 5.4.

Table 5-3. Execution Commands (Continued)

Command Controls	Format of Entry (Note 1)	Description of Command	Class. (Note 2)
Pause	T*nnn.nn*	Delay processing subsequent commands in buffer for indicated number of seconds (0-999.99)	B
	TR*xxxx*	Trigger Pause in processing commands	B
	PS	Pause and Wait for the CONTINUE entry	B
	U	Pause and wait for CONTINUE entry	I
	C	CONTINUE (End Pause)	I
Loop	L	Loop continuously	B
	L*nnnnnnn*	Set loop counter (0-9,999,999)	B
	N	End of loop	B
Conditional Branch	SKE*xxxx*	Skip on Trigger Equal	B
	SKN*xxxx*	Skip on Trigger Not Equal	B
Motor Shutdown	ST*b*	Energize or De-energize Drive (b = 1 or 0)	B
Reset	Z	Software equivalent of Power-on reset	I
	X0	Reset Cumulative Position Counter	B
	PZ	Reset Absolute Position Counter	B

NOTE 1. All entries must be upper case, with a delimiter.
"*nnn*" indicates variable numeric parameter
"*xxx*" indicates variable parameter, x=1 or 0 (or X)
(the maximum number of parameter digits is shown)

2. B = Buffered command that is stored and executed in sequence.
I = Immediate command that will be executed as soon as a delimiter is entered.

GH±*nn.nnn* -- GO HOME

This command causes the controller to seek the home position. The sign is optional (a "+" or CW is assumed if omitted). The "nn.nnn" parameter sets the velocity for the Homing function, and has the same range as the "*V*" command. Acceleration rate is set with the *GA* command. For example:

GA55 GH-.3

The indexer response is to accelerate (at 55 rev/sec/sec) in the counterclockwise direction (to .3 rev/sec velocity), and run until either the Home Limit signal or and End of Travel limit signal is encountered. If the Home Limit is encountered, the 372 will stop and back up as discussed in section 3.2.5. If an End of Travel limit is encountered, then the indexer stops hard, reverses, and searches in the other direction. Encountering the other End of Travel limit will cause termination of the search.

The Absolute position (as reported by the "*PR*" command) is set to zero once the Home position is located.

XR*n* -- EXECUTE STORED SEQUENCE

The "*XR*" command is also a Memory command in that it causes a stored sequence to be recalled from non-volatile memory. The sequence is then executed. See section 5.6 regarding the other Memory commands.

Example: *XR1* (execute sequence number 1)

5.3.2. LOAD COMMANDS

These two commands are functional only if the LOAD AND GO function has been enabled with the "*SS*" command. The LOAD AND GO function operates only in the incremental Preset (Normal) mode.

> Note
> The symbol "u" is used in this manual to indicate that a Unit Number (Device Address) must be entered for a "Device Specific" command.

The LOAD command "*I*" or "u*IC* " is used for the LOAD AND GO option. Prior to any move, the microprocessor in the Indexer must calculate a number of internal move constants. Consequently, there can be a delay of up to 25 milliseconds (msec) from entry of the GO command until the first step pulse is generated. In some applications this delay may be excessive. The LOAD AND GO function allows the 372 to precalculate internal constants. The precalculations may be performed at any time

using the "*I*" or "u*IC*" commands. Once calculations are complete, the next GO command will commence pulse generation within 2 msec. In the LOAD AND GO mode, the Load command is required before any new acceleration, velocity, or distance parameter is incorporated.

> Note
> The 25-millisecond set-up period must be available prior to the GO to take full advantage of this option. This applies for any change of move parameters that require a new LOAD command.

If you wish the Indexer to transmit a carriage return to the host when it has finished parameter calculations, use the "u*IC* " command. (See section 5.4 for details on the use of Unit Numbers and Status Requests.)

5.3.3. STOP MOVE COMMANDS

The STOP MOVE commands presented in Table 5-3 are arranged in the order of severity of response. The RESET "*Z*" command is the most abrupt and severe stop command you can use. The STOP AT END OF CURRENT LOOP command "Y" is the least abrupt.

Y -- STOP AT END OF CURRENT LOOP

This command will not halt processing until command processing reaches the command character "*N*" at the end of the command loop. At that time, the Indexer will execute the next command in the buffer after *N*, if any. The command loop cannot be restarted.

Q -- STOP AT END OF CURRENT COMMAND

This command clears any commands awaiting execution in the buffer after the current command has been processed.

S -- CONTROLLED STOP COMMAND

In the PRESET or CONTINUOUS mode this command will decelerate the motor to a stop at the last-used acceleration rate.

The response to "*S*" in the ALTERNATE mode ordinarily is to stop at the end of the current Alternate cycle. The "SS " command allows changing this so that the motor is decelerated immediately. If set to ON, the motor

An "*S*" command will always cause a deceleration to velocity zero at the last used acceleration rate.

The "*S*" command clears any remaining commands in the command buffer unless prevented from doing so via the "*SS*" command.

Note

Normally, the motor is decelerated to a stop at the same rate it was accelerated. A different deceleration rate may be programmed in CONTINUOUS mode only by calling for a velocity of zero rev/sec with a new acceleration and executing a "*G*" rather than a "*S*" command. Example: *"MC A1 V10 G...A100 V0 G "*

The advisability of a steeper deceleration curve should be checked by computing the rotational momentum of the load in question and the amount of energy which may be taken from the load without the motor losing synchronism.

K -- EMERGENCY STOP COMMAND

The "*K*" or KILL command causes the output pulses to cease immediately. This should not be used under anything but **emergency situations** since the motor and load are not brought to a controlled stop. For large loads at high speeds, the load could be driven past limit switches and cause damage to the mechanism and possibly the operator. The "*K*" command always clears the remainder of the Command buffer.

Z -- SOFTWARE RESET

The "*Z*" command is equivalent to cycling the AC power to the Indexer; that is, it disables the communications interface and returns all internal settings to their power-on values. The command buffer is cleared. Like the "*K*" command, "*Z*" causes an immediate cessation of output pulses to the motor.

Note

When the "*Z*" command is used, the Indexer is busy for 500 msec and will ignore any commands.

5.3.4. PAUSE COMMANDS

Four command choices are available for temporarily interrupting 372 command execution.

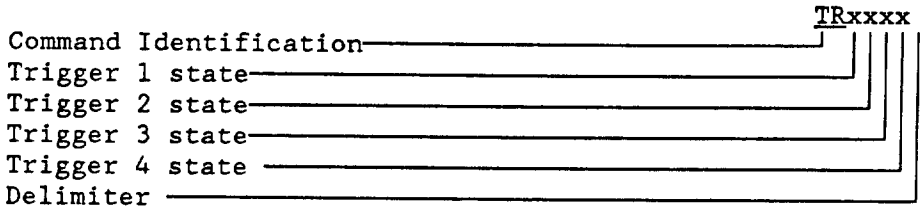
T*nnn.nn* -- Pause in processing commands for a given number of seconds and then continue. Range is .01 to 999.99 seconds.

TR*xxxx* -- Pause in processing commands until the state of the input Triggers matches the designated pattern. The Indexer has four external inputs used to coordinate command execution without the intervention of the host. These inputs are normally off, a state represented by "0". When current is provided to turn the input on, it switches to the "1" state.

Note

The *TR* command functions independently of Input Mode. Care must be taken to avoid confusion with Trigger inputs when Input Mode 4 is employed. The Trigger inputs makeup the third BCD input digit in Input Mode 4.

The TRIGGER command syntax is:



Example: *TR1001*

The four characters used for "xxxx" are variables and can be one of the following:

0 = low; 1 = high; X = don't care

Triggers are used to synchronize 372 operations with external events. They can be used to implement a "handshaking" function with other devices.

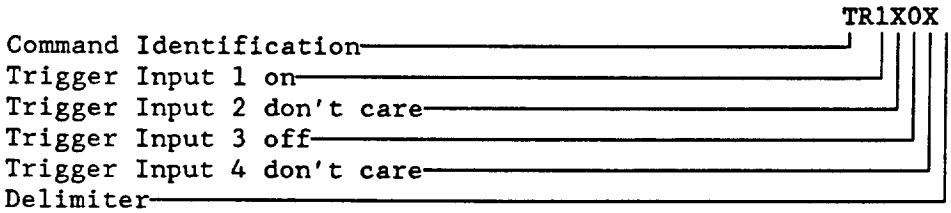
PS -- Pause and wait for a CONTINUE command. A common use of the Buffered Pause PS is to hold execution of individual commands in a Command Loop until the entire Loop has been loaded. This command is typically imbedded in a buffered command string or sequence to allow operator or other external control.

U -- Hold (Pause) Now and wait for CONTINUE. This is an Immediate command. This command is typically used to temporarily halt the execution of a loop or sequence as for a "coffee break".

The CONTINUE command "*C*" is used following execution of a PAUSE command, "*PS*" or "*U*". CONTINUE will allow execution of the next command waiting in the command buffer if any. The *SS* command can be used to configure the Start input to clear the Pause condition.

Pause status of the 372 can be checked with the "*RB*" command. See section 5.4.7.

Example Four: TRIGGER Command



The Indexer would interpret this command as PAUSE until the Trigger 1 input is On and Trigger 3 is off. (Triggers 2 and 4 are ignored.)

Example: *" TR1X0X G "*
(Wait for Trigger 1 to go high, and Trigger 3 to go low, then move.)

5.3.5. LOOP COMMANDS

The LOOP command allows a cycle to be repeated continuously "*L*" or a given number of times "*Lnnnnnnn*". The END-OF-LOOP command "*N*" indicates where the loop ends. The END-OF-LOOP command can be used to indicate that the Indexer should proceed with further commands after the designated numbers of loops have been executed, or in combination with the "*Y*" command to indicate where execution is to stop. The "*U*" command may be used to temporarily halt Loop execution.

5.3.6. CONDITIONAL BRANCHING COMMANDS

The conditional branch instructions involve skipping a command in a list of commands. Both branch commands use a syntax similar to the *TR* command, in which the four Trigger inputs are tested. When the 372 encounters this command, it evaluates the Trigger expression, and either skips the following command, or not.

These are very powerful instructions when used in sequences with the *XR* command. With a combination of these commands, the indexer can be programmed to execute any of a number of sequences depending on the four Trigger inputs.

SKE*xxxx* -- Skip on Trigger Equal: if the state of the four Trigger inputs matches the pattern specified in the command, the command following the *SKE* command is skipped (ignored). Otherwise, the *SKE* command will have no effect.

(x = 0, 1 or X, meaning Off, On, or Ignore, respectively)

SKN*xxxx* -- Skip on Trigger Not Equal: if the state of the four Trigger inputs does not match the pattern specified in the command, the command following the *SKN* command is skipped. If the pattern matches, the *SKN* command will be ignored.

(x = 0, 1 or X, meaning Off, On, or Ignore, respectively)

Example: *MN A1 V1 D100000 L G SKNXXX0 GH-1 N*

(The motor is moved repeatedly until a signal is received on Trigger input 4 at which time the indexer Goes Home and starts over.)

Note

The *SK* commands function independently of Input Mode. Care must be taken to avoid confusion of Trigger inputs with BCD inputs when the Remote Data feature is employed using BCD digit number 3.

5.3.7. MOTOR SHUTDOWN -- ST Command

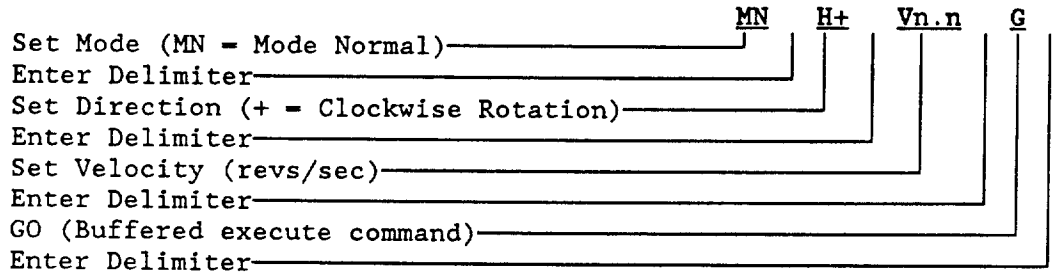
This command allows the de-energizing of the motor. This can be useful when manual adjustment of the motor position is required, or if motor heating needs to be reduced.

ST0 - De-energize (remove motor current)
ST1 - Energize (restore motor current)

5.3.8. EXAMPLES OF EXECUTION COMMANDS

The examples presented in section 5.2.3 dealt solely with individual commands for motion parameters. The examples in this section demonstrate the inclusion of Mode commands and Execution commands. Keep in mind that the absence of Execution commands in examples one, two, and three means the motion parameter commands were executed without any shaft movement. An Execution command MUST follow any buffered commands entered for a move to be executed.

