ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Galil Motion Control DMC-1580* at our website: *Click HERE*

USER MANUAL

DMC-1500

Manual Rev. 2.0xg

By Galil Motion Control, Inc.

Galil Motion Control, Inc.
270 Technology Way
Rocklin, California 95765
Phone: (916) 626-0101
Fax: (916) 626-0102
Internet Address: support@galilmc.com
URL: www.galilmc.com

Rev 03-10

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using This Manual

This user manual provides information for proper operation of the DMC-1500 controller. A separate supplemental manual, the Command Reference, contains a description of the commands available for use with this controller.

Your DMC-1500 motion controller has been designed to work with both servo and stepper type motors. In addition, the DMC-1500 has a daughter board for controllers with more than 4 axes. Installation and system setup will vary depending upon whether the controller will be used with stepper motors, or servo motors, and whether the controller has more than 4 axes of control. To make finding the appropriate instructions faster and easier, icons will be next to any information that applies exclusively to one type of system. Otherwise, assume that the instructions apply to all types of systems. The icon legend is shown below.



Attention: Pertains to servo motor use.



Attention: Pertains to stepper motor use.



Attention: Pertains to controllers with more than 4 axes.

Please note that many examples are written for the DMC-1540 four-axis controller or the DMC-1580 eight axes controller. Users of the DMC-1530 3-axis controller, DMC-1520 2-axis controller or DMC-1510 1-axis controller should note that the DMC-1530 uses the axes denoted as XYZ, the DMC-1520 uses the axes denoted as XY, and the DMC-1510 uses the X-axis only.

Examples for the DMC-1580 denote the axes as A,B,C,D,E,F,G,H. Users of the DMC-1550 5-axis controller, DMC-1560 6-axis controller or DMC-1570, 7-axis controller should note that the DMC-1550 denotes the axes as A,B,C,D,E, the DMC-1560 denotes the axes as A,B,C,D,E,F and the DMC-1570 denotes the axes as A,B,C,D,E,F,G. The axes A,B,C,D may be used interchangeably with X,Y,Z,W.

This manual was written for the DMC-1500 firmware revision 2.0 and later. For controllers with firmware previous to revision 2.0, please consult the original manual for your hardware. The later revision firmware was previously specified as DMC-1500-18.

WARNING: Machinery in motion can be dangerous! It is the responsibility of the user to design effective error handling and safety protection as part of the machine. Galil shall not be liable or responsible for any incidental or consequential damages.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Firmware Updates

New feature for Rev 2.0h April 1998:

Feature	Description
1. CMDERR enhanced to support multitasking:	If CMDERR occurs on thread 1,2 or 3, thread will be holted. Thread can be re-started with XQ _ED2, _ED1, 1 for retry XQ _ED3, _ED1, 1 for next instruction
2. _VM returns instantaneous commanded vector velocity	
3. FA resolution increased to 0.25.	

New feature for Rev 2.0g November 1997:

Feature	Description
1. CR radius now has range of 16 million	Allows for large circular interpolation radii
2. _AB returns abort input	Allows for monitoring of abort input
3. CW,1 When output FIFO full application program will not pause but data will be lost	Allows for output FIFO buffer to fill up without affecting the execution of a program
4. List Variable (LV), List Array (LA), List app program labels (LL)	Allows for the user to interrogate RAM

New feature for Rev 2.0e May 1997:

Feature	Description
1. ER now accepts argument < 0	Disables error output (LED and Error Output does not turn on for that axis)
2. During a PR decel can now be changed on an unnatural stop	Allows for monitoring of abort input

New feature for Rev 2.0d February 1997:

Feature	Description
1. AP, MF, MR in stepper now uses _DE instead of _RP	Trippoints based on register after buffer
2. \ now terminates QD	Download array no longer requires control sequence to end
3. KS can now be fraction (down to .5)	Allows for smaller stepper motor smoothing delay (due to filter)
4. New arguments for MT of 2.5 and -2.5	Reverses the direction of motion from MT 2 and MT -2
5. MG now can go to 80 characters	Increased message size

New feature for Rev 2.0c October 1996:

Feature	Description
1. MC now works for steppers	More accurate trippoint for stepper motor completion

New feature for Rev 2.0b September 1996:

Feature	Description
1. Operand '&' and '|' for conditional statements	Allows for multiple conditional statements in jump routines IE. (A>=3) & (B<55) | (C=78)

New feature for Rev 2.0 March 1996. (This revision is also designated DMC-1500-18).

DAC resolution increased to 16-bits.
Step motor control method improved.

Command	Description
KS	Step Motor Smoothing

New feature for Rev 1.1

Electronic Cam

New commands:

Command	Description
EA	Choose ECAM master
EM	Cam Cycle Command
EP	Cam table interval and starting point
ET	ECAM table entry
EB	Enable ECAM
EG	Engage ECAM cycle
EQ	Disengage ECAM

New features added Jan 1995:

Allow circular array recording.

New commands added July 1994 Rev 1.4:

Command	Description
RI,N	N is a new interrupt mask which allows changing the interrupt mask
QU	Upload array
QD	Download array
MF x,y,z,w	Trippoint for motion - forward direction
MR x,y,z,w	Trippoint for motion - reverse direction
MC XYZW	In position trippoint
TW x,y,z,w	Sets timeout for in position
VR r	Sets speed ratio for VS

New commands added January 1994 Rev 1.3:

Can specify parameters with axis designator. For example:

Command	Description
KPZ=10	Set Z axis gain to 10
KP*=10	Set all axes gains to 10

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

(KPXZ=10 is invalid. Only one or all axes can be specified at a time).

New commands added July 1993 Rev 1.2:

Command	Description
_UL	Gives available variables
_DL	Give available labels
@COM[n]	2's complement function

New commands added March 1993: Rev 1.2

Command	Description
_CS	Segment counter in LM, VM and CM modes
_AV	Return distance travelled in LM and VM modes
_VPX	Return the coordinate of the last point in a motion sequence, LM or VM
VP x,y<n	Can specify vector speed with each vector segment Where <n sets vector speed

New commands added January 1993:

Command	Description
HX	Halt execution for multitasking
AT	At time trippoint for relative time from reference
ES	Ellipse scale factor
OB n,expression	Defines output n where expression is logical operation, such as I1 & I6, variable or array element
XQ#Label,n	Where n = 0 through 3 and is program thread for multitasking
DV	Dual velocity for Dual Loop

Feature	Description
1.	Allows gearing and coordinated move simultaneously
2.	Multitasking for up to four independent programs
3.	Velocity Damping from auxiliary encoder for dual loop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contents

Chapter 1 Overview 1

Introduction 1
Overview of Motor Types 2
Standard Servo Motors with +/- 10 Volt Command Signal 2
Stepper Motor with Step and Direction Signals 2
DMC-1500 Functional Elements 2
Microcomputer Section 3
Motor Interface 3
Communication 3
General I/O 3
System Elements 3
Motor 4
Amplifier (Driver) 4
Encoder or Position Sensor 4
Watch Dog Timer 4

Chapter 2 Getting Started 7

Elements You Need 7
Installing the DMC-1500 8
Step 1. Determine Overall Motor Configuration 8
Step 2. Install Jumpers on the DMC-1500 8
Step 3. Configure DIP switches on the DMC-1500 9
Step 4. Connect AC Power to the Controller 9
Step 5. Install Communications Software 9
Step 6. Establish Communications with Galil Software 10
Step 7. Connect Amplifiers and Encoders. 11
Step 8a. Connect Standard Servo Motors 13
Step 8b. Connect Step Motors 16
Step 9. Tune the Servo System 17
Design Examples 18
Example 1 - System Set-up 18
Example 2 - Profiled Move 18
Example 3 - Multiple Axes 19
Example 4 - Independent Moves 19
Example 5 - Position Interrogation 19
Example 6 - Absolute Position 20
Example 7 - Velocity Control 20
Example 8 - Operation Under Torque Limit 20

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 9 - Interrogation....21
Example 10 - Operation in the Buffer Mode....21
Example 11 - Motion Programs....21
Example 12 - Motion Programs with Loops....22
Example 13 - Motion Programs with Trippoints....22
Example 14 - Control Variables....22
Example 15 - Linear Interpolation....23
Example 16 - Circular Interpolation....23

Chapter 3 Connecting Hardware 25

Overview....25
Using Opto-isolated Inputs....25
Limit Switch Input....25
Home Switch Input....26
Abort Input....26
Uncommitted Digital Inputs....27
Wiring the Optoisolated Inputs....27
Using an Isolated Power Supply....28
Bypassing the Opto-Isolation:....30
Changing Optoisolated Inputs From Active Low to Active High....30
Amplifier Interface....30
TTL Inputs....31
Analog Inputs....32
TTL Outputs....32
Offset Adjustment....32

Chapter 4 Communication 33

Introduction....33
RS232 Ports....33
RS232 - Main Port {P1} DATATERM....33
RS232 - Auxiliary Port {P2} DATASET....33
*RS422 - Main Port {P1}....33
*RS422 - Auxiliary Port {P2}....34
Configuration....34
Baud Rate Selection....34
Daisy-Chaining....35
Daisy Chain Example:....35
Synchronizing Sample Clocks....36
Controller Response to DATA....36
Galil Software Tools and Libraries....36

Chapter 5 Command Basics 37

Introduction....37
Command Syntax....37
Coordinated Motion with more than 1 axis....38
Program Syntax....38
Controller Response to DATA....38
Interrogating the Controller....39
Interrogation Commands....39
Summary of Interrogation Commands....39
Additional Interrogation Methods.....40
Operands....40
Command Summary....40

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 6 Programming Motion 41

Overview 41
Independent Axis Positioning 41
Command Summary - Independent Axis 42
Operand Summary - Independent Axis 42
Independent Jogging 44
Command Summary - Jogging 44
Operand Summary - Independent Axis 44
Linear Interpolation Mode 45
Specifying Linear Segments 45
Specifying Vector Acceleration, Deceleration and Speed: 46
Additional Commands 46
Command Summary - Linear Interpolation 47
Operand Summary - Linear Interpolation 48
Vector Mode: Linear and Circular Interpolation Motion 50
Specifying Vector Segments 50
Specifying Vector Acceleration, Deceleration and Speed: 51
Additional Commands 51
Command Summary - Vector Mode Motion 53
Operand Summary - Vector Mode Motion 53
Electronic Gearing 54
Command Summary - Electronic Gearing 55
Operand Summary - Electronic Gearing 55
Electronic Cam 57
Command Summary - ECAM Mode 61
Operand Summary - ECAM Mode 61
Contour Mode 63
Specifying Contour Segments 63
Additional Commands 64
Command Summary - Contour Mode 64
Teach (Record and Play-Back) 66
Stepper Motor Operation 67
Specifying Stepper Motor Operation 68
Using an Encoder with Stepper Motors 69
Command Summary - Stepper Motor Operation 69
Operand Summary - Stepper Motor Operation 69
Dual Loop (Auxiliary Encoder) 70
Backlash Compensation 70
Command Summary - Using the Auxiliary Encoder 72
Operand Summary - Using the Auxiliary Encoder 72
Motion Smoothing 72
Using the IT and VT Commands (S curve profiling): 72
Using the KS Command (Step Motor Smoothing): 74
Homing 74
High Speed Position Capture (The Latch Function) 77

Chapter 7 Application Programming 79

Overview 79
Using the DMC-1500 Editor to Enter Programs 79
Edit Mode Commands 80
Program Format 81
Using Labels in Programs 81
Special Labels 82

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Commenting Programs 82
Executing Programs & Multitasking 83
Debugging Programs 84
Debugging Programs 86
Commands 86
Operands 86
Program Flow Commands 87
Event Triggers & Trippoints 87
DMC-1500 Event Triggers 88
Event Trigger Examples: 88
Conditional Jumps 91
Subroutines 94
Stack Manipulation 94
Auto-Start Routine 95
Automatic Subroutines for Monitoring Conditions 95
Mathematical and Functional Expressions 98
Mathematical Expressions 98
Bit-Wise Operators 99
Functions 100
Variables 100
Assigning Values to Variables: 101
Operands 102
Special Operands (Keywords) 103
Arrays 103
Defining Arrays 103
Assignment of Array Entries 104
Automatic Data Capture into Arrays 105
Command Summary - Automatic Data Capture 105
Data Types for Recording: 105
Operand Summary - Automatic Data Capture 106
Deallocating Array Space 106
Input of Data (Numeric and String) 107
Input of Data 107
Operator Data Entry Mode 108
Using Communication Interrupt 108
Output of Data (Numeric and String) 110
Sending Messages 110
Specifying the Serial Port for Messages: 111
Formatting Messages 111
Using the MG Command to Configure Terminals 112
Summary of Message Functions: 112
Displaying Variables and Arrays 112
Interrogation Commands 112
Formatting Variables and Array Elements 114
Converting to User Units 115
Programmable Hardware I/O 115
Digital Outputs 115
Digital Inputs 116
Input Interrupt Function 117
Analog Inputs 118
Example Applications 119
Wire Cutter 119
X-Y Table Controller 120
Speed Control by Joystick 122
Position Control by Joystick 123

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Backlash Compensation by Sampled Dual-Loop 123

Chapter 8 Hardware & Software Protection **126**

Introduction 126
Hardware Protection 126
Output Protection Lines 126
Input Protection Lines 126
Software Protection 127
Programmable Position Limits 127
Off-On-Error 127
Automatic Error Routine 128
Limit Switch Routine 128

Chapter 9 Troubleshooting **130**

Overview 130
Installation 130
Communication 131
Stability 131
Operation 131

Chapter 10 Theory of Operation **132**

Overview 132
Operation of Closed-Loop Systems 134
System Modeling 135
Motor-Amplifier 136
Encoder 138
DAC 139
Digital Filter 139
ZOH 139
System Analysis 140
System Design and Compensation 142
The Analytical Method 142

Appendices **146**

Electrical Specifications 146
Servo Control 146
Stepper Control 146
Input/Output 146
Power 147
Performance Specifications 147
Card Level Layout 148
Connectors for DMC-1500 Main Board 149
J2 - Main (60 pin IDC) 149
J5 - General I/O (26 pin IDC) 150
J3 - Aux Encoder (20 pin IDC) 150
J4 - Driver (20 pin IDC) 151
J6 - Daughter Board Connector (60 pin) 151
J7 - 10 pin 151
Connectors for Auxiliary Board (Axes E,F,G,H) 152
JD2 - Main (60 pin IDC) 152
JD5 - I/O (26 pin IDC) 153
JD3 - 20 pin IDC - Auxiliary Encoders 153

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

JD4 - 20 pin IDC - Amplifiers 154
JD6 - Daughterboard Connector (60 pin) 154
Cable Connections for DMC-1500 154
Standard RS-232 Specifications 154
DMC-1500 Serial Cable Specifications 155
Pin-Out Description for DMC-1500 157
Configuration Description for DMC-1500 159
Jumpers 159
Address Configuration Jumpers 159
Front Panel Baud Rate Switches 159
Adjustment Pots 160
Dip Switch Settings 160
Offset Adjustments for DMC-1500 160
Accessories and Options 161
ICM-1100 Interconnect Module 162
AMP/ICM-1100 Connections 162
J2 - Main (60 pin IDC) 165
J3 - Aux Encoder (20 pin IDC) 165
J4 - Driver (20 pin IDC) 165
J5 - General I/O (26 pin IDC) 165
JX6, JY6, JZ6, JW6 - Encoder Input (10 pin IDC) 165
ICM-1100 Drawing 166
AMP-11x0 Mating Power Amplifiers 167
TERM-1500 Operator Terminal 167
DB-15072 OPTO-22 Expansion Option 175
Configuring the I/O for the DB-15072 175
Connector Description of the DB-15072 176
Coordinated Motion - Mathematical Analysis 179
DMC-700/DMC-1500 Comparison 182
List of Other Publications 184
Contacting Us 184
WARRANTY 185

Index **191**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 1 Overview

Introduction

The DMC-1500 Series are packaged motion controllers designed for stand-alone operation. Features include coordinated motion profiling, uncommitted inputs and outputs, non-volatile memory for stand-alone operation and RS232/RS422 communication. Extended performance capability over the previous generation of controllers includes: fast 8 MHz encoder input frequency, precise 16-bit motor command output DAC, +/-2 billion counts total travel per move, faster sample rate, and multitasking of up to four programs. The controllers provide increased performance and flexibility and yet are smaller in size and lower in cost than the previous generation. The DMC-1500 is also available as a cost-effective, card-level product making it ideal for OEM applications.

Designed for maximum system flexibility, the DMC-1500 is available for one to eight axes and can be interfaced to a variety of motors and drives including step motors, servo motors and hydraulics.

Each axis accepts feedback from a quadrature linear or rotary encoder with input frequencies up to 8 million quadrature counts per second. For dual-loop applications that require one encoder on both the motor and the load, auxiliary encoder inputs are included for each axis.

The powerful controller provides many modes of motion including jogging, point-to-point positioning, linear and circular interpolation with infinite vector feed, electronic gearing and user-defined path following. Several motion parameters can be specified including acceleration and deceleration rates, and slew speed. The DMC-1500 also provides S-curve acceleration for motion smoothing.

For synchronizing motion with external events, the DMC-1500 includes 8 opto-isolated inputs, 8 programmable outputs and 7 analog inputs. For controllers with 5 or more axes, the DMC-1500 has an additional 8 opto-isolated inputs and 8 TTL inputs. I/O expansion boards provide additional inputs and outputs or interface to OPTO 22 racks. Event triggers can automatically check for elapsed time, distance and motion complete.

Despite its full range of sophisticated features, the DMC-1500 is easy to program. Instructions are represented by two letter commands such as BG for Begin and SP for Speed. Conditional Instructions, Jump Statements, and arithmetic functions are included for writing self-contained applications programs. An internal editor allows programs to be quickly entered and edited, and support software such as the Servo Design Kit allows quick system set-up and tuning.

To prevent system damage during machine operation, the DMC-1500 provides several error handling features. These include software and hardware limits, automatic shut-off on excessive error, abort input, and user-definable error and limit routines.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Overview of Motor Types

The DMC-1500 can provide the following types of motor control:

Standard servo motors with +/- 10 volt command signals

Step motors with step and direction signals

Other actuators such as hydraulics - For more information, contact Galil.

The user can configure each axis for any combination of motor types, providing maximum flexibility.

Standard Servo Motors with +/- 10 Volt Command Signal

The DMC-1500 achieves superior precision through use of a 16-bit motor command output DAC and a sophisticated PID filter that features velocity and acceleration feedforward, an extra pole filter and integration limits.

The controller is configured by the factory for standard servo motor operation. In this configuration, the controller provides an analog signal (+/- 10Volt) to connect to a servo amplifier. This connection is described in Chapter 2.

Stepper Motor with Step and Direction Signals

The DMC-1500 can control stepper motors. In this mode, the controller provides two signals to connect to the stepper motor: Step and Direction. For stepper motor operation, the controller does not require an encoder and operates the stepper motor in an open loop fashion. Chapter 2 describes the proper connection and procedure for using stepper motors.

DMC-1500 Functional Elements

The DMC-1500 circuitry can be divided into the following functional groups as shown in Figure 1.1 and discussed in the following.

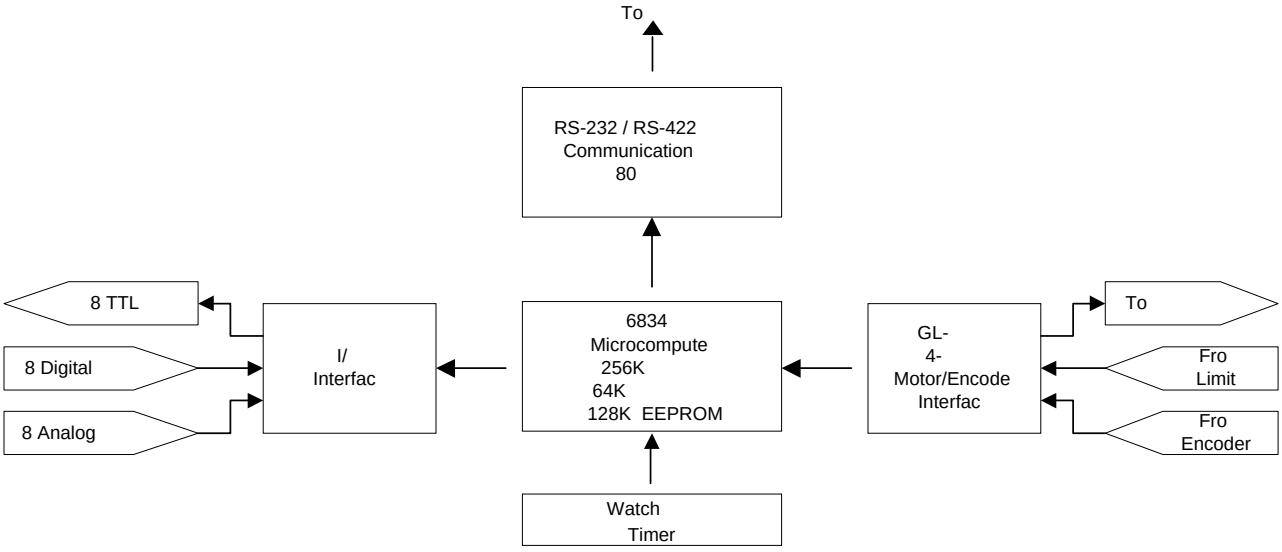


Figure 1.1 - DMC-1500 Functional Elements

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Microcomputer Section

The main processing unit of the DMC-1500 is a specialized 32-bit Motorola 68340 Series Microcomputer with 256K RAM, 64 K EPROM and 128 K bytes EEPROM. The RAM provides memory for variables, array elements and application programs. The EPROM stores the firmware of the DMC-1500. The EEPROM allows parameters and programs to be saved in non-volatile memory upon power down.

Motor Interface

For each axis, a GL-1800 custom gate array performs quadrature decoding of the encoders at up to 8 MHz, generates the +/-10 Volt analog signal (16 Bit DAC) for input to a servo amplifier, and generates step and direction signal for step motor drivers.

Communication

Communication to the DMC-1500 is via two separately addressable RS232 ports. The ports may also be configured by the factory for RS422. The serial ports may be daisy-chained to other DMC-1500 controllers.

General I/O

The DMC-1500 provides interface circuitry for eight optoisolated inputs, eight general outputs and seven analog inputs (12 Bit ADC with option for 16 Bit ADC).

An auxiliary board, the DB-15072 provides interface to up to three OPTO 22 racks with 24 I/O modules each. 24 bits can be configured for interface to output or input modules and the remaining 48 for input modules.

1580 Controllers with 5 or more axes provide 24 inputs and 16 outputs.

System Elements

As shown in Fig. 1.2, the DMC-1500 is part of a motion control system which includes amplifiers, motors and encoders. These elements are described below

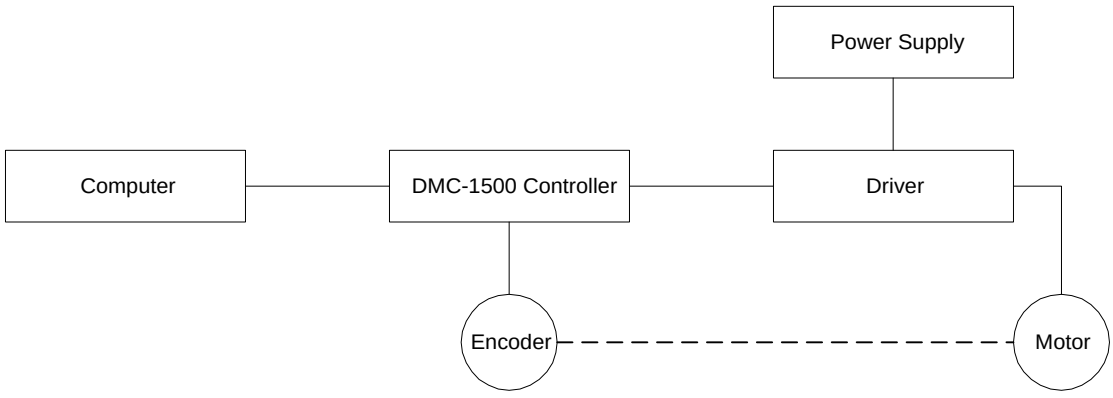


Figure 1.2 - Elements of Servo systems

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Motor

A motor converts current into torque which produces motion. Each axis of motion requires a motor sized properly to move the load at the desired speed and acceleration. Galil's Motion Component Selector software can help you calculate motor size and drive size requirements. Contact Galil at 800-377-6329 if you would like this product.

The motor may be a step or servo motor and can be brush-type or brushless, rotary or linear. For step motors, the controller can be configured to control full-step, half-step, or microstep drives.

Amplifier (Driver)

For each axis, the power amplifier converts a +/-10 Volt signal from the controller into current to drive the motor. The amplifier should be sized properly to meet the power requirements of the motor. For brushless motors, an amplifier that provides electronic commutation is required. The amplifiers may be either pulse-width-modulated (PWM) or linear. They may also be configured for operation with or without a tachometer. For current amplifiers, the amplifier gain should be set such that a 10 Volt command generates the maximum required current. For example, if the motor peak current is 10A, the amplifier gain should be 1 A/V. For velocity mode amplifiers, 10 Volts should run the motor at the maximum speed.



For stepper motors, the amplifier converts step and direction signals into current.

Encoder or Position Sensor

An encoder translates motion into electrical pulses which are fed back into the controller. The DMC-1500 accepts feedback from either a rotary or linear encoder. Typical encoders provide two channels in quadrature, known as CHA and CHB. This type of encoder is known as a quadrature encoder. Quadrature encoders may be either single-ended (CHA and CHB) or differential (CHA,CHA-,CHB,CHB-). The DMC-1500 decodes either type into quadrature states or four times the number of cycles. Encoders may also have a third channel (or index) for synchronization.

The DMC-1500 can also interface to encoders with pulse and direction signals.

There is no limit on encoder line density, however, the input frequency to the controller must not exceed 2,000,000 full encoder cycles/second or 8,000,000 quadrature counts/sec. For example, if the encoder line density is 10,000 cycles per inch, the maximum speed is 200 inches/second.

The standard voltage level is TTL (zero to five volts), however, voltage levels up to 12 Volts are acceptable. If using differential signals, 12 Volts can be input directly to the DMC-1500. Single-ended 12 Volt signals require a bias voltage input to the complementary inputs.

The DMC-1500 can accept analog feedback instead of an encoder for any axis. Note: the DMC-1580 controller must be modified by the factory to allow for analog feedback on axis H. For more information see description of analog feedback in Chapter 2 under section entitled "Test the encoder operation".

To interface with other types of position sensors such as resolvers or absolute encoders, Galil can customize the controller and command set. Please contact Galil to talk to one of our applications engineers about your particular system requirements.

Watch Dog Timer

The DMC-1500 provides an internal watch dog timer which checks for proper microprocessor operation. The timer toggles the Amplifier Enable Output (AEN) which can be used to switch the amplifiers off in the event of a serious DMC-1500 failure. The AEN output is normally high. During power-up and if the microprocessor ceases to function properly, the AEN output will go low. The

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

error light for each axis will also turn on at this stage. A reset is required to restore the DMC-1500 to normal operation. Consult the factory for a Return Materials Authorization (RMA) Number if your DMC-1500 is damaged.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE LEFT BLANK INTENTIONALLY

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 2 Getting Started

Elements You Need

Before you start, you will need the following system elements:

1. DMC-1500 Motion Controller and included cables, RS232, 60 pin ribbon cable and 26-pin ribbon cable.

1a. For stepper motor operation, you will need an additional 20-pin ribbon cable, J4.

2. Servo motors with Optical Encoder (one per axis) or step motors

3. Power Amplifiers for motors

4. Power Supply for Amplifiers

5. PC (Personal Computer with RS232 port)

Software from Galil (Optional - but strongly recommended for first time users)

Communication Disk (COMMdisk)

-AND -

WSDK-16 Servo Design Software for Windows 3.1, and 3.11 for Workgroups

-OR -

WSDK-32 for Windows 95 or NT

ICM-1100 Interface Module (Optional, but strongly recommended). The Galil ICM-1100 is an interconnect module with screw type terminals that directly interfaces to the DMC-1500 controller. Note: An additional ICM-1100 is required for the DMC-1550 through DMC-1580.

The motors may be servo (brush type or brushless) or steppers. The amplifiers should be suitable for the motor and may be linear or pulse-width-modulated. An amplifier may have current feedback or voltage feedback.



For servo motors, the amplifiers should accept an analog signal in the +/-10 Volt range as a command. The amplifier gain should be set so that a +10V command will generate the maximum required current. For example, if the motor peak current is 10A, the amplifier gain should be 1 A/V. For velocity mode amplifiers, a command signal of 10 Volts should run the motor at the maximum required speed.



For step motors, the amplifiers should accept step and direction signals.

The WSDK software is highly recommended for first time users of the DMC-1500. It provides step-by-step instructions for system connection, tuning and analysis.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Installing the DMC-1500

Installation of a complete, operational DMC-1500 system consists of 9 steps.

Step 1. Determine overall motor configuration.

Step 2. Install jumpers on the DMC-1500.

Step 3. Configure the DIP switches on the DMC-1500.

Step 4. Connect AC power to controller

Step 5. Install communications software.

Step 6. Establish communications with Galil Software.

Step 7. Connect amplifiers and Encoders.

Step 8a. Connect standard servo motors.

Step 8b. Connect step motors.

Step 9. Tune the servo system

Step 1. Determine Overall Motor Configuration

Before setting up the motion control system, the user must determine the desired motor configuration. The DMC-1500 can control any combination of standard servo motors, and stepper motors. Other types of actuators, such as hydraulics can also be controlled, please consult Galil.

The following configuration information is necessary to determine the proper motor configuration:

Standard Servo Motor Operation:

The DMC-1500 has been setup by the factory for standard servo motor operation providing an analog command signal of +/- 10V. No hardware or software configuration is required for standard servo motor operation.

Stepper Motor Operation:

To configure the DMC-1500 for stepper motor operation, the controller requires a jumper for each stepper motor and the command, MT, must be given. The installation of the stepper motor jumper is discussed in the following section entitled "Installing Jumpers on the DMC-1500". Further instruction for stepper motor connections are discussed in Step 8b.

Step 2. Install Jumpers on the DMC-1500

The DMC-1500 has jumpers inside the controller box which may need to be installed. To access these jumpers, the cover of the controller box must be removed. The following describes each of the jumpers.

WARNING: Never open the controller box when AC power is applied to it.