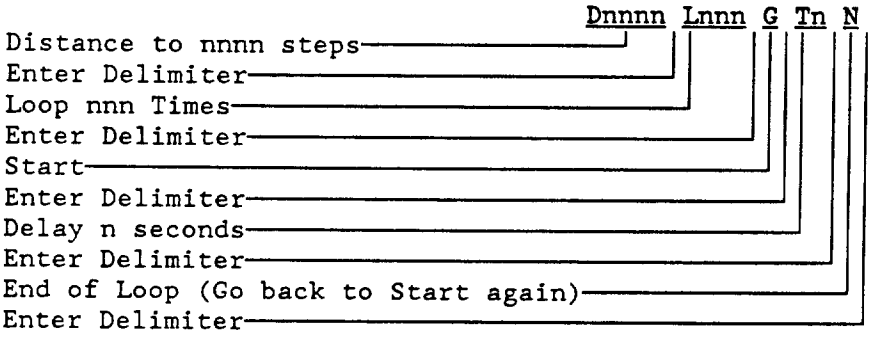
Example Five: Execution Command
Function: Set mode, direction, and velocity; and start move



Sample Command: MC H- V2.9 G

Example Six: Loop Command

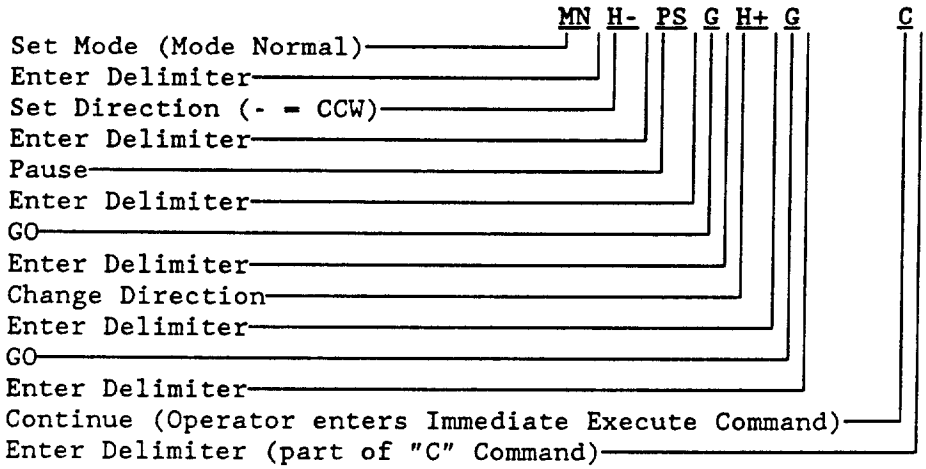
Function: Set Loop; move; wait; repeat cycle.



Sample Command: D1476 L265 G T2 N

Example Seven: Pause Command with Continue

Function: Change Mode and direction: set Pause; execute, change direction and execute when operator sends Continue



Sample Command: MN H+ PS G H- G C

5.4. STATUS REQUEST COMMANDS

You can determine the status of the Indexer using the Status Request commands summarized in Table 5-4. In reply to a Status Request command, the Indexer will transmit the responses. The format and interpretation of the responses to Status Request commands are described in sections 5.4.2 through 5.4.16.

The ordinary way to determine what the indexer is doing is to send it the appropriate status request. In some applications where a host computer requires timely status information, it is required to "poll" the indexer repeatedly to see if an expected event has occurred. This is often an undesirable way to use computer time, therefore the 372 supports two commands which allow it to "notify" a host computer when specific conditions occur.

The *"QS"* command allows the 372 to transmit a "message" when any of five different operating conditions occurs. See section 5.5.6

The *"CR"* command is buffered and may be embedded in a string of commands to send the host a carriage return indicating that the prior instruction has just been completed. See section 5.4.14

5.4.1. IDENTIFICATION OF UNITS (UNIT NUMBERS)

When a single communications port is used to control more than one indexer, much confusion is possible. This is because all indexers in a daisy chain receive and echo the same commands, and will execute them unless prevented from doing so. This becomes critical if any indexer is instructed to transmit some information itself.

To prevent all of the units on the line from responding to an instruction, that instruction must be preceded by the Unit Number of the designated indexer. If a command has no unit number, every indexer will execute it. Unit numbers are set to 1 at the factory.

To avoid having multiple indexers transmitting status information at the same time, status request commands are given the classification "Device Specific". This means the command must be preceded by a Unit Number. No indexer will execute a Device Specific command without its own Unit Number. The Device Specific commands include both Buffered and Immediate commands.

Each indexer should be set to a unique Unit Number. The Unit Number is entered by the operator as part of an individual command. Unit Numbers apply only to individual commands.

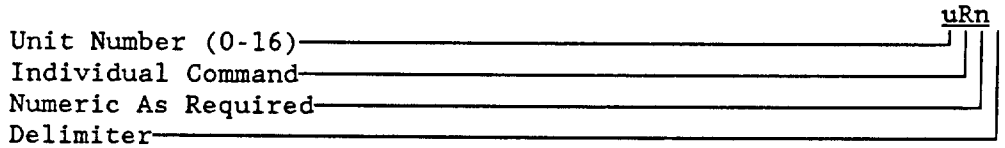
or example, *3A10* indicates the command is for Unit Number 3.

Table 5-4. Status Request Commands

Command Controls	Format of Entry	Description	Response Format	Class
Operating Status Requests	uB	Buffer Status	*a<CR>	I
	uR	Execution Status	*a<CR>	I
	uRA	Limit Switch Status	*a<CR>	I
	uRC	Closed Loop Status	*a<CR>	I
	uTS	Trigger Status	xxx	I
	uW	Position:		I
		uW1 binary	bbbb	I
		uW2 Hexadecimal	*hhhhhhhh<CR>	I
		uW3 Hexadecimal	*hhhhhhhh<CR>	I
Buffered Status Requests	uP	Incremental Position (decimal value)	±nnnnnnnn<CR>	B
	uPB	Incremental Position (binary)	bbbb	B
	uPR	Absolute position (decimal value)	±nnnnnnnn<CR>	B
	uX1	Cumulative position (decimal value)	±nnnnnnnn<CR>	B
	uX1B	Cumulative position (binary)	bbbb	B
	uX2	Cumulative Encoder position (decimal)	±nnnnnnnn<CR>	B
	uCR	Command Execution	<CR>	B
Setup Status Requests	uDR	Default Parameters	see 5.4.15	I
	uXU	Sequence Contents	see 5.4.16	I
	uXN	Sequence Memory	see 5.4.16	I
	uXV	Memory Integrity	see 5.4.16	I
	uSR	Switch status 1	xxxxxxxx<CR>	I
	uOR	Switch status 2	xxxxxxxx<CR>	I
	uQR	Signal status	xxxxxxxx<CR>	I
	uFR	Encoder Function	xxxxxxxx<CR>	I
	uUR	Scale Factor	nnn<CR>	I
	uRV	Software revision	see 5.4.23	B

NOTE 1. All entries must be upper case, with a delimiter
"*nnn*" indicates numeric response
"*hhh*" indicates hexadecimal response
"*xxx*" indicates digital response, x=1 or 0
"*bbb*" indicates binary response
"*a*" indicates a single alpha character response
2. B = Buffered command that is executed in sequence.
I = Immediate command that will be executed as soon as a delimiter is entered.

All status requests must include the Unit Number. When the Unit Number is used, an individual command has the syntax shown below:



5.4.2. FORMAT OF STATUS REQUEST RESPONSES

Except for the *DR* and *XU* commands, the status request command response will be in one of the following formats:

a. A set of three characters: an asterisk (*), an @ sign or letter, and a carriage return (<CR>). Typical examples are **A<CR>*, **M<CR>*, **@<CR>*. This format is used for status reports other than position reports. A special case is used when the Indexer transmits a single carriage return.
On a remote terminal these returns would be displayed as **A*, **M*, **@*, respectively. If a computer is used to control the Indexer, the computer would decode the responses and present the information for the operator or use the data to command additional operations. Refer to section 5.4.3 for a description of the interpretation process for this type of return.

b. A set of ten characters for position reporting consisting of a plus or minus sign (+ -), eight numbers, and a carriage return. On a remote terminal these returns would be displayed as *+12123434*, *-98765432*, etc. Refer to section 5.4.11 for a description of the interpretation process for this type of return. (*P*, *PR*, *X1*, and *X2* commands)

c. A set of eight characters representing eight hexadecimal characters that may be used for the relative position report obtained in response to the u*W2* or u*W3* status request commands. If a remote terminal is used, the operator must interpret this return to determine the equivalent decimal value. If a computer is used, the computer would have to perform the interpretation. A typical example of this type of return is **000433AE<CR>*. Refer to section 5.4.12 for a description of the interpretation process for this type of return.

d. A set of four 8-bit characters representing a 32-bit binary word can be used for responses to buffered position report request commands. The four bytes must be concatenated to determine the binary value of the response. If a remote terminal is used, the operator will be unable to interpret this return to determine the equivalent decimal value. If a computer is used, the computer

would have to perform the translation operation. Typical examples of this type of return (if displayed by a remote terminal) are **x/&d, OH#!, ABCD**, where the characters can be any one of the full set of ASCII characters, including nonprintable characters. Refer to section 5.4.13 for a description of the interpretation process for this type of return.

e. Four digits with a value of 0 or 1 followed by a carriage return (Trigger status).

f. Three decimal digits followed by a carriage return (Scale Factor status only).

g. Eight digits with a value of 0 or 1 followed by a carriage return (Special status, Encoder function status).

Note

You may have to enter a line feed command to keep the responses separate. The carriage return alone will only return the cursor to the start of the line. All Status Requests except binary reports are followed by a carriage return character.

The *DR* command response provides a large amount of information and is primarily a setup and test function rather than an operational status request. See section 5.4.15. The *XU* command response is a direct playback of buffered indexer commands that have been previously entered. The other "*X*" responses do not conform to the above formats either. See section 5.4.16.

5.4.3. INTERPRETING RESPONSES OF THE FORM *A<CR>

When you connect a "Full Duplex" terminal to the Indexer RS-232C interface, any character command you enter on the remote terminal keyboard shows on the terminal display. The displayed character does not originate within the terminal. The displayed character is transmitted from the Indexer back to the terminal (echoed back) so that the operator can verify what command the Indexer received in response to the keyboard entry.

An asterisk (*) is placed in front of the responses to many Status Request commands to differentiate these responses from Echoed commands. The carriage return at the end of the response message serves the same purpose as the command delimiter and signifies the end of the response.

The characters that are transmitted by the Indexer are represented by an 8-bit (or 7-bit) binary code, the ASCII code. For example:

Carriage return = 0000 1010 (13)

* = 0010 1010 (42)	1 = 0011 0001 (49)	@ = 0100 0000 (64)
+ = 0010 1011 (43)	2 = 0011 0010 (50)	A = 0100 0001 (65)
- = 0010 1101 (45)	3 = 0011 0011 (51)	B = 0100 0010 (66)

The decimal equivalent of each binary code is shown in parentheses.

Single character (@ or letters) replies have a specific meaning for each type of status request. You must know the status request command that caused the reply before you can decode the meaning of the single character A specific response table for each status request is indicated in this manual for each single character response. See sections 5.4.4 through 5.4.8.

5.4.4. BUFFER STATUS RESPONSE -- uB Command

Format of response --*A<CR>*

B<CR> Buffer 90% Full
R<CR> Buffer less than 90% Full

5.4.5. INDEXER STATUS RESPONSE -- uR Command

Format of response --*A<CR>*

B<CR> = Busy, No Attention
C<CR> = Busy, Attention
R<CR> = Ready, No Attention
S<CR> = Ready, Attention
F<CR> = Sequence in process, No Attention
G<CR> = Sequence in process, Attention

Busy and Ready refer to move execution, pauses, loops, and time delays. For Preset mode, Busy means that the Preset move is still in progress. The Continuous mode will give a Busy reply when the motor is being either accelerated or decelerated, and a Ready reply while running at the designated velocity.

Attention is indicated when an End-of-Travel limit switch was tripped in the previous move, or if a Stall condition exists. To determine which, use the **RA**, **RB**,or **RC** commands. See sections 5.4.6 through 5.4.8.

5.4.6. LIMIT SWITCH STATUS -- uRA Command

Format of response --**A<CR>*

The reply indicates both the current state of the Limit Switch inputs and their effect on the previous move, if any. Refer to the *RA* Response Table 5-5.

5.4.7. PAUSE STATUS -- uRB Command

Format of response -- **A<CR>*

This command indicates what, if anything, is inhibiting indexer execution. This command is useful when the 372 seemingly refuses to do anything, and a Limit switch condition is not indicated. It provides status information regarding the Pause, Hold, and Trigger commands, as well as the Shutdown output and Stop input.

Refer to the *RB* Response Table 5-6.