For each axis that will be driving a stepper motor, a stepper mode (SM) jumper must be connected.

1580

If you using a controller with more than 4 axis, you will have two pc-cards inside the controller box. In this case, you will have 2 sets of stepper motor jumpers, one on each card. The jumpers on the bottom card will be for axes X,Y,Z and W (or A,B,C, and D) and the top will be E,F,G and H. To access the bottom card, the top card must be carefully removed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The stepper mode jumpers are located next to the GL-1800 which is the largest IC on the board. The jumper set is labeled JP40 and the individual stepper mode jumpers are labeled SMX, SMY, SMZ, SMW. The fifth jumper of the set, OPT, is for use by Galil technicians only.

The jumper set, J41, can be used to connect the controllers internal power supply to the optoisolated inputs. This may be desirable if your system will be using limit switches, home inputs digital inputs, or hardware abort **and** optoisolation is not necessary for your system. For a further explanation, see section *Bypassing the Opto-Isolation* in Chapter 3.

Step 3. Configure DIP switches on the DMC-1500

Located on the outside of the controller box is a set of 5 DIP switches.

Switch 1 is the Master Reset switch. When this switch is on, the controller will perform a master reset upon PC power up. Whenever the controller has a master reset, all programs and motion control parameters stored in EEPROM will be ERASED. During normal operation, this switch should be off.

Switch 2,3 and 4 are used to configure the baud rate of the main RS232 serial port. See section *Configuration* in Chapter 4.

Switch 5 is used to configure both serial ports for hardware handshake mode. Set this switch on for handshake mode. Please note that the Galil communication software requires that hardware handshake mode be enabled.

Step 4. Connect AC Power to the Controller

Before applying power, connect the 60-pin and 26-pin ribbons between the DMC-1500 and ICM-1100 interconnect module. The DMC-1500 requires a single AC supply voltage, single phase, 50 Hz or 60 Hz. from 90 volts to 260 volts.

WARNING: Dangerous voltages, current, temperatures and energy levels exist in this product and in its associated amplifiers and servo motor(s). Extreme caution should be exercised in the application of this equipment. Only qualified individuals should attempt to install, set up and operate this equipment.

WARNING: Never open the controller box when AC power is applied to it.

Applying power will turn on the green light power indicator.

Step 5. Install Communications Software

After you have installed the DMC-1500 controller and turned the power on to your computer, you should install software that enables communication between the controller and PC. There are several ways to do this. The easiest way is to use the communication disks available from Galil (COMMDISK VOL1 FOR DOS AND VOL2 FOR WINDOWS).

Using the COMMdisk Vol1 for Dos:

To use this disk, insert the COMMDISK VOL 1 in drive A. Type INSTALL and follow the directions.

Using the COMMdisk Vol2 for Windows (16 bit and 32 bit versions):

For Windows3.x, run the installation program, setup16.exe. For Windows 95 or Windows NT, run the installation program, setup32.exe.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step 6. Establish Communications with Galil Software

Use the supplied 9-pin RS232 ribbon cable to connect the MAIN DMC-1500 serial port to your computer or terminal at COMPORT 1. The DMC-1500 main serial port is configured as DATASET. Your computer or terminal must be configured as a DATATERM for full duplex, no parity, 8 bits data, one start bit and one stop bit.

Select the baud rate switches for 19.2 KB, 9600 B or 1200 B. The default setting is 19.2 KB.

Your computer needs to be configured as a "dumb" terminal which sends ASCII characters as they are typed to the DMC-1500. The COMMdisk from Galil provides a terminal emulator program for your computer. Follow the steps below to install and run the terminal emulator.

Dos Users:

To communicate with the DMC-1500, type TALK2DMC at the prompt. Once you have established communication, the terminal display should show a colon, :. If you do not receive a colon, press the carriage return. If a colon prompt is not returned, there is most likely an incorrect setting of the serial communications port. The user must ensure that the correct communication port and baud rate are specified when attempting to communicate with the controller. Please note that the serial port on the controller must be set for handshake mode for proper communication with Galil software. The user must also insure that the proper serial cable is being used, see appendix for pin-out of serial cable.

Windows Users:

In order for the windows software to communicate with a Galil controller, the controller must be registered in the Galil Registry. The Galil Registry is simply a list of controllers. Registration consists of telling the software the model of the controller, the address of the controller, and other information. To do this, run the program DMCREG16 for Windows 3.x or DMCREG32 for Windows 95 and NT. The DMCREG window will appear. Select Registry from the menu.

Note: If you are using DMCREG for the first time, no controllers will exist in the Galil Register. This is normal.

The registry window is equipped with buttons to **Add, Change,** or **Delete** a controller. Pressing any of these buttons will bring up the Set Registry Information window. (It should be noted that if you wish to change information on any existing controller, it should be selected before clicking **Change**, even if it is the only controller listed in the Registry.)

Use the **Add** button to add a new entry to the Registry. You will need to supply the Galil Controller type. For any address changes to take effect, a model number must be entered. If you are changing an existing controller, this field will already have an entry. If you are adding a controller, it will not. Pressing the down arrow to the right of this field will reveal a menu of valid controller types. *You should choose DMC-1500.* The registry information will show a default comm port of 2 and a default baud rate of 9600 appears. This information should be changed as necessary to reflect the computers comm port and the baud rate as set by the controller's DIP switches. The registry entry also displays timeout and delay information. These are advanced parameters which should only be modified by advanced users (see software documentation for more information).

Once you have set the appropriate Registry information for your controller, exit from the DMCREG program. You will now be able to run communication software.

If you are using Windows 3.x, run the program DTERM16.EXE and if you are using Windows 95 or Windows NT, run the program DTERM32.EXE. From the file menu, select Startup. You will now see the registry information. Select the entry for your controller. Note: If you have only one entry, you still must select this controller for the software to establish communications. Once the entry has been selected, click on the **OK** button. If the software has successfully established communications with the controller, the registry entry will be displayed at the top of the screen.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If you are not properly communicating with the controller, the program will pause for 3-15 seconds. The top of the screen will display the message "Status: not connected with Galil motion controller" and the following error will appear: "STOP - Unable to establish communication with the Galil controller. A time-out occurred while waiting for a response from the Galil controller." If this message appears, you must click OK. In this case, there is most likely an incorrect setting of the serial communications port. The user must ensure that the correct communication port and baud rate are specified when attempting to communicate with the controller. Please note that the serial port on the controller must be set for handshake mode for proper communication with Galil software. The user must also insure that the proper serial cable is being used, see appendix for pin-out of serial cable.

Once you establish communications, click on the menu for terminal and you will receive a colon prompt. Communicating with the controller is described in later sections.

Sending Test Commands to the Terminal:

After you connect your terminal, press <carriage return> or the <enter> key on your keyboard. In response to carriage return (CR), the controller responds with a colon, :

Now type

TPX (CR)

This command directs the controller to return the current position of the X axis. The controller should respond with a number such as

0000000

The RS232 communication is established.

Step 7. Connect Amplifiers and Encoders.

Once you have established communications between the software and the DMC-1500, you are ready to connect the rest of the motion control system. The motion control system typically consists of an ICM-1100 Interface Module, an amplifier for each axis of motion, and a motor to transform the current from the amplifier into torque for motion. Galil also offers the AMP-11X0 series Interface Modules which are ICM-1100's equipped with servo amplifiers for brush type DC motors.

If you are using an ICM-1100, connect the 100-pin ribbon cable to the DMC-1500 and to the connector located on the AMP-11X0 or ICM-1100 board. The ICM-1100 provides screw terminals for access to the connections described in the following discussion.

1580

Motion Controllers with more than 4 axes require a second ICM-1100 or AMP-11X0 and second 100-pin cable.

System connection procedures will depend on system components and motor types. Any combination of motor types can be used with the DMC-1500.

Here are the first steps for connecting a motion control system:

Step A. Connect the motor to the amplifier *with no connection to the controller*. Consult the amplifier documentation for instructions regarding proper connections. Connect and turn-on the amplifier power supply. If the amplifiers are operating properly, the motor should stand still even when the amplifiers are powered up.

Step B. Connect the amplifier enable signal.

Before making any connections from the amplifier to the controller, you need to verify that the ground level of the amplifier is either floating or at the same potential as earth.

WARNING: When the amplifier ground is not isolated from the power line or when it has a different potential than that of the computer ground, serious damage may result to the computer controller and amplifier.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If you are not sure about the potential of the ground levels, connect the two ground signals (amplifier ground and earth) by a 10 KΩ resistor and measure the voltage across the resistor. Only if the voltage is zero, connect the two ground signals directly.

The amplifier enable signal is used by the controller to disable the motor. It will disable the motor when the watchdog timer activates, the motor-off command, MO, is given, or the position error exceeds the error limit with the "Off-On-Error" function enabled (see the command OE for further information).

The standard configuration of the AEN signal is TTL active high. In other words, the AEN signal will be high when the controller expects the amplifier to be enabled. The polarity and the amplitude can be changed if you are using the ICM-1100 interface board. To change the polarity from active high (5 volts = enable, zero volts = disable) to active low (zero volts = enable, 5 volts = disable), replace the 7407 IC with a 7406. Note that many amplifiers designate the enable input as 'inhibit'.

To change the voltage level of the AEN signal, note the state of the resistor pack on the ICM-1100. When Pin 1 is on the 5V mark, the output voltage is 0-5V. To change to 12 volts, pull the resistor pack and rotate it so that Pin 1 is on the 12 volt side. If you remove the resistor pack, the output signal is an open collector, allowing the user to connect an external supply with voltages up to 24V.

On the ICM-1100, the amplifier enable signal is labeled AENX for the X axis. Connect this signal to the amplifier (figure 2.3) and issue the command, MO, to disable the motor amplifiers - often this is indicated by an LED on the amplifier.

Step C. Connect the encoders

For stepper motor operation, an encoder is optional.

For servo motor operation, if you have a preferred definition of the forward and reverse directions, make sure that the encoder wiring is consistent with that definition.

The DMC-1500 accepts single-ended or differential encoder feedback with or without an index pulse. If you are not using the AMP-11X0 or the ICM-1100 you will need to consult the appendix for the encoder pinouts for connection to the motion controller. The AMP-11X0 and the ICM-1100 can accept encoder feedback from a 10-pin ribbon cable or individual signal leads. For a 10-pin ribbon cable encoder, connect the cable to the protected header connector labeled X ENCODER (repeat for each axis necessary). For individual wires, simply match the leads from the encoder you are using to the encoder feedback inputs on the interconnect board. The signal leads are labeled XA+ (channel A), XB+ (channel B), and XI+. For differential encoders, the complement signals are labeled XA-, XB-, and XI-.

Note: When using pulse and direction encoders, the pulse signal is connected to CHA and the direction signal is connected to CHB. The controller must be configured for pulse and direction with the command CE. See the command summary for further information on the command CE.

Step D. Verify proper encoder operation.

Start with the X encoder first. Once it is connected, turn the motor shaft and interrogate the position with the instruction TPX <return>. The controller response will vary as the motor is turned.

At this point, if TPX does not vary with encoder rotation, there are three possibilities:

1. The encoder connections are incorrect - check the wiring as necessary.
2. The encoder has failed - using an oscilloscope, observe the encoder signals. Verify that both channels A and B have a peak magnitude between 5 and 12 volts. Note that if only one encoder channel fails,

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

the position reporting varies by one count only. If the encoder failed, replace the encoder. If you cannot observe the encoder signals, try a different encoder.

3. There is a hardware failure in the controller- connect the same encoder to a different axis. If the problem disappears, you probably have a hardware failure. Consult the factory for help.

Step 8a. Connect Standard Servo Motors

The following discussion applies to connecting the DMC-1500 controller to standard servo motor amplifiers:

The motor and the amplifier may be configured in the torque or the velocity mode. In the torque mode, the amplifier gain should be such that a 10 Volt signal generates the maximum required current. In the velocity mode, a command signal of 10 Volts should run the motor at the maximum required speed.

Step by step directions on servo system setup are also included on the WSDK (Windows Servo Design Kit) software offered by Galil. See section on WSDK for more details.

Step A. Check the Polarity of the Feedback Loop

It is assumed that the motor and amplifier are connected together and that the encoder is operating correctly (Step B). Before connecting the motor amplifiers to the controller, read the following discussion on setting Error Limits and Torque Limits. Note that this discussion only uses the X axis as an examples.

Step B. Set the Error Limit as a Safety Precaution

Usually, there is uncertainty about the correct polarity of the feedback. The wrong polarity causes the motor to run away from the starting position. Using a terminal program, such as DMCTERM, the following parameters can be given to avoid system damage:

Input the commands:

ER 2000 <CR> Sets error limit on the X axis to be 2000 encoder counts

OE 1 <CR> Disables X axis amplifier when excess position error exists

If the motor runs away and creates a position error of 2000 counts, the motor amplifier will be disabled. **Note:** This function requires the AEN signal to be connected from the controller to the amplifier.

Step C. Set Torque Limit as a Safety Precaution

To limit the maximum voltage signal to your amplifier, the DMC-1500 controller has a torque limit command, TL. This command sets the maximum voltage output of the controller and can be used to avoid excessive torque or speed when initially setting up a servo system.

When operating an amplifier in torque mode, the voltage output of the controller will be directly related to the torque output of the motor. The user is responsible for determining this relationship using the documentation of the motor and amplifier. The torque limit can be set to a value that will limit the motors output torque.

When operating an amplifier in velocity or voltage mode, the voltage output of the controller will be directly related to the velocity of the motor. The user is responsible for determining this relationship using the documentation of the motor and amplifier. The torque limit can be set to a value that will limit the speed of the motor.

For example, the following command will limit the output of the controller to 1 volt on the X axis:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TL 1 <CR>

Note: Once the correct polarity of the feedback loop has been determined, the torque limit should, in general, be increased to the default value of 9.99. The servo will not operate properly if the torque limit is below the normal operating range. See description of TL in the command reference.

Step D. Connect the Motor

Once the parameters have been set, connect the analog motor command signal (ACMD) to the amplifier input.

To test the polarity of the feedback, command a move with the instruction:

PR 1000 <CR> Position relative 1000 counts

BGX <CR> Begin motion on X axis

When the polarity of the feedback is wrong, the motor will attempt to run away. The controller should disable the motor when the position error exceeds 2000 counts. If the motor runs away, the polarity of the loop must be inverted.

Note: Inverting the Loop Polarity

When the polarity of the feedback is incorrect, the user must invert the loop polarity and this may be accomplished by several methods. If you are driving a brush-type DC motor, the simplest way is to invert the two motor wires (typically red and black). For example, switch the M1 and M2 connections going from your amplifier to the motor. When driving a brushless motor, the polarity reversal may be done with the encoder. If you are using a single-ended encoder, interchange the signal CHA and CHB. If, on the other hand, you are using a differential encoder, interchange only CHA+ and CHA-. The loop polarity and encoder polarity can also be affected through software with the MT, and CE commands. For more details on the MT command or the CE command, see the Command Reference section.

Note: Reversing the Direction of Motion

If the feedback polarity is correct but the direction of motion is opposite to the desired direction of motion, reverse the motor leads AND the encoder signals.

When the position loop has been closed with the correct polarity, the next step is to adjust the PID filter parameters, KP, KD and KI. It is necessary to accurately tune your servo system to ensure fidelity of position and minimize motion oscillation as described in the next section.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

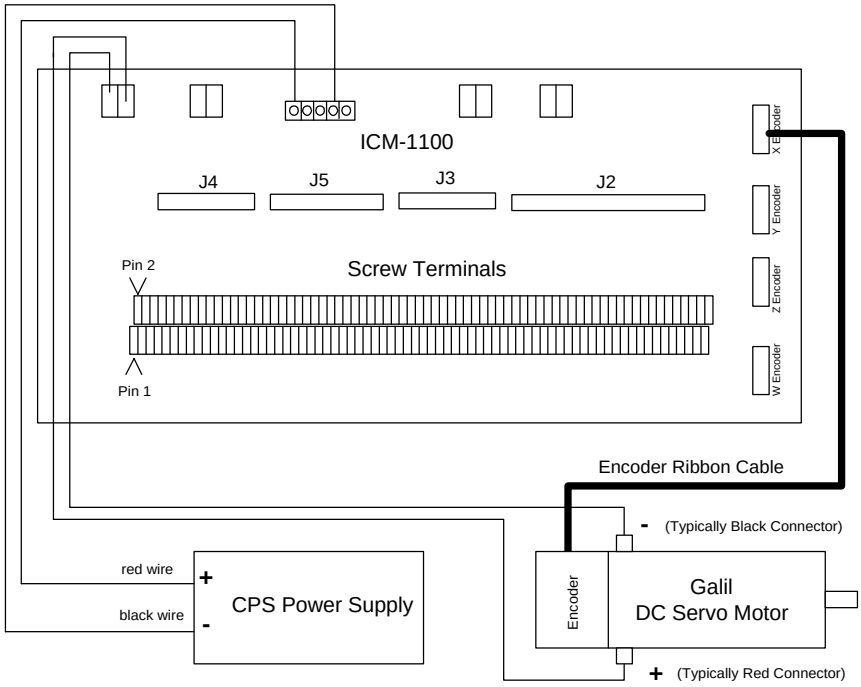


Figure 2-2 - System Connections with the AMP-1100Amplifier. Note: this figure shows a Galil Motor and Encoder which uses a flat ribbon cable to connect to the AMP-1100 unit.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

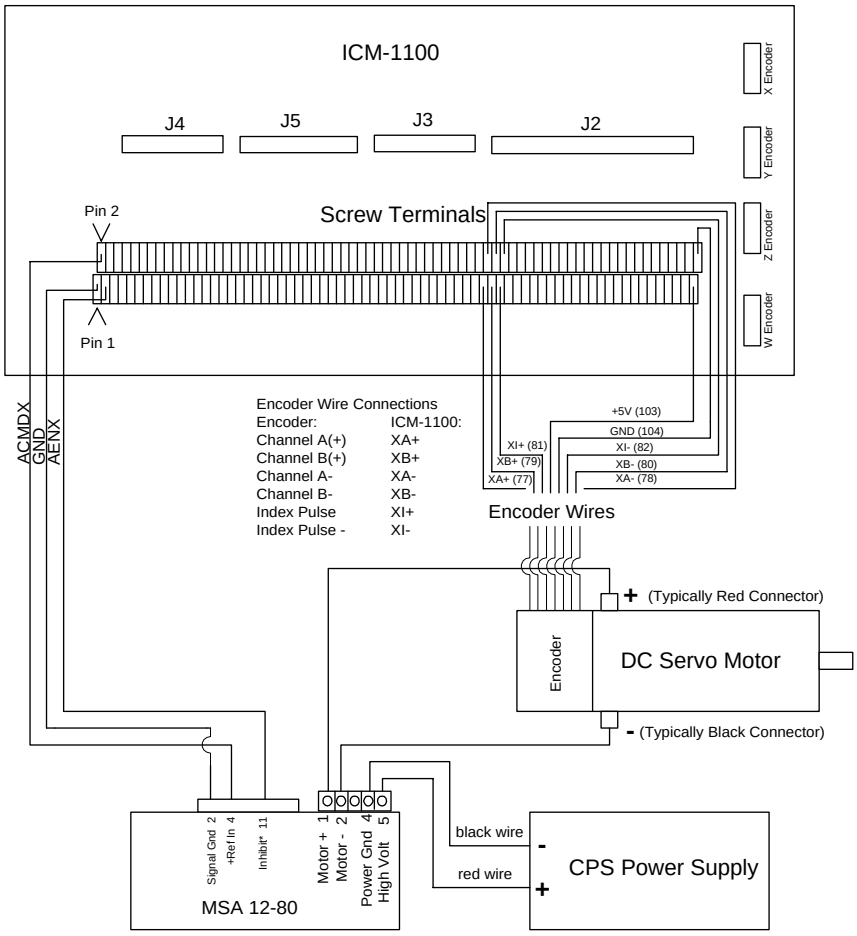


Figure 2-3 System Connections with a separate amplifier (MSA 12-80). This diagram shows the connections for a standard DC Servo Motor and encoder.

Step 8b. Connect Step Motors

In Stepper Motor operation, the pulse output signal has a 50% duty cycle. Step motors operate open loop and do not require encoder feedback. When a stepper is used, the auxiliary encoder for the corresponding axis is unavailable for an external connection. If an encoder is used for position feedback, connect the encoder to the main encoder input corresponding to that axis. The commanded position of the stepper can be interrogated with RP or DE. The encoder position can be interrogated with TP.

The frequency of the step motor pulses can be smoothed with the filter parameter, KS. The KS parameter has a range between 0.5 and 8, where 8 implies the largest amount of smoothing. *See Command Reference regarding KS.*

The DMC-1500 profiler commands the step motor amplifier. All DMC-1500 motion commands apply such as PR, PA, VP, CR and JG. The acceleration, deceleration, slew speed and smoothing are also used. Since step motors run open-loop, the PID filter does not function and the position error is not generated.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To connect step motors with the DMC-1500 you must follow this procedure:

Step A. Install SM jumpers

Each axis of the DMC-1500 that will operate a stepper motor must have the corresponding stepper motor jumper installed. For a discussion of SM jumpers, see step 2.

Step B. Connect step and direction signals.

Make connections from controller to motor amplifiers. (These signals are labeled PULSX and DIRX for the x-axis on the ICM-1100). Consult the documentation for your step motor amplifier.

Step C. Configure DMC-1500 for motor type using MT command. You can configure the DMC-1500 for active high or active low pulses. Use the command MT 2 for active high step motor pulses and MT -2 for active low step motor pulses. *See description of the MT command in the Command Reference.*

Step 9. Tune the Servo System

Adjusting the tuning parameters is required when using servo motors. The system compensation provides fast and accurate response. The following presentation suggests a simple and easy way for compensation. More advanced design methods are available with software design tools from Galil, such as the Servo Design Kit (SDK software)

The filter has three parameters: the damping, KD; the proportional gain, KP; and the integrator, KI. The parameters should be selected in this order.

To start, set the integrator to zero with the instruction

KI 0 (CR) Integrator gain

and set the proportional gain to a low value, such as

KP 1 (CR) Proportional gain

KD 100 (CR) Derivative gain

For more damping, you can increase KD (maximum is 4095). Increase gradually and stop after the motor vibrates. A vibration is noticed by audible sound or by interrogation. If you send the command

TE X (CR) Tell error

a few times, and get varying responses, especially with reversing polarity, it indicates system vibration. When this happens, simply reduce KD.

Next you need to increase the value of KP gradually (maximum allowed is 1023). You can monitor the improvement in the response with the Tell Error instruction

KP 10 (CR) Proportion gain

TE X (CR) Tell error

As the proportional gain is increased, the error decreases.

Again, the system may vibrate if the gain is too high. In this case, reduce KP. Typically, KP should not be greater than KD/4. (Only when the amplifier is configured in the current mode).

Finally, to select KI, start with zero value and increase it gradually. The integrator eliminates the position error, resulting in improved accuracy. Therefore, the response to the instruction

TE X (CR)

becomes zero. As KI is increased, its effect is amplified and it may lead to vibrations. If this occurs, simply reduce KI. Repeat tuning for the Y, Z and W axes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For a more detailed description of the operation of the PID filter and/or servo system theory, see Chapter 10 - Theory of Operation.

Design Examples

Here are a few examples for tuning and using your controller. These examples have remarks next to each command - these remarks must not be included in the actual program.

Example 1 - System Set-up

This example assigns the system filter parameters, error limits and enables the automatic error shut-off.

Instruction	Interpretation
KP10,10,10,10,10,10,10,10	Set gains for A,B,C,D,E,and H axes
KP10,10,10,10,10,10,10,10	Set gains for A,B,C,D,E,and H axes
KP*=10	Alternate method for setting gain on all axes
KPX=10	Alternate method for setting X (or A) axis gain
KPA=10	Alternate method for setting A (or X) axis gain

1580

The X,Y,Z and W axes can also be referred to as the A,B,C, and D axes.

Instruction	Interpretation
OE 1,1,1,1,1,1,1,1	Enable automatic Off on Error function for all axes
ER*=1000	Set error limit for all axes to 1000 counts
KP10,10,10,10,10,10,10,10	Set gains for A,B,C,D,E,and H axes
KP*=10	Alternate method for setting gain on all axes
KPX=10	Alternate method for setting X (or A) axis gain
KPA=10	Alternate method for setting A (or X) axis gain
KPZ=10	Alternate method for setting Z axis gain
KPD=10	Alternate method for setting D axis gain
KPH=10	Alternate method for setting H axis gain

Example 2 - Profiled Move

Objective: Rotate the X axis a distance of 10,000 counts at a slew speed of 20,000 counts/sec and an acceleration and deceleration rates of 100,000 counts/s2. In this example, the motor turns and stops:

Instruction	Interpretation
PR 10000	Distance
SP 20000	Speed
DC 100000	Deceleration
AC 100000	Acceleration
BG X	Start Motion

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 3 - Multiple Axes

Objective: Move the four axes independently.

Instruction	Interpretation
PR 500,1000,600,-400	Distances of X,Y,Z,W
SP 10000,12000,20000,10000	Slew speeds of X,Y,Z,W
AC 100000,10000,100000,100000	Accelerations of X,Y,Z,W
DC 80000,40000,30000,50000	Decelerations of X,Y,Z,W
BG XZ	Start X and Z motion
BG YW	Start Y and W motion

Example 4 - Independent Moves

The motion parameters may be specified independently as illustrated below.

Instruction	Interpretation
PR ,300,-600	Distances of Y and Z
SP ,2000	Slew speed of Y
DC ,80000	Deceleration of Y
AC, 100000	Acceleration of Y
SP ,,40000	Slew speed of Z
AC ,,100000	Acceleration of Z
DC ,,150000	Deceleration of Z
BG Z	Start Z motion
BG Y	Start Y motion

Example 5 - Position Interrogation

The position of the four axes may be interrogated with the instruction, TP.

Instruction	Interpretation
TP	Tell position all four axes
TP X	Tell position - X axis only
TP Y	Tell position - Y axis only
TP Z	Tell position - Z axis only
TP W	Tell position - W axis only

The position error, which is the difference between the commanded position and the actual position can be interrogated with the instruction TE.

Instruction	Interpretation
TE	Tell error - all axes

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TE X	Tell error - X axis only
TE Y	Tell error - Y axis only
TE Z	Tell error - Z axis only
TE W	Tell error - W axis only

Example 6 - Absolute Position

Objective: Command motion by specifying the absolute position.

Instruction	Interpretation
DP 0,2000	Define the current positions of X,Y as 0 and 2000
PA 7000,4000	Sets the desired absolute positions
BG X	Start X motion
BG Y	Start Y motion

After both motions are complete, the X and Y axes can be command back to zero:

PA 0,0	Move to 0,0
BG XY	Start both motions

Example 7 - Velocity Control

Objective: Drive the X and Y motors at specified speeds.

Instruction	Interpretation
JG 10000,-20000	Set Jog Speeds and Directions
AC 100000, 40000	Set accelerations
DC 50000,50000	Set decelerations
BG XY	Start motion

after a few seconds, send the following command:

JG -40000	New X speed and Direction
TV X	Returns X speed

and then

JG ,20000	New Y speed
TV Y	Returns Y speed

These cause velocity changes including direction reversal. The motion can be stopped with the instruction

ST	Stop

Example 8 - Operation Under Torque Limit

The magnitude of the motor command may be limited independently by the instruction TL.

Instruction	Interpretation
TL 0.2	Set output limit of X axis to 0.2 volts
JG 10000	Set X speed
BG X	Start X motion

In this example, the X motor will probably not move since the output signal will not be sufficient to overcome the friction. If the motion starts, it can be stopped easily by a touch of a finger.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Increase the torque level gradually by instructions such as

Instruction	Interpretation
TL 1.0	Increase torque limit to 1 volt.
TL 9.98	Increase torque limit to maximum, 9.98 Volts.

The maximum level of 10 volts provides the full output torque.

Example 9 - Interrogation

The values of the parameters may be interrogated. Some examples …

Instruction	Interpretation
KP ?	Return gain of X axis.
KP ,,?	Return gain of Z axis.
KP ?,?,?,?	Return gains of all axes.

Many other parameters such as KI, KD, FA, can also be interrogated. The command reference denotes all commands which can be interrogated.

Example 10 - Operation in the Buffer Mode

The instructions may be buffered before execution as shown below.

Instruction	Interpretation
PR 600000	Distance
SP 10000	Speed
WT 10000	Wait 10000 milliseconds before reading the next instruction
BG X	Start the motion

Example 11 - Motion Programs

Motion programs may be edited and stored in the controllers on-board memory.

The instruction

ED	Edit mode

moves the operation to the editor mode where the program may be written and edited. The editor provides the line number. For example, in response to the first ED command, the first line is zero.

Line #	Instruction	Interpretation
000	#A	Define label
001	PR 700	Distance
002	SP 2000	Speed
003	BGX	Start X motion
004	EN	End program

To exit the editor mode, input <cntrl>Q. The program may be executed with the command.

XQ #A	Start the program running

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example 12 - Motion Programs with Loops

Motion programs may include conditional jumps as shown below.

Instruction	Interpretation
#A	Label
DP 0	Define current position as zero
V1=1000	Set initial value of V1
#Loop	Label for loop
PA V1	Move X motor V1 counts
BG X	Start X motion
AM X	After X motion is complete
WT 500	Wait 500 ms
TP X	Tell position X
V1=V1+1000	Increase the value of V1
JP #Loop,V1<10001	Repeat if V1<10001
EN	End

After the above program is entered, quit the Editor Mode, <cntrl>Q. To start the motion, command:

XQ #A	Execute Program #A

Example 13 - Motion Programs with Trippoints

The motion programs may include trippoints as shown below.

Instruction	Interpretation
#B	Label
DP 0,0	Define initial positions
PR 30000,60000	Set targets
SP 5000,5000	Set speeds
BGX	Start X motion
AD 4000	Wait until X moved 4000
BGY	Start Y motion
AP 6000	Wait until position X=6000
SP 2000,50000	Change speeds
AP ,50000	Wait until position Y=50000
SP ,10000	Change speed of Y
EN	End program

To start the program, command:

XQ #B	Execute Program #B

Example 14 - Control Variables

Objective: To show how control variables may be utilized.

Instruction	Interpretation
#A;DP0	Label; Define current position as zero

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PR 4000	Initial position
SP 2000	Set speed
BGX	Move X
AMX	Wait until move is complete
WT 500	Wait 500 ms
#B	
V1 = _TPX	Determine distance to zero
PR -V1/2	Command X move 1/2 the distance
BGX	Start X motion
AMX	After X moved
WT 500	Wait 500 ms
V1=	Report the value of V1
JP #C, V1=0	Exit if position=0
JP #B	Repeat otherwise
#C	Label #C
EN	End of Program

To start the program, command

XQ #A	Execute Program #A

This program moves X to an initial position of 1000 and returns it to zero on increments of half the distance. Note, _TPX is an internal variable which returns the value of the X position. Internal variables may be created by preceding a DMC-1500 instruction with an underscore, _.

Example 15 - Linear Interpolation

Objective: Move X,Y,Z motors distance of 7000,3000,6000, respectively, along linear trajectory. Namely, motors start and stop together.

Instruction	Interpretation
LM XYZ	Specify linear interpolation axes
LI 7000,3000,6000	Relative distances for linear interpolation
LE	Linear End
VS 6000	Vector speed
VA 20000	Vector acceleration
VD 20000	Vector deceleration
BGS	Start motion

Example 16 - Circular Interpolation

Objective: Move the XY axes in circular mode to form the path shown on Fig. 2-4. Note that the vector motion starts at a local position (0,0) which is defined at the beginning of any vector motion sequence. See application programming for further information.

Instruction	Interpretation
VM XY	Select XY axes for circular interpolation
VP -4000,0	Linear segment
CR 2000,270,-180	Circular segment

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VP 0,4000	Linear segment
CR 2000,90,-180	Circular segment
VS 1000	Vector speed
VA 50000	Vector acceleration
VD 50000	Vector deceleration
VE	End vector sequence
BGS	Start motion

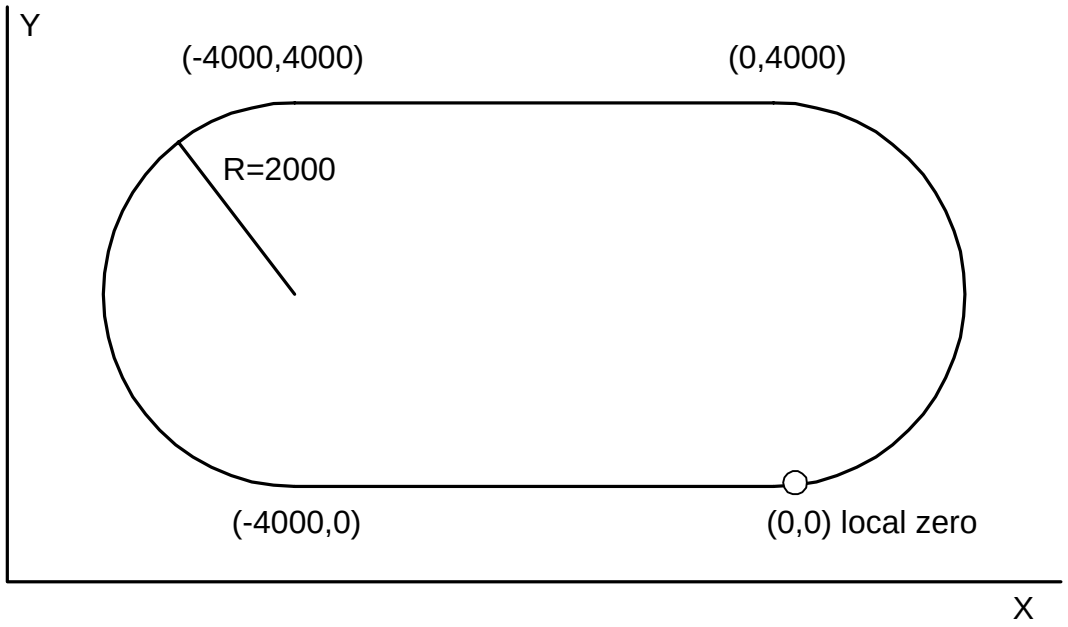


Figure 2-4 Motion Path for Example 16

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 3 Connecting Hardware

Overview

The DMC-1500 provides optoisolated digital inputs for forward limit, reverse limit, home, and abort signals. The controller also has 8 optoisolated, uncommitted inputs (for general use) as well as 8 TTL outputs and 7 analog inputs configured for voltages between +/- 10 volts.

1580

Controllers with 5 or more axes have an additional 8 TTL level inputs and 8 TTL level outputs.

This chapter describes the inputs and outputs and their proper connection.

To access the analog inputs or general inputs 5-8 or all outputs except OUT1, connect the 26-pin ribbon cable to the 26-pin J5 IDC connector from the DMC-1500 to the AMP-11X0 or ICM-1100 board.

If you plan to use the auxiliary encoder feature of the DMC-1500, you must also connect a 20-pin ribbon cable from the 20-pin J3 header connector on the DMC-1500 to the 26-pin J3 header connector on the AMP-11X0 or ICM-1100. This cable is not shipped unless requested when ordering.

Using Opto-isolated Inputs

Limit Switch Input

The forward limit switch (FLSx) inhibits motion in the forward direction immediately upon activation of the switch. The reverse limit switch (RLSx) inhibits motion in the reverse direction immediately upon activation of the switch. If a limit switch is activated during motion, the controller will make a decelerated stop using the deceleration rate previously set with the DC command. The motor will remain in a servo state after the limit switch has been activated and will hold motor position.

When a forward or reverse limit switch is activated, the current application program that is running will be interrupted and the controller will automatically jump to the #LIMSWI subroutine if one exists. This is a subroutine which the user can include in any motion control program and is useful for executing specific instructions upon activation of a limit switch.

After a limit switch has been activated, further motion in the direction of the limit switch will not be possible until the logic state of the switch returns back to an inactive state. This usually involves physically opening the tripped switch. Any attempt at further motion before the logic state has been reset will result in the following error: “022 - Begin not possible due to limit switch” error.

The operands, _LFx and _LRx, return the state of the forward and reverse limit switches, respectively (x represents the axis, X,Y,Z,W etc.). The value of the operand is either a ‘0’ or ‘1’ corresponding to the logic state of the limit switch. Using a terminal program, the state of a limit switch can be printed to the screen with the command, MG _LFx or MG _LFx. This prints the value of the limit switch operands for the 'x' axis. The logic state of the limit switches can also be interrogated with the TS command. For more details on TS see the Command Reference.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Home Switch Input

The Home inputs are designed to provide mechanical reference points for a motion control application. A transition in the state of a Home input alerts the controller that a particular reference point has been reached by a moving part in the motion control system. A reference point can be a point in space or an encoder index pulse.

The Home input detects any transition in the state of the switch and toggles between logic states 0 and 1 at every transition. A transition in the logic state of the Home input will cause the controller to execute a homing routine specified by the user.

There are three homing routines supported by the DMC-1500: Find Edge (FE), Find Index (FI), and Standard Home (HM).

The Find Edge routine is initiated by the command sequence: FEX <return>, BGX <return>. The Find Edge routine will cause the motor to accelerate, then slew at constant speed until a transition is detected in the logic state of the Home input. The motor will then decelerate to a stop. The acceleration rate, deceleration rate and slew speed are specified by the user, prior to the movement, using the commands AC, DC, and SP. *It is recommended that a high deceleration value be used so the motor will decelerate rapidly after sensing the Home switch.*

The Find Index routine is initiated by the command sequence: FIX <return>, BGX <return>. Find Index will cause the motor to accelerate to the user-defined slew speed (SP) at a rate specified by the user with the AC command and slew until the controller senses a change in the index pulse signal from low to high. The motor then decelerates to a stop at the rate previously specified by the user with the DC command. *Although Find Index is an option for homing, it is not dependent upon a transition in the logic state of the Home input, but instead is dependent upon a transition in the level of the index pulse signal.*

The Standard Homing routine is initiated by the sequence of commands HMX <return>, BGX <return>. Standard Homing is a combination of Find Edge and Find Index homing. Initiating the standard homing routine will cause the motor to slew until a transition is detected in the logic state of the Home input. The motor will accelerate at the rate specified by the command, AC, up to the slew speed. After detecting the transition in the logic state on the Home Input, the motor will decelerate to a stop at the rate specified by the command, DC. After the motor has decelerated to a stop, it switches direction and approaches the transition point at the speed of 256 counts/sec. When the logic state changes again, the motor moves forward (in the direction of increasing encoder count) at the same speed, until the controller senses the index pulse. After detection, it decelerates to a stop and defines this position as 0. The logic state of the Home input can be interrogated with the command MG _HMX. This command returns a 0 or 1 if the logic state is low or high, respectively. The state of the Home input can also be interrogated indirectly with the TS command.

For examples and further information about Homing, see command HM, FI, FE of the Command Reference and the section entitled ‘Homing’ in the Programming Motion Section of this manual.

Abort Input

The function of the Abort input is to immediately stop the controller upon transition of the logic state.

NOTE: The response of the abort input is significantly different from the response of an activated limit switch. When the abort input is activated, the controller stops generating motion commands immediately, whereas the limit switch response causes the controller to make a decelerated stop.

NOTE: The effect of an Abort input is dependent on the state of the off-on-error function for each axis. If the Off-On-Error function is enabled for any given axis, the motor for that axis will be turned off when the abort signal is generated. This could cause the motor to ‘coast’ to a stop since it is no longer under servo control. If the Off-On-Error function is disabled, the motor will decelerate to a stop as fast as mechanically possible and the motor will remain in a servo state.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

All motion programs that are currently running are terminated when a transition in the Abort input is detected. For information on setting the Off-On-Error function, see the Command Reference, OE.

NOTE: The error LED does not light up when the Abort Input is active.

Uncommitted Digital Inputs

The DMC-1500 has 8 uncommitted opto-isolated inputs. These inputs are specified as INx where x specifies the input number, 1 through 24. These inputs allow the user to monitor events external to the controller. For example, the user may wish to have the x-axis motor move 1000 counts in the positive direction when the logic state of IN1 goes high.

1580

Controllers with 5 or more axes have 16 opto-isolated inputs and 8 TTL level inputs. .

For controllers with more than 4 axes, the inputs 9-16 and the limit switch inputs for the additional axes are accessed through the second 100-pin connector.

IN9-IN16	INCOM
FLE,RLE,HOMEE	LSCOM
FLF,RLF,HOMEF	
FLG,RLG,HOMEG	
FLH,RLH,HOMEH	

A logic zero is generated when at least 1mA of current flows from the common to the input. A positive voltage (with respect to the input) must be supplied at the common. This can be accomplished by connecting a voltage in the range of +5V to +28V into INCOM of the input circuitry from a separate power supply.

Wiring the Optoisolated Inputs

The default state of the controller configures all inputs to be interpreted as a logic one without any connection. The inputs must be brought low to be interpreted as a zero. With regard to limit switches, a limit switch is considered to be activated when the input is brought low (or a switch is closed to ground). Some inputs can be configured to be active when the input is high - see section *Changing Optoisolated Inputs from Active High to Active Low.*

The optoisolated inputs are organized into groups. For example, the general inputs, IN1-IN8, and the ABORT input are one group. Each group has a common signal which supplies current for the inputs in the group. In order to use an input, the associated common signal must be connected to voltage between +5 and +18 volts, see discussion below.

The optoisolated inputs are connected in the following groups (these inputs are accessed through the 26-pin J5 header).

Group	Common Signal
IN1-IN8, ABORT	INCOM
FLX,RLX,HOMEX	LSCOM
FLY,RLY,HOMEY	
FLZ,RLZ,HOMEZ	
FLW,RLW,HOMEW	

1580

For controllers with more than 4 axes, the inputs 9-16 and the limit switch inputs for the additional axes are accessed through a separate connector, JD5.

Group	Common Signal

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IN9-IN16	INCOM
FLE,RLE,HOMEE	LSCOM
FLF,RLF,HOMEF	
FLG,RLG,HOMEG	
FLH,RLH,HOMEH	

A logic zero is generated when at least 1mA of current flows from the common signal to the input. A positive voltage (with respect to the input) must be supplied at the common. This can be accomplished by connecting a voltage in the range of +5V to +18V into INCOM of the input circuitry from a separate power supply

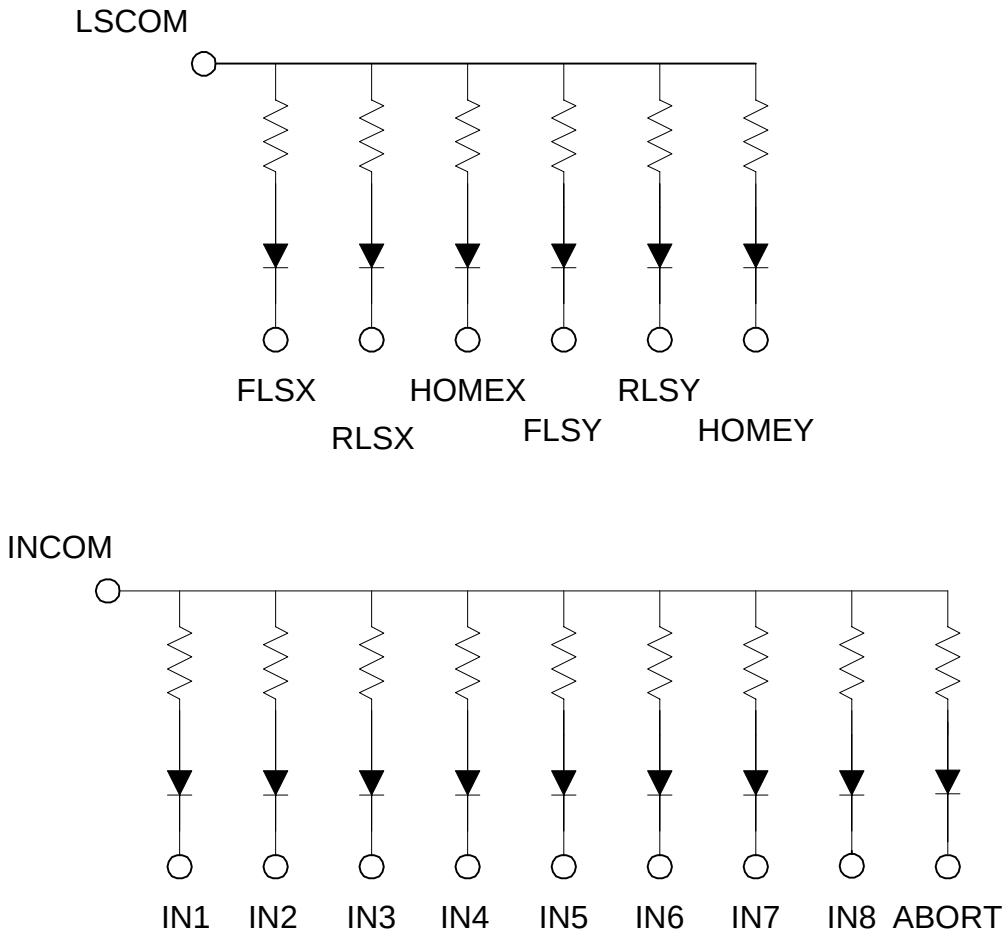


Figure 3-1. The Optoisolated Inputs

Using an Isolated Power Supply

To take full advantage of opto-isolation, an isolated power supply should be used to provide the voltage at the input common connection. When using an isolated power supply, do not connect the ground of the isolated power to the ground of the controller. A power supply in the voltage range between 5 to 18 Volts may be applied directly (see Figure 3-2). For voltages greater than 18 Volts, a resistor, R, is needed in series with the input such that

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1 mA < V supply/(R + 1.5KΩ) < 15 mA

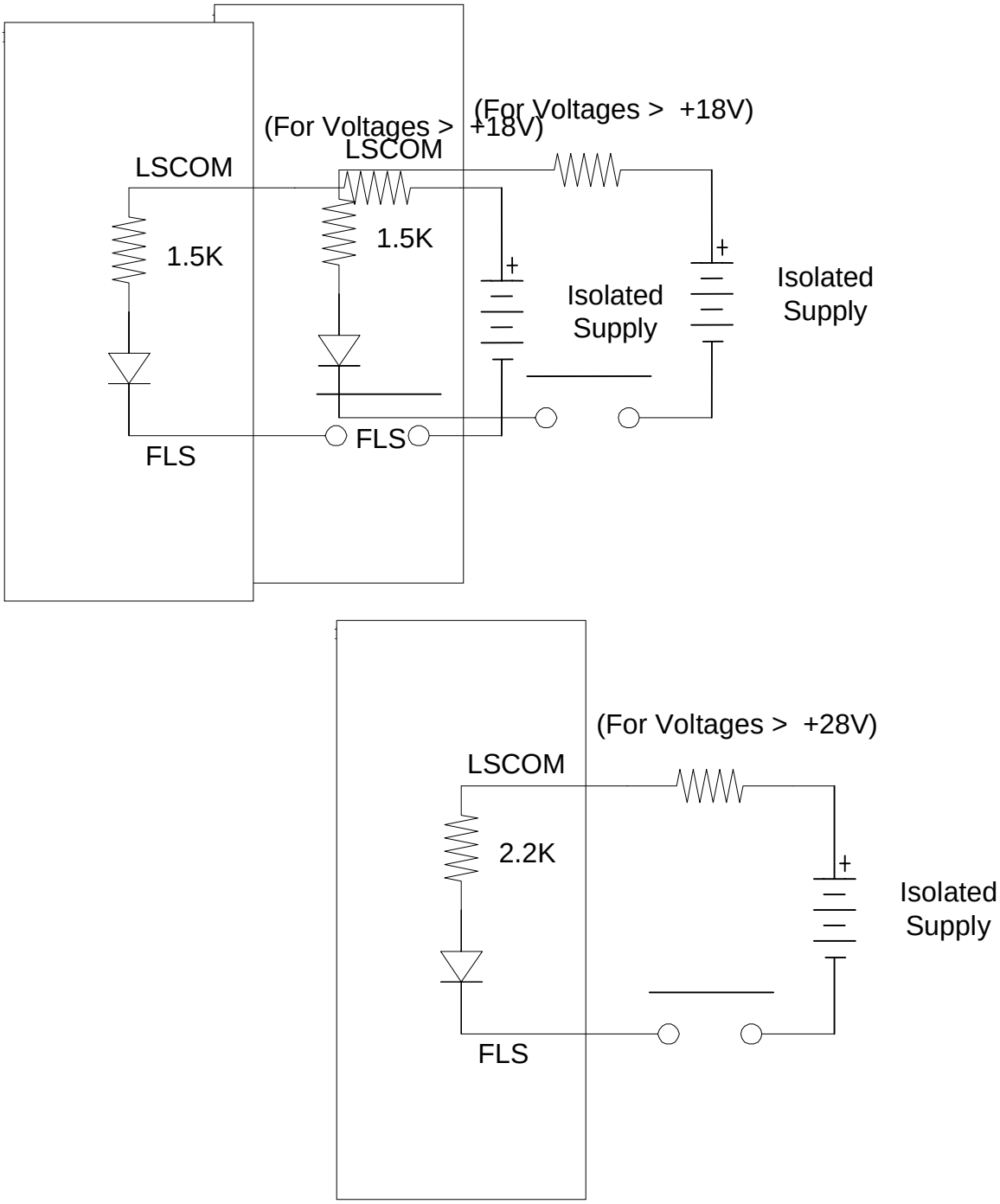


Figure 3-2. Connecting a single Limit or Home Switch to an Isolated Supply

NOTE: As stated in Chapter 2, the wiring is simplified when using the ICM-1100 or AMP-11x0 interface board. This board accepts the signals from the ribbon cables of the DMC-1500 and provides phoenix-type screw terminals. A picture of the ICM-1100 can be seen on pg. 2-14. The user must wire the system directly off the ribbon cable if the ICM-1100 or equivalent breakout board is not available.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Bypassing the Opto-Isolation:

If no isolation is needed, the internal 5 Volt supply may be used to power the switches, as shown in Figure 3-3. This can be done by connecting a jumper between the pins LSCOM or INCOM and 5V, labeled J9. These jumpers can be added on either the ICM-1100 or the DMC-1500. This can also be done by connecting wires between the 5V supply and common signals using the screw terminals on the ICM-1100 or AMP-11x0.

To close the circuit, wire the desired input to any ground (GND) terminal.

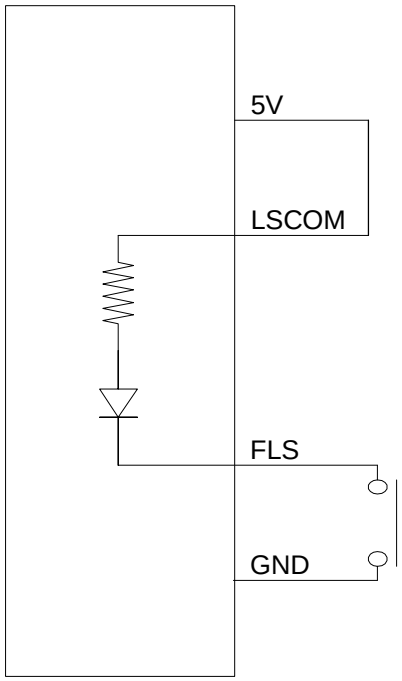


Figure 3-3 - Connecting Limit switches to the internal 5V supply

Changing Optoisolated Inputs From Active Low to Active High

Some users may prefer that the optoisolated inputs be active high. For example, the user may wish to have the inputs be activated with a logic one signal. The limit, home and latch inputs can be configured through software to be active high or low with the CN command. For more details on the CN see Command Reference manual.

The Abort input *cannot* be configured in this manner.

Amplifier Interface

The DMC-1500 analog command voltage, ACMD, ranges between +/-10V. This signal, along with GND, provides the input to the power amplifiers. The power amplifiers must be sized to drive the motors and load. For best performance, the amplifiers should be configured for a current mode of operation with no additional compensation. The gain should be set such that a 10 Volt input results in the maximum required current.

The DMC-1500 also provides an amplifier enable signal, AEN. This signal changes under the following conditions: the watchdog timer activates, the motor-off command, MO, is given, or the

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OE1command (Enable Off-On-Error) is given and the position error exceeds the error limit. As shown in Figure 3-4, AEN can be used to disable the amplifier for these conditions.

The standard configuration of the AEN signal is TTL active high. In other words, the AEN signal will be high when the controller expects the amplifier to be enabled. The polarity and the amplitude can be changed if you are using the ICM-1100 interface board. To change the polarity from active high (5 volts = enable, zero volts = disable) to active low (zero volts = enable, 5 volts= disable), replace the 7407 IC with a 7406. Note that many amplifiers designate the enable input as 'inhibit'.

To change the voltage level of the AEN signal, note the state of the resistor pack on the ICM-1100. When Pin 1 is on the 5V mark, the output voltage is 0-5V. To change to 12 volts, pull the resistor pack and rotate it so that Pin 1 is on the 12 volt side. If you remove the resistor pack, the output signal is an open collector, allowing the user to connect an external supply with voltages up to 24V.

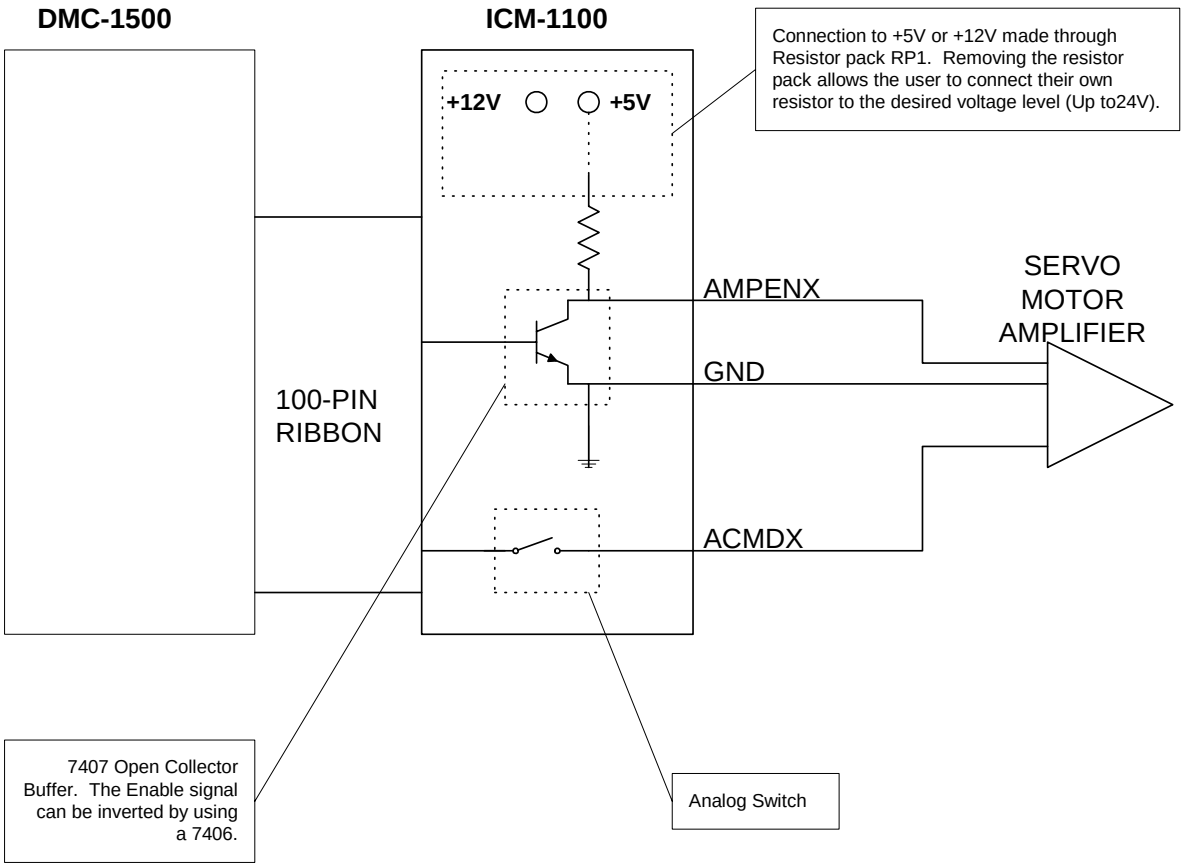


Figure 3-4 - Connecting AEN to the motor amplifier

TTL Inputs

1580

As previously mentioned, the DMC-1500 has 8 uncommitted TTL level inputs for controllers with 5 or more axes. These are specified as INx where x ranges from 17 thru 24. The reset input is also a TTL level, non-isolated signal and is used to locally reset the DMC-1500 without resetting the PC.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Analog Inputs

The DMC-1500 has seven analog inputs configured for the range between -10V and 10V. The inputs are decoded by a 12-bit A/D converter giving a voltage resolution of approximately .005V. The impedance of these inputs is 10 KΩ. The analog inputs are specified as AN[x] where x is a number 1 thru 7. Galil can supply the DMC-1500 with a 16-bit A/D converter as an option.

TTL Outputs

The DMC-1500 provides eight general use outputs and an error signal output.

The general use outputs are TTL and are accessible by connections to OUT1 thru OUT8. These outputs can be turned On and Off with the commands, SB (Set Bit), CB (Clear Bit), OB (Output Bit), and OP (Output Port). For more information about these commands, see the Command Summary. The value of the outputs can be checked with the operand _OP and the function @OUT[] (see Chapter 7, Mathematical Functions and Expressions).

1580

Controllers with 5 or more axes have an additional eight general use TTL outputs (connector JD5).

The error signal output is available on the main connector (J2, pin 3). This is a TTL signal which is low when the controller has an error. This signal is not available through the phoenix connectors of the ICM-1100.

Note: When the error signal is active, the LED on the controller will be on. An error condition indicates one of the following conditions:

1. At least one axis has a position error greater than the error limit. The error limit is set by using the command ER.
2. The reset line on the controller is held low or is being affected by noise.
3. There is a failure on the controller and the processor is resetting itself.
4. There is a failure with the output IC which drives the error signal.

Offset Adjustment

For each axis, the DMC-1500 provides offset correction potentiometers to compensate for any offset in the analog output. These potentiometers have been adjusted at the factory to produce 0 Volts output for a zero digital motor command. Before making any adjustment to the offset, send the motor off command, MO, to the DMC-1500. This causes a zero digital motor command. Connect an oscilloscope or voltmeter to the motor command pin. You should measure zero volts. If not, adjust the offset potentiometer on the DMC-1500 until zero volts is observed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 4 Communication

Introduction

The DMC-1500 has two RS232 ports. The main port is the data set and the auxiliary port is the data term. The main port can be configured through the switches on the front panel, and the auxiliary port can be configured with the software command CC. The auxiliary port can either be configured as a general port or for daisy-chain communications. The auxiliary port configuration can be saved using the Burn (BN) instruction. The RS232 ports also have a clock synchronizing line that allows synchronization of motion on more than one controller.

RS232 Ports

The RS232 pin-out description for the main and auxiliary port is given below. Note, the auxiliary port is essentially the same as the main port except inputs and outputs are reversed. The DMC-1500 may also be configured by the factory for RS422. These pin-outs are also listed below.

Note: If you are connecting the RS232 auxiliary port to a terminal or any device which is a DATATERM, it is necessary to use a connector adapter, which changes a dataterm to a dataset . This cable is also known as a 'null' modem cable.

RS232 - Main Port {P1} DATATERM

1	CTS - output	6	CTS - output
2	Transmit Data - output	7	RTS - input
3	Receive Data - input	8	CTS - output
4	RTS - input	9	No connect (Can be connected to +5V or sample clock with jumpers)
5	Ground		

RS232 - Auxiliary Port {P2} DATASET

1	CTS - input	6	CTS - input
2	Transmit Data - input	7	RTS - output
3	Receive Data - output	8	CTS - input
4	RTS - output	9	5V (Can be disconnected or connected to sample clock with jumpers)
5	Ground		

*RS422 - Main Port {P1}

1	CTS - output	6	CTS+ output
2	Transmit Data - output	7	Transmit+ output

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3 Receive Data - input	8 Receive+ input
4 RTS - input	9 RTS+ input
5 Ground	

*RS422 - Auxiliary Port {P2}

1 CTS - input	6 CTS+ input
2 Receive Data - input	7 Receive+ input
3 Transmit Data - output	8 Transmit+ output
4 RTS - output	9 RTS+ output
5 Ground	

***Default configuration is RS232. RS422 configuration available by factory.**

Configuration

Configure your PC for 8-bit data, one start-bit, one stop-bit, full duplex and no parity. The baud rate for the RS232 communication can be selected by setting the proper switch configuration on the front panel according to the table below.