Table 5-5. *RA* Response -- Limit Switch Status

Response Character	Last Move Terminated by		Current Move Limited by	
	CW Limit	CCW Limit	CW Limit	CCW Limit
@	NO	NO	NO	NO
A	YES	NO	NO	NO
B	NO	YES	NO	NO
C	YES	YES	NO	NO
D	NO	NO	YES	NO
E	YES	NO	YES	NO
F	NO	YES	YES	NO
G	YES	YES	YES	NO
H	NO	NO	NO	YES
I	YES	NO	NO	YES
J	NO	YES	NO	YES
K	YES	YES	NO	YES
L	NO	NO	YES	YES
M	YES	NO	YES	YES
N	NO	YES	YES	YES
O	YES	YES	YES	YES

Table 5-6. *RB* Response -- Pause Status

Response	Shutdown Output	Pause *"PS"*	Hold *"U"*	Waiting for Trigger
@	OFF	NO	NO	NO
A	ON	NO	NO	NO
B	OFF	YES	NO	NO
C	ON	YES	NO	NO
D	OFF	NO	YES	NO
E	ON	NO	YES	NO
F	OFF	YES	YES	NO
G	ON	YES	YES	NO
H	OFF	NO	NO	YES
I	ON	NO	NO	YES
J	OFF	YES	NO	YES
K	ON	YES	NO	YES
L	OFF	NO	YES	YES
M	ON	NO	YES	YES
N	OFF	YES	YES	YES
O	ON	YES	YES	YES
S	STOP Input is Asserted			

5.4.8. CLOSED LOOP STATUS -- uRC Command

The "*RC*" command has the same response format as "*RA*" and "*RB*". The four conditions for which status is indicated are:

1. Static Position Loss - In this condition, the indexer has detected motion of the load while the motor was stopped. The indexer was not able to correct the position, resulting in Position Maintenance being disabled.

2. Post Move Position Loss -- In this condition, the indexer has detected a deviation between actual and desired position at the end of a move which exceeds the backlash/deadband parameter. This may involve a Stall, or slipping of the load short of a Stall.

3. Homing Function Failure -- In this condition, the indexer has encountered both End-of-travel limits or one of several possible Stop commands or conditions. Go Home motion was concluded, but not at Home.

4. Stall -- In this condition, the indexer has detected a deviation between motor and encoder position larger than one pole of the motor while running, or a deviation larger than that plus the backlash parameter following a direction change.

Table 5-7. *RC* Response

Response Character	Static Position Loss Between Moves	Position Loss Detected	Go Home Sequence Terminated Normally	Stall During Last Move Attempt
@	NO	NO	YES	NO
A	NO	NO	YES	YES
B	NO	NO	NO	NO
C	NO	NO	NO	YES
D	NO	YES	YES	NO
E	NO	YES	YES	YES
F	NO	YES	NO	NO
G	NO	YES	NO	YES
H	YES	NO	YES	NO
I	YES	NO	YES	YES
J	YES	NO	NO	NO
K	YES	NO	NO	YES
L	YES	YES	YES	NO
M	YES	YES	YES	YES
N	YES	YES	NO	NO
O	YES	YES	NO	YES

5.4.9. TRIGGER INPUT STATUS -- uTS Command

Format of response -- *bbbb<CR>* (b = 0 or 1).

The four digits indicate the status of trigger inputs 1, 2, 3, and 4, respectively. (0 = off, 1 = on)

5.4.10. POSITION REPORTING

Position reports are based on counting the number of pulses sent to the motor/drive and are independent of the scale factor.

Three different basic position report request commands are available as follows:

a. u*W1* , u*W2* , or u*W3* -- Immediate commands that provide the current incremental distance from the position where the motor was last at rest. As immediate commands, these respond regardless of any other instructions that are being executed, and are typically used to obtain position data "on the fly" (while moving).

The *W1* command is fastest, providing a four character response (bbbb) which is interpreted as a 32 bit binary number (see section 5.4.13). This value, like the *W2* response is unsigned (always positive), and gives no information as to the direction of motor rotation. *W3* supercedes the *W2* command and reports incremental position in two's complement signed notation. Counterclockwise is negative. Both *W2* and *W3* respond in hexadecimal format (**hhhhhhhh<CR>*) (see section 5.4.12). The *W2* command is not operative in Encoder Step Mode, *W3* must be used.

All three commands are Immediate and Device Specific.

CAUTION

Do not use the four digit binary format when multiple 372s are daisy-chained on a single RS-232C Interface.

b. u*P* or u*PB* -- Buffered commands that provide the incremental distance travelled during the last move, where the *P* command provides the decimal-values format (±ddddddddd<CR>) (see section 5.4.11) and the *PB* command provides the four character binary-number format (see section 5.4.13).

c. u*PR* -- Buffered command that provides absolute position; cumulative count from power-up, from the position at which a *PZ* command is given, or the Home position. The *PR* command provides decimal values the same as the *P* command.

d. u*X1* or u*X1B* -- Buffered commands that provide cumulative position (cumulative count from power-up or from the position at which an *X0* command is given) where the *X* command provides decimal values the same as the *P* command and the *XB* command provides four character binary numbers the same as the *PB* command.

All the above buffered position reports will report in either Encoder steps or motor steps depending on the mode established with the *FS* command. In Encoder mode, these commands are likely to report the last command position, even if external forces have since caused motion of the load. To get up to date Encoder position information, the "*X2*" command should be used.

e. u*X2* -- Buffered command that provides updated cumulative encoder position (cumulative count from power-up or from the position at which an *X0* command is given) where the *X2* command provides decimal values the same as the *X1* command, in Encoder Step mode only.

Your computer must have an 8-bit capability to use the *W1*, *PB*, and *X1B* options. Only single axis systems should use binary reports.

The rate at which a new position can be reported depends upon the baud rate selected and can be up to 500 times a second (every 7 to 8 msec.)

5.4.11. INTERPRETING DECIMAL POSITION REPORTS

The *X1*, *P* and *PR* instructions provide the position report as a decimal value. The report has the form ±dddddddd<CR> consisting of ten characters; an eight digit number preceded by sign (+/-), and followed by a carriage return. The value represents either cumulative position in steps (X1), the number of steps generated in the last move (*P*), or the absolute distance from Home. The sign indicates either which side of the zero (*X1*) or Home (*PR*) position the motor is on, or the direction of motion of the last move (*P*). Neither instruction provides current information when the motor is moving in the CONTINUOUS mode.

The "*X2*" command is valid only when ENCODER STEP Mode is used. Then, it is identical to the *X1* command except that the encoder position is read and the cumulative encoder position is updated before any response is sent. It may be used only while the motor is not moving. The zero reference position may be different than that used by the "*PR*" command.

The "*X0*" command sets the position counter to zero for "*X1*" and "*X2*" commands, the "*PZ*" command or the Homing function will do the same for the "*PR*" command.

5.4.12. INTERPRETING HEXADECIMAL POSITION REPORTS

This form of position report (***hhhhhhhh<CR>**) consists of an asterisk followed by eight hexadecimal characters; 0 through 9 and A through F. The position report is followed by a carriage return.

The decimal value of the hexadecimal expression can be determined using the technique demonstrated in figures 5-2 and 5-3 below. The *W2* report is always positive regardless of direction. The *W3* response is in "two's complement" notation reflecting direction; negative numbers imply counterclockwise motion. Both commands are designed for use in a computer controlled situation where the computer can translate from the hexadecimal. The *W3* command must be used in Encoder mode.

5.4.13. INTERPRETING BINARY POSITION REPORTS

This form of position report (**bbbb**), consists of four bytes that must be linked together (concatenated). Assume that the response is equivalent to the ASCII characters ^A#0/<CR> ("^A" refers to "Control A", an unprintable character). The actual binary code for this would be assembled as

```
     ^A       #       0       /
00000001001000110011000000101111
```

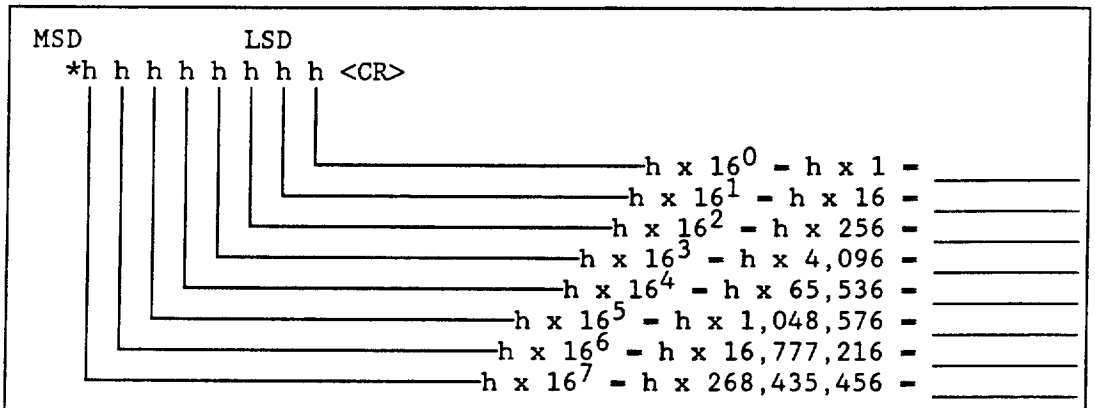
This code would then have to be interpreted by the computer. A terminal will interpret the four bytes as ASCII characters. The operator may not be able to read the response, since many characters are unprintable. The four characters must be converted to their ASCII code numbers, and multiplied by a power of 256. The first character received is the most significant byte. Refer to Table 5-8 for a conversion technique.

Table 5-8. Binary Position Conversion

Response	ASCII value	character multiplier		character value
^A#0/				
	47	x 1	(256 to the 0 power) =	57
	48	x 256	(256 to the 1st power) =	12,288
	35	x 65,536	(256 to the 2nd power) =	2,293,760
	1	x 16,777,216	(256 to the 3rd power) =	16,777,216
			Total =	19,083,321

The *PB* and *X1B* responses provide direction information. If the direction of the last move was counterclockwise (*PB*), or the current cumulative position is negative (*X1B*), then the most significant bit of the 32 bit response will be set. That is, the first (most significant) character will have an ASCII code number greater than 127.

Figure 5-2. *W2* and *W3* Response Interpretation



where h can be one of the following values

Decimal Value	Hexadecimal Value (h)	Decimal Value	Hexadecimal Value (h)
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

Example:

Assume the response was *000433AE. The decimal value would be:

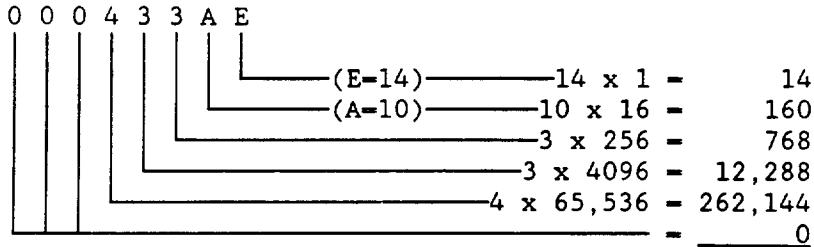
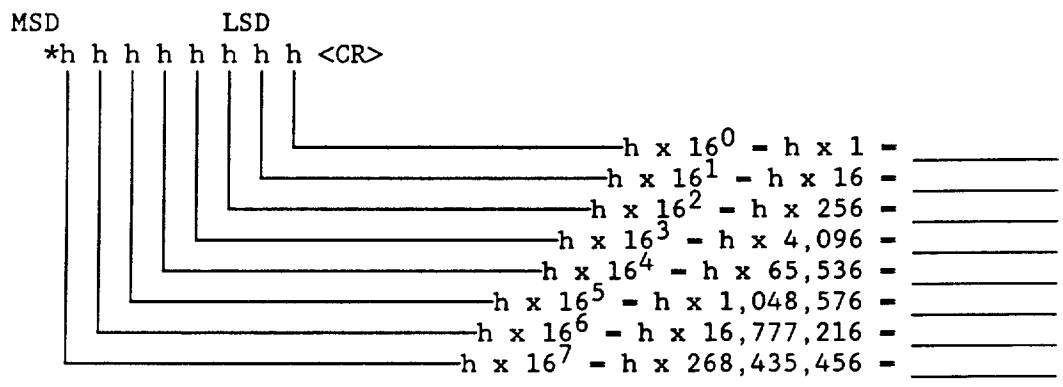


Figure 5-2. *W2* and *W3* Response Interpretation



where h can be one of the following values

Decimal Value	Hexadecimal Value (h)	Decimal Value	Hexadecimal Value (h)
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

Example:

Assume the response was *000433AE. The decimal value would be:

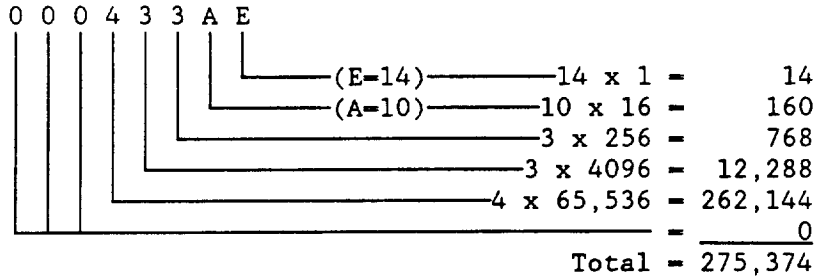


Figure 5-3. *W3* Response Interpretation

If the first digit of the response is an "F" then response represents a "two's complement" negative number. Any response but:

Fhhhhhhhh

should be interpreted per Figure 5-2. There are several ways to convert a two's complement hexadecimal number to decimal:

- The binary approach:

  1. Convert the hexadecimal response to binary form.
  2. Complement the binary number.
  3. Add 1 to the binary result.
  4. Convert the binary result to decimal value with a minus sign placed ahead of the decimal value.

- The computer approach:

  Subtract the hexadecimal number from 16^8 (2^{32}) (4,294,967,296)

- The easy way:

  1. Chop off all the leading "F"'s, then convert to decimal
  2. Subtract the next largest power of 16.

Example: the indexer responds to *W3* as follows:

*FFFF9E58

1. Chop off the F's:	9E58 hex	=	40,536
2. Subtract from 16^4	10000 hex	=	65,536
	Result:	=	-25,000

5.4.14. COMMAND EXECUTION STATUS -- uCR Command

This status request command is a special case where the indexer transmits a carriage return alone. When the CR command is included in a sequence of commands, the indexer will send a carriage return character after prior commands have been executed. The *CR* command is used to notify a host computer that a sequence of instructions has been completed. For example:

MC A10 L V4 G T2 V0 G 1CR N

In this example, the motor repetitively starts, moves at the designated speed for two seconds and stops. The carriage return is transmitted each time the Loop is executed to indicate completion of the motion.

5.4.15. DEFAULT PARAMETER STATUS -- uDR Command

This status request is typically used only in a setup situation. It provides complete data on indexer configuration. The response to a *DR* command for a factory configured 372 is as follows:

```
MN   H+   GH+  IM1   CM0  MV7  CG4  MS1
FS00000000   QS00000000    SS00000000
OS00000000   MR10          ER4000
D25000       US1           DW250
V.2          LA999         VS0
A100         GH.2          GA100
JV.1         JA100         XP0
```

This parameter report reflects the current setting of all motion parameters. The report reflects the setting of non-volatile memory defaults ONLY if no parameters have been changed since power up or reset. Parameter changes can be stored using the Save Defaults command: "*SA*"

The parameter commands from the *DR* report are described briefly below. Refer to the description for each command in the sections listed on the right.

MN	- motion mode	5.2.1
H+	- motion direction	5.2.2
GH+	- Go Home direction (Off Line)	3.4.2
IM1	- Input Mode	5.5.7
CM0	- Position Maintenance correction mode	5.2.6
MV7	- Position Maintenance velocity limit	5.2.6
CG4	- Position Maintenance correction gain	5.2.6
MS1	- Messages on or off	5.5.8
FS00000000	- Enable various encoder functions	5.2.6
QS00000000	- Transmit a signal on various conditions	5.5.6
SS00000000	- Enable various configuration options	5.5.4
OS00000000	- Enable various other configuration options	5.5.5
MR10	- Match the motor resolution	5.2.3
D25000	- Move distance	5.2.2
US1	- Distance units	5.2.3
V.2	- Move velocity	5.2.2
LA999	- Limit stop deceleration rate	5.2.2
VS0	- Stop/Start rate for low resolution motors	5.2.2
A100	- Move acceleration rate	5.2.2
GH.2	- Go Home velocity (Off Line)	5.2.2
GA100	- Go Home acceleration rate	5.2.2
JV.1	- Jog velocity	5.2.2
JA100	- Jog acceleration	5.2.2
XP0	- Sequence number for automatic execution	5.6.3

5.4.16. SEQUENCE REPORTS uXU*n*, u*XN*, u*XV* Command

-The *XU* command lists all the valid commands in a sequence. The variable parameter "*n*" above designates which sequence is to be listed. The response to this command depends on how the specified sequence has been programmed. The sequence will be listed as it was programmed. An invalid or unprogrammed sequence will yield an error message or no response.

-The *XN* command reports how much sequence memory is unused, and which sequence numbers have already been assigned. This command is useful for determining whether the 372 has room for a new sequence, and what number may be assigned to it.

The *XN* response format is to list each number of each stored sequence in a two digit format with a space between. At the end of this list, an asterisk and another number appear, indicating how many blocks of sequence memory remain unused. Each block is 32 characters in size.

Example: 01 02 07 *43

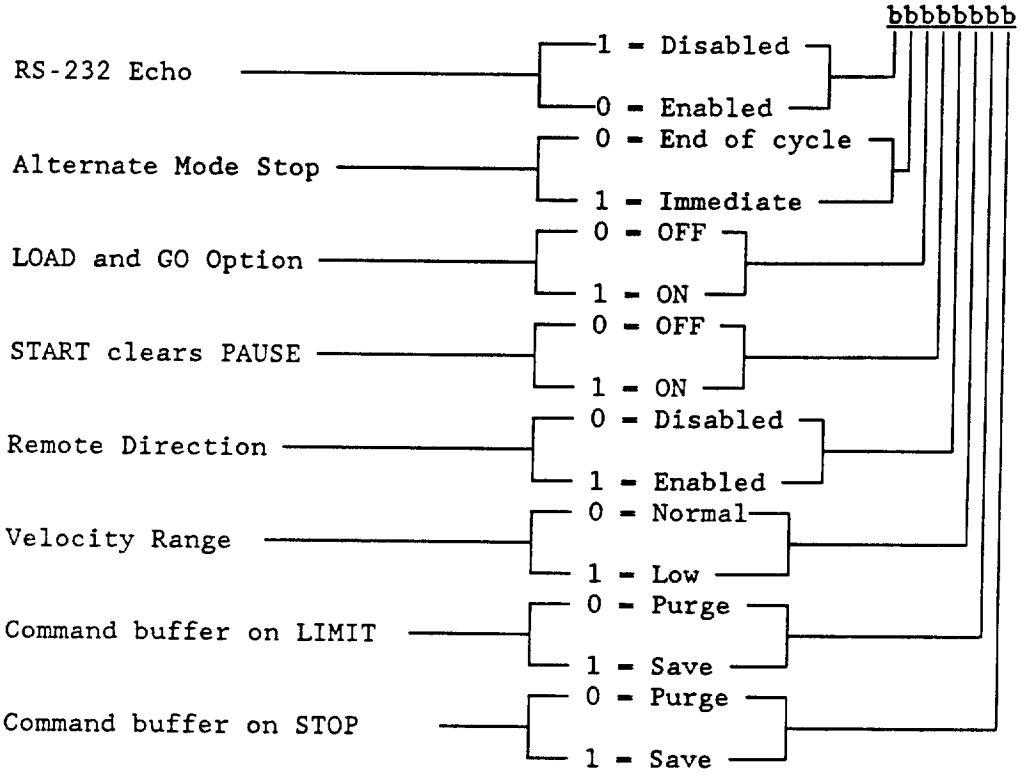
(sequences 1, 2, and 7 are used, 43x32=1376 bytes or characters left)

-The *XV* command is used to verify the integrity of the non-volatile memory devices used in the 372. Because these "EEPROM" devices have a finite capacity for rewriting, their integrity should be checked periodically in applications where data is constantly being stored.

The response to the *XV* command is an asterisk "*" if memory passes the test. Otherwise, an error message is generated.

5.4.17. SOFTWARE SWITCH STATUS -- uSR Command

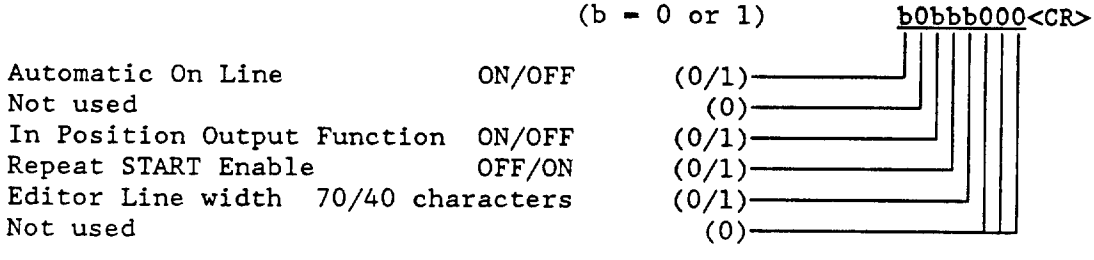
The format of the response is bbbbbbbb<CR>, where b = 0 or 1. The response can be interpreted from the diagram that follows.



Refer to section 5.5.4 on the *SS* command for additional details.

5.4.18. OPTION SWITCH STATUS -- uOR Command

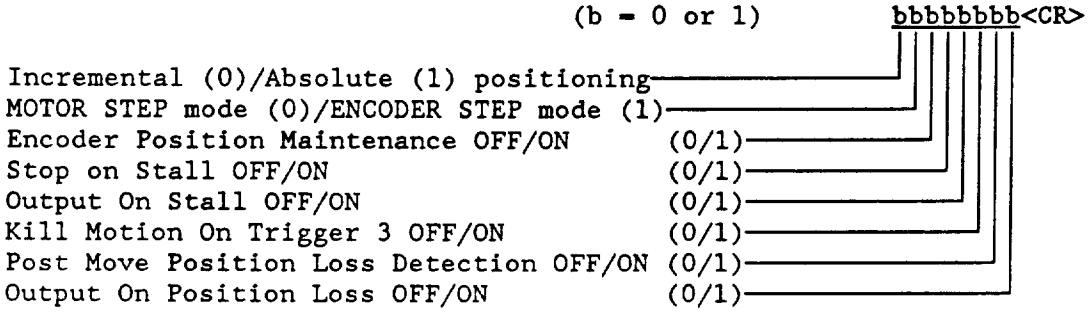
This command is a request for the status of the functions set or cleared by the "*OS*" command. The reply is eight digits, five of which represent the functions listed below from left to right:



Refer to section 5.5.5 on the *OS* command for more details.

5.4.19. ENCODER FUNCTIONS STATUS -- uFR Command

This command is a request for the status of the functions set or cleared by the "*FS*" command. The reply is eight digits, each representing one of the listed functions as shown below:



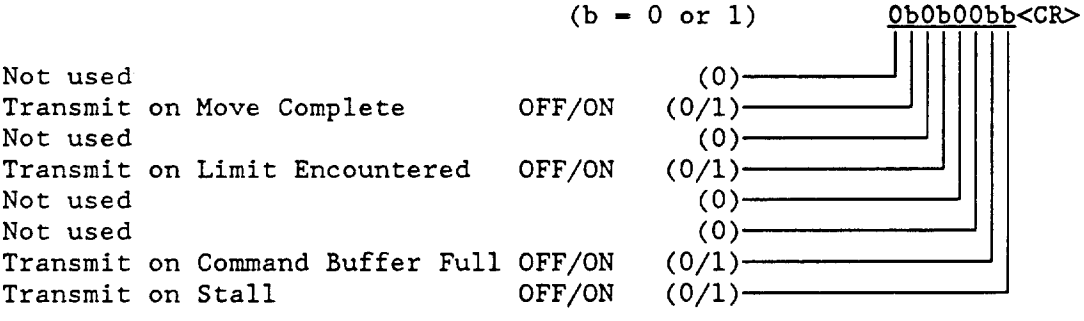
Example: the 372 responds "00011000<CR>"

It is set to stop and signal if the motor stalls.

Refer to section 5.2.6 on the *FS* command for details about each of these optional conditions.

5.4.20. SIGNAL ENABLE STATUS -- uQR Command

This command is a request for the status of the functions enabled or disabled by the "*QS*" command. The reply is eight digits, indicating which of five signal functions are active, as shown below:



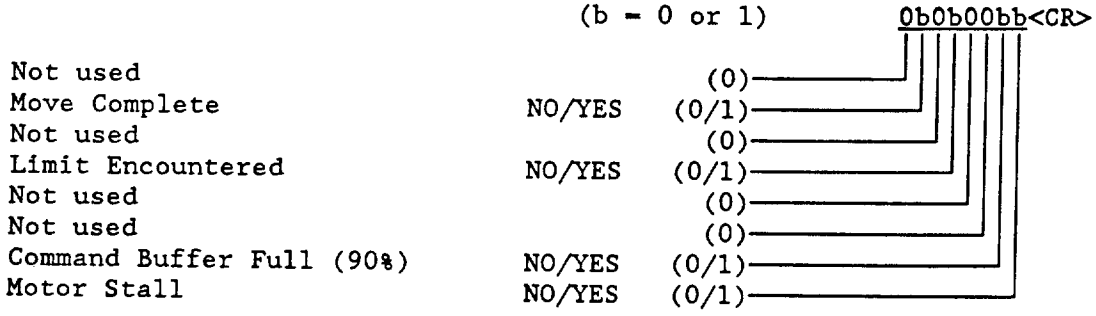
Example: the 372 responds "00010001<CR>"

It is set to send a signal if the motor stalls or hits a Limit.

Refer to section 5.5.6 for more details on the *QS* command.

5.4.21. SIGNAL SOURCE STATUS -- uQI Command

This command is a request for the status of the functions which can cause the *QS* command signal to be sent. The reply is eight digits, each of functions, left to right, 1-8 as shown below:



Refer to section 5.4.6 on the QS command.

5.4.22. SCALE FACTOR STATUS -- uUR Command

The format of the *UR* response is *nnn*<CR>. The "nnn" will be a three digit value between 1 and 255. This number reflects the position scale factor set using command *US* (section 5.2.3). The actual number of steps sent to the motor/drive during a Preset move will be the current Distance parameter setting multiplied by this scale factor.

5.4.23. SOFTWARE REVISION -- uRV Command

This command returns the part number and revision level of the software which the indexer is running. The response is 14 characters indicating the part number and revision level as follows:

*1_92_06883-A<CR>

This command is Bufferred and Device Specific.

5.5. GENERAL PURPOSE COMMANDS

These commands are summarized in Table 5-9 and described in sections 5.5.1 through 5.5.7.