Baud Rate Selection

Switch Setting			Interpretation
1200	**9600**	**19.2K**	
ON	ON	OFF	300 Baud rate
ON	OFF	OFF	1200 Baud rate
ON	OFF	ON	4800 Baud rate
OFF	ON	OFF	9600 Baud rate
OFF	OFF	ON	19200 Baud rate
OFF	ON	ON	38400 Baud
ON	ON	ON	SELF TEST

The RS232 main port can be configured for handshake or non-handshake mode. Set the HSHK switch to ON to select the handshake mode. In this mode, the RTS and CTS lines are used. The CTS line will go high whenever the DMC-1500 is not ready to receive additional characters. The RTS line will inhibit the DMC-1500 from sending additional characters. Note, the RTS line goes high for inhibit. The handshake should be turned on to ensure proper communication especially at higher baud rates.

The auxiliary port of the DMC-1500 can be configured either as a general port or for the daisy-chain. When configured as a general port, the port can be commanded to send ASCII messages to another DMC-1500 controller or to a display terminal or panel.

(Configure Communication) at port 2. The command is in the format of:

CC m,n,r,p

where m sets the baud rate, n sets for either handshake or non-handshake mode, r sets for general port or the auxiliary port, and p turns echo on or off.

m - Baud Rate - 300,1200,4800,9600,19200,38400

n - Handshake - 0=No; 1=Yes

r - Mode - 0=General Port; 1=Daisy-chain

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

p - Echo - 0=Off; 1=On; Valid only if r=0

Note, for the handshake of the auxiliary port, the roles for the RTS and CTS lines are reversed.

Example:

CC 1200,0,0,1	Configure auxiliary communication port for 1200 baud, no handshake, general port mode and echo turned on.

Daisy-Chaining

Up to eight DMC-1500 controllers may be connected in a daisy-chain allowing for multiple controllers to be commanded from a single serial port. One DMC-1500 is connected to the host terminal via the RS232 at port 1 or the main port. Port 2 or the auxiliary port of that DMC-1500 is then brought into port 1 of the next DMC-1500, and so on. The address of each DMC-1500 is configured by setting the three address jumpers (ADR4,ADR2,ADR1) located inside the box near the main processor IC.

When connecting multiple controllers in a daisy-chain, the cable between controllers should be made to female 0.89 with all wires connected straight through.

ADR1 represents the 2^0 bit, ADR2 represents 2^1 bit, and ADR4 represents 2^2 bit of the address. The eight possible addresses, 0 through 7, are set as follows:

ADR4	ADR2	ADR1	ADDRESS
OFF	OFF	OFF	0
OFF	OFF	ON	1
OFF	ON	OFF	2
OFF	ON	ON	3
ON	OFF	OFF	4
ON	OFF	ON	5
ON	ON	OFF	6
ON	ON	ON	7

To communicate with any one of the DMC-1500 units, give the command “%A”, where A is the address of the board. All instructions following this command will be sent only to the board with that address. Only when a new %A command is given will the instruction be sent to another board. The only exception is "!" command. To talk to all the DMC-1500 boards in the daisy-chain at one time, insert the character "!" before the software command. All boards receive the command, but only address 0 will echo.

Note: The CC command must be specified to configure the port P2 of each unit.

Daisy Chain Example:

Objective: Control a 7-axis motion system using two controllers, a DMC-1540 4 axis controller and a DMC-1530 3 axis controller. Address 0 is the DMC-1540 and address 1 is the DMC-1530.

Desired motion profile:

Address 0 (DMC-1540)	X Axis is 500 counts
	Y Axis is 1000 counts
	Z Axis is 2000 counts
	W Axis is 1500 counts
Address 1 (DMC-1530)	X Axis is 700 counts
	Y Axis is 1500 counts

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Z Axis is 2500 counts

Command	Interpretation
%0	Talk only to controller 0 (DMC-1540)
PR 500,1000,2000,1500	Specify X,Y,Z,W distances
%1	Talk only to controller board 1 (DMC-1530)
PR 700,1500,2500	Specify X,Y,Z distances
!BG	Begin motion on both controllers

Synchronizing Sample Clocks

It is possible to synchronize the sample clocks of all DMC-1500's in the daisy-chain. This involves burning in the command, TM-1, in all DMC-1500's except for one DMC-1500 which will be the source. It is also necessary to put a jumper on pins 7 and 9 of the JP30 and JP31 jumper blocks.

Controller Response to DATA

Most DMC-1500 instructions are represented by two characters followed by the appropriate parameters. Each instruction must be terminated by a carriage return or semicolon.

Instructions are sent in ASCII, and the DMC-1500 decodes each ASCII character (one byte) one at a time. It takes approximately .5 msec for the controller to decode each command. However, the PC can send data to the controller at a much faster rate because of the FIFO buffer.

After the instruction is decoded, the DMC-1500 returns a colon (:) if the instruction was valid or a question mark (?) if the instruction was not valid.

For instructions that return data, such as Tell Position (TP), the DMC-1500 will return the data followed by a carriage return, line feed and : .

It is good practice to check for : after each command is sent to prevent errors. An echo function is provided to enable associating the DMC-1500 response with the data sent. The echo is enabled by sending the command EO 1 to the controller.

Galil Software Tools and Libraries

API (Application Programming Interface) software is available from Galil. The API software is written in C and is included in the Galil COMM disks. They can be used for development under DOS and Windows environments (16 and 32 bit Windows). With the API's, the user can incorporate already existing library functions directly into a C program.

Galil has also developed a Visual Basic Toolkit. This provides VBXs, 16-bit OCXs and 32-bit OCXs for handling all of the DMC-1500 communications including support of interrupts. These objects install directly into Visual Basic and are part of the run-time environment. For more information, contact Galil.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 5 Command Basics

Introduction

The DMC-1500 provides over 100 commands for specifying motion and machine parameters. Commands are included to initiate action, interrogate status and configure the digital filter.

The DMC-1500 instruction set is BASIC-like and easy to use. Instructions consist of two uppercase letters that correspond phonetically with the appropriate function. For example, the instruction BG begins motion, and ST stops the motion.

Commands can be sent "live" over the bus for immediate execution by the DMC-1500, or an entire group of commands can be downloaded into the DMC-1500 memory for execution at a later time. Combining commands into groups for later execution is referred to as Applications Programming and is discussed in the following chapter.

This section describes the DMC-1500 instruction set and syntax. A summary of commands as well as a complete listing of all DMC-1500 instructions is included in the *Command Reference* Manual.

Command Syntax

DMC-1500 instructions are represented by two ASCII upper case characters followed by applicable arguments. A space may be inserted between the instruction and arguments. A semicolon or <enter> is used to terminate the instruction for processing by the DMC-1500 command interpreter. Note: If you are using a Galil terminal program, commands will not be processed until an <enter> command is given. This allows the user to separate many commands on a single line and not begin execution until the user gives the <enter> command.

IMPORTANT: All DMC-1500 commands must be sent in upper case.

For example, the command

PR 4000 <enter> Position relative

PR is the two character instruction for position relative. 4000 is the argument which represents the required position value in counts. The <enter> terminates the instruction. The space between PR and 4000 is optional.

For specifying data for the X,Y,Z and W axes, commas are used to separate the axes. If no data is specified for an axis, a comma is still needed as shown in the examples below. If no data is specified for an axis, the previous value is maintained. The space between the data and instruction is optional. For controllers with 5 or more axes, the axes are referred to as A,B,C,D,E,F,G,H where X,Y,Z,W and A,B,C,D may be used interchangeably.

The DMC-1500 provides an alternative method for specifying data. Here data is specified individually using a single axis specifier such as X,Y,Z or W (or A,B,C,D,E,F,G or H for the DMC-1580). An equals sign is used to assign data to that axis. For example:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PRX=1000	Specify a position relative movement for the X axis of 1000
ACY=200000	Specify acceleration for the Y axis as 200000

Instead of data, some commands request action to occur on an axis or group of axes. For example, ST XY stops motion on both the X and Y axes. Commas are not required in this case since the particular axis is specified by the appropriate letter X Y Z or W. If no parameters follow the instruction, action will take place on all axes. Here are some examples of syntax for requesting action:

BG X	Begin X only
BG Y	Begin Y only
BG XYZW	Begin all axes
BG YW	Begin Y and W only
BG	Begin all axes

1580

For controllers with 5 or more axes, the axes are referred to as A,B,C,D,E,F,G,H. The specifiers X,Y,Z,W and A,B,C,D may be used interchangeably:

BG ABCDEFGH	Begin all axes
BG D	Begin D only

Coordinated Motion with more than 1 axis

When requesting action for coordinated motion, the letter S is used to specify the coordinated motion. For example:

BG S	Begin coordinated sequence
BG SW	Begin coordinated sequence and W axis

Program Syntax

Chapter 7 explains the how to write and execute motion control programs.

Controller Response to DATA

The DMC-1500 returns a : for valid commands.

The DMC-1500 returns a ? for invalid commands.

For example, if the command BG is sent in lower case, the DMC-1500 will return a ?.

:bg <enter>	invalid command, lower case
?	DMC-1500 returns a ?

When the controller receives an invalid command the user can request the error code. The error code will specify the reason for the invalid command response. To request the error code type the command: TC1 For example:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

?TC1 <enter>	Tell Code command
1 Unrecognized command	Returned response

There are many reasons for receiving an invalid command response. The most common reasons are: unrecognized command (such as typographical entry or lower case), command given at improper time (such as during motion), or a command out of range (such as exceeding maximum speed). A complete list of all error codes can be found with the description of the TC command in the Command Reference Manual.

Interrogating the Controller

Interrogation Commands

The DMC-1500 has a set of commands that directly interrogate the controller. When the command is entered, the requested data is returned in decimal format on the next line followed by a carriage return and line feed. The format of the returned data can be changed using the Position Format (PF), Variable Format (VF) and Leading Zeros (LZ) command. See Chapter 7 and the Command Reference Manual.

Summary of Interrogation Commands

RP	Report Command Position
RL	Report Latch
^R ^V	Firmware Revision Information
SC	Stop Code
TB	Tell Status
TC	Tell Error Code
TD	Tell Dual Encoder
TE	Tell Error
TI	Tell Input
TP	Tell Position
TR	Trace
TS	Tell Switches
TT	Tell Torque
TV	Tell Velocity

For example, the following example illustrates how to display the current position of the X axis:

TP X <enter>	Tell position X
0000000000	Controllers Response
TP XY <enter>	Tell position X and Y
0000000000,0000000000	Controllers Response

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Additional Interrogation Methods.

Most commands can be interrogated by using a question mark. For information specific to a particular axis, type the command followed by a ? for each axis requested.

PR ?,?,?,?	Request X,Y,Z,W values
PR ,?	Request Y value only

The controller can also be interrogated with operands.

Operands

Most DMC-1500 commands have corresponding operands that can be used for interrogation. Operands must be used inside of valid DMC expressions. For example, to display the value of an operand, the user could use the command:

MG ‘operand’ where 'operand' is a valid DMC operand

All of the command operands begin with the underscore character (_). For example, the value of the current position on the X axis can be assigned to the variable, V, with the command:

V=_TPX

The Command Reference denotes all commands which have an equivalent operand as "Used as an Operand". For further information, see description of operands in Chapter 7.

Command Summary

For a complete command summary, see *Command Reference* manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 6 Programming Motion

Overview

The DMC-1500 can be commanded to do the following modes of motion: Absolute and relative independent positioning, jogging, linear interpolation (up to 8 axes), linear and circular interpolation (2 axes with 3rd axis of tangent motion), electronic gearing, electronic cam motion and contouring. These modes are discussed in the following sections.

The DMC-1510 is a single axis controller and uses X-axis motion only. Likewise, the DMC-1520 uses X and Y, the DMC-1530 uses X,Y and Z, and the DMC-1540 uses X,Y,Z and W. The DMC-1550 uses A,B,C,D, and E. The DMC-1560 uses A,B,C,D,E, and F. The DMC-1570 uses A,B,C,D,E,F and G. The DMC-1580 uses the axes A,B,C,D,E,F,G, and H.

The example applications described below will help guide you to the appropriate mode of motion.

1580

For controllers with 5 or more axes, the specifiers, ABCDEFGH, are used. XYZ and W may be interchanged with ABCD.

Independent Axis Positioning

In this mode, motion between the specified axes is independent, and each axis follows its own profile. The user specifies the desired absolute position (PA) or relative position (PR), slew speed (SP), acceleration ramp (AC), and deceleration ramp (DC), for each axis. On begin (BG), the DMC-1500 profiler generates the corresponding trapezoidal or triangular velocity profile and position trajectory. The controller determines a new command position along the trajectory every sample period until the specified profile is complete. Motion is complete when the last position command is sent by the DMC-1500 profiler. Note: The actual motor motion may not be complete when the profile has been completed, however, the next motion command may be specified.

The Begin (BG) command can be issued for all axes either simultaneously or independently. XYZ or W axis specifiers are required to select the axes for motion. When no axes are specified, this causes motion to begin on all axes.

The speed (SP) and the acceleration (AC) can be changed at any time during motion, however, the deceleration (DC) and position (PR or PA) cannot be changed until motion is complete. Remember, motion is complete when the profiler is finished, not when the actual motor is in position. The Stop command (ST) can be issued at any time to decelerate the motor to a stop before it reaches its final position.

An incremental position movement (IP) may be specified during motion as long as the additional move is in the same direction. Here, the user specifies the desired position increment, n. The new target is equal to the old target plus the increment, n. Upon receiving the IP command, a revised profile will be generated for motion towards the new end position. The IP command does not require a begin. Note: If the motor is not moving, the IP command is equivalent to the PR and BG command combination.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command Summary - Independent Axis

COMMAND	DESCRIPTION
PR X,Y,Z,W	Specifies relative distance
PA x,y,z,w	Specifies absolute position
SP x,y,z,w	Specifies slew speed
AC x,y,z,w	Specifies acceleration rate
DC x,y,z,w	Specifies deceleration rate
BG XYZW	Starts motion
ST XYZW	Stops motion before end of move
IP x,y,z,w	Changes position target
IT x,y,z,w	Time constant for independent motion smoothing
AM XYZW	Trippoint for profiler complete
MC XYZW	Trippoint for "in position"

The lower case specifiers (x,y,z,w) represent position values for each axis. For controllers with more than 4 axes, the position values would be represented as a,b,c,d,e,f,g,h.

Operand Summary - Independent Axis

OPERAND	DESCRIPTION
_ACx	Return acceleration rate for the axis specified by ‘x’
_DCx	Return deceleration rate for the axis specified by ‘x’
_SPx	Returns the speed for the axis specified by ‘x’
_PAx	Returns current destination if ‘x’ axis is moving, otherwise returns the current commanded position if in a move.
_PRx	Returns current incremental distance specified for the ‘x’ axis

Example - Absolute Position Movement

PA 10000,20000	Specify absolute X,Y position
AC 1000000,1000000	Acceleration for X,Y
DC 1000000,1000000	Deceleration for X,Y
SP 50000,30000	Speeds for X,Y
BG XY	Begin motion

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example - Multiple Move Sequence

Required Motion Profiles:

X-Axis	500 counts	Position
	10000 count/sec	Speed
	500000 counts/sec^2	Acceleration
Y-Axis	1000 counts	Position
	15000 count/sec	Speed
	500000 counts/sec^2	Acceleration
Z-Axis	100 counts	Position
	5000 counts/sec	Speed
	500000 counts/sec	Acceleration

This example will specify a relative position movement on X, Y and Z axes. The movement on each axis will be separated by 20 msec. Fig. 6.1 shows the velocity profiles for the X,Y and Z axis.

#A	Begin Program
PR 2000,500,100	Specify relative position movement of 1000, 500 and 100 counts for X,Y and Z axes.
SP 15000,10000,5000	Specify speed of 10000, 15000, and 5000 counts / sec
AC 500000,500000,500000	Specify acceleration of 500000 counts / sec^2 for all axes
DC 500000,500000,500000	Specify deceleration of 500000 counts / sec^2 for all axes
BG X	Begin motion on the X axis
WT 20	Wait 20 msec
BG Y	Begin motion on the Y axis
WT 20	Wait 20 msec
BG Z	Begin motion on Z axis
EN	End Program

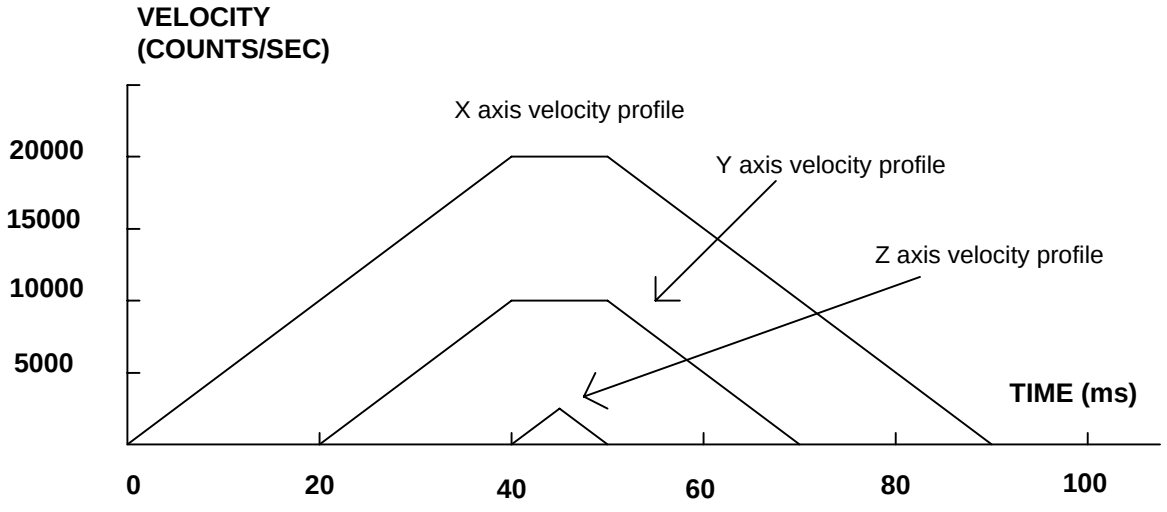


Figure 6.1 - Velocity Profiles of XYZ

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Notes on fig 6.1: The X and Y axis have a 'trapezoidal' velocity profile, while the Z axis has a 'triangular' velocity profile. The X and Y axes accelerate to the specified speed, move at this constant speed, and then decelerate such that the final position agrees with the command position, PR. The Z axis accelerates, but before the specified speed is achieved, must begin deceleration such that the axis will stop at the commanded position. All 3 axes have the same acceleration and deceleration rate, hence, the slope of the rising and falling edges of all 3 velocity profiles are the same.

Independent Jogging

The jog mode of motion allows the user to change speed, direction and acceleration during motion. The user specifies the jog speed (JG), acceleration (AC), and the deceleration (DC) rate for each axis. The direction of motion is specified by the sign of the JG parameters. When the begin command is given (BG), the motor accelerates up to speed and continues to jog at that speed until a new speed or stop (ST) command is issued. If the jog speed is changed during motion, the controller will make a accelerated (or decelerated) change to the new speed.

An instant change to the motor position can be made with the use of the IP command. Upon receiving this command, the controller commands the motor to a position which is equal to the specified increment plus the current position. This command is useful when trying to synchronize the position of two motors while they are moving.

Note that the controller operates as a closed-loop position controller while in the jog mode. The DMC-1500 converts the velocity profile into a position trajectory and a new position target is generated every sample period. This method of control results in precise speed regulation with phase lock accuracy.

Command Summary - Jogging

COMMAND	DESCRIPTION
AC x,y,z,w	Specifies acceleration rate
BG X,Y,Z,W	Begins motion
DC x,y,z,w	Specifies deceleration rate
IP x,y,z,w	Increments position instantly
IT x,y,z,w	Time constant for independent motion smoothing
JG +/-x,y,z,w	Specifies jog speed and direction
ST XYZW	Stops motion

Parameters can be set with individual axes specifiers such as JGY+2000(set jog speed for X axis to 2000) or ACYH=40000 (set acceleration for Y and H axes to 400000) .

Operand Summary - Independent Axis

OPERAND	DESCRIPTION
_ACx	Return acceleration rate for the axis specified by 'x'
_DCx	Return deceleration rate for the axis specified by 'x'
_SPx	Returns the jog speed for the axis specified by 'x'
_TVx	Returns the actual velocity of the axis specified by 'x' (averaged over.25 sec)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example - Jog in X Only

Jog X motor at 50000count/s. After X motor is at its jog speed, begin jogging Z in reverse direction at 25000 count/s.

```
#A
AC 20000,,20000          Specify X,Z acceleration of 20000 cts/sec
DC 20000,,20000          Specify X,Z deceleration of 20000 cts/sec
JG 50000,,-25000         Specify jog speed and direction for X and Z axis
BG XY                    Begin X motion
AS X                     Wait until X is at speed
BG Z                     Begin Z motion
EN
```

Example - Joystick Jogging

The jog speed can also be changed using an analog input such as a joystick. Assume that for a 10 Volt input the speed must be 50000 counts/sec.

```
#JOY                     Label
JG0                      Set in Jog Mode
BGX                      Begin motion
#B                       Label for Loop
V1 = @AN[1]              Read analog input
VEL = V1*50000/2047      Compute speed
JG VEL                   Change JG speed
JP #B                    Loop
```

Linear Interpolation Mode

The DMC-1500 provides a linear interpolation mode for 2 or more axes (up to 8 axes for the DMC-1580). In linear interpolation mode, motion between the axes is coordinated to maintain the prescribed vector speed, acceleration, and deceleration along the specified path. The motion path is described in terms of incremental distances for each axis. An unlimited number of incremental segments may be given in a continuous move sequence, making the linear interpolation mode ideal for following a piece-wise linear path. There is no limit to the total move length.

The LM command selects the Linear Interpolation mode and axes for interpolation. For example, LM YZ selects only the Y and Z axes for linear interpolation.

When using the linear interpolation mode, the LM command only needs to be specified once unless the axes for linear interpolation change.

Specifying Linear Segments

The command LI x,y,z,w or LI a,b,c,d,e,f,g,h specifies the incremental move distance for each axis. This means motion is prescribed with respect to the current axis position. Up to 511 incremental move segments may be given prior to the Begin Sequence (BGS) command. Once motion has begun, additional LI segments may be sent to the controller.

The clear sequence (CS) command can be used to remove LI segments stored in the buffer prior to the start of the motion. To stop the motion, use the instructions STS or AB. The command, ST, causes a decelerated stop. The command, AB, causes an instantaneous stop and aborts the program, and the command AB1 aborts the motion only.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The Linear End (LE) command must be used to specify the end of a linear move sequence. This command tells the controller to decelerate to a stop following the last LI command. If an LE command is not given, an Abort AB1 must be used to abort the motion sequence.

It is the responsibility of the user to keep enough LI segments in the DMC-1500 sequence buffer to ensure continuous motion. If the controller receives no additional LI segments and no LE command, the controller will stop motion instantly at the last vector. There will be no controlled deceleration. LM? or _LM returns the available spaces for LI segments that can be sent to the buffer. 511 returned means the buffer is empty and 511 LI segments can be sent. A zero means the buffer is full and no additional segments can be sent. As long as the buffer is not full, additional LI segments can be sent at PC bus speeds.

The instruction _CS returns the segment counter. As the segments are processed, _CS increases, starting at zero. This function allows the host computer to determine which segment is being processed.

Specifying Vector Acceleration, Deceleration and Speed:

The commands VS n, VA n, and VD n are used to specify the vector speed, acceleration and deceleration. The DMC-1500 computes the vector speed based on the axes specified in the LM mode. For example, LM XYZ designates linear interpolation for the X,Y and Z axes. The vector speed for this example would be computed using the equation:

$VS^2=XS^2+YS^2+ZS^2$, where XS, YS and ZS are the speed of the X,Y and Z axes.

The controller always uses the axis specifications from LM, not LI, to compute the speed.

In cases where the acceleration causes the system to 'jerk', the DMC-1500 provides a vector motion smoothing function. VT is used to set the S-curve smoothing constant for coordinated moves.

Additional Commands

The DMC-1500 provides commands for additional control of vector motion and program control. Note: Many of the commands used in Linear Interpolation motion also applies Vector motion described in the next section.

Trippoints

The command AV n is the ‘After Vector’ trippoint, which halts program execution until the vector distance of n has been reached.

In this example, the XY system is required to perform a 90° turn. In order to slow the speed around the corner, we use the AV 4000 trippoint, which slows the speed to 1000 count/s. Once the motors reach the corner, the speed is increased back to 4000 cts / s.

Instruction	Interpretation
#LMOVE	Label
DP ,,0,0	Define position of Z and W axes to be 0
LMXY	Define linear mode between X and Y axes.
LI 5000,0	Specify first linear segment
LI 0,5000	Specify second linear segment
LE	End linear segments
VS 4000	Specify vector speed
BGS	Begin motion sequence

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AV 4000	Set trippoint to wait until vector distance of 4000 is reached
VS 1000	Change vector speed
AV 5000	Set trippoint to wait until vector distance of 5000 is reached
VS 4000	Change vector speed
EN	Program end

Specifying Vector Speed for Each Segment

The instruction VS has an immediate effect and, therefore, must be given at the required time. In some applications, such as CNC, it is necessary to attach various speeds to different motion segments. This can be done by the instruction

LI x,y,z,w < n

This instruction attaches the vector speed, n, to the motion segment LI. As a consequence, the program #LMOVE can be written in the alternative form:

Instruction	Interpretation
#ALT	Label for alternative program
DP 0,0	Define Position of X and Y axis to be 0
LMXY	Define linear mode between X and Y axes.
LI 4000,0 <4000	Specify first linear segment with a vector speed of 4000
LI 1000,0 < 1000	Specify second linear segment with a vector speed of 1000
LI 0,5000 < 4000	Specify third linear segment with a vector speed of 4000
LE	End linear segments
BGS	Begin motion sequence
EN	Program end

Changing Feedrate:

The command VR n allows the feedrate, VS, to be scaled between 0 and 10 with a resolution of .0001. This command takes effect immediately and causes VS to be scaled. VR also applies when the vector speed is specified with the '<' operator. This is a useful feature for feedrate override. VR does not ratio the accelerations. For example, VR .5 results in the specification VS 2000 to be divided in half.

Command Summary - Linear Interpolation

COMMAND	DESCRIPTION
LM xyzw LM abcdefgh	Specify axes for linear interpolation (same) controllers with 5 or more axes
LM?	Returns number of available spaces for linear segments in DMC-1500 sequence buffer. Zero means buffer full. 512 means buffer empty.
LI x,y,z,w < n LI a,b,c,d,e,f,g,h < n	Specify incremental distances relative to current position, and assign vector speed n.
VS n	Specify vector speed
VA n	Specify vector acceleration
VD n	Specify vector deceleration
VR n	Specify the vector speed ratio
BGS	Begin Linear Sequence
CS	Clear sequence

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LE	Linear End- Required at end of LI command sequence
LE?	Returns the length of the vector (resets after 2147483647)
AMS	Trippoint for After Sequence complete
AV n	Trippoint for After Relative Vector Distance, n
VT	S curve smoothing constant for vector moves

Operand Summary - Linear Interpolation

OPERAND	DESCRIPTION
_AV	Return distance traveled
_CS	Segment counter - returns number of the segment in the sequence, starting at zero.
_LE	Returns length of vector (resets after 2147483647)
_LM	Returns number of available spaces for linear segments in DMC-1500 sequence buffer. Zero means buffer full. 512 means buffer empty.
_VPm	Return the absolute coordinate of the last data point along the trajectory. (m=X,Y,Z or W or A,B,C,D,E,F,G or H)

To illustrate the ability to interrogate the motion status, consider the first motion segment of our example, #LMOVE, where the X axis moves toward the point X=5000. Suppose that when X=3000, the controller is interrogated using the command 'MG _AV'. The returned value will be 3000. The value of _CS, _VPX and _VPY will be zero.

Now suppose that the interrogation is repeated at the second segment when Y=2000. The value of _AV at this point is 7000, _CS equals 1, _VPX=5000 and _VPY=0.

Example - Linear Move

Make a coordinated linear move in the ZW plane. Move to coordinates 40000,30000 counts at a vector speed of 100000 counts/sec and vector acceleration of 1000000 counts/sec^2.

Instruction	Interpretation
#TEST	Label
LM ZW	Specify axes for linear interpolation
LI,,40000,30000	Specify ZW distances
LE	Specify end move
VS 100000	Specify vector speed
VA 1000000	Specify vector acceleration
VD 1000000	Specify vector deceleration
BGS	Begin sequence
AMS	After motion sequence ends
EN	End program

Note that the above program specifies the vector speed, VS, and not the actual axis speeds VZ and VW. The axis speeds are determined by the DMC-1500 from:

$$VS = \sqrt{VZ^2 + VW^2}$$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The resulting profile is shown in Figure 6.2.

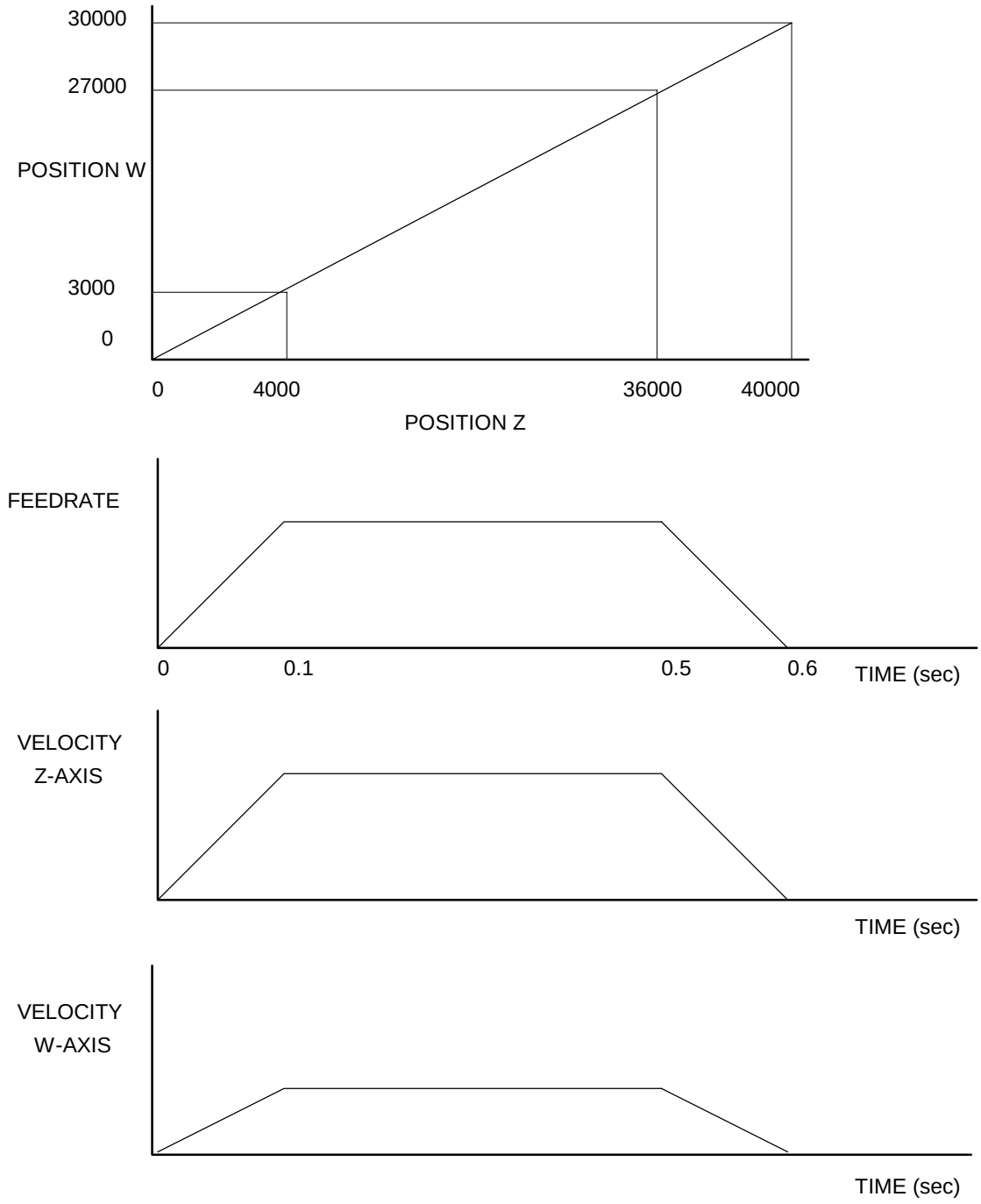


Figure 6.2 - Linear Interpolation

Example - Multiple Moves

This example makes a coordinated linear move in the XY plane. The Arrays VX and VY are used to store 750 incremental distances which are filled by the program #LOAD.

Instruction **Interpretation**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

#LOAD	Load Program
DM VX [750],VY [750]	Define Array
COUNT=10	Initialize Counter
N=10	Initialize position increment
#LOOP	LOOP
VX [COUNT]=N	Fill Array VX
VY [COUNT]=N	Fill Array VY
N=N+10	Increment position
COUNT=COUNT+1	Increment counter
JP #LOOP,COUNT<750	Loop if array not full
#A	Label
LM XY	Specify linear mode for XY
COUNT=0	Initialize array counter
#LOOP2;JP#LOOP2,_LM=0	If sequence buffer full, wait
JS#C,COUNT=500	Begin motion on 500th segment
LI VX[COUNT],VY[COUNT]	Specify linear segment
COUNT=COUNT+1	Increment array counter
JP #LOOP2,COUNT<750	Repeat until array done
LE	End Linear Move
AMS	After Move sequence done
MG "DONE"	Send Message
EN	End program
#C;BGS;EN	Begin Motion Subroutine

Vector Mode: Linear and Circular Interpolation Motion

The DMC-1500 allows a long 2-D path consisting of linear and arc segments to be prescribed. Motion along the path is continuous at the prescribed vector speed even at transitions between linear and circular segments. The DMC-1500 performs all the complex computations of linear and circular interpolation, freeing the host PC from this time intensive task.

The coordinated motion mode is similar to the linear interpolation mode. Any pair of two axes may be selected for coordinated motion consisting of linear and circular segments. In addition, a third axis can be controlled such that it remains tangent to the motion of the selected pair of axes. Note that only one pair of axes can be specified for coordinated motion at any given time.

The command VM m,n,p where ‘m’ and ‘n’ are the coordinated pair and p is the tangent axis (Note: the commas which separate m,n and p are not necessary). For example, VM XWZ selects the XW axes for coordinated motion and the Z-axis as the tangent.

Specifying Vector Segments

The motion segments are described by two commands; VP for linear segments and CR for circular segments. Once a set of linear segments and/or circular segments have been specified, the sequence is ended with the command VE. This defines a sequence of commands for coordinated motion. Immediately prior to the execution of the first coordinated movement, the controller defines the current position to be zero for all movements in a sequence. Note: This ‘local’ definition of zero does not affect the absolute coordinate system or subsequent coordinated motion sequences.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The command, VP xy specifies the coordinates of the end points of the vector movement with respect to the starting point. Non-sequential axes do not require comma delineation. The command, CR r,q,d define a circular arc with a radius r, starting angle of q, and a traversed angle d. The notation for q is that zero corresponds to the positive horizontal direction, and for both q and d, the counter-clockwise (CCW) rotation is positive.

Up to 511 segments of CR or VP may be specified in a single sequence and must be ended with the command VE. The motion can be initiated with a Begin Sequence (BGS) command. Once motion starts, additional segments may be added.

The Clear Sequence (CS) command can be used to remove previous VP and CR commands which were stored in the buffer prior to the start of the motion. To stop the motion, use the instructions STS or AB1. ST stops motion at the specified deceleration. AB1 aborts the motion instantaneously.

The Vector End (VE) command must be used to specify the end of the coordinated motion. This command requires the controller to decelerate to a stop following the last motion requirement. If a VE command is not given, an Abort (AB1) must be used to abort the coordinated motion sequence.

It is the responsibility of the user to keep enough motion segments in the DMC-1500 sequence buffer to ensure continuous motion. If the controller receives no additional motion segments and no VE command, the controller will stop motion instantly at the last vector. There will be no controlled deceleration. LM? or _LM returns the available spaces for motion segments that can be sent to the buffer. 511 returned means the buffer is empty and 511 segments can be sent. A zero means the buffer is full and no additional segments can be sent. As long as the buffer is not full, additional segments can be sent at PC bus speeds.

The operand _CS can be used to determine the value of the segment counter.

Specifying Vector Acceleration, Deceleration and Speed:

The commands VS n, VA n, and VD n are used to specify the vector speed, acceleration and deceleration. The DMC-1500 computes the vector speed based on the two axes specified in the VM mode. For example, VM YZ designates vector mode for the Y and Z axes. The vector speed for this example would be computed using the equation:

$VS^2=YS^2+ZS^2$, where YS and ZS are the speed of the Y and Z axes.

In cases where the acceleration causes the system to 'jerk', the DMC-1500 provides a vector motion smoothing function. VT is used to set the S-curve smoothing constant for coordinated moves.

Additional Commands

The DMC-1500 provides commands for additional control of vector motion and program control. Note: Many of the commands used in Vector Mode motion also applies Linear Interpolation motion described in the previous section.

Trippoints

The command AV n is the ‘After Vector’ trippoint, which halts program execution until the vector distance of n has been reached.

Specifying Vector Speed for Each Segment

The vector speed may be specified by the immediate command VS. It can also be attached to a motion segment with the instructions

VP x,y, < n

CR r,θ,δ < n

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Both cases assign a vector speed of n count/s to the corresponding motion segment.

Changing Feedrate:

The command VR n allows the feedrate, VS, to be scaled between 0 and 10 with a resolution of .0001. This command takes effect immediately and causes VS scaled. VR also applies when the vector speed is specified with the '<' operator. This is a useful feature for feedrate override. VR does not ratio the accelerations. For example, VR .5 results in the specification VS 2000 to be divided in half.

Compensating for Differences in Encoder Resolution:

By default, the DMC-1500 uses a scale factor of 1:1 for the encoder resolution when used in vector mode. If this is not the case, the command, ES can be used to scale the encoder counts. The ES command accepts two arguments which represent the number of counts for the two encoders used for vector motion. The smaller ratio of the two numbers will be multiplied by the higher resolution encoder. For more information, see ES command in Chapter 11, Command Summary.

Tangent Motion:

Several applications, such as cutting, require a third axis (i.e. a knife blade), to remain tangent to the coordinated motion path. To handle these applications, the DMC-1500 allows one axis to be specified as the tangent axis. The VM command provides parameter specifications for describing the coordinated axes and the tangent axis.

VM m,n,p	m,n specifies coordinated axes p specifies tangent axis such as X,Y,Z,W or A,B,C,D,E,F,G,H p=N turns off tangent axis

Before the tangent mode can operate, it is necessary to assign an axis via the VM command and define its offset and scale factor via the TN m,n command. m defines the scale factor in counts/degree and n defines the tangent position that equals zero degrees in the coordinated motion plane. The _TN can be used to return the initial position of the tangent axis.

Example - XY Table Control

Assume an XY table with the Z-axis controlling a knife. The Z-axis has a 2000 quad counts/rev encoder and has been initialized after power-up to point the knife in the +Y direction. A 180° circular cut is desired, with a radius of 3000, center at the origin and a starting point at (3000,0). The motion is CCW, ending at (-3000,0). Note that the 0° position in the XY plane is in the +X direction. This corresponds to the position -500 in the Z-axis, and defines the offset. The motion has two parts. First, X,Y and Z are driven to the starting point, and later, the cut is performed. Assume that the knife is engaged with output bit 0.

Instruction	Interpretation
#EXAMPLE	Example program
VM XYZ	XY coordinate with Z as tangent
TN 2000/360,-500	2000/360 counts/degree, position -500 is 0 degrees in XY plane
CR 3000,0,180	3000 count radius, start at 0 and go to 180 CCW
VE	End vector
CB0	Disengage knife
PA 3000,0,_TN	Move X and Y to starting position, move Z to initial tangent position
BG XYZ	Start the move to get into position

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AM XYZ	When the move is complete
SB0	Engage knife
WT50	Wait 50 msec for the knife to engage
BGS	Do the circular cut
AMS	After the coordinated move is complete
CB0	Disengage knife
MG "ALL DONE"	
EN	End program

Command Summary - Vector Mode Motion

COMMAND	DESCRIPTION
VM m,n	Specifies the axes for the planar motion where m and n represent the planar axes and p is the tangent axis.
VP m,n	Return coordinate of last point, where m=X,Y,Z or W.
CR r,Θ, ±ΔΘ	Specifies arc segment where r is the radius, Θ is the starting angle and ΔΘ is the travel angle. Positive direction is CCW.
VS n	Specify vector speed or feedrate of sequence.
VA n	Specify vector acceleration along the sequence.
VD n	Specify vector deceleration along the sequence.
VR n	Specify vector speed ratio
BGS	Begin motion sequence.
CS	Clear sequence.
AV n	Trippoint for After Relative Vector distance, n.
AMS	Holds execution of next command until Motion Sequence is complete.
TN m,n	Tangent scale and offset.
ES m,n	Ellipse scale factor.
VT	S curve smoothing constant for coordinated moves
LM?	Return number of available spaces for linear and circular segments in DMC-1500 sequence buffer. Zero means buffer is full. 512 means buffer is empty.

Operand Summary - Vector Mode Motion

OPERAND	DESCRIPTION
_VPM	The absolute coordinate of the axes at the last intersection along the sequence.
_AV	Distance traveled.
_LM	Number of available spaces for linear and circular segments in DMC-1500 sequence buffer. Zero means buffer is full. 512 means buffer is empty.
_CS	Segment counter - Number of the segment in the sequence, starting at zero.
_VE	Vector length of coordinated move sequence.

When AV is used as an operand, _AV returns the distance traveled along the sequence.

The operands _VPX and _VPY can be used to return the coordinates of the last point specified along the path.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example:

Traverse the path shown in Fig. 6.3. Feedrate is 20000 counts/sec. Plane of motion is XY

Instruction	Interpretation
VM XY	Specify motion plane
VS 20000	Specify vector speed
VA 1000000	Specify vector acceleration
VD 1000000	Specify vector deceleration
VP -4000,0	Segment AB
CR 1500,270,-180	Segment BC
VP 0,3000	Segment CD
CR 1500,90,-180	Segment DA
VE	End of sequence
BGS	Begin Sequence

The resulting motion starts at the point A and moves toward points B, C, D, A. Suppose that we interrogate the controller when the motion is halfway between the points A and B.

The value of _AV is 2000

The value of _CS is 0

_VPX and _VPY contain the absolute coordinate of the point A

Suppose that the interrogation is repeated at a point, halfway between the points C and D.

The value of _AV is 4000+1500π+2000=10,712

The value of _CS is 2

_VPX,_VPY contain the coordinates of the point C

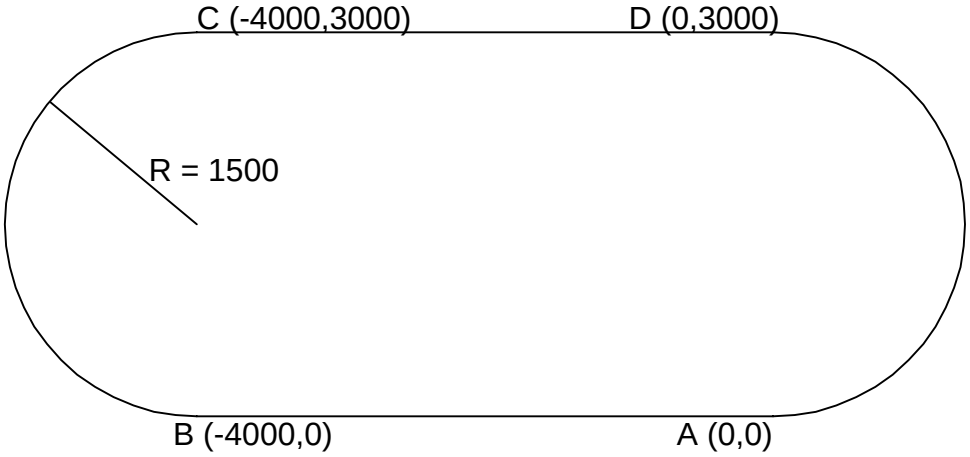


Figure 6.3 - The Required Path

Electronic Gearing

This mode allows up to 8 axes to be electronically geared to one master axis. The master may rotate in both directions and the geared axes will follow at the specified gear ratio. The gear ratio may be different for each axis and changed during motion.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The command GAX or GAY or GAZ or GAW (or GAA or GAB or GAC or GAD or GAE or GAF or GAG or GAH for DMC-1580) specifies the master axis. There may only be one master. GR x,y,z,w specifies the gear ratios for the slaves where the ratio may be a number between +/-127.9999 with a fractional resolution of .0001. GR 0,0,0,0 turns off electronic gearing for any set of axes. A limit switch will also disable electronic gearing for that axis. GR causes the specified axes to be geared to the actual position of the master. The master axis is commanded with motion commands such as PR, PA or JG.

When the master axis is driven by the controller in the jog mode or an independent motion mode, it is possible to define the master as the command position of that axis, rather than the actual position. The designation of the commanded position master is by the letter, C. For example, GACX indicates that the gearing is the commanded position of X.

An alternative gearing method is to synchronize the slave motor to the commanded vector motion of several axes performed by GAS. For example, if the X and Y motor form a circular motion, the Z axis may move in proportion to the vector move. Similarly, if X,Y and Z perform a linear interpolation move, W can be geared to the vector move.

Electronic gearing allows the geared motor to perform a second independent or coordinated move in addition to the gearing. For example, when a geared motor follows a master at a ratio of 1:1, it may be advanced an additional distance with PR, or JG, commands, or VP, or LI.

Command Summary - Electronic Gearing

COMMAND	DESCRIPTION
GA n	Specifies master axis for gearing where: n = X,Y,Z or W or A,B,C,D,E,F,G,H for main encoder as master
	n = XC,YC,ZC or WC or AC, BC, CC, DC, EC, FC,GC,HC for commanded position.
	n = DX,DY,DZ or DW or DA, DB, DC, DD, DE, DF,DG,DH for auxiliary encoders
	n = S vector move as master
GR x,y,z,w	Sets gear ratio for slave axes. 0 disables electronic gearing for specified axis.
GR a,b,c,d,e,f,g,h	Sets gear ratio for slave axes. 0 disables electronic gearing for specified axis.
MR x,y,z,w	Trippoint for reverse motion past specified value. Only one field may be used.
MF x,y,z,w	Trippoint for forward motion past specified value. Only one field may be used.

Operand Summary - Electronic Gearing

COMMAND	DESCRIPTION
GA n	Specifies master axis for gearing where: n = X,Y,Z or W or A,B,C,D,E,F,G,H for main encoder as master
	n = XC,YC,ZC or WC or AC, BC, CC, DC, EC, FC,GC,HC for commanded position.
	n = DX,DY,DZ or DW or DA, DB, DC, DD, DE, DF,DG,DH for auxiliary encoders
	n = S vector move as master
GR x,y,z,w	Sets gear ratio for slave axes. 0 disables electronic gearing for specified axis.
GR a,b,c,d,e,f,g,h	Sets gear ratio for slave axes. 0 disables electronic gearing for specified axis.
MR x,y,z,w	Trippoint for reverse motion past specified value. Only one field may be used.
MF x,y,z,w	Trippoint for forward motion past specified value. Only one field may be used.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example - Simple Master Slave

Master axis moves 10000 counts at slew speed of 100000 counts/sec. Y is defined as the master. X,Z,W are geared to master at ratios of 5,-.5 and 10 respectively.

Instruction	Interpretation
GAY	Specify master axes as Y
GR 5,,-.5,10	Set gear ratios
PR ,10000	Specify Y position
SP ,100000	Specify Y speed
BGY	Begin motion

Example - Electronic Gearing

Objective: Run two geared motors at speeds of 1.132 and -0.045 times the speed of an external master. The master is driven at speeds between 0 and 1800 RPM (2000 counts/rev encoder).

Solution: Use a DMC-1530 controller, where the Z-axis is the master and X and Y are the geared axes.

MO Z	Turn Z off, for external master
GA Z	Specify master axis
GR 1.132,-.045	Specify gear ratios

Now suppose the gear ratio of the X-axis is to change on-the-fly to 2. This can be achieved by commanding:

GR 2	Specify gear ratio for X axis to be 2

In applications where both the master and the follower are controlled by the DMC-1500 controller, it may be desired to synchronize the follower with the commanded position of the master, rather than the actual position. This eliminates the coupling between the axes which may lead to oscillations.

For example, assume that a gantry is driven by two axes, X,Y, on both sides. The X-axis is the master and the Y-axis is the follower. To synchronize Y with the commanded position of X, use the instructions:

GA XC	Specify master as commanded position of X
GR,1	Set gear ratio for Y as 1:1
PR 3000	Command X motion
BG X	Start motion on X axis

You may also perform profiled position corrections in the electronic gearing mode. Suppose, for example, that you need to advance the slave 10 counts. Simply command

IP ,10	Specify an incremental position movement of 10 on Y axis.

Under these conditions, this IP command is equivalent to:

PR,10	Specify position relative movement of 10 on Y axis
BGY	Begin motion on Y axis

Often the correction is quite large. Such requirements are common when synchronizing cutting knives or conveyor belts.

Example - Synchronize two conveyor belts with trapezoidal velocity correction.

Instruction	Interpretation

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

GAX	Define master axis as X
GR,2	Set gear ratio 2:1 for Y
PR,300	Specify correction distance
SP,5000	Specify correction speed
AC,100000	Specify correction acceleration
DC,100000	Specify correction deceleration
BGY	Start correction

Electronic Cam

The electronic cam is a motion control mode which enables the periodic synchronization of several axes of motion. Up to 7 axes can be slaved to one master axis. The master axis encoder must be input through a main encoder port.

The electronic cam is a more general type of electronic gearing which allows a table-based relationship between the axes. It allows synchronizing all the controller axes. For example, the DMC-1580 controller may have one master and up to seven slaves. To simplify the presentation, we will limit the description to a 4-axis controller.

To illustrate the procedure of setting the cam mode, consider the cam relationship for the slave axis Y, when the master is X. Such a graphic relationship is shown in Figure 6.8.

Step 1. Selecting the master axis

The first step in the electronic cam mode is to select the master axis. This is done with the instruction

EAp where p = X,Y,Z,W

p is the selected master axis

Step 2. Specify the master cycle and the change in the slave axis(es).

In the electronic cam mode, the position of the master is always expressed modulo one cycle. In this example, the position of x is always expressed in the range between 0 and 6000. Similarly, the slave position is also redefined such that it starts at zero and ends at 1500. At the end of a cycle when the master is 6000 and the slave is 1500, the positions of both x and y are redefined as zero. To specify the master cycle and the slave cycle change, we use the instruction EM.

EM x,y,z,w

where x,y,z,w specify the cycle of the master and the total change of the slaves over one cycle.

The cycle of the master is limited to 8,388,607 whereas the slave change per cycle is limited to 2,147,483,647. If the change is a negative number, the absolute value is specified. For the given example, the cycle of the master is 6000 counts and the change in the slave is 1500. Therefore, we use the instruction:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EM 6000,1500

Step 3. Specify the master interval and starting point.

Next we need to construct the ECAM table. The table is specified at uniform intervals of master positions. Up to 256 intervals are allowed. The size of the master interval and the starting point are specified by the instruction:

EP m,n

where m is the interval width in counts, and n is the starting point.

For the given example, we can specify the table by specifying the position at the master points of 0, 2000, 4000 and 6000. We can specify that by

EP 2000,0

Step 4. Specify the slave positions.

Next, we specify the slave positions with the instruction

ET[n]=x,y,z,w

where n indicates the order of the point.

The value, n, starts at zero and may go up to 256. The parameters x,y,z,w indicate the corresponding slave position. For this example, the table may be specified by

ET[0]=,0
ET[1]=,3000
ET[2]=,2250
ET[3]=,1500

This specifies the ECAM table.

Step 5. Enable the ECAM

To enable the ECAM mode, use the command

EB n

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

where n=1 enables ECAM mode and n=0 disables ECAM mode.

Step 6. Engage the slave motion

To engage the slave motion, use the instruction

EG x,y,z,w

where x,y,z,w are the master positions at which the corresponding slaves must be engaged.

If the value of any parameter is outside the range of one cycle, the cam engages immediately. When the cam is engaged, the slave position is redefined, modulo one cycle.

Step 7. Disengage the slave motion

To disengage the cam, use the command

EQ x,y,z,w

where x,y,z,w are the corresponding slave axes are disengaged.

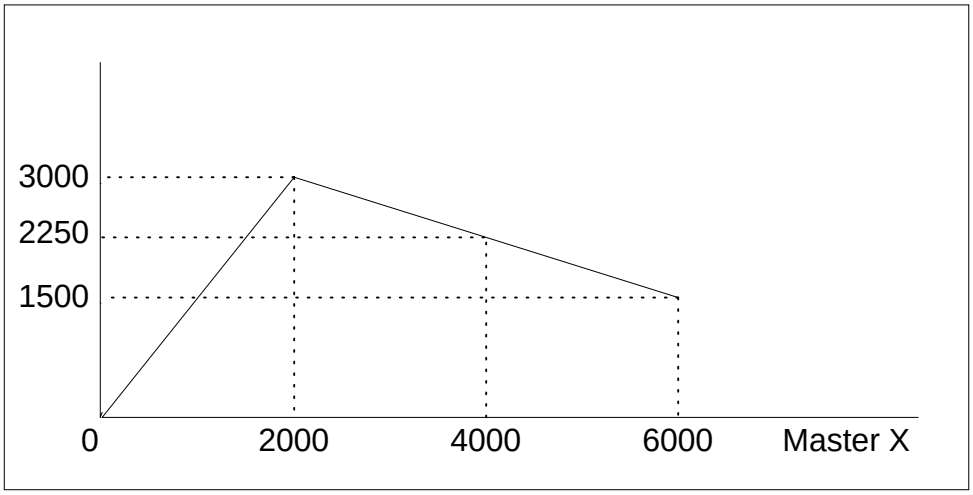


Figure 6.8: Electronic Cam Example

This disengages the slave axis at a specified master position. If the parameter is outside the master cycle, the stopping is instantaneous.

Programmed start and stop can be used only when the master moves forward.

Some Examples

To illustrate the complete process, consider the cam relationship described by

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

the equation:

$Y = 0.5 * X + 100 \sin(0.18*X)$ where X is the master, with a cycle of 2000 counts.

The cam table can be constructed manually, point by point, or automatically by a program. The following program includes the set-up.

The instruction EAX defines X as the master axis. The cycle of the master is

2000. Over that cycle, X varies by 1000. This leads to the instruction EM 2000,1000.

Suppose we want to define a table with 100 segments. This implies increments of 20 counts each. If the master points are to start at zero, the required instruction is EP 20,0.

The following routine computes the table points. As the phase equals 0.18X and X varies in increments of 20, the phase varies by increments of 3.6°. The program then computes the values of Y according to the equation and assigns the values to the table with the instruction ET[N] = ,Y.

Instruction	Interpretation
#SETUP	Label
EAX	Select X as master
EM 2000,1000	Cam cycles
EP 20,0	Master position increments
N = 0	Index
#LOOP	Loop to construct table from equation
*P = N*3.6*	Note 3.6 = 0.18*20
*S = @SIN [P] *100*	Define sine position
*Y = N *10+S*	Define slave position
ET [N] =, Y	Define table
N = N+1	
JP #LOOP, N<=100	Repeat the process
EN	

Now suppose that the slave axis is engaged with a start signal, input 1, but that both the engagement and disengagement points must be done at the center of the cycle: X = 1000 and Y = 500. This implies that Y must be driven to that point to avoid a jump.

This is done with the program:

Instruction	Interpretation
#RUN	Label
EB1	Enable cam
PA,500	starting position
SP,5000	Y speed
BGY	Move Y motor
AM	After Y moved
AI1	Wait for start signal
EG,1000	Engage slave
AI - 1	Wait for stop signal
EQ,1000	Disengage slave
EN	End

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Command Summary - ECAM Mode

command	description
EA ABCDEFG or H	Specify ECAM master axis
EB n (n = 0 or 1)	Enable ECAM
EG a,b,c,d,e,f,g,h	ECAM go - Specifies position for engaging ECAM
EM a,b,c,d,e,f,g,h	Specify cam cycle
EP m,n	Specifies cam table interval and starting point
EQ a,b,c,d,e,f,g,h	Quit ECAM
ET[n]	Specify ECAM table entry

Operand Summary - ECAM Mode

operand	Description
_EB	Contains the state of ECAM mode (0 = disabled, 1 = enabled)
_EGx	Contains ecam status for specified axis (0 = engaged, 1= disengaged)
_EMx	Contains the cycle of the specified axis
_EP	Contains the value of the interval
_EQx	Contains status of ECAM mode for specified axis

Example - Using ECAM

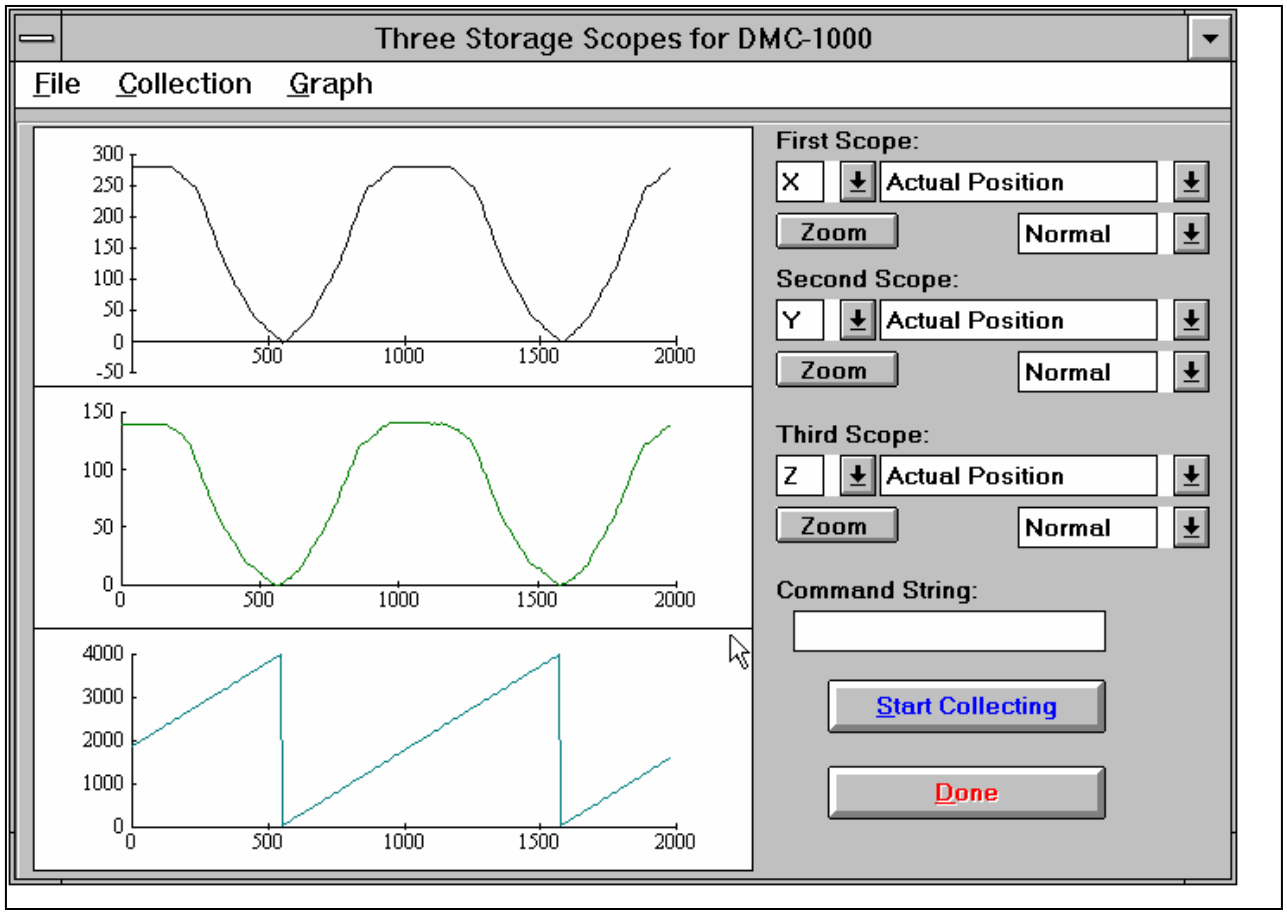
The following example illustrates a cam program with a master axis, Z, and two slaves, X and Y.