Table 5-9. General Purpose Commands

Command Controls	Format of Entry (Note 1)	Description of Command	Class (Note 2)
Communi-cation Interface	**E**	Enable Communications	I
	F	Disable Communications	I
Keyboard Entries	**^H**	(Control H) Backspace and Delete character	I
Program-mable Outputs	O*bb*	Set programmable output	B
Position Scale Factor	US*nnn*	Position multiplier	B
Special Functions	SS*ab*	Set Special Function 1 (a = A to H, b = 0 or 1)	B
	OS*ab*	Set Special Function 2 (a = A to H, b = 0 or 1)	B
Signal On Condition	QS*ab*	Set Message Function (a = A to H, b = 0 or 1)	B
Input Mode	IM*n*	Set Input Mode (n = 1 to 4)	B
Messages	MS*n*	Operator Messages (n = 0, 1, or 2)	I

NOTE 1. All entries must be upper case, with a delimiter.
"*nnn*" indicates numeric parameter
"*a*" indicates a single alpha character parameter
"bbb" indicates digital parameter, x=1 or 0

2. B = Buffered command that is stored and executed in sequence.
I =Immediate command that will be executed as soon as a delimiter is entered.

5.5.1. COMMUNICATIONS INTERFACE ENABLE/DISABLE -- E and F

In the factory configuration, the 372 is "On Line" following power on (or Reset). It is listening for commands, and echoing every character it receives. The "*F*" command instructs the indexer to go "Off Line". In this mode, it still echoes all characters it receives, but ignores commands. Only an "*E* " command will result in any action.

When the indexer is off line, AUXILIARY command inputs are active. The START input may be used to initiate Continuous, Alternating or incremental Preset moves. The Go Home input is also operational. Move parameters may be loaded On-Line before the "*F* " command is issued.

The external ON LINE indicator will come on in response to the "*E* " command, and goes off following an "*F* " command.

The *OS* command may be used to program the 372 to come up in the Off Line state. The *SS* command may be used to stop the indexer from echoing characters. See sections 5.5.4 and 5.5.5 below.

5.5.2. BACKSPACE -- ^H

The "^H" command backspaces one character. A new character may be entered at that position to replace the existing character. (^H indicates that the CONTROL or CTRL key is held down when the H key is pressed.)

The effect of this command character is to cause the 372 to back up one character in the command buffer regardless of what appears on the terminal. On some terminals pressing the BACKSPACE key will produce the same character.

5.5.3. PROGRAMMABLE OUTPUTS -- O*bb* Command

Two outputs are available that can be remotely controlled. Either output can be set low or high. Both outputs will sink or source 30 mA.

The "*b*" above designates the state of outputs 1 and 2 in that order (*b*= 0, 1, or X). When a "0" is specified for an output it is set low (sink), a "1" sets it high (source). An "X" leaves the prior output state intact.

NOTE: To insure that the outputs perform as expected, alternate output function assignments should be disabled. These functions include "Output on Stall" (*FSE*) and "Moving/Stopped" (*OSC*).

Example: *O1X* (set output 1 high, don't change output 2)

Note that each of the two programmable outputs may be assigned alternate functions. Output 1 may be programmed to indicate closed loop status with the *FS* command. See section 5.2.6. Output 2 is programmed at the factory to indicate whether or not the motor is moving. This function can be turned off with the *OS* command.

5.5.4. SOFTWARE SWITCH -- SS Command

This command sets up the functions in the table that follows. The form of the command is "*SSab*" where "a" indicates one of the letters from the table below is to be entered. Enter b = 0 or 1 as shown in the table below to obtain the desired result.

		b	
a	Parameter	0	1
A	Interface Echo	YES	NO
B	Alternate Mode Stop	End of Cycle	Immediate
C	LOAD and GO Option	OFF	ON
D	START signal Clears Pause	NO	YES
E	Off Line Direction Parameter	Internal	Remote
F	Velocity Range	Normal	Low
G	Save Command Buffer After Limit Stop	NO	YES
H	Save Command Buffer After STOP Command	NO	YES

Notes:

- A Inhibits echo. Only responses from the non-echoing unit are transmitted. Only the last unit in a multiaxis system can be set for no echo, no other 372 in the chain can report status.
- B Controls whether or not the indexer returns to its starting point when a Stop is received.
- C LOAD AND GO option is discussed in section 5.3.2.
- D Allows operator to clear a Pause using the Remote START input in place of the CONTINUE command.
- E Allows the direction of motion for Off Line Homing and indexing functions to be controlled by the Direction input
- F Reduces or increases the velocity range by a factor of 10.
- G, H Determines buffer response on Limit or Stop. b = 1 saves the contents of the Indexer command buffer.

Example: *SSF1* (switch to Low velocity range)
SSA0 (switch interface Echo back on)

5.5.5. OPTION SWITCH -- OS Command

This command sets up the functions in the table that follows. The form of the command is "*OSab*" where "a" indicates one of the letters from the table below is to be entered. Enter b = 0 or 1 as shown in the table below to obtain the desired result.

a	Parameter	b = 0	b = 1
A	Power Up On Line/Off Line	ON	OFF
B	Not used		
C	Output 2 Indicates Moving/Stopped	YES	NO
D	Continuous START Repeat	Disabled	Enabled
E	Editing mode display width (in characters)	40	70
F,G,H	Not Used		

Notes:

A *OSA1* - Configures the unit to run from the AUXILIARY inputs on power up, in the Off Line state. An *E* command is required to give it RS-232 commands.
OSA0 - Configures the unit to come up On Line. (default)

C *OSC1* - Disables the "Moving/Stopped" function on Output 2.
OSC0 - Enables the "Moving/Stopped" function. (default)

D *OSD1* - Configures "Level sensitive" START input: Move or sequence will repeat as long as START is asserted.
OSD0 - Configures "Edge sensitive" START input: START input must go from Off to On to initiate action. (default)

E *OSE1* - Limits the Sequence editing display width to 40 characters to accommodate terminals with 40 column width.
OSE0 - Switches display width to 70 characters. (default)

5.5.6. SIGNAL ON CONDITION -- QS Command

The "Signal on Condition" function is as close as the 372 can get to providing an interrupt to the Host computer when attention is required. The *QS* command allows the 372 to send a signal on one or more of the five conditions listed below. As such, the *QS* function waits in the background during normal operation until the specified condition occurs. At that point, the signal is transmitted.

The host computer may discover the cause of the signal using the *QI* command. See section 5.4.21.

The signal consists of an asterisk, the Unit Number, an exclamation mark, and a carriage return. The exclamation mark is not used anywhere else in the 372 syntax.

The signal is repeated each time the condition occurs until the signal function is switched off. The form of the command is "*QSB1*" to turn on the signal function, and "*QSB0*" to turn it off.

a	Parameter	b 0	1
A	Not Used		
B	Transmit Signal on Move Complete	Disable	Enable
C	Not used		
D	Transmit Signal on Limit Input	Disable	Enable
E	Not Used		
F	Not Used		
G	Transmit Signal on Command Buffer Full	Disable	Enable
H	Transmit Signal on Stall	Disable	Enable

5.5.7. INPUT MODE -- **IM** Command

This command is used to set the function of 12 programmable inputs on connectors AUXILIARY 1 and AUXILIARY 2. The input modes are defined in Section 3.5. The syntax for the command is:

IM1	(select Off Line discrete command functions)
IM2	(select Off Line discrete Sequence execute command)
IM3	(select Off Line BCD Sequence execute command)
IM4	(disable input functions for Input Mode 1, 2, and 3)

The Input Mode command is buffered.

5.5.8. MESSAGES CONTROL -- **MS** Command

The 372 has three levels of Message support for the operator or the controlling computer. The default level 2 provides full operator message support including error messages. Level 1 provides only error codes, and level 0 suppresses all messages. Command syntax is:

MS2	(error codes)
MS1	(full messages, default)
MS0	(no messages)

All message and error code responses begin with the same character sequence. Each message begins with a carriage return, linefeed, asterisk character, and the unit number. Following these characters is either an English message for Level 2, or an underline character, followed by "ER" and the error code in the case of Level 1.

Refer to Appendix B for a listing of Error codes and additional details of message syntax.

5.6. NON-VOLATILE MEMORY CONTROL COMMANDS

The following commands are used to store, edit, and execute sequences and setup parameters in the non-volatile EEPROM memory of the 372. Sequences and Stored Parameters are discussed in Section 3.3.

These commands are summarized in Table 5-10 and described in sections 5.6.1 through 5.6.3.

Note that the three Status Requests are described in section 5.4.16

Note also that the more radical commands that write to non-volatile memory are Device Specific. The Unit Number must precede these instructions (for safety).

Table 5-10. Memory Control Commands

Command Controls	Format of Entry	Description of Command	Class
Sequence Memory Status Requests	uXU*n*	Report Sequence *n*	I,D
	uXN	Report Memory Use	I,D
	uXV	Test Checksum	I,D
Sequence Edit	XD*n*	Enter or Edit Sequence *n*	I
	XT	Exit Edit Mode and Save[1]	I
	XK	Exit Edit Mode, Don't Save	I
	uXE	Erase a Sequence	I,D
	uZS	Erase All Sequences	I,D
Parameter Memory	SA	Save Current Parameters	I
	uZD	Restore Factory Defaults	I,D
Execute	XR*n*	Execute Sequence *n*	I
	XP*n*	Select Sequence *n* for automatic execution on Power up	I

NOTE 1. All entries must be upper case, with a delimiter. "*n*" indicates numeric parameter

2. I =Immediate command that will be executed as soon as a delimiter is entered.

5.6.1. PARAMETER STORAGE

The 372 has two commands that pertain to the maintenance of default parameters in non-volatile memory. In addition, the special "=" syntax allows saving single parameters in non-volatile memory. See 5.1.6.

The *SA* command is used to save the entire set of default parameters for automatic loading each time the power is turned on, or the units is reset. (The *Z* command resets the unit).

The Device Specific u*ZD* command restores all the default parameters in non-volatile memory to their factory configuration.

---

[1] Note: When nothing is entered in sequence "XT' will provide error message "cannot save while moving".

Refer to section 3.3.1 and the *DR* command (section 5.4.15) for a listing of default parameters and their factory values.

5.6.2. SEQUENCE EDITING

The *XD* command is used to enter a new sequence or to edit an existing one. The *XD* command includes the sequence number as the third character (and fourth, if any) of the command. Following such a command, the 372 prompts the operator for entry of the sequence commands. The commands may be typed straight in; the indexer will not execute any commands while in sequence editing mode. In this mode, the 372 provides several Control character commands that allow moving the cursor around within the sequence, and deleting or inserting characters. Control characters require that the Control (or "CTRL") key be held down while the alpha key is struck. These Control character functions are described below.

Two commands are provided to exit from the sequence editing mode.

The *XT* command exits from editing mode and stores the edited sequence in non-volatile memory.

The *XK* command exits from editing mode without storing. The edited sequence is abandoned unchanged.

Two additional commands are provided for removing unwanted sequences.

The *XE* command erases a sequence where the number of the sequence to be erased follows the command.

The u*ZS* command erases all the sequences stored in the unit. This command is Device Specific.

Editing Mode

The 372 Editing Mode is designed to accommodate portable terminal devices with limited display capability. As such, the display output of the editor restricts itself to a single display line, to accommodate displays that are either 40 or 80 characters wide. The *OSE* command is used to set display width, the default is for an 80 character device.

Following the *XD* command, the 372 displays the last 56 (or 26) characters of the sequence if the sequence is that long. The display is like a "window" and the sequence may be moved left and right under the window.

The Control characters for operating on the sequence are:

Space - Move the cursor to the right
^H - Move the cursor to the left (and exit insert mode)
^A - Move the cursor to the beginning of the sequence
^B - Move the cursor to the left side of the display
^F - Move the cursor to the right side of the display
^E - Move the cursor to the end of the sequence
^I - Enter insert mode
^D - Delete the cursor character (and exit insert mode)

Note: The Space character serves both as a delimiter and for cursor control. If a second delimiter is received while entering commands, the editor will exit insert mode and a control character ^I or ^E will be needed to commence command entry.

In editing mode, to add commands to the end of a sequence, type ^E and begin entering the commands. If the keyboard freezes, type ^E again to clear the fault.

To add commands in the middle of the sequence, use the control characters to move the cursor to the spot where insertion is required, and type ^I. The 372 displays a "$" character at the insertion point while insert mode is active. Command characters may then be typed in. The 372 will blank everything to the right of the cursor while insertion takes place. Any control character except ^D will cause an exit from insert mode.

To delete characters in the middle of the sequence, use the control characters to move the cursor to the spot where deletion is required, and type ^D. The character under the cursor will be deleted. The 372 will then rewrite the line, showing the deletion, but the cursor will no longer be in the position where the deletion took place. Even so, additional ^D's will continue to delete characters from the same position as the first.

5.6.3. SEQUENCE EXECUTION COMMANDS

The *XR* command is used to execute an existing sequence. The number of the sequence directly follows the command. The *R* command can be used to indicate whether a sequence is running or not.

Example: *XR7* (run sequence number 7)

The 372 can be configured to automatically run a sequence following power on or a reset. The *XP* command specifies which sequence is to executed automatically. The command has the same format as the *XR* command. The *DR* command response indicates which sequence is set up for automatic execution. The default is *XP0* meaning no sequence. The command *XP0* disable automatic sequence execution.