Instruction	Interpretation
#A;V1=0	Label; Initialize variable
PA 0,0;BGXY;AMXY	Go to position 0,0 on X and Y axes
EA Z	Z axis as the Master for ECAM
EM 0,0,4000	Change for Z is 4000, zero for X, Y
EP400,0	ECAM interval is 400 counts with zero start
ET[0]=0,0	When master is at 0 position; 1st point.
ET[1]=40,20	2nd point in the ECAM table
ET[2]=120,60	3rd point in the ECAM table
ET[3]=240,120	4th point in the ECAM table
ET[4]=280,140	5th point in the ECAM table
ET[5]=280,140	6th point in the ECAM table
ET[6]=280,140	7th point in the ECAM table
ET[7]=240,120	8th point in the ECAM table
ET[8]=120,60	9th point in the ECAM table
ET[9]=40,20	10th point in the ECAM table
ET[10]=0,0	Starting point for next cycle
EB 1	Enable ECAM mode
JGZ=4000	Set Z to jog at 4000
EG 0,0	Engage both X and Y when Master = 0
BGZ	Begin jog on Z axis
#LOOP;JP#LOOP,V1=0	Loop until the variable is set

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EQ2000,2000	Disengage X and Y when Master = 2000
MF,, 2000	Wait until the Master goes to 2000
ST Z	Stop the Z axis motion
EB 0	Exit the ECAM mode
EN	End of the program

The above example shows how the ECAM program is structured and how the commands can be given to the controller. The next page provides the results captured by the WSDK program. This shows how the motion will be seen during the ECAM cycles. The first graph is for the X axis, the second graph shows the cycle on the Y axis and the third graph shows the cycle of the Z axis.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Contour Mode

The DMC-1500 also provides a contouring mode. This mode allows any arbitrary position curve to be prescribed for 1 to 8 axes. This is ideal for following computer generated paths such as parabolic, spherical or user-defined profiles. The path is not limited to straight line and arc segments and the path length may be infinite.

Specifying Contour Segments

The Contour Mode is specified with the command, CM. For example, CMXZ specifies contouring on the X and Z axes. Any axes that are not being used in the contouring mode may be operated in other modes.

A contour is described by position increments which are described with the command, CD x,y,z,w over a time interval, DT n. The parameter, n, specifies the time interval. The time interval is defined as 2^n ms, where n is a number between 1 and 8. The controller performs linear interpolation between the specified increments, where one point is generated for each millisecond.

Consider, for example, the trajectory shown in Fig. 6.4. The position X may be described by the points:

Point 1	X=0 at T=0ms
Point 2	X=48 at T=4ms
Point 3	X=288 at T=12ms
Point 4	X=336 at T=28ms

The same trajectory may be represented by the increments

Increment 1	DX=48	Time=4	DT=2
Increment 2	DX=240	Time=8	DT=3
Increment 3	DX=48	Time=16	DT=4

When the controller receives the command to generate a trajectory along these points, it interpolates linearly between the points. The resulting interpolated points include the position 12 at 1 msec, position 24 at 2 msec, etc.

The programmed commands to specify the above example are:

Instruction	Interpretation
#A	
CMX	Specifies X axis for contour mode
DT 2	Specifies first time interval, 2^2 ms
CD 48;WC	Specifies first position increment
DT 3	Specifies second time interval, 2^3 ms
CD 240;WC	Specifies second position increment
DT 4	Specifies the third time interval, 2^4 ms
CD 48;WC	Specifies the third position increment
DT0;CD0	Exits contour mode
EN	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

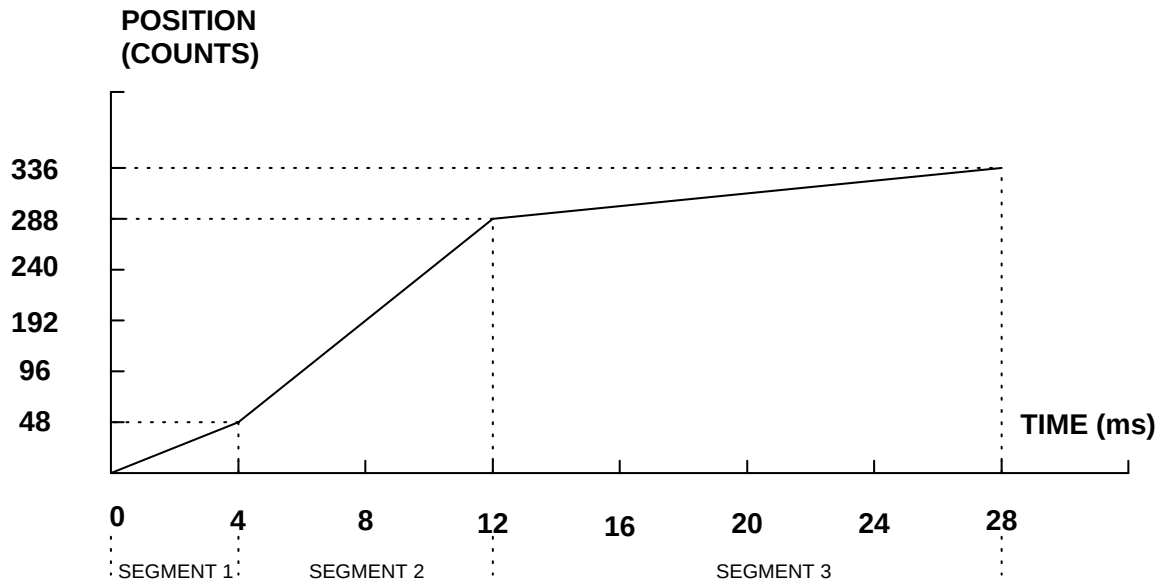


Figure 6.4 - The Required Trajectory

Additional Commands

The command, WC, is used as a trippoint "When Complete". This allows the DMC-1500 to use the next increment only when it is finished with the previous one. Zero parameters for DT followed by zero parameters for CD exit the contour mode.

If no new data record is found and the controller is still in the contour mode, the controller waits for new data. No new motion commands are generated while waiting. If bad data is received, the controller responds with a ?.

Command Summary - Contour Mode

COMMAND	DESCRIPTION
CM XYZW	Specifies which axes for contouring mode. Any non-contouring axes may be operated in other modes.
CM ABCDEFGH	Contour axes for DMC-1580
CD x,y,z,w	Specifies position increment over time interval. Range is +/-32,000. Zero ends contour mode.
CD a,b,c,d,e,f,g,h	Position increment data for DMC-1580
DT n	Specifies time interval 2^n msec for position increment, where n is an integer between 1 and 8. Zero ends contour mode. If n does not change, it does not need to be specified with each CD.
WC	Waits for previous time interval to be complete before next data record is processed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

General Velocity Profiles

The Contour Mode is ideal for generating any arbitrary velocity profiles. The velocity profile can be specified as a mathematical function or as a collection of points.

The design includes two parts: Generating an array with data points and running the program.

Generating an Array - An Example

Consider the velocity and position profiles shown in Fig. 6.5. The objective is to rotate a motor a distance of 6000 counts in 120 ms. The velocity profile is sinusoidal to reduce the jerk and the system vibration. If we describe the position displacement in terms of A counts in B milliseconds, we can describe the motion in the following manner:

$$\omega = \frac{A}{B}\left(1 - \cos(2\pi/B)\right)$$

$$X = \frac{AT}{B} - \frac{A}{2\pi}\sin(2\pi/B)$$

Note: ω is the angular velocity; X is the position; and T is the variable, time, in milliseconds.

In the given example, A=6000 and B=120, the position and velocity profiles are:

$$X = 50T - (6000/2\pi) \sin (2\pi\ T/120)$$

Note that the velocity, ω, in count/ms, is

$$\omega = 50\ [1 - \cos 2\pi\ T/120]$$

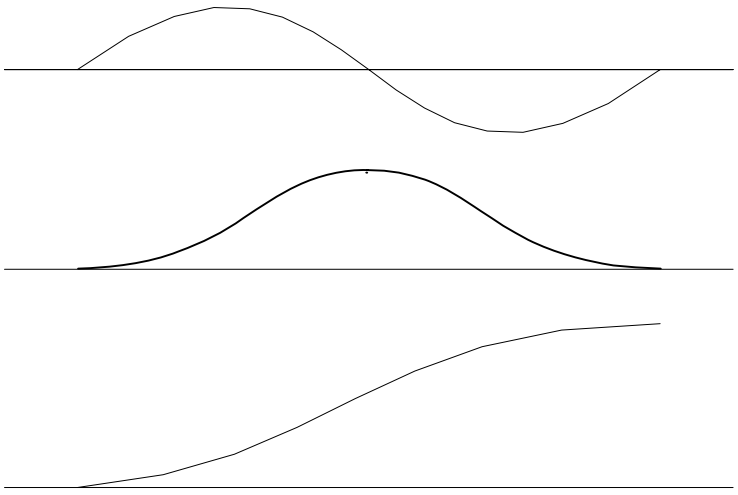


Figure 6.5 - Velocity Profile with Sinusoidal Acceleration

The DMC-1500 can compute trigonometric functions. However, the argument must be expressed in degrees. Using our example, the equation for X is written as:

$$X = 50T - 955 \sin 3T$$

A complete program to generate the contour movement in this example is given below. To generate an array, we compute the position value at intervals of 8 ms. This is stored at the array POS. Then,

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

the difference between the positions is computed and is stored in the array DIF. Finally the motors are run in the contour mode.

Contour Mode Example

Instruction	Interpretation
#POINTS	Program defines X points
DM POS[16]	Allocate memory
DM DIF[15]	
C=0	Set initial conditions, C is index
T=0	T is time in ms
#A	
V1=50*T	
V2=3*T	Argument in degrees
V3=-955*@SIN[V2]+V1	Compute position
V4=@INT[V3]	Integer value of V3
POS[C]=V4	Store in array POS
T=T+8	
C=C+1	
JP #A,C<16	
#B	Program to find position differences
C=0	
#C	
D=C+1	
DIF[C]=POS[D]-POS[C]	Compute the difference and store
C=C+1	
JP #C,C<15	
EN	End first program
#RUN	Program to run motor
CMX	Contour Mode
DT3	4 millisecond intervals
C=0	
#E	
CD DIF[C]	Contour Distance is in DIF
WC	Wait for completion
C=C+1	
JP #E,C<15	
DT0	
CD0	Stop Contour
EN	End the program

Teach (Record and Play-Back)

Several applications require teaching the machine a motion trajectory. Teaching can be accomplished using the DMC-1500 automatic array capture feature to capture position data. The captured data may then be played back in the contour mode. The following array commands are used:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DM C[n]	Dimension array
RA C[]	Specify array for automatic record (up to 8 arrays)
RD _TPX	Specify data for capturing (such as _TPX or _TPZ)
RC n,m	Specify capture time interval where n is 2n msec, m is number of records to be captured
RC? or _RC	Returns a 1 if recording

Record and Playback Example:

Instruction	Interpretation
#RECORD	Begin Program
DP0	Define position for X axis to be 0
DA*[ ]	De-allocate all arrays
DM XPOS [501]	Dimension 501 element array called XPOS
RA XPOS [ ]	Record Elements into XPOS array
RD_TPX	Element to be recorded is encoder position of X axis
MOX	Motor off for X axis
RC2	Begin Recording with a sample rate of 2 msec
#LOOP1;JP#LOOP1,_RC=1	Loop until all elements have been recorded
#COMPUTE	Routine to determine the difference between consecutive points
DM DX [500]	Dimension a 500 element array to hold contour points
I = 0	Set loop counter
#LOOP2	Loop to calculate the difference
DX[I]=XPOS[I+1]-XPOS[I]	Calculate difference
I=I+1	Update loop counter
JP#LOOP2,I<500	Continue looping until DX is full
#PLAYBK	Routine to play back motion that was recorded
SHX	Servo Here
WT1000	Wait 1 sec (1000 msec)
CMX	Specify contour mode on X axis
DT2	Set contour data rate to be 2 msec
I=0	Set array index to 0
#LOOP3	Subroutine to execute contour points
CD DX[I];WC	Contour data command; Wait for next contour point
I=I+1	Update index
JP#LOOP3,I<500	Continue until all array elements have been executed
DT0	Set contour update rate to 0
CD0	Disable the contour mode (combination of DT0 and CD0)
EN	End program

For additional information about Automatic Array Capture, see Chapter 7, Arrays.

Stepper Motor Operation

When configured for stepper motor operation, several commands are interpreted differently than from servo mode. The following describes operation with stepper motors.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Specifying Stepper Motor Operation

In order to command stepper motor operation, the appropriate stepper mode jumpers must be installed. See chapter 2 for this installation.

Stepper motor operation is specified by the command MT. The argument for MT is as follows:

2 specifies a stepper motor with active low step output pulses

-2 specifies a stepper motor with active high step output pulses

2.5 specifies a stepper motor with active low step output pulses and reversed direction

-2.5 specifies a stepper motor with active high step output pulse and reversed direction

Stepper Motor Smoothing

The command, KS, provides stepper motor smoothing. The effect of the smoothing can be thought of as a simple Resistor-Capacitor (single pole) filter. The filter occurs after the motion profiler and has the effect of smoothing out the spacing of pulses for a more smooth operation of the stepper motor. Use of KS is most applicable when operating in full step or half step operation. KS will cause the step pulses to be delayed in accordance with the time constant specified.

When operating with stepper motors, you will always have some amount of stepper motor smoothing, KS. Since this filtering effect occurs after the profiler, the profiler may be ready for additional moves before all of the step pulses have gone through the filter. It is important to consider this effect since steps may be lost if the controller is commanded to generate an additional move before the previous move has been completed. See the discussion below, *Monitoring Generated Pulses vs Commanded Pulses*.

The general motion smoothing command, IT, can also be used. The purpose of the command, IT, is to smooth out the motion profile and decrease 'jerk' due to acceleration.

Monitoring Generated Pulses vs Commanded Pulses

For proper controller operation, it is necessary to make sure that the controller has completed generating all step pulses before making additional moves. This is most particularly important if you are moving back and forth. For example, when operating with servo motors, the trippoint AM (After Motion) is used to determine when the motion profiler is complete and is prepared to execute a new motion command. However when operating in stepper mode, the controller may still be generating step pulses when the motion profiler is complete. This is caused by the stepper motor smoothing filter, KS. To understand this, consider the steps the controller executes to generate step pulses:

First, the controller generates a motion profile in accordance with the motion commands.

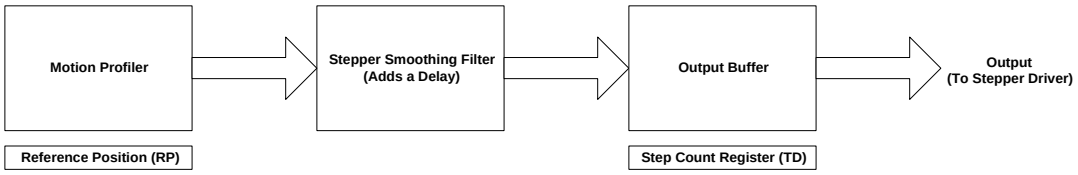
Second, the profiler generates pulses as prescribed by the motion profile. The pulses that are generated by the motion profiler can be monitored by the command, RP (Reference Position). RP gives the absolute value of the position as determined by the motion profiler. The command, DP, can be used to set the value of the reference position. For example, DP 0, defines the reference position of the X axis to be zero.

Third, the output of the motion profiler is filtered by the stepper smoothing filter. This filter adds a delay in the output of the stepper motor pulses. The amount of delay depends on the parameter which is specified by the command, KS. As mentioned earlier, there will always be some amount of stepper motor smoothing. The default value for KS is 2 which corresponds to a time constant of 6 sample periods.

Fourth, the output of the stepper smoothing filter is buffered and is available for input to the stepper motor driver. The pulses which are generated by the smoothing filter can be monitored by the command, TD (Tell Dual). TD gives the absolute value of the position as determined by actual output

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

of the buffer. The command, DP sets the value of the step count register as well as the value of the reference position. For example, DP 0, defines the reference position of the X axis to be zero.



Motion Complete Trippoint

When used in stepper mode, the MC command will hold up execution of the proceeding commands until the controller has generated the same number of steps out of the step count register as specified in the commanded position. The MC trippoint (Motion Complete) is generally more useful than AM trippoint (After Motion) since the step pulses can be delayed from the commanded position due to stepper motor smoothing.

Using an Encoder with Stepper Motors

An encoder may be used on a stepper motor to check the actual motor position with the commanded position. If an encoder is used, it must be connected to the main encoder input. Note: The auxiliary encoder is not available while operating with stepper motors. The position of the encoder can be interrogated by using the command, TP. The position value can be defined by using the command, DE. Note: Closed loop operation with a stepper motor is not possible.

Command Summary - Stepper Motor Operation

COMMAND	DESCRIPTION
DE	Define Encoder Position (When using an encoder)
DP	Define Reference Position and Step Count Register
IT	Motion Profile Smoothing - Independent Time Constant
KS	Stepper Motor Smoothing
MT	Motor Type (2,-2,2.5 or -2.5 for stepper motors)
RP	Report Commanded Position
TD	Report number of step pulses generated by controller
TP	Tell Position of Encoder

Operand Summary - Stepper Motor Operation

OPERAND	DESCRIPTION
_DEx	Contains the value of the step count register
_DPx	Contains the value of the main encoder
_ITx	Contains the value of the Independent Time constant for the 'x' axis
_KS	Contains the value of the Stepper Motor Smoothing Constant for the 'x' axis
_MT	Contains the motor type value for the 'x' axis
_RP	Contains the commanded position generated by the profiler
_TD	Contains the value of the step count register

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

_TP	Contains the value of the main encoder

Dual Loop (Auxiliary Encoder)

The DMC-1500 provides an interface for a second encoder for each axis except for axes configured for stepper motor operation. When used, the second encoder is typically mounted on the motor or the load, but may be mounted in any position. The most common use for the second encoder is backlash compensation, described below.

The auxiliary encoder may also be used for gearing. In this case, the auxiliary encoder input is used to monitor an encoder which is not under control of the DMC-1500. To use the auxiliary encoder for gearing, the master axis is specified as the auxiliary encoder and GR is used to specify the gear ratios. For more information, see previous section Electronic Gearing on page 54.

The second encoder may be a standard quadrature type, or it may provide pulse and direction. The controller also offers the provision for inverting the direction of the encoder rotation. The main and the auxiliary encoders are configured with the CE command. The command form is CE x,y,z,w (or a,b,c,d,e,f,g,h for controllers with more than 4 axes) where the parameters x,y,z,w each equal the sum of two integers m and n. m configures the main encoder and n configures the auxiliary encoder.

Using the CE Command

m=	Main Encoder	n=	Second Encoder
0	Normal quadrature	0	Normal quadrature
1	Pulse & direction	4	Pulse & direction
2	Reverse quadrature	8	Reversed quadrature
3	Reverse pulse & direction	12	Reversed pulse & direction

For example, to configure the main encoder for reversed quadrature, m=2, and a second encoder of pulse and direction, n=4, the total is 6, and the command for the X axis is

```
CE 6
```

Additional Commands for the Auxiliary Encoder

The command, DE x,y,z,w, can be used to define the position of the auxiliary encoders. For example,

```
DE 0,500,-30,300
```

sets their initial values.

The positions of the auxiliary encoders may be interrogated with the command, DE?. For example

```
DE ?,,?
```

returns the value of the X and Z auxiliary encoders.

The auxiliary encoder position may be assigned to variables with the instructions

```
V1= _DEX
```

The command, TD XYZW, returns the current position of the auxiliary encoder.

The command, DV XYZW, configures the auxiliary encoder to be used for backlash compensation.

Backlash Compensation

There are two methods for backlash compensation using the auxiliary encoders:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Continuous dual loop

Sampled dual loop

To illustrate the problem, consider a situation in which the coupling between the motor and the load has a backlash. To compensate for the backlash, position encoders are mounted on both the motor and the load.

The continuous dual loop combines the two feedback signals to achieve stability. This method requires careful system tuning, and depends on the magnitude of the backlash. However, once successful, this method compensates for the backlash continuously.

The second method, the sampled dual loop, reads the load encoder only at the end point and performs a correction. This method is independent of the size of the backlash. However, it is effective only in point-to-point motion systems which require position accuracy only at the endpoint.

Example - Continuous Dual Loop

Note: In order to have a stable continuous dual loop system, the encoder on the motor must be of equal or higher resolution than the encoder on the load.

Connect the load encoder to the main encoder port and connect the motor encoder to the dual encoder port. The dual loop method splits the filter function between the two encoders. It applies the KP (proportional) and KI (integral) terms to the position error, based on the load encoder, and applies the KD (derivative) term to the motor encoder. This method results in a stable system.

The dual loop method is activated with the instruction DV (Dual Velocity), where

```
DV      1,1,1,1
```

activates the dual loop for the four axes and

```
DV      0,0,0,0
```

disables the dual loop.

Note that the dual loop compensation depends on the backlash magnitude, and in extreme cases will not stabilize the loop. The proposed compensation procedure is to start with KP=0, KI=0 and to maximize the value of KD under the condition DV1. Once KD is found, increase KP gradually to a maximum value, and finally, increase KI, if necessary.

Example - Sampled Dual Loop

In this example, we consider a linear slide which is run by a rotary motor via a lead screw. Since the lead screw has a backlash, it is necessary to use a linear encoder to monitor the position of the slide. For stability reasons, it is best to use a rotary encoder on the motor.

Connect the rotary encoder to the X-axis and connect the linear encoder to the auxiliary encoder of X. Assume that the required motion distance is one inch, and that this corresponds to 40,000 counts of the rotary encoder and 10,000 counts of the linear encoder.

The design approach is to drive the motor a distance, which corresponds to 40,000 rotary counts. Once the motion is complete, the controller monitors the position of the linear encoder and performs position corrections.

This is done by the following program.

Instruction	Interpretation
#DUALOOP	Label
CE 0	Configure encoder
DE0	Set initial value
PR 40000	Main move

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
BGX                     Start motion
#Correct                Correction loop
AMX                     Wait for motion completion
V1=10000-_DEX           Find linear encoder error
V2=-_TEX/4+V1           Compensate for motor error
JP#END,@ABS[V2]<2       Exit if error is small
PR V2*4                 Correction move
BGX                     Start correction
JP#CORRECT              Repeat
#END
EN
```

Command Summary - Using the Auxiliary Encoder

COMMAND	DESCRIPTION
CE	Configure Encoder Type
DE	Define dual (auxiliary) encoder position
DV	Set continuous dual loop mode - see description below
GA	Set master axis for gearing - the auxiliary encoder input can be used for gearing
GR	Set gear ratio for gearing - the auxiliary encoder input can be used for gearing
TD	Tell dual (auxiliary) encoder input position.

Operand Summary - Using the Auxiliary Encoder

OPERAND	DESCRIPTION
_CEx	Contains the encoder configuration for the specified axis
_DEx	Contains the current position of the specified auxiliary encoder
_DVx	Contains a '1' or '0' if the specified axis is in continuous dual loop operation.
_GRx	Contains the value of the gear ratio for the specified axis
_TDx	Contains the position of the specified auxiliary encoder.

Motion Smoothing

The DMC-1500 controller allows the smoothing of the velocity profile to reduce the mechanical vibration of the system.

Trapezoidal velocity profiles have acceleration rates which change abruptly from zero to maximum value. The discontinuous acceleration results in jerk which causes vibration. The smoothing of the acceleration profile leads to a continuous acceleration profile and reduces the mechanical shock and vibration.

Using the IT and VT Commands (S curve profiling):



When operating with servo motors, motion smoothing can be accomplished with the IT and VT command. These commands filter the acceleration and deceleration functions to produce a smooth velocity profile. The resulting velocity profile, known as S curve, has continuous acceleration and results in reduced mechanical vibrations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The smoothing function is specified by the following commands:

IT x,y,z,w	Independent time constant
VT n	Vector time constant

The command, IT, is used for smoothing independent moves of the type JG, PR, PA and the command, VT, is used to smooth vector moves of the type VM and LM.

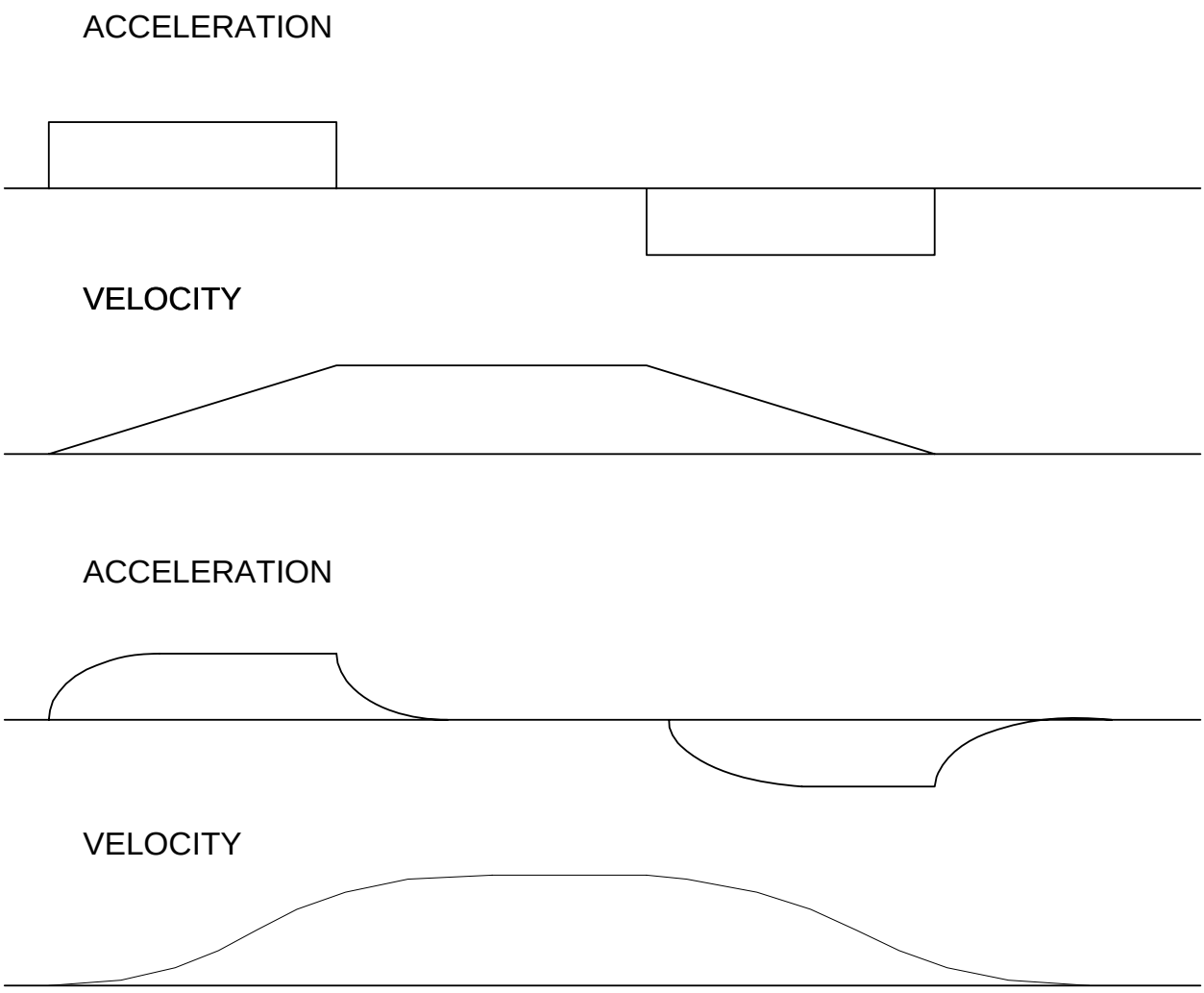
The smoothing parameters, x,y,z,w and n are numbers between 0 and 1 and determine the degree of filtering. The maximum value of 1 implies no filtering, resulting in trapezoidal velocity profiles. Smaller values of the smoothing parameters imply heavier filtering and smoother moves.

The following example illustrates the effect of smoothing. Fig. 6.6 shows the trapezoidal velocity profile and the modified acceleration and velocity.

Note that the smoothing process results in longer motion time.

Example - Smoothing

Instruction	Interpretation
PR 20000	Position
AC 100000	Acceleration
DC 100000	Deceleration
SP 5000	Speed
IT .5	Filter for S-curve
BG X	Begin



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 6.6 - Trapezoidal velocity and smooth velocity profiles

Using the KS Command (Step Motor Smoothing):

When operating with step motors, motion smoothing can be accomplished with the command, KS. The KS command smoothes the frequency of step motor pulses. Similar to the commands, IT and VT, this produces a smooth velocity profile.

The step motor smoothing is specified by the following command:

KS x,y,z,w where x,y,z,w is an integer from 1 to 16 and represents the amount of smoothing

The command, IT, is used for smoothing independent moves of the type JG, PR, PA and the command, VT, is used to smooth vector moves of the type VM and LM.

The smoothing parameters, x,y,z,w and n are numbers between 0 and 16 and determine the degree of filtering. The minimum value of 1 implies no filtering, resulting in trapezoidal velocity profiles. Larger values of the smoothing parameters imply heavier filtering and smoother moves.

Note that KS is valid only for step motors.

Homing

The Find Edge (FE) and Home (HM) instructions may be used to home the motor to a mechanical reference. This reference is connected to the Home input line. The HM command initializes the motor to the encoder index pulse in addition to the Home input. The configure command (CN) is used to define the polarity of the home input.

The Find Edge (FE) instruction is useful for initializing the motor to a home switch. The home switch is connected to the Homing Input. When the Find Edge command and Begin is used, the motor will accelerate up to the slew speed and slew until a transition is detected on the Homing line. The motor will then decelerate to a stop. A high deceleration value must be input before the find edge command is issued for the motor to decelerate rapidly after sensing the home switch. The velocity profile generated is shown in Fig. 6.7.

The Home (HM) command can be used to position the motor on the index pulse after the home switch is detected. This allows for finer positioning on initialization. The command sequence HM and BG causes the following sequence of events to occur.

Upon begin, motor accelerates to the slew speed. The direction of its motion is determined by the state of the homing input. A zero (GND) will cause the motor to start in the forward direction; +5V will cause it to start in the reverse direction. The CN command is used to define the polarity of the home input.

Upon detecting the home switch changing state, the motor begins decelerating to a stop.

The motor then traverses very slowly back until the home switch toggles again.

The motor then traverses forward until the encoder index pulse is detected.

The DMC-1500 defines the home position (0) as the position at which the index was detected.

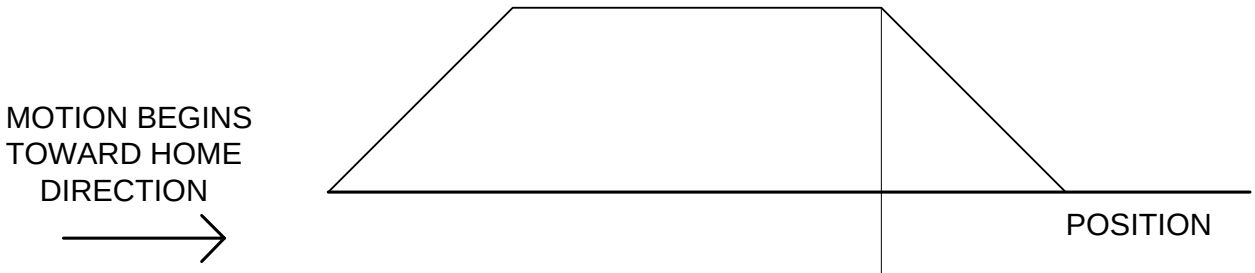
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

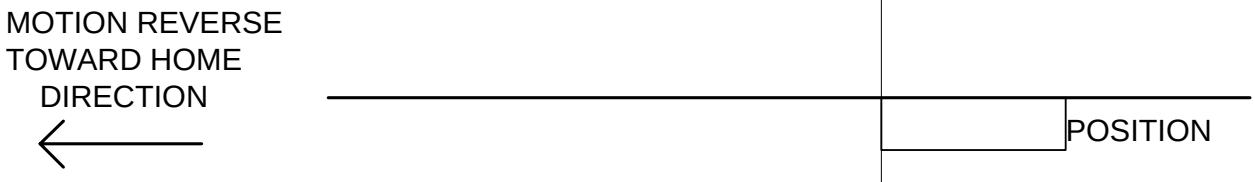
Example 1 - Using Home Command

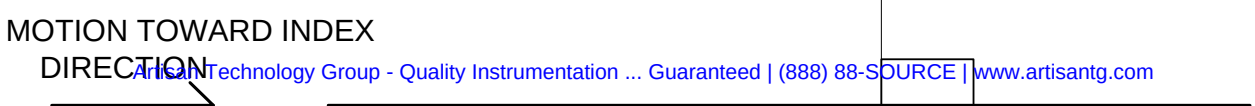
Instruction	Interpretation
#HOME	Label
AC 1000000	Acceleration Rate
DC 1000000	Deceleration Rate
SP 5000	Speed for Home Search
HM X	Home X
BG X	Begin Motion
AM X	After Complete
MG "AT HOME"	Send Message
EN	End

Example 2 - Using Find Edge Command

Instruction	Interpretation
#EDGE	Label
AC, 2000000	Acceleration rate
DC, 2000000	Deceleration rate
SP, 8000	Speed
FE Y	Find edge command
BG Y	Begin motion
AM Y	After complete
MG "FOUND HOME"	Print message
DP,0	Define position as 0
EN	End







Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 6.7 - Motion intervals in the Home sequence

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

High Speed Position Capture (The Latch Function)

Often it is desirable to capture the position precisely for registration applications. The DMC-1500 provides a position latch feature. This feature allows the position of X,Y,Z or W to be captured within 25 microseconds of an external low input signal. The general inputs 1 through 4, and 9 through 12 correspond to each axis.

IN1 X-axis latch	IN 9 E-axis latch
IN2 Y-axis latch	IN10 F-axis latch
IN3 Z-axis latch	IN11 G-axis latch
IN4 W-axis latch	IN12 H-axis latch

Note: To insure a position capture within 25 microseconds, the input signal must be a transition from high to low.

The DMC-1500 software commands, AL and RL, are used to arm the latch and report the latched position. The steps to use the latch are as follows:

Give the AL XYZW command, or ABCDEFGH for DMC-1580, to arm the latch for the specified axis or axes.

Test to see if the latch has occurred (Input goes low) by using the _AL X or Y or Z or W command. Example, V1=_ALX returns the state of the X latch into V1. V1 is 1 if the latch has not occurred.

After the latch has occurred, read the captured position with the RL XYZW command or _RL XYZW.

Note: The latch must be re-armed after each latching event.

Example - Using Position Capture Function

Instruction	Interpretation
#Latch	Latch program
JG,5000	Jog Y
BG Y	Begin motion on Y axis
AL Y	Arm Latch for Y axis
#Wait	#Wait label for loop
JP #Wait,_ALY=1	Jump to #Wait label if latch has not occurred
Result=_RLY	Set value of variable ‘Result’ equal to the report position of y axis
Result=	Print result
EN	End

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE LEFT BLANK INTENTIONALLY

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 7 Application Programming

Overview

The DMC-1500 provides a powerful programming language that allows users to customize the controller for their particular application. Programs can be downloaded into the DMC-1500 memory freeing the host computer for other tasks. However, the host computer can send commands to the controller at any time, even while a program is being executed.

In addition to standard motion commands, the DMC-1500 provides commands that allow the DMC-1500 to make its own decisions. These commands include conditional jumps, event triggers and subroutines. For example, the command JP#LOOP, n<10 causes a jump to the label #LOOP if the variable n is less than 10.

For greater programming flexibility, the DMC-1500 provides user-defined variables, arrays and arithmetic functions. For example, with a cut-to-length operation, the length can be specified as a variable in a program which the operator can change as necessary.

The following sections in this chapter discuss all aspects of creating applications programs.

Using the DMC-1500 Editor to Enter Programs

Application programs for the DMC-1500 may be created and edited either locally using the DMC-1500 editor or remotely using another editor and then downloading the program into the controller. (Galil's Terminal and SDK software provides an editor and UPLOAD and DOWNLOAD utilities).

The DMC-1500 provides a line Editor for entering and modifying programs. The Edit mode is entered with the ED instruction. The ED command can only be given when the controller is not running a program.

In the Edit Mode, each program line is automatically numbered sequentially starting with 000. If no parameter follows the ED command, the editor prompter will default to the last line of the program in memory. If desired, the user can edit a specific line number or label by specifying a line number or label following ED.

Instruction	Interpretation
ED	Puts Editor at end of last program
ED 5	Puts Editor at line 5
ED #BEGIN	Puts Editor at label #BEGIN

The program memory space for the DMC-1500 is 1000 lines x 80 characters per line

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Line numbers appear as 000,001,002 and so on. Program commands are entered following the line numbers. Multiple commands may be given on a single line as long as the total number of characters doesn't exceed the limits given above.

While in the Edit Mode, the programmer has access to special instructions for saving, inserting and deleting program lines. These special instructions are listed below:

Edit Mode Commands

<RETURN>

Typing the return or enter key causes the current line of entered instructions to be saved. The editor will automatically advance to the next line. Thus, hitting a series of <RETURN> will cause the editor to advance a series of lines. Note, changes on a program line will not be saved unless a <return> is given.

<cntrl>P

The <cntrl>P command moves the editor to the previous line.

<cntrl>I

The <cntrl>I command inserts a line above the current line. For example, if the editor is at line number 2 and <cntrl>I is applied, a new line will be inserted between lines 1 and 2. This new line will be labeled line 2. The old line number 2 is renumbered as line 3.

<cntrl>D

The <cntrl>D command deletes the line currently being edited. For example, if the editor is at line number 2 and <cntrl>D is applied, line 2 will be deleted. The previous line number 3 is now renumbered as line number 2.

<cntrl>Q

The <cntrl>Q quits the editor mode. In response, the DMC-1500 will return a colon.

After the Edit session is over, the user may list the entered program using the LS command. If no number or label follows the LS command, the entire program will be listed. The user can start listing at a specific line or label. A range of program lines can also be displayed. For example;

Instruction	Interpretation
LS	List entire program
LS 5	Begin listing at line 5
LS 5,9	List lines 5 through 9
LS #A,9	List line label #A through line 9

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Program Format

A DMC-1500 program consists of DMC-1500 instructions combined to solve a machine control application. Action instructions, such as starting and stopping motion, are combined with Program Flow instructions to form the complete program. Program Flow instructions evaluate real-time conditions, such as elapsed time or motion complete, and alter program flow accordingly.

Each DMC-1500 instruction in a program must be separated by a delimiter. Valid delimiters are the semicolon (;) or carriage return. The semicolon is used to separate multiple instructions on a single program line where the maximum number of instructions on a line is limited by 38 characters. A carriage return enters the final command on a program line.

Using Labels in Programs

All DMC-1500 programs must begin with a label and end with an End (EN) statement. Labels start with the pound (#) sign followed by a maximum of seven characters. The first character must be a letter; after that, numbers are permitted. Spaces are not permitted.

The maximum number of labels allowed on the DMC-1500 series controller is 254.

Valid Labels

Label
#BEGIN
#SQUARE
#X1
#Begin1

Invalid Labels

Label	Problem
#1Square	Can not use number to begin a label
#SQUAREPEG	Can not use more than 7 characters in a label

Program Example:

Instruction	Interpretation
#START	Beginning of the Program
PR 10000,20000	Specify relative distances on X and Y axes
BG XY	Begin Motion
AM	Wait for motion complete
WT 2000	Wait 2 sec
JP #START	Jump to label START
EN	End of Program

The above program moves X and Y 10000 and 20000 units. After the motion is complete, the motors rest for 2 seconds. The cycle repeats indefinitely until the stop command is issued.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Special Labels

The DMC-1500 has some special labels, which are used to define input interrupt subroutines, limit switch subroutines, error handling subroutines, and command error subroutines. See section on "Automatic Subroutines for Monitoring Conditions" on page 95.

#AUTO	Label for auto program start
#ININT	Label for Input Interrupt subroutine
#LIMSWI	Label for Limit Switch subroutine
#POSERR	Label for excess Position Error subroutine
#MCTIME	Label for timeout on Motion Complete trip point
#CMDERR	Label for incorrect command subroutine
#COMINT	Label for communication interrupt subroutine

Commenting Programs

There are two methods for commenting programs. The first method uses the NO command and allows for comments to be embedded into Galil programs. The second method used the REM statement and requires the use of Galil software.

Using the Command, NO

The DMC-1500 provides a command, NO, for commenting programs. This command allows the user to include up to 77 characters on a single line after the NO command and can be used to include comments from the programmer as in the following example:

```
#PATH
NO 2-D CIRCULAR PATH
VMXY
NO VECTOR MOTION ON X AND Y
VS 10000
NO VECTOR SPEED IS 10000
VP -4000,0
NO BOTTOM LINE
CR 1500,270,-180
NO HALF CIRCLE MOTION
VP 0,3000
NO TOP LINE
CR 1500,90,-180
NO HALF CIRCLE MOTION
VE
NO END VECTOR SEQUENCE
BGS
NO BEGIN SEQUENCE MOTION
EN
NO END OF PROGRAM
```

Note: The NO command is an actual controller command. Therefore, inclusion of the NO commands will require process time by the controller.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using REM Statements with the Galil Terminal Software.

If you are using Galil software to communicate with the DMC-1500 controller, you may also include REM statements. 'REM' statements begin with the word 'REM' and may be followed by any comments which are on the same line. The Galil terminal software will remove these statements when the program is downloaded to the controller. For example:

```
#PATH
REM 2-D CIRCULAR PATH
VMXY
REM VECTOR MOTION ON X AND Y
VS 10000
REM VECTOR SPEED IS 10000
VP -4000,0
REM BOTTOM LINE
CR 1500,270,-180
REM HALF CIRCLE MOTION
VP 0,3000
REM TOP LINE
CR 1500,90,-180
REM HALF CIRCLE MOTION
VE
REM END VECTOR SEQUENCE
BGS
REM BEGIN SEQUENCE MOTION
EN
REM END OF PROGRAM
```

These REM statements will be removed when this program is downloaded to the controller.

Executing Programs & Multitasking

The DMC-1500 can run up to four independent programs simultaneously. These programs are called threads and are numbered 0 through 3, where 0 is the main one. Multitasking is useful for executing independent operations such as PLC functions that occur independently of motion.

The main thread differs from the others in the following ways:

1. Only the main thread may use the input command, IN.
2. When input interrupts are implemented for limit switches, position errors or command errors, the automatic subroutines, #LIMSWI, #POSERR, and #CMDERR are executed in thread 0. For more information, see section "*Automatic Subroutines for Monitoring Conditions*" on page 95.

To begin execution of the various programs, use the following instruction:

```
XQ #A, n
```

Where n indicates the thread number. If the XQ command is given with no parameters, the first program in memory will be executed in thread 0.

To halt the execution of any thread, use the instruction

```
HX n
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

where n is the thread number.

Note that both the XQ and HX commands can be performed by an executing program.

Multitasking Example: Producing Waveform on Output 1 Independent of a Move.

Instruction	Interpretation
#TASK1	Task1 label
AT0	Initialize reference time
CB1	Clear Output 1
#LOOP1	Loop1 label
AT 10	Wait 10 msec from reference time
SB1	Set Output 1
AT -40	Wait 40 msec from reference time, then initialize reference
CB1	Clear Output 1
JP #LOOP1	Repeat Loop1
#TASK2	Task2 label
XQ #TASK1,1	Execute Task1
#LOOP2	Loop2 label
PR 1000	Define relative distance
BGX	Begin motion
AMX	After motion done
WT 10	Wait 10 msec
JP #LOOP2,@IN[2]=1	Repeat motion unless Input 2 is low
HX	Halt all tasks

The program above is executed with the instruction XQ #TASK2,0 which designates TASK2 as the main thread (i.e. Thread 0). #TASK1 is executed within TASK2.

Debugging Programs

The DMC-1500 provides commands and operands which are useful in debugging application programs. These commands include interrogation commands to monitor program execution, determine the state of the controller and the contents of the controllers program, array, and variable space. Operands also contain important status information which can help to debug a program.

Trace Commands

The trace command causes the controller to send each line in a program to the host computer immediately prior to execution. Tracing is enabled with the command, TR1. TR0 turns the trace function off. Note: When the trace function is enabled, the line numbers as well as the command line will be displayed as each command line is executed.

Data which is output from the controller is stored in an output FIFO buffer. The output FIFO buffer can store up to 512 characters of information. In normal operation, the controller places output into the FIFO buffer. The software on the host computer monitors this buffer and reads information as needed. When the trace mode is enabled, the controller will send information to the FIFO buffer at a very high rate. In general, the FIFO will become full since the software is unable to read the information fast enough. When the FIFO becomes full, program execution will be delayed until it is cleared. If the user wants to avoid this delay, the command CW,1 can be given. This command

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

causes the controller to throw away the data which can not be placed into the FIFO. In this case, the controller does not delay program execution.

Error Code Command

When there is a program error, the DMC-1500 halts the program execution at the point where the error occurs. To display the last line number of program execution, issue the command, MG _ED.

The user can obtain information about the type of error condition that occurred by using the command, TC1. This command reports back a number and a text message which describes the error condition. The command, TC0 or TC, will return the error code without the text message. For more information about the command, TC, see the Command Reference.

Stop Code Command

The status of motion for each axis can be determined by using the stop code command, SC. This can be useful when motion on an axis has stopped unexpectedly. The command SC will return a number representing the motion status. See the command reference for further information. The command SC1 will return the number and the textual explanation of the motion status.

RAM Memory Interrogation Commands

For debugging the status of the program memory, array memory, or variable memory, the DMC-1500 has several useful commands. The command, DM ?, will return the number of array elements currently available. The command, DA ?, will return the number of arrays which can be currently defined. For example, a standard DMC-1500 controller will have a maximum of 8000 array elements in up to 30 arrays. If an array of 100 elements is defined, the command DM ? will return the value 7900 and the command DA ? will return 29.

To list the contents of the variable space, use the interrogation command LV (List Variables). To list the contents of array space, use the interrogation command, LA (List Arrays). To list the contents of the Program space, use the interrogation command, LS (List). To list the application program labels only, use the interrogation command, LL (List Labels).

Operands

In general, all operands provide information which may be useful in debugging an application program. Below is a list of operands which are particularly valuable for program debugging. To display the value of an operand, the message command may be used. For example, since the operand, _ED contains the last line of program execution, the command MG _ED will display this line number.

_ED contains the last line of program execution. Useful to determine where program stopped.

_DL contains the number of available labels.

_UL contains the number of available variables.

_DA contains the number of available arrays.

_DM contains the number of available array elements.

_AB contains the state of the Abort Input

_FLx contains the state of the forward limit switch for the 'x' axis

_RLx contains the state of the reverse limit switch for the 'x' axis

Debugging Example:

The following program has an error. It attempts to specify a relative movement while the X-axis is already in motion. When the program is executed, the controller stops at line 003. The user can then

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

query the controller using the command, TC1. The controller responds with the corresponding explanation:

:ED	Edit Mode
000 #A	Program Label
001 PR1000	Position Relative 1000
002 BGX	Begin
003 PR5000	Position Relative 5000
004 EN	End
<cntrl> Q	Quit Edit Mode
:XQ #A	Execute #A
?003 PR5000	Error on Line 3
:TC1	Tell Error Code
?7 Command not valid while running.	Command not valid while running
:ED 3	Edit Line 3
003 AMX;PR5000;BGX	Add After Motion Done
<cntrl> Q	Quit Edit Mode
:XQ #A	Execute #A

Debugging Programs

Commands

The DMC-1500 provides trace and error code commands which are used in debugging programs. The trace command causes the controller to send each line in a program to the host computer immediately prior to execution. Tracing is enabled with the command, TR1. TR0 turns the trace function off. Note: When the trace function is enabled, the line numbers as well as the command line will be displayed as each command line is executed.

When there is a program error, the DMC-1500 halts the program execution at the point where the error occurs. The line number is then displayed.

The user can obtain information about the type of error condition that occurred by using the command, TC1. This command reports back a number and a text message which describes the error condition. The command, TC0 or TC, will return the error code without the text message. For more information about the command, TC, see the Command Reference.

The DMC-1500 provides the capability to check the available program memory and array memory. The command, DM ?, will return the number of array elements currently available. The command, DA ?, will return the number of arrays currently available. For example, a standard DMC-1510 will have a maximum of 8000 array elements in up to 30 arrays. If an array of 100 elements is defined, the command DM ? will return the value 1500 and the command DA ? will return 13.

Operands

The operand _ED will return the value of the last line executed and can be used to determine where an error occurred. For example, the command MG _ED will display the line number in the program that failed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The operand _DL returns the number of available labels.

The operand _UL returns the number of available variables.

The operand _DA returns the number of available arrays.

The operand _DM returns the number of available array elements.

Debugging Example:

The following program has an error. It attempts to specify a relative movement while the X-axis is already in motion. When the program is executed, the controller stops at line 003. The user can then query the controller using the command, TC1. The controller responds with the corresponding explanation:

Instruction	Interpretation
:ED	Edit Mode
000 #A	Program Label
001 PR1000	Position Relative 1000
002 BGX	Begin
003 PR5000	Position Relative 5000
004 EN	End
<cntrl> Q	Quit Edit Mode
:XQ #A	Execute #A
?003 PR5000	Error on Line 3
:TC1	Tell Error Code
?7 Command not valid while running.	Command not valid while running
:ED 3	Edit Line 3
003 AMX;PR5000;BGX	Add After Motion Done
<cntrl> Q	Quit Edit Mode
:XQ #A	Execute #A

Program Flow Commands

The DMC-1500 provides instructions to control program flow. The DMC-1500 program sequencer normally executes program instructions sequentially. The program flow can be altered with the use of event triggers, trippoints, and conditional jump statements.

Event Triggers & Trippoints

To function independently from the host computer, the DMC-1500 can be programmed to make decisions based on the occurrence of an event. Such events include waiting for motion to be complete, waiting for a specified amount of time to elapse, or waiting for an input to change logic levels.

The DMC-1500 provides several event triggers that cause the program sequencer to halt until the specified event occurs. Normally, a program is automatically executed sequentially one line at a time. When an event trigger instruction is decoded, however, the actual program sequence is halted. The program sequence does not continue until the event trigger is "tripped". For example, the motion complete trigger can be used to separate two move sequences in a program. The commands for the second move sequence will not be executed until the motion is complete on the first motion sequence.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In this way, the DMC-1500 can make decisions based on its own status or external events without intervention from a host computer.

DMC-1500 Event Triggers

Command	Function
AM X Y Z W or S (A B C D E F G H)	Halts program execution until motion is complete on the specified axes or motion sequence(s). AM with no parameter tests for motion complete on all axes. This command is useful for separating motion sequences in a program.
AD X or Y or Z or W (A or B or C or D or E or F or G or H)	Halts program execution until position command has reached the specified relative distance from the start of the move. Only one axis may be specified at a time.
AR X or Y or Z or W (A or B or C or D or E or F or G or H)	Halts program execution until after specified distance from the last AR or AD command has elapsed. Only one axis may be specified at a time.
AP X or Y or Z or W (A or B or C or D or E or F or G or H)	Halts program execution until after absolute position occurs. Only one axis may be specified at a time.
MF X or Y or Z or W (A or B or C or D or E or F or G or H)	Halt program execution until after forward motion reached absolute position. Only one axis may be specified. If position is already past the point, then MF will trip immediately. Will function on geared axis.
MR X or Y or Z or W (A or B or C or D or E or F or G or H)	Halt program execution until after reverse motion reached absolute position. Only one axis may be specified. If position is already past the point, then MR will trip immediately. Will function on geared axis.
MC X or Y or Z or W (A or B or C or D or E or F or G or H)	Halt program execution until after the motion profile has been completed and the encoder has entered or passed the specified position. TW x,y,z,w sets timeout to declare an error if not in position. If timeout occurs, then the trippoint will clear and the stopcode will be set to 99. An application program will jump to label #MCTIME.
AI +/- n	Halts program execution until after specified input is at specified logic level. n specifies input line. Positive is high logic level, negative is low level. n=1 through 24.
AS X Y Z W S (A B C D E F G H)	Halts program execution until specified axis has reached its slew speed.
AT +/-n	Halts program execution until n msec from reference time. AT 0 sets reference. AT n waits n msec from reference. AT -n waits n msec from reference and sets new reference after elapsed time.
AV n	Halts program execution until specified distance along a coordinated path has occurred.
WT n	Halts program execution until specified time in msec has elapsed.

Event Trigger Examples:

Event Trigger - Multiple Move Sequence

The AM trippoint is used to separate the two PR moves. If AM is not used, the controller returns a ? for the second PR command because a new PR cannot be given until motion is complete.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Instruction	Interpretation
#TWOMOVE	Label
PR 2000	Position Command
BGX	Begin Motion
AMX	Wait for Motion Complete
PR 4000	Next Position Move
BGX	Begin 2nd move
EN	End program

Event Trigger - Set Output after Distance

Set output bit 1 after a distance of 1000 counts from the start of the move. The accuracy of the trippoint is the speed multiplied by the sample period.

Instruction	Interpretation
#SETBIT	Label
SP 10000	Speed is 10000
PA 20000	Specify Absolute position
BGX	Begin motion
AD 1000	Wait until 1000 counts
SB1	Set output bit 1
EN	End program

Event Trigger - Repetitive Position Trigger

To set the output bit every 10000 counts during a move, the AR trippoint is used as shown in the next example.

Instruction	Interpretation
#TRIP	Label
JG 50000	Specify Jog Speed
BGX;n=0	Begin Motion
#REPEAT	# Repeat Loop
AR 10000	Wait 10000 counts
TPX	Tell Position
SB1	Set output 1
WT50	Wait 50 msec
CB1	Clear output 1
n=n+1	Increment counter
JP #REPEAT,n<5	Repeat 5 times
STX	Stop
EN	End

Event Trigger - Start Motion on Input

This example waits for input 1 to go low and then starts motion. Note: The AI command actually halts execution of the program until the input occurs. If you do not want to halt the program sequences, you can use the Input Interrupt function (II) or use a conditional jump on an input, such as JP #GO,@IN[1] = -1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Instruction	Interpretation
#INPUT	Program Label
AI-1	Wait for input 1 low
PR 10000	Position command
BGX	Begin motion
EN	End program

Event Trigger - Set Output When At Speed

Instruction	Interpretation
#ATSPEED	Program Label
JG 50000	Specify jog speed
AC 10000	Acceleration rate
BGX	Begin motion
ASX	Wait for at slew speed 50000
SB1	Set output 1
EN	End program

Event Trigger - Change Speed Along Vector Path

The following program changes the feedrate or vector speed at the specified distance along the vector. The vector distance is measured from the start of the move or from the last AV command.

Instruction	Interpretation
#VECTOR	Label
VMXY;VS 5000	Coordinated path
VP 10000,20000	Vector position
VP 20000,30000	Vector position
VE	End vector
BGS	Begin sequence
AV 5000	After vector distance
VS 1000	Reduce speed
EN	End

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Event Trigger - Multiple Move with Wait

This example makes multiple relative distance moves by waiting for each to be complete before executing new moves.

Instruction	Interpretation
#MOVES	Label
PR 12000	Distance
SP 20000	Speed
AC 100000	Acceleration
BGX	Start Motion
AD 10000	Wait a distance of 10,000 counts
SP 5000	New Speed
AMX	Wait until motion is completed
WT 200	Wait 200 ms
PR -10000	New Position
SP 30000	New Speed
AC 150000	New Acceleration
BGX	Start Motion
EN	End

Example - Creating an Output Waveform Using AT

The following program causes Output 1 to be high for 10 msec and low for 40 msec. The cycle repeats every 50 msec.

Instruction	Interpretation
#OUTPUT	Program label
AT0	Initialize time reference
SB1	Set Output 1
#LOOP	Loop
AT 10	After 10 msec from reference,
CB1	Clear Output 1
AT -40	Wait 40 msec from reference and reset reference
SB1	Set Output 1
JP #LOOP	Jump to location #LOOP and continue executing commands
EN	End of program

Conditional Jumps

The DMC-1500 provides Conditional Jump (JP) and Conditional Jump to Subroutine (JS) instructions for branching to a new program location. Program execution will continue at the location specified by the JP or JS command if the conditional statement is satisfied. If no conditional statement is provided, the jump will occur automatically. See description of conditional statements below.

Conditional jumps are useful for testing events in real-time since they allow the DMC-1500 to make decisions without a host computer. For example, the DMC-1500 can begin execution at a specified label or line number based on the state of an input line.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the JP Command:

The JP command will cause the controller to execute commands at the location specified by the label or line number if the condition of the jump statement is satisfied. If no condition is specified, program execution will automatically jump to the specified line. If the condition is not satisfied, the controller will continue to execute the next commands in the program sequence.

Using the JS Command:

The JS command is significantly different from the JP command. When the condition specified by the JS command is satisfied, the controller will begin execution at the program location specified by the line or label number. However, when the controller reaches an end statement, EN, the controller will jump back to the location of the JS command and resume executing the next commands. This is known as jumping to a subroutine. For more information, see section *Subroutines*.

Each jump to a subroutine causes the controller to save the line number of the jump statement. This information is saved in an area of program memory called the program stack. The program stack can save up to 16 line numbers allowing a program to nest up to 16 jumps to subroutines. If it is necessary to remove entries from the program stack, use the command ZS. For example, while executing a subroutine, the program can be kept from jumping back to the original program line by issuing the command ZS. This will remove all entries in the program stack and continue executing at the current line. See section *Stack Manipulation on page 94* and the Command Reference in Chapter 11.

Conditional Statements

The conditional statement is satisfied if it evaluates to any value other than zero. The conditional statement can be any valid DMC-1500 numeric operand, including variables, array elements, numeric values, functions, keywords, and arithmetic expressions. If no conditional statement is given, the jump will always occur.

Examples:

Number	V1=6
Numeric Expression	V1=V7*6
	@ABS[V1]>10
Array Element	V1<Count[2]
Variable	V1<V2
Internal Variable	_TPX=0
	_TVX>500
I/O	V1>@AN[2]
	@IN[1]=0

Multiple Conditional Statements

The DMC-1500 will accept multiple conditions in a single jump statement. The conditional statements are combined in pairs using the operands “&” and “|”. The “&” operand between any two conditions, requires that both statements must be true for the combined statement to be true. The “|” operand between any two conditions, requires that only one statement be true for the combined statement to be true. *Note: Each condition must be placed in parenthesis for proper evaluation by the controller. In addition, the DMC-1500 will execute operations from left to right. For further information on Mathematical Expressions and the bit-wise operators ‘&’ and ‘|’, see pg. 7- 98.*

For example, using variables named V1, V2, V3 and V4:

JP #TEST, (V1<V2) & (V3<V4)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In this example, this statement will cause the program to jump to the label #TEST if V1 is less than V2 and V3 is less than V4. To illustrate this further, consider this same example with an additional condition:

JP #TEST, ((V1<V2) & (V3<V4)) | (V5<V6)

This statement will cause the program to jump to the label #TEST under two conditions; 1. If V1 is less than V2 and V3 is less than V4. OR 2. If V5 is less than V6.

Examples Using JP and JS

Instruction	Interpretation
JP #Loop, COUNT<10	Jump to #Loop if the variable, COUNT, is less than 10
JS #MOVE2,@IN[1]=1	Jump to subroutine #MOVE2 if input 1 is logic level high. After the subroutine MOVE2 is executed, the program sequencer returns to the main program location where the subroutine was called.
JP #BLUE,@ABS[V2]>2	Jump to #BLUE if the absolute value of variable, V2, is greater than 2
JP #C,V1*V7<=V8*V2	Jump to #C if the value of V1 times V7 is less than or equal to the value of V8*V2
JP#A	Jump to #A

Example Using JP command:

Move the X motor to absolute position 1000 counts and back to zero ten times. Wait 100 msec between moves.

Instruction	Interpretation
#BEGIN	Begin Program
COUNT=10	Initialize loop counter
#LOOP	Begin loop
PA 1000	Position absolute 1000
BGX	Begin move
AMX	Wait for motion complete
WT 100	Wait 100 msec
PA 0	Position absolute 0
BGX	Begin move
AMX	Wait for motion complete
WT 100	Wait 100 msec
COUNT=COUNT-1	Decrement loop counter
JP #LOOP,COUNT>0	Test for 10 times through loop
EN	End Program

Command Format - JP and JS

FORMAT:	DESCRIPTION
JS destination, logical condition	Jump to subroutine if logical condition is satisfied
JP destination, logical condition	Jump to location if logical condition is satisfied

The destination is a program line number or label where the program sequencer will jump if the specified condition is satisfied. Note that the line number of the first line of program memory is 0. The comma designates "IF". The logical condition tests two operands with logical operators.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Logical Operators:

OPERATOR	DESCRIPTION
<	less than
>	greater than
=	equal to
<=	less than or equal to
>=	greater than or equal to
<>	not equal

Subroutines

A subroutine is a group of instructions beginning with a label and ending with an end command (EN). Subroutines are called from the main program with the jump subroutine instruction JS, followed by a label or line number, and conditional statement. Up to 8 subroutines can be nested. After the subroutine is executed, the program sequencer returns to the program location where the subroutine was called unless the subroutine stack is manipulated as described in the following section.