APPENDIX A: 372 CIRCUIT BOARD JUMPERS

The 372 has 13 jumper posts for plug on jumpers to configure the board. Most of these should be left alone, but three of them control functions that may be needed. Except for JU9, JU10, and JU13, these jumper positions are in pairs for selecting either one configuration or a single alternative. Default jumper locations are underlined in the list below.

JU1/JU2 - Motor/Driver Direction Output - Differential/Single Ended
JU3/JU4 - Motor/Driver Step Output - Differential/Single Ended

JU5/JU6 - RAM Size - DO NOT REMOVE
JU7/JU8 - EEPROM Size - DO NOT REMOVE

JU9 - Reserved - DO NOT REMOVE
JU10 - Reserved - DO NOT REMOVE

JU11/JU12 - Z Channel Encoder Input - Active Low/Active high

JU13 - Watchdog timer disable - Normally absent

APPENDIX B: ERROR CODES

When the 372 Full Message function is on (*MS1*), the 372 will catch most command entry errors and provide an error message explaining the problem in general terms. This results in more text than a computer application needs or wants.

The intermediate message mode is provided to allow a compuetr to analyze command errors without having to process text.

In the Error Code message mode, (*MS2*), error mesages have the following form:

*1_ER01<LF><CR>

where <LF> and <CR> are the line feed and carriage return characters.

The error codes include:

1. Illegal sequence number (1 to 48)
2. Sequence too long (245 characters)
3. Command Buffer Full (1200 characters)
4. Edit Mode while moving
5. Command Buffer Overrun (characters too fast)
6. EEPROM error detected (hardware failure)
7. Command Syntax error
8. Command illegal while moving
9. Insufficient Sequence memory
10. RAM error detected (hardware failure)
11. Non-existent Sequence number
12. Checksum error detected (hardware failure)
13. EEPROM Default error detected (factory defaults restored)

APPENDIX C: SAMPLE AUXILIARY CONNECTOR WIRING

Figure C-1: JOG and TRIGGER Connections

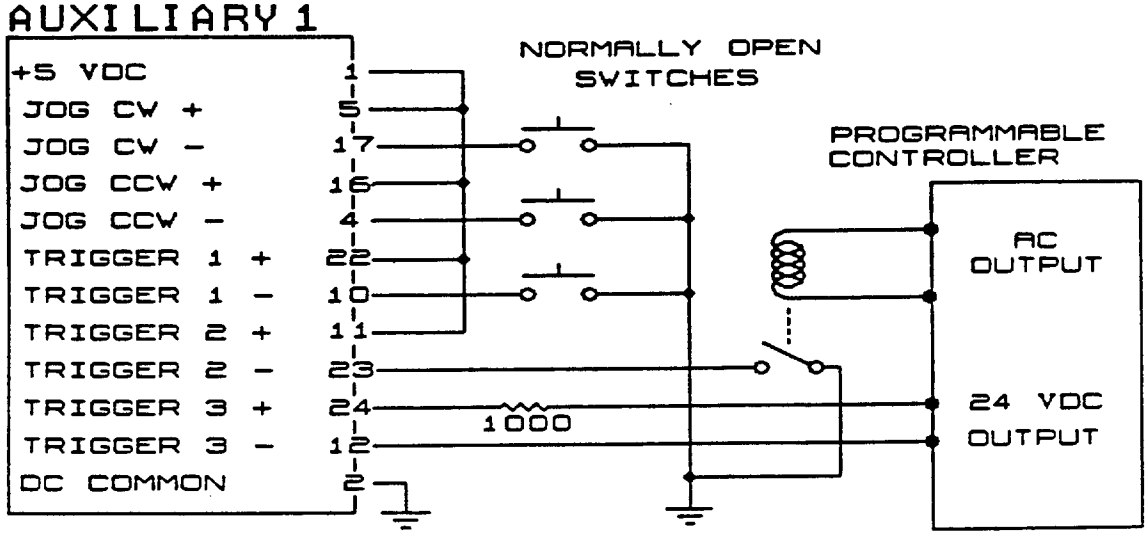
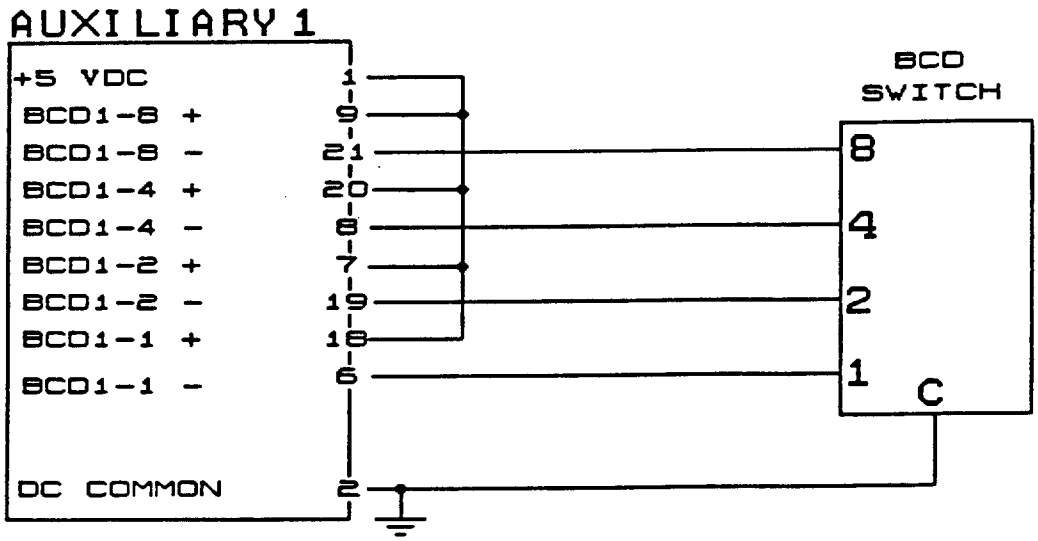


Figure C-2: Sample BCD Connections for Input Mode 3 and 4



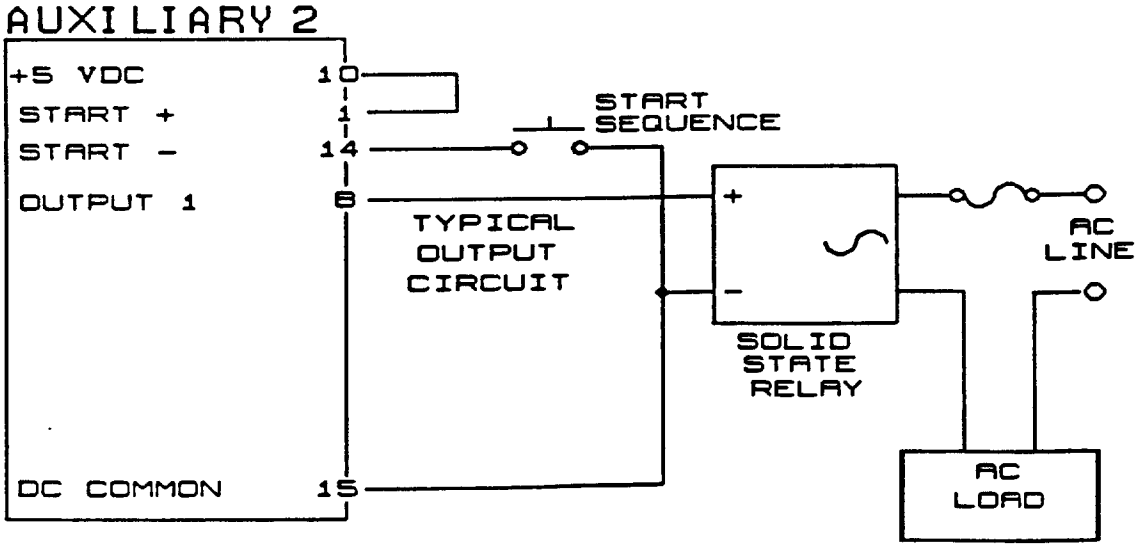
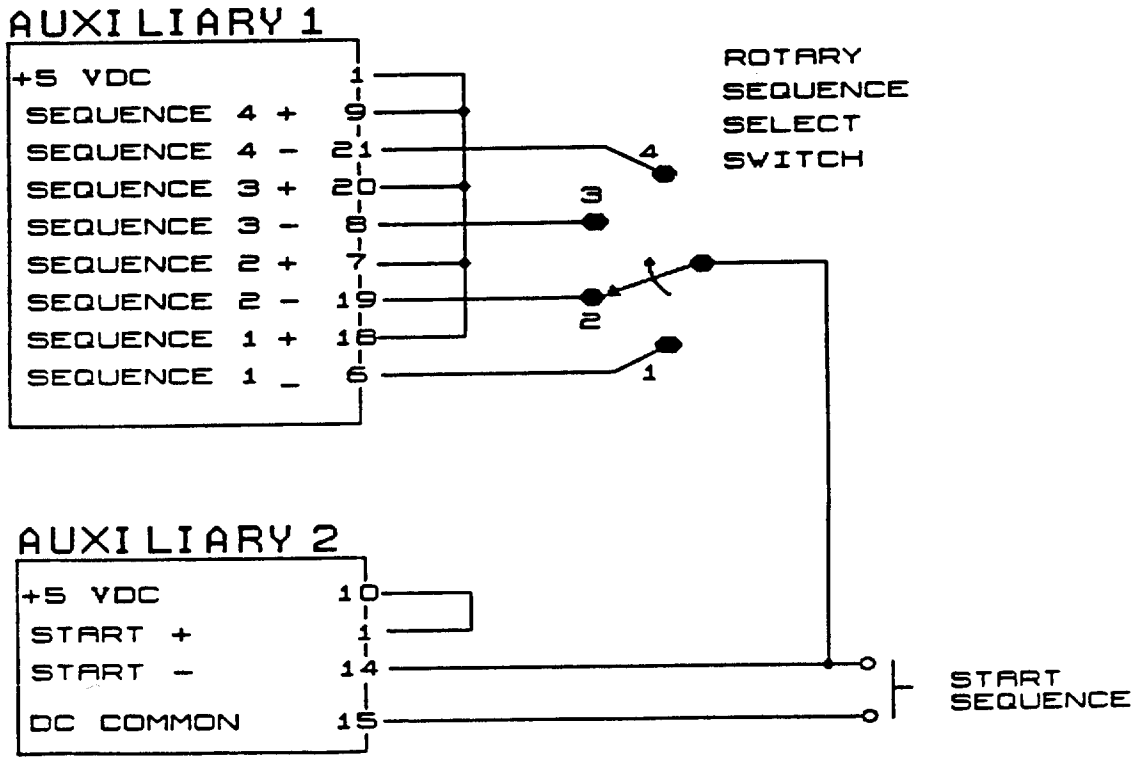


Figure C-3: Sample Wiring for Input Mode 2

Note: Connecting the Sequence Select signal through the START switch insures that the Sequence Select switch will not interfere with any Input Mode 1 functions such as Shutdown Request.



APPENDIX D: Summary of Connectors

AUXILIARY 1:

INPUT MODE 1	2	3	4
1. +5 VDC	<--	<--	<--
2. DC Common	<--	<--	<--
3. Shutdown +	Sequence 6 +	BCD2-2 +	<--
4. Jog CCW -	<--	<--	BCD2-8 -
5. Jog CW +	<--	<--	BCD2-4 +
6. Direction -	Sequence 1 -	BCD1-1 -	<--
7. Not used +	Sequence 2 +	BCD1-2 +	<--
8. Not Used -	Sequence 3 -	BCD1-4 -	<--
9. Not Used +	Sequence 4 +	BCD1-8 +	<--
10. Trigger 1 -	<--	<--	BCD3-1 - *
11. Trigger 2 +	<--	<--	BCD3-2 + *
12. Trigger 3 -	<--	<--	BCD3-4 - *
13. Trigger 4 +	<--	<--	BCD3-8 + *
14. DC Common	<--	<--	<--
15. Shutdown -	Sequence 6 -	BCD2-2 -	<--
16. Jog CCW +	<--	<--	BCD2-8 +
17. Jog CW -	<--	<--	BCD2-4 -
18. Direction +	Sequence 1 +	BCD1-1 +	<--
19. Not used -	Sequence 2 -	BCD1-2 -	<--
20. Not Used +	Sequence 3 +	BCD1-4 +	<--
21. Not Used -	Sequence 4 -	BCD1-8 -	<--
22. Trigger 1 +	<--	<--	BCD3-1 + *
23. Trigger 2 -	<--	<--	BCD3-2 - *
24. Trigger 3 +	<--	<--	BCD3-4 + *
25. Trigger 4 -	<--	<--	BCD3-8 - *

NOTE: the "<--" arrow notation implies that the input function in that column is the same as that of the column to the left.

* Although the third BCD input digit is connected to the Trigger inputs, the Trigger function is not disabled by Input Mode 4.

An additional multifunction input appears on the **AUXILIARY 2** connector.

AUXILIARY 2:

INPUT MODE 1	2	3	4
1. START +	<--	<--	<--
2. CCW Limit -	<--	<--	<--
3. CW Limit -	<--	<--	<--
4. STOP +	<--	<--	<--
5. Home Enable -	<--	<--	<--
6. Go Home -	<--	<--	<--
7. Output #2	<--	<--	<--
8. Output #1	<--	<--	<--
9. Unused input	Sequence 5 +	BCD2-1 +	<--
10. +5 VDC	<--	<--	<--
11. Reserved	<--	<--	<--
12. Reserved	<--	<--	<--
13. DC Common	<--	<--	<--
14. START -	<--	<--	<--
15. DC Common	<--	<--	<--
16. DC Common	<--	<--	<--
17. STOP -	<--	<--	<--
18. Home Enable+	<--	<--	<--
19. Go Home +	<--	<--	<--
20. DC Common	<--	<--	<--
21. DC Common	<--	<--	<--
22. Unused input	Sequence 5 -	BCD2-1 -	<--
23. NC	<--	<--	<--
24. Reserved	<--	<--	<--
25. Reserved	<--	<--	<--

NOTE: the "<--" arrow notation implies that the input function in that column is the same as that of the column to the left.

MOTOR/DRIVER Connector Pinout

1. Step +
2. Direction +
3. CW Step (open collector)
4. CCW Step (open collector)
5. Shield
6. Reserved output Return
7. Fault Reset +
8. + 5 VDC
9. Drive Fault input +
10. Slip Fault input +
11. Boost output +
12. DC Common
13. NC
14. Step Return
15. Direction Return
16. Drive Shutdown +
17. Drive Shutdown Return
18. Reserved output +
19. Fault Reset Return
20. DC Common
21. Drive Fault Common
22. Slip Fault Common
23. Boost Return
24. DC Common
25. NC

ENCODER Connector Pinout

1. Quadrature Channel A+
2. Quadrature Channel A-
3. Quadrature Channel B+
4. Quadrature Channel B-
5. Z Channel+
6. Z Channel-
7. NC
8. Shield (case ground)
9. NC
10. NC
11. NC
12. NC
13. NC
14. Ground
15. Ground
16. Ground
17. Ground
18. Ground
19. Ground
20. Ground
21. NC
22. NC
23. +5 VDC
24. +5 VDC
25. +5 VDC

RS-232C Connector Pinout

1. Earth (Shield)
2. Receive Input
3. Transmit Output
4,5,6. No Connection
7. Signal Ground
8 through 25. No Connection

APPENDIX E: STATUS REQUEST RESPONSE SUMMARY

RA Response -- Limit Switch Status

Response Character	Last Move Terminated by		Current Move Limited by	
	CW Limit	CCW Limit	CW Limit	CCW Limit
@	NO	NO	NO	NO
A	YES	NO	NO	NO
B	NO	YES	NO	NO
C	YES	YES	NO	NO
D	NO	NO	YES	NO
E	YES	NO	YES	NO
F	NO	YES	YES	NO
G	YES	YES	YES	NO
H	NO	NO	NO	YES
I	YES	NO	NO	YES
J	NO	YES	NO	YES
K	YES	YES	NO	YES
L	NO	NO	YES	YES
M	YES	NO	YES	YES
N	NO	YES	YES	YES
O	YES	YES	YES	YES

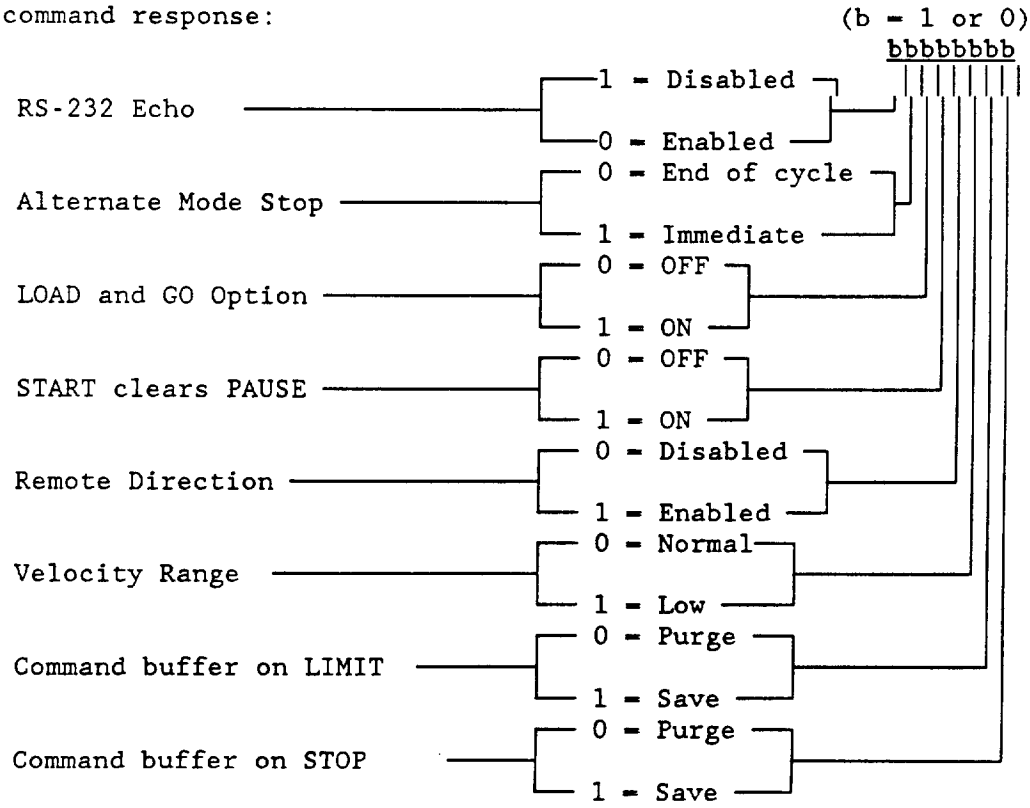
RB Response -- Pause Status

Response	Shutdown Output	Pause *"PS"*	Hold *"U"*	Waiting for Trigger
@	OFF	NO	NO	NO
A	ON	NO	NO	NO
B	OFF	YES	NO	NO
C	ON	YES	NO	NO
D	OFF	NO	YES	NO
E	ON	NO	YES	NO
F	OFF	YES	YES	NO
G	ON	YES	YES	NO
H	OFF	NO	NO	YES
I	ON	NO	NO	YES
J	OFF	YES	NO	YES
K	ON	YES	NO	YES
L	OFF	NO	YES	YES
M	ON	NO	YES	YES
N	OFF	YES	YES	YES
O	ON	YES	YES	YES
S	STOP Input is Asserted			

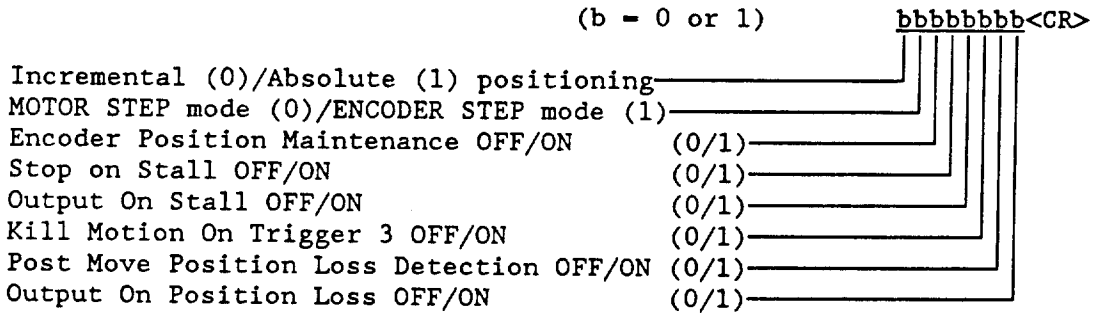
RC Response --Closed Loop Status

Response Character	Static Position Loss Between Moves	Position Loss Detected	Go Home Sequence Terminated Normally	Stall During Last Move Attempt
@	NO	NO	YES	NO
A	NO	NO	YES	YES
B	NO	NO	NO	NO
C	NO	NO	NO	YES
D	NO	YES	YES	NO
E	NO	YES	YES	YES
F	NO	YES	NO	NO
G	NO	YES	NO	YES
H	YES	NO	YES	NO
I	YES	NO	YES	YES
J	YES	NO	NO	NO
K	YES	NO	NO	YES
L	YES	YES	YES	NO
M	YES	YES	YES	YES
N	YES	YES	NO	NO
O	YES	YES	NO	YES

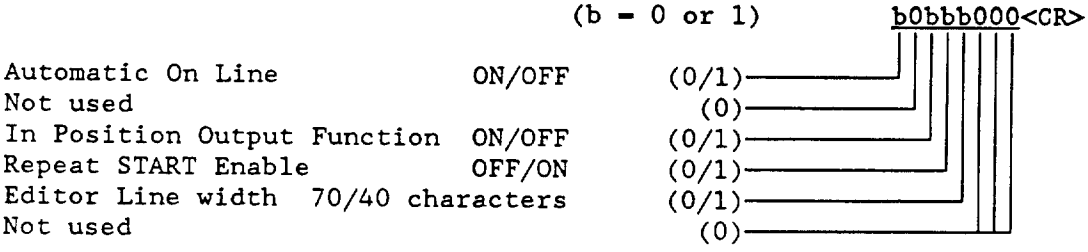
u*SR* command response:



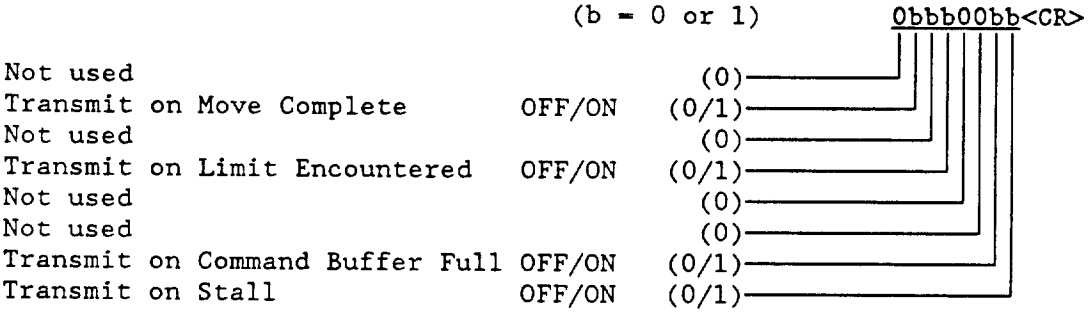
u*FR* command response:



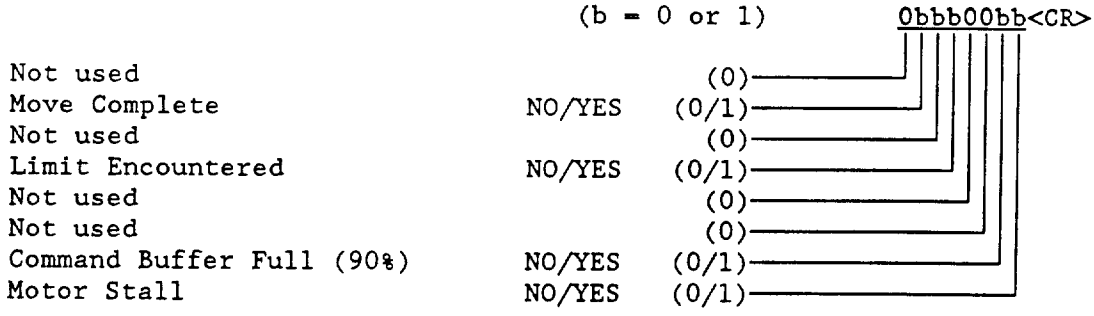
u*OR* command response:



u*QR* command response:



u*QI* command response:



APPENDIX F: COMMAND SUMMARY

The 372 commands are listed below first in a functional grouping and again alphabetically below that. The pertinent pages of the manual are listed with the alphabetical commands.

Motion	**MN**	PRESET MODE (Go to Position)
Mode	**MC**	CONTNIUOUS MODE
	MA	ALTERNATING MODE
	MP*a*	Mode (a=A/I, Absolute/Incremental)
	Q*b*	Velocity Mode (b=1/0, enable/disable)
Parameters	**D**$\pm$*nnnnnnnn*	Set Distance (or Position) Value
	H	Reverse Motor Direction
	H$\pm$	Set Direction of Motion (+=CW)
	V*nn.nnn*	Set Velocity
	VS*nn.nnn*	Set STOP/START Velocity
	JV*nn.nnn*	Set Jog Velocity
	A*nnn.nn*	Set Acceleration/Deceleration
	GA*nnn.nn*	Set Go Home Acceleration
	JA*nnn.nn*	Set Jog Acceleration
	LA*nnn.nn*	Set Limit Acceleration
	RM*hhhh*	Direct Velocity Input
Setup	**MR***nn*	Match Indexer to Motor
	LD*n*	Define Limit Function
	US*nnn*	Set Position Scale Factor
Position	**ER***nnnnnn*	Set Encoder Resolution
Tracking	**FS***ab*	Set Encoder function
	DW*nnnnnn*	Set Backlash window
	CM*n*	Set Correction Mode
	CG*n*	Set Correction Gain
	MV*n*	Set Correction Velocity
Execute	**G**	GO (start executing move)
	GH$\pm$*nn.nnn*	Go Home (seek Home signal)
	XR*n*	Execute stored sequence
Load Data	**I**	Load Parameters & Compute
	uIC	Load, Compute,Transmit <CR>
Stop	**Y**	Exit from Loop
	Q	Clear buffer
	S	Stop
	K	Kill
	Z	Reset
	ST*b*	Energize or Shutdown Drive (b = 1 or 0)
Pause	**T***nnn.nn*	Delay (in seconds)
	TR*xxxx*	Wait for Triggers
	PS	Pause for CONTINUE
	U	Pause for CONTINUE
	C	CONTINUE
Loop	**L***nnnnnnn*	Loop (0-9,999,999 times)
	N	End of loop
Branch	**SKE***xxxx*	Skip on Trigger Equal

	SKNxxxx	Skip on Trigger Not Equal
Reset	Z	Software reset
	XO	Reset Cumulative Position Counter
	PZ	Reset Absolute Position Counter
Immediate	uB	Buffer Status
Status	uR	Execution Status
Requests	uRA	Limit Switch Status
	uRC	Closed Loop Status
	uTS	Trigger Status
Immediate	uW1	Binary Position
Position	uW2, uW3	Hexadecimal Position
Buffered	uP	Incremental Position
Status	uPB	Incremental binary Position
Requests	uPR	Absolute position
	uX1	Cumulative position
	uX1B	Cumulative binary position
	uX2	Cumulative Encoder position
	uCR	Command Execution Status
Setup	uDR	Default Parameters
Status	uXU	Sequence Contents
Requests	uXN	Sequence Memory
	uXV	Memory Integrity
	uSR	Special status 1
	uOR	Special status 2
	uQR	Transmit Signal
	uFR	Encoder Function
	uUR	Scale Factor
	uRV	Software Revision
Interface	E	Enable Communications
	F	Disable Communications
Keyboard	^H	Backspace (Control H)
Outputs	O*bb*	Set programmable output
Special	SS*ab*	Set Special Function 1
Functions	OS*ab*	Set Special Function 2
Signal	QS*ab*	Set Message Function
Input Mode	IM*n*	Set Input Mode (n = 1 to 4)
Messages	MS*n*	Operator Messages (n = 0, 1, or 2)
Memory	uXU*n*	Report Sequence *n*
Status	uXN	Report Memory Use
Requests	uXV	Test Checksum
Sequence	XD*n*	Enter or Edit Sequence *n*
Edit	XT	Exit Edit Mode and Save
	XK	Exit Edit Mode and Don't Save
	uXE	Erase a Sequence
	uZS	Erase All Sequences
Parameter	SA	Save Current Parameters
Memory	uZD	Restore Factory Defaults
	XP*n*	Select Power on Sequence

Alphabetical listing:

Annn.nn Acceleration 80
B Busy, No Attention. 101
C Continue. 92
CGn Correction Gain Factor. 85
CMn Correction Mode for Position Maintenance. 84
CR Command Execution Status 109
D±nnnnnnnn Distance 79
DR Default Parameter Status 110
DWnnnnn Backlash Deadband Width 83
E Communication Interface Enable/Disable. 116
ERnnnnn Encoder Resolution Select 83
F Communication Interface Enable/Disable. 116
FR Encoder Functions Status 113
FSab Encoder Function Set 84
G Go Command. 85
GAnn.nnn Go Home Acceleration 80
GH±nn.nnn Go Home 89
H Change Motor Direction. 79
H± Direction of Motion. 79
I,IC Load and Go 89
IM Input Mode 120
JAnn.nnn Jog Acceleration 80
JVnn.nnn Set Jog Velocity 80
K Emergency Stop Command 91
LAnn.nnn Limit Acceleration 80
LDn Limit Definition 81
Lnnnnnnn Loop 93
MA Alternate Mode 77
MC Continuous Mode. 77
MN Normal Mode. 77
MPa Positioning Mode. 77
MRnn Motor Resolution Select 81
MS Messages Control 120
MVn Maximum Correction Velocity 85
N End of Loop 93
Obb Programmable Outputs. 116
OR Option Switch Status 112
OS Option Switch. 118
P,PB Incremental Distance Traveled 105
PR Absolute Position 105
PS Pause and Wait for a CONTINUE 92
PZ Reset Absolute Position Counter. 88
Q Stop at End of Current Command. 90
Qb Direction Velocity Mode 77
QI Signal Source Status 114
QR Signal Enable Status 113
QS Signal on Condition 119
R Indexer Status Response 101

RA Limit Switch Status. 102
RB Pause Status 102
RC Closed Loop Status 104
RMhhhh Direction Velocity Parameter 80
RV Software Revision. 114
S Controlled Stop Command 90
SA Save Default Parameters. 121
SKExxxx Skip on Trigger Equal 93
SKNxxxx Skip on Trigger not Equal 94
SR Software Switch Status 112
SS Software Switch. 117
ST Motor Shutdown 94
Tnnn.nn Pause in Processing Command 91
TRxxxx Pause until Input Trigger matches Designated Pattern 92
TS Trigger Input Status 105
U Hold (Pause) Now and Wait for Continue. 92
UR Scale Factor Status. 114
USnnn Position Scale Factor 81
Vnn.nnn Velocity 80
VSnn.nnn Set Start/Stop Velocity. 80
W1,W2,W3 Current Incremental Distance 105
X0 Set Position Counter 106
X1,X1B Cumulative Position 106
X2 Updated Cumulative Encoder Position. 106
XD Sequence Editing 122
XE Erase a Sequence 122
XK Exit from Editing Mode without Storing 122
XP Sequence Execution Command 123
XRn Execute Stored Sequence 89
XT Exit from Editing Mode 122
XUn,XN,XV Sequence Reports. 111
Y Stop at End of Current Loop 90
Z Software Rest 91
ZD Enter or Edit Sequence 121
ZS Erase all Sequences stored in Unit 122

INDEX

Absolute 22, 46, 47, 84
Acceleration 5
Address 60
Alternating 23
ASCII 56
AUXILIARY 30, 52, 120
Backlash 44, 69, 70, 83
Backspace 116
Baud 56, 60
BCD 34, 38, 76
Binary report 106
Buffer 75, 117, 133
Buffered 72
Cable 51, 56
Carriage return 72, 109
Character 56
Closed loop 41, 65
Command 72
    buffer 75, 117
    Buffered 72
    Immediate 73
Computer 11, 56
Conditional branch 93
Connectors 51
Console 60, 61
Continuous 24
Daisy-chain 56
DC Common 52
DCE 56
Deadband 44
Deceleration 5
default 28, 77, 110, 121
Delimiter 72
Device Specific 97
Differential 51, 65
Direction 37, 53, 112, 117, 133
Direction input 32
Distance 5
DTE 56
Echo 61, 112, 117, 133
Encoder 32, 41, 65, 69, 84
    Installation 65
Encoder Step Mode 42
Equal 76
Error code 120
Exclamation mark 119
Feedback 84
Full Duplex 61

Absolute 22, 45, 47, 84
Fuse 51
Gain 43, 85
GO 85
GO HOME 89
GO HOME input 24, 32
Half Duplex 61
Handshaking 56
Hexadecimal 107
HOME ENABLE input 24, 32
Home Limit 24
Homing function 24, 32, 46, 69, 89, 104
Incremental 22, 46, 65, 84
Indexing 21
    Mode 77
Inertia 5
Input 52, 120
    differential 65
    single-ended 65
Input Mode 31, 34, 36, 120
    Input Mode 2 37
    Input Mode 3 38
    Input Mode 4 38
Inputs 30
Installation 48
Interface 56
JOG 35
Jumpers 48
KILL 91
LED indicators 21
Limit switch 8, 12, 30, 53
Line-feed 73
LOAD 89
LOAD and GO 117
Loop 93
Message 120
Motor 8, 52, 77
    resolution 8, 22
    size 8
Mounting 48
Multi-axis 45, 56, 60, 75, 97
Non-volatile memory 29, 121
Off Line 30, 34, 116, 118
On Line 116, 118
Optically isolated 52
Output 35, 44, 45, 51, 84, 116, 118
    differential 51
    single-ended 51
Parameter 4, 22, 29, 62, 77, 82, 110, 121
    Acceleration 5

Acceleration 5
default 29, 121
encoder 66
position multiplier 22
scale factor 22
Stored 28, 29
velocity 5
Velocity Range 117
Parameters 79
Parentheses 76
Parity 60
Pause 91, 117
Position Maintenance 42, 84
Position report 106
Position Tracking 41
Power 50
Preset 23
Profile 5
Programmable inputs 34
Protocol 60
Quadrature 65
Receive 56
Remote Data 34, 38
Remote Data 76
Reset 91
Resolution 4, 8, 62, 66
encoder 66, 69
motor 22, 81
RS-232 . . . . . 11, 20, 56, 64, 116
Baud 60
Cable 56
Console 11
Off Line 30
On Line 21
protocol 60
terminal 11
Scale factor 22, 114
Sequence 29, 37, 89, 111, 122
editing 122
Servo 42, 84
Shutdown 37
Signal 119
Single-ended 51, 65
Skip 93
SPECIAL INPUT mode 27
Stall 41, 43, 84, 104
START 118
Start input 31
Status 97
Status Request 97

Address 60
Stop 90
Stop input 31
Switch 60
Terminal 56, 61
Torque 5
Transmission 70
Transmit 56, 60, 119
Trapezoidal 5
Triangular 5
Trigger 32, 92, 93, 105
Troubleshooting 64, 69
Unit Number 60, 97
Velocity 5
Velocity Range 117
Velocity stream 27
Voltage 51, 52
Z Channel 24, 48
Zero 47
Alternating 23
ASCII 56
AUXILIARY 30, 52, 120
Backlash 44, 69, 70, 83
Backspace 116
Baud 56, 60
BCD 34, 38, 76
Binary report 106
Buffer 75, 117, 133
Buffered 72
Cable 51, 56
Carriage return 72, 109
Character 56
Closed loop 41, 65
Command 72
    buffer 75, 117
    Buffered 72
    Immediate 73
Computer 11, 56
Conditional branch 93
Connectors 51
Console 60, 61
Continuous 24
Daisy-chain 56
DC Common 52
DCE 56
Deadband 44
Deceleration 5
default 28, 77, 110, 121
Delimiter 72
Device Specific 97

Differential 51, 65
Direction 37, 53, 112, 117, 133
Direction input 32
Distance 5
DTE 56
Echo 61, 112, 117, 133
Encoder 32, 41, 65, 69, 84
    Installation 65
Encoder Step Mode 42
Equal 76
Error code 120
Exclamation mark 119
Feedback 84
Full Duplex 61
Fuse 51
Gain 43, 85
Alternating 23
GO 85
GO HOME 89
GO HOME input 24, 32
Half Duplex 61
Handshaking 56
Hexadecimal 107
HOME ENABLE input 24, 32
Home Limit 24
Homing function 24, 32, 45, 69, 89, 104
Incremental 22, 45, 65, 84
Indexing 21
    Mode 77
Inertia 5
Input 52, 120
    differential 65
    single-ended 65
Input Mode 31, 34, 36, 120
    Input Mode 2 37
    Input Mode 3 38
    Input Mode 4 38
Inputs 30
Installation 48
Interface 56
JOG 35
Jumpers 48
KILL 91
LED indicators 21
Limit switch 8, 12, 30, 53, 81
Line-feed 73
LOAD 89
LOAD and GO 117
Loop 93
Message 120

Motor 8, 52, 77
resolution 8, 22
size 8
Mounting 48
Multi-axis 45, 56, 60, 75, 97
Non-volatile memory 29, 121
Off Line 30, 34, 116, 118
On Line 116, 118
Optically isolated 52
Output 35, 44, 45, 51, 84, 116, 118
differential 51
single-ended 51
Parameter 4, 22, 29, 62, 77, 82, 110, 121
Acceleration 5
default 29, 121
encoder 66
position multiplier 22
scale factor 22
Analog 47
Stored 28, 29
velocity 5
Velocity Range 117
Parameters 79
Parentheses 76
Parity 60
Pause 91, 117
Position Maintenance 42, 84
Position report 106
Position Tracking 41
Power 50
Preset 23
Profile 5
Programmable inputs 34
Protocol 60
Quadrature 65
Receive 56
Remote Data 34, 38
Remote Data 76
Reset 91
Resolution 4, 8, 62, 66
encoder 66, 69
motor 22, 81
RS-232 . . . . . 11, 20, 56, 64, 116
Baud 60
Cable 56
Console 11
Off Line 30
On Line 21
protocol 60
terminal 11

Scale factor 22, 114
Sequence 29, 37, 89, 111, 122
editing 122
Servo 42, 84
Shutdown 37
Signal 119
Single-ended 51, 65
Skip 93
SPECIAL INPUT mode 27
Stall 41, 43, 84, 104
START 118
Start input 31
Status 97
Status Request 97
Stop 90
Stop input 31
Switch 60
Terminal 56, 61
Torque 5
Transmission 70
ASCII 56
Transmit 56, 60, 119
Trapezoidal 5
Triangular 5
Trigger 32, 92, 93, 105
Troubleshooting 64, 69
Unit Number 60, 97
Velocity 5
Velocity Range 117
Velocity stream 27
Voltage 51, 52
Z Channel 24, 48
Zero 47