Example - Using a Subroutine

Subroutine to draw a square 500 counts on each side. The square starts at vector position 1000,1000.

Instruction	Interpretation
#M	Begin main program
CB1	Clear Output Bit 1 (pick up pen)
VMXY	Specify vector motion between X and Y axes
VP 1000,1000;VE;BGS	Define vector position; move pen
AMS	Wait for after motion trippoint
SB1	Set Output Bit 1 (put down pen)
JS #Square;CB1	Jump to square subroutine
EN	End main program
#Square	Square subroutine
V1=500;JS #L	Define length of side, Jump to subroutine #L
V1=-V1;JS #L	Switch direction, Jump to subroutine #L
EN	End subroutine #Square
#L;PR V1,V1;BGX	Subroutine #L, Define relative position movement on X and Y; Begin motion
AMX;BGY;AMY	After motion on X, Begin Y, Wait for motion on Y to complete
EN	End subroutine #L

Stack Manipulation

It is possible to manipulate the subroutine stack by using the ZS command. Every time a JS instruction, interrupt or automatic routine (such as #POSERR or #LIMSWI) is executed, the subroutine stack is incremented by 1. Normally the stack is restored with an EN instruction. Occasionally it is desirable not to return back to the program line where the subroutine or interrupt was called. The ZS1 command clears 1 level of the stack. This allows the program sequencer to continue to the next line. The ZS0 command resets the stack to its initial value. For example, if a limit occurs and the #LIMSWI routine is executed, it is often desirable to restart the program sequence instead of

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

returning to the location where the limit occurred. To do this, give a ZS command at the end of the #LIMSWI routine.

Auto-Start Routine

The DMC-1500 has a special label for automatic program execution. A program which has been saved into the controllers non-volatile memory can be automatically executed upon power up or reset by beginning the program with the label #AUTO. The program must be saved into non-volatile memory using the command, BP.

Automatic Subroutines for Monitoring Conditions

Often it is desirable to monitor certain conditions continuously without tying up the host or DMC-1500 program sequences. The DMC-1500 can monitor several important conditions in the background. These conditions include checking for the occurrence of a limit switch, a defined input, position error, or a command error. Automatic monitoring is enabled by inserting a special, predefined label in the applications program. The pre-defined labels are:

SUBROUTINE	DESCRIPTION
#LIMSWI	Limit switch on any axis goes low
#ININT	Input specified by II goes low
#POSERR	Position error exceeds limit specified by ER
#MCTIME	Motion Complete timeout occurred. Timeout period set by TW command
#CMDERR	Bad command given
#COMINT	Communication Interrupt Routine

For example, the #POSERR subroutine will automatically be executed when any axis exceeds its position error limit. The commands in the #POSERR subroutine could decode which axis is in error and take the appropriate action. In another example, the #ININT label could be used to designate an input interrupt subroutine. When the specified input occurs, the program will be executed automatically.

NOTE: An application program must be running for automatic monitoring to function.

Example - Limit Switch

This program prints a message upon the occurrence of a limit switch. Note, for the #LIMSWI routine to function, the DMC-1500 must be executing an applications program from memory. This can be a very simple program that does nothing but loop on a statement, such as #LOOP;JP #LOOP;EN. Motion commands, such as JG 5000 can still be sent from the PC even while the "dummy" applications program is being executed.

Instruction	Interpretation
#LOOP	Dummy Program
JP #LOOP;EN	Jump to Loop
#LIMSWI	Limit Switch Label
MG "LIMIT OCCURRED"	Print Message
RE	Return to main program
XQ #LOOP	Execute Dummy Program
JG 5000	Jog X axis at rate of 5000 counts / sec
BGX	Begin motion on X axis

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Now, when a forward limit switch occurs on the X axis, the #LIMSWI subroutine will be executed.

NOTE: The RE command is used to return from the #LIMSWI subroutine.

NOTE: The #LIMSWI will continue to be executed until the limit switch is cleared (goes high).

Example - Position Error

Instruction	Interpretation
#LOOP	Dummy Program
JP #LOOP;EN	Loop
#POSERR	Position Error Routine
V1=_TEX	Read Position Error
MG "EXCESS POSITION ERROR"	Print Message
MG "ERROR=",V1=	Print Error
RE	Return from Error

While running the 'dummy' program, if the position error on the X axis exceeds that value specified by the ER command, the #POSERR routine will execute.

NOTE: The RE command is used to return from the #POSERR subroutine

NOTE: The #POSERR routine will continue to be executed until the position error is cleared (is less than the ER limit).

Example - Input Interrupt

Instruction	Interpretation
#A	Label
II1	Input Interrupt on 1
JG 30000,,,60000	Jog
BGXW	Begin Motion
#LOOP;JP#LOOP;EN	Loop
#ININT	Input Interrupt
STXW;AM	Stop Motion
#TEST;JP #TEST, @IN[1]=0	Test for Input 1 still low
JG 30000,,,6000	Restore Velocities
BGXW;RI	Begin motion and Return to Main Program
EN	

NOTE: Use the RI command to return from #ININT subroutine.

Example - Motion Complete Timeout

Instruction	Interpretation
#BEGIN	Begin main program
TW 1000	Set the time out to 1000 ms
PA 10000	Position Absolute command
BGX	Begin motion
MCX	Motion Complete trip point
EN	End main program
#MCTIME	Motion Complete Subroutine

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MG “X fell short”	Send out a message
EN	End subroutine without restoring trippoint

This simple program will issue the message “X fell short” if the X axis does not reach the commanded position within 1 second of the end of the profiled move.

Example - Bad Command

Instruction	Interpretation
#BEGIN	Begin main program
IN "ENTER SPEED", SPEED	Prompt for speed
JG SPEED;BGX;	Begin motion
JP #BEGIN	Repeat
EN	End main program
#CMDERR	Command error utility
JP#DONE,_ED<>2	Check if error on line 2
JP#DONE,_TC<>6	Check if out of range
MG "SPEED TOO HIGH"	Send message
MG "TRY AGAIN"	Send message
ZS1	Adjust stack
JP #BEGIN	Return to main program
#DONE	End program if other error
ZS0	Zero stack
EN1	End program and restore trippoint

The above program prompts the operator to enter a jog speed. If the operator enters a number out of range (greater than 8 million), the #CMDERR routine will be executed prompting the operator to enter a new number.

Example - Communication Interrupt

A DMC-1510 is used to move the X axis back and forth from 0 to 10000. This motion can be paused, resumed and stopped via input from an auxiliary port terminal.

Instruction	Interpretation
#BEGIN	Label for beginning of program
CC 9600,0,0,0	Setup communication configuration for auxiliary serial port
CI ,2	Setup communication interrupt for auxiliary serial port
MG {P2}"Type 0 to stop motion"	Message out of auxiliary port
MG {P2}"Type 1 to pause motion"	Message out of auxiliary port
MG {P2}"Type 2 to resume motion"	Message out of auxiliary port
RATE=2000	Variable to remember speed
SPX=RATE	Set speed of X axis motion
#LOOP	Label for Loop
PAX=10000	Move to absolute position 10000
BGX	Begin Motion on X axis
AMX	Wait for motion to be complete
PAX=0	Move to absolute position 0

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BGX	Begin Motion on X axis
AMX	Wait for motion to be complete
JP #LOOP	Continually loop to make back and forth motion
EN	End main program
#COMINT	Interrupt Routine
CI0	Clear interrupt
JP #STOP,P2CH="0"	Check for S (stop motion)
JP #PAUSE,P2CH="1"	Check for P (pause motion)
JP #RESUME,P2CH="2"	Check for R (resume motion)
EN1,1	Do nothing
#STOP	Routine for stopping motion
STX;ZS;EN	Stop motion on X axis; Zero program stack; End Program
#PAUSE	Routine for pausing motion
RATE=_SPX	Save current speed setting of X axis motion
SPX=0	Set speed of X axis to zero (allows for pause)
EN1,1	Re-enable trip-point and communication interrupt
#RESUME	Routine for resuming motion
SPX=RATE	Set speed on X axis to original speed
EN1,1	Re-enable trip-point and communication interrupt

For additional information, see section *Operator Data Entry Mode* on page 108.

Mathematical and Functional Expressions

Mathematical Expressions

For manipulation of data, the DMC-1500 provides the use of the following mathematical operators:

Operator	Function
+	Addition
-	Subtraction
*	Multiplication
/	Division
&	Logical And (Bit-wise)
|	Logical Or (On some computers, a solid vertical line appears as a broken line)
()	Parenthesis

The numeric range for addition, subtraction and multiplication operations is +/-2,147,483,647.9999. The precision for division is 1/65,000.

Mathematical operations are executed from left to right. Calculations within a parentheses have precedence.

Examples of Mathematical Expressions

SPEED=7.5*V1/2	The variable, SPEED, is equal to 7.5 multiplied by V1 and divided by 2
COUNT=COUNT+2	The variable, COUNT, is equal to the current value plus 2.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RESULT=_TPX-(@COS[45]*40)	Puts the position of X - 28.28 in RESULT. 40 * cosine of 45° is 28.28
TEMP=@IN[1]&@IN[2]	TEMP is equal to 1 only if Input 1 and Input 2 are high

Bit-Wise Operators

The mathematical operators & and | are bit-wise operators. The operator, &, is a Logical And. The operator, |, is a Logical Or. These operators allow for bit-wise operations on any valid DMC-1500 numeric operand, including variables, array elements, numeric values, functions, keywords, and arithmetic expressions. The bit-wise operators may also be used with strings.

Bit-wise operators are useful for separating characters from an input string. When using the input command for string input, the input variable holds 6 bytes of data. Each byte is eight bits, so a number represented as 32 bits of integer and 16 bits of fraction. Each ASCII character is represented as one byte (8 bits), therefore the input variable can hold a six character string. The first character of the string will be placed in the top byte of the variable and the last character will be placed in the lowest significant byte of the fraction. The characters can be individually separated by using bit-wise operations as illustrated in the following example:

Instruction	Interpretation
#TEST	Begin main program
IN "ENTER",LEN{S6}	Input character string up to 6 characters into variable ‘LEN’
FLEN=@FRAC[LEN]	Define variable ‘FLEN’ as fractional part of variable ‘LEN’
FLEN=$10000*FLEN	Shift FLEN by 32 bits (Convert fraction, FLEN, to integer)
LEN1=(FLEN&$00FF)*$1000000	Set 4th byte of FLEN = 1st byte of variable LEN1
LEN2=(FLEN&$FF00)*$10000	Set 3rd byte of FLEN = 1st byte of variable of LEN2
LEN3=(LEN&$000000FF)*$1000000	Set 1st byte of variable LEN3 = 4th byte of LEN
LEN4=(LEN&$0000FF00)*$10000	Set 1st byte of variable LEN4 = 3rd byte of LEN
LEN5=(LEN&$00FF0000)*$100	Set 1st byte of variable LEN5 = 2nd byte of LEN
LEN6=(LEN&$FF000000)	Set 1st byte of variable LEN6 = 1st byte of LEN
MG LEN6 {S1}	Display ‘LEN6’ as string message of 1 char
MG LEN5 {S1}	Display ‘LEN5’ as string message of 1 char
MG LEN4 {S1}	Display ‘LEN4’ as string message of 1 char
MG LEN3 {S1}	Display ‘LEN3’ as string message of 1 char
MG LEN2 {S1}	Display ‘LEN2’ as string message of 1 char
MG LEN1 {S1}	Display ‘LEN1’ as string message of 1 char
EN	

This program will accept a string input of up to 6 characters, parse each character, and then display each character. Notice also that the values used for masking are represented in hexadecimal (as denoted by the preceding ‘$’). For more information, see section *Sending Messages.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To illustrate further, if the user types in the string “TESTME” at the input prompt, the controller will respond with the following:

T	Response from command MG LEN6 {S1}
E	Response from command MG LEN5 {S1}
S	Response from command MG LEN4 {S1}
T	Response from command MG LEN3 {S1}
M	Response from command MG LEN2 {S1}
E	Response from command MG LEN1 {S1}

Functions

FUNCTION	DESCRIPTION
@SIN[n]	Sine of n (n in degrees with range of -32768 to 32767 and 16-bit fractional resolution)
@COS[n]	Cosine of n (n in degrees with range of -32768 to 32767 and 16-bit fractional resolution)
@COM[n]	1’s Complement of n
@ABS[n]	Absolute value of n
@FRAC[n]	Fraction portion of n
@INT[n]	Integer portion of n
@RND[n]	Round of n (Rounds up if the fractional part of n is .5 or greater)
@SQR[n]	Square root of n (Accuracy is +/-.004)
@IN[n]	Return status of digital input n
@OUT[n]	Return status of digital output n
@AN[n]	Return voltage measured at analog input n

Functions may be combined with mathematical expressions. The order of execution of mathematical expressions is from left to right and can be over-ridden by using parentheses.

Examples - Using Functions

V1=@ABS[V7]	The variable, V1, is equal to the absolute value of variable V7.
V2=5*@SIN[POS]	The variable, V2, is equal to five times the sine of the variable, POS.
V3=@IN[1]	The variable, V3, is equal to the digital value of input 1.
V4=2*(5+@AN[5])	The variable, V4, equals the value of analog input 5 plus 5, then multiplied by 2.

Variables

The maximum number of variables available with a DMC-1500 is 254. These variables can be numbers or strings. Variables are useful in applications where specific parameters, such as position or speed, must be able to change. Variables can be assigned by an operator or determined by program calculations. For example, a cut-to-length application may require that a cut length be variable.

Each variable is defined by a name which can be up to eight characters. The name must start with an alphabetic character, however, numbers are permitted in the rest of the name. Spaces are not permitted. Variable names should not be the same as DMC-1500 instructions. For example, PR is not a good choice for a variable name.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Examples - Valid Variable Names

POSX

POS1

SPEEDZ

Examples - Invalid Variable Names

Variable	Problem
REALLONGNAME	Cannot have more than 8 characters
124	Cannot begin variable name with a number
SPEED Z	Cannot have spaces in the name

Assigning Values to Variables:

Assigned values can be numbers, internal variables and keywords, functions, controller parameters and strings;

Variables hold 6 bytes of data, 4 bytes of integer (2^{31})followed by two bytes of fraction providing a range of values of +/-2,147,483,647.9999.

Numeric values can be assigned to programmable variables using the equal sign.

Any valid DMC-1500 function can be used to assign a value to a variable. For example, V1=@ABS[V2] or V2=@IN[1]. Arithmetic operations are also permitted.

To assign a string value, the string must be in quotations. String variables can contain up to six characters which must be in quotations.

Variable values may be assigned to controller parameters such as PR or SP.

Examples - Assigning values to variables

Instruction	Interpretation
POSX=_TPX	Assigns returned value from TPX command to variable POSX.
SPEED=5.75	Assigns value 5.75 to variable SPEED
INPUT=@IN[2]	Assigns logical value of input 2 to variable INPUT
V2=V1+V3*V4	Assigns the value of V1 plus V3 times V4 to the variable V2.
VAR="CAT"	Assign the string, CAT, to VAR
PR V1	Assign value of variable V1 to PR command for X axis
SP VS*2000	Assign VS*2000 to SP command

Displaying the Value of Variables at the Terminal

Variables may be sent to the screen using the format, variable=. For example, V1= , returns the value of the variable V1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example - Using Variables for Joystick

The example below reads the voltage of an X-Y joystick and assigns it to variables VX and VY to drive the motors at proportional velocities, where

10 Volts = 3000 rpm = 200000 c/sec

Speed/Analog input = 200000/10 = 20000.

Instruction	Interpretation
#JOYSTIK	Label
JG 0,0	Set in Jog mode
BGXY	Begin Motion
#LOOP	Loop
VX=@AN[1]*20000	Read joystick X
VY=@AN[2]*20000	Read joystick Y
JG VX,VY	Jog at variable VX,VY
JP#LOOP	Repeat
EN	End

Operands

Operands allow motion or status parameters of the DMC-1500 to be incorporated into programmable variables and expressions. An operand contains data and must be used in a valid expression or function. Most DMC-1500 commands have an equivalent operand - which are designated by adding an underscore (_) prior to the DMC-1500 command. Commands which have an associated operand are listed in the Command Reference as "Used as an Operand" .. Yes.

Status commands such as Tell Position return actual values, whereas action commands such as GN or SP return the values in the DMC-1500 registers. The axis designation is required following the command.

Examples of Operand Usage

POSX=_TPX	Assigns value from Tell Position X to the variable POSX.
GAIN=_GNZ*2	Assigns value from GNZ multiplied by two to variable, GAIN.
JP #LOOP,_TEX>5	Jump to #LOOP if the position error of X is greater than 5
JP #ERROR,_TC=1	Jump to #ERROR if the error code equals 1.

Operands can be used in an expression and assigned to a programmable variable, but they cannot be assigned a value. For example: _GNX=2 is invalid.

The value of an operand can be output to the computer with the message command, MG. IE. MG _TEX sends the current position error value on axis X to the computer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Special Operands (Keywords)

The DMC-1500 provides a few operands which give access to internal variables that are not accessible by standard DMC-1500 commands.

Keyword	Function
_BGn	*Is equal to a 1 if motion on axis 'n' is complete, otherwise equal to 0.
_DA	*Is equal to the number of arrays available
_DL	*Is equal to the number of available labels for programming
_DM	*Is equal to the available array memory
_HMn	*Is equal to status of Home Switch (equals 0 or 1)
_LFn	Is equal to status of Forward Limit switch input of axis 'n' (equals 0 or 1)
_LRX	Is equal to status of Reverse Limit switch input of axis 'n' (equals 0 or 1)
_UL	*Is equal to the number of available variables
TIME	Free-Running Real Time Clock (off by 2.4% - Resets with power-on). Note: TIME does not use an underscore character (_) as other keywords.

* - These keywords have corresponding commands while the keywords _LF, _LR, and TIME do not have any associated commands. All keywords are listed in the Command Summary, Chapter 11.

Examples of Keywords

Instruction	Interpretation
V1=_LFX	Assign V1 the logical state of the Forward Limit Switch on the X-axis
V3=TIME	Assign V3 the current value of the time clock
V4=_HMW	Assign V4 the logical state of the Home input on the W-axis

Arrays

For storing and collecting numerical data, the DMC-1500 provides array space for 8000 elements. The arrays are one dimensional and up to 30 different arrays may be defined. Each array element has a numeric range of 4 bytes of integer (2^{31})followed by two bytes of fraction (+/-2,147,483,647.9999).

Arrays can be used to capture real-time data, such as position, torque and analog input values. In the contouring mode, arrays are convenient for holding the points of a position trajectory in a record and playback application.

Defining Arrays

An array is defined with the command DM. The user must specify a name and the number of entries to be held in the array. An array name can contain up to eight characters, starting with an uppercase alphabetic character. The number of entries in the defined array is enclosed in [ ].

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example - Using the Command, DM

Instruction	Interpretation
DM POSX[7]	Defines an array names POSX with seven entries
DM SPEED[100]	Defines an array named speed with 100 entries
DM POSX[0]	Frees array space

Assignment of Array Entries

Like variables, each array element can be assigned a value. Assigned values can be numbers or returned values from instructions, functions and keywords.

Array elements are addressed starting at count 0. For example the first element in the POSX array (defined with the DM command, DM POSX[7]) would be specified as POSX[0].

Values are assigned to array entries using the equal sign. Assignments are made one element at a time by specifying the element number with the associated array name.

NOTE: Arrays must be defined using the command, DM, before assigning entry values.

Examples - Assigning Values to Array Entries

Instruction	Interpretation
DM SPEED[10]	Dimension Speed Array
SPEED[1]=7650.2	Assigns the first element of the array, SPEED the value 7650.2
SPEED[1]=	Returns array element value
POSX[10]=_TPX	Assigns the 10th element of the array POSX the returned value from the tell position command.
CON[2]=@COS[POS]*2	Assigns the second element of the array CON the cosine of the variable POS multiplied by 2.
TIMER[1]=TIME	Assigns the first element of the array timer the returned value of the TIME keyword.

Using a Variable to Address Array Elements

An array element number can also be a variable. This allows array entries to be assigned sequentially using a counter. For example;

Instruction	Interpretation
#A	Begin Program
COUNT=0;DM POS[10]	Initialize counter and define array
#LOOP	Begin loop
WT 10	Wait 10 msec
POS[COUNT]=_TPX	Record position into array element
POS[COUNT]=	Report position
COUNT=COUNT+1	Increment counter
JP #LOOP,COUNT<10	Loop until 10 elements have been stored
EN	End Program

The above example records 10 position values at a rate of one value per 10 msec. The values are stored in an array named POS. The variable, COUNT, is used to increment the array element counter. The above example can also be executed with the automatic data capture feature described below.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Uploading and Downloading Arrays to On Board Memory

Arrays may be uploaded and downloaded using the QU and QD commands.

QU array[],start,end,delim

QD array[],start,end

where array is an array name such as A[].

Start is the first element of array (default=0)

End is the last element of array (default=last element)

Delim specifies whether the array data is separated by a comma (delim=1) or a carriage return (delim=0).

The file is terminated using <control>Z, <control>Q, <control>D or \.

Automatic Data Capture into Arrays

The DMC-1500 provides a special feature for automatic capture of data such as position, position error, inputs or torque. This is useful for teaching motion trajectories or observing system performance. Up to eight types of data can be captured and stored in eight arrays. The capture rate or time interval may be specified. Recording can done as a one time event or as a circular continuous recording.

Command Summary - Automatic Data Capture

Command	Description
RA n[],m[],o[],p[]	Selects up to eight arrays for data capture. The arrays must be defined with the DM command.
RD type1,type2,type3,type4	Selects the type of data to be recorded, where type1, type2, type3, and type 4 represent the various types of data (see table below). The order of data type is important and corresponds with the order of n,m,o,p arrays in the RA command.
RC n,m	The RC command begins data collection. Sets data capture time interval where n is an integer between 1 and 8 and designates 2n msec between data. m is optional and specifies the number of elements to be captured. If m is not defined, the number of elements defaults to the smallest array defined by DM. When m is a negative number, the recording is done continuously in a circular manner. _RD is the recording pointer and indicates the address of the next array element. n=0 stops recording.
RC?	Returns a 0 or 1 where, 0 denotes not recording, 1 specifies recording in progress

Data Types for Recording:

Data Type	Description
_DEX	2nd encoder position (dual encoder)
_TPX	Encoder position
_TEX	Position error
_SHX	Commanded position
_RLX	Latched position
_TI	Inputs
_OP	Output

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

_TSX	Switches (only bit 0-4 valid)
_SCX	Stop code
_NOX	Status bits
_TTX	Torque (reports digital value +/-8097)

Note: X may be replaced by Y,Z or W for capturing data on other axes, or A,B,C,D,E,F,G,H for DMC-1580.

Operand Summary - Automatic Data Capture

Operand	Description
_RC	Returns a 0 or 1 where, 0 denotes not recording, 1 specifies recording in progress
_RD	Returns address of next array element.

Example - Recording into An Array

During a position move, store the X and Y positions and position error every 2 msec.

Instruction	Interpretation
#RECORD	Begin program
DM XPOS[300],YPOS[300]	Define X,Y position arrays
DM XERR[300],YERR[300]	Define X,Y error arrays
RA XPOS[],XERR[],YPOS[],YERR[]	Select arrays for capture
RD _TPX,_TEX,_TPY,_TEY	Select data types
PR 10000,20000	Specify move distance
RC1	Start recording now, at rate of 2 msec
BG XY	Begin motion
#A;JP #A,RC=1	Loop until done
MG "DONE"	Print message
EN	End program
#PLAY	Play back
N=0	Initial Counter
JP# DONE,N>300	Exit if done
N=	Print Counter
X POS[N]=	Print X position
Y POS[N]=	Print Y position
XERR[N]=	Print X error
YERR[N]=	Print Y error
N=N+1	Increment Counter
#DONE	Done

Deallocating Array Space

Array space may be deallocated using the DA command followed by the array name. DA*[0] deallocates all the arrays.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Input of Data (Numeric and String)

Input of Data

The command, IN, is used to prompt the user to input numeric or string data. Using the IN command, the user may specify a message prompt by placing a message in quotations. When the controller executes an IN command, the controller will wait for the input of data. The input data is assigned to the specified variable or array element.

Example - Inputting Numeric Data

Instruction	Interpretation
#A	Program label
IN "Enter Length", LENX	Use input command, IN, to query the user
EN	End the program

In this example, the message “Enter Length” is displayed on the computer screen. The controller waits for the operator to enter a value. The operator enters the numeric value which is assigned to the variable, LENX.

Cut-to-Length Example

In this example, a length of material is to be advanced a specified distance. When the motion is complete, a cutting head is activated to cut the material. The length is variable, and the operator is prompted to input it in inches. Motion starts with a start button which is connected to input 1.

The load is coupled with a 2 pitch lead screw. A 2000 count/rev encoder is on the motor, resulting in a resolution of 4000 counts/inch. The program below uses the variable LEN, to length. The IN command is used to prompt the operator to enter the length, and the entered value is assigned to the variable LEN.

Instruction	Interpretation
#BEGIN	LABEL
AC 800000	Acceleration
DC 800000	Deceleration
SP 5000	Speed
LEN=3.4	Initial length in inches
#CUT	Cut routine
AI1	Wait for start signal
IN "enter Length(IN)", LEN	Prompt operator for length in inches
PR LEN *4000	Specify position in counts
BGX	Begin motion to move material
AMX	Wait for motion done
SB1	Set output to cut
WT100;CB1	Wait 100 msec, then turn off cutter
JP #CUT	Repeat process
EN	End program

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Operator Data Entry Mode

The Operator Data Entry Mode permits data to be entered at anytime. In this mode, the input will not be interpreted as DMC commands and input such as ST or JG will not be recognized as commands. In this mode, the DMC-1500 provides a buffer for receiving characters. This mode may only be used when executing an applications program.

The Operator Data Entry Mode may be specified for either Port 1 or Port 2 or both. The mode may be exited with the \ or <escape> key.

NOTE: This is not used for high rate data transfer.

For Port 1:	Use the third field of the CI command to set the Data Mode. A 1 specifies Operator Data Mode, a 0 disables the Data Mode.
For Port 2:	Use the third field of the CC command to set the Data Mode. A 0 configures P2 as a general port for the Operator Data Mode.

To capture and decode characters in the Operator Data Mode, the DMC-1500 provides four special keywords for Port 1 (P1) and Port 2 (P2).

Port 1 (Main) Keyword	Port 2 (Aux) Keyword	Function
P1CH	P2CH	Contains the last character received
P1ST	P2ST	Contains the received string
P1NM	P2NM	Contains the received number
P1CD	P2CD	Contains the status code: -1 mode disabled 0 nothing received 1 received character, but not <enter> 2 received string, not a number 3 received number

Note: The value of P1CD and P2CD returns to zero after the corresponding string or number is read.

These keywords may be used in an applications program to decode data. They may be used in conditional statements with logical operators.

Examples:

JP #LOOP,P2CD< >3	Checks to see if status code is 3 (number received)
JP #P,P1CH="V"	Checks if last character received was a V
PR P2NM	Assigns received number to position
JS #XAXIS,P1ST="X"	Checks to see if received string is X

Using Communication Interrupt

The DMC-1500 provides a special interrupt for communication allowing the application program to be interrupted by input from the user. The interrupt is enabled using the CI command. The syntax for the command is CI m,n,o:

m=0	Don't interrupt Port 1
1	Interrupt on <enter> Port 1
2	Interrupt on any character Port 1
-1	Clear any characters in buffer

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

n=0	Don't interrupt Port 2
1	Interrupt on <enter> Port 2
2	Interrupt on any character Port 2
-1	Clear any characters in buffer

o=0	Disable operator data mode for P1
1	Enable operator data mode for P1

The #COMINT label is used for the communication interrupt. For example, the DMC-1500 can be configured to interrupt on any character received on Port 2. The #COMINT subroutine is entered when a character is received and the subroutine can decode the characters. At the end of the routine the EN command is used. EN,1 will re-enable the interrupt and return to the line of the program where the interrupt was called, EN will just return to the line of the program where it was called without re-enabling the interrupt. As with any automatic subroutine, a program must be running in thread 0 at all times for it to be enabled.

Example - Using the #COMINT Routine:

A DMC-1520 is used to jog the X and Y axis. This program automatically begins upon power-up and allows the user to input values from the main serial port terminal. The speed of either axis may be changed during motion by specifying the axis letter followed by the new speed value. An S stops motion on both axes.

Command	Interpretation
#AUTO	Label for Auto Execute
SPEEDX=10000	Initial X speed
SPEEDY=10000	Initial Y speed
CI 2,,1	Set Port 1 for Character Interrupt
JG SPEEDX,SPEEDY	Specify jog mode speed for X and Y axis
BGXY	Begin motion
#PRINT	Routine to print message to terminal
MG "TO CHANGE SPEEDS"	Print message
MG "TYPE X OR Y"	
MG "TYPE S TO STOP"	
#JOGLOOP	Loop to change Jog speeds
JG SPEEDX,SPEEDY	Set new jog speed
JP #JOGLOOP	
EN	End of main program
#COMINT	Interrupt routine
CI0	Clear interrupt
JP #A,P1CH="X"	Check for X
JP #B,P1CH="Y"	Check for Y
JP #C,P1CH="S"	Check for S
ZS1;CI2;JP#JOGLOOP	Jump if not X,Y,S
#A;JS#NUM	
SPEEDX=VAL	New X speed

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
ZS1;CI2;JP#PRINT                    Jump to Print
#B;JS#NUM
SPEEDY=VAL                          New Y speed
ZS1;CI2;JP#PRINT                    Jump to Print
#C;ST;AMX;CI-1                      Stop motion on S
MG{^8}, "THE END"
ZS;EN,1                             End-Re-enable interrupt
#NUM                                Routine for entering new jog speed
MG "ENTER",PICH{S},"AXIS SPEED" {N} Prompt for value
#NUMLOOP; CI-1                      Check for enter
#NMLP                               Routine to check input from terminal
JP #NMLP,P1CD<2                     Jump to error if string
JP #ERROR,P1CD=2                    Read value
VAL=P1NM
EN                                  End subroutine
#ERROR;CI-1                         Error Routine
MG "INVALID-TRY AGAIN"              Error message
JP #NMLP
EN                                  End
```

Inputting String Variables

String variables with up to six characters may be input using the specifier, {Sn} where n represents the number of string characters to be input. If n is not specified, six characters will be accepted. For example, IN "Enter X,Y or Z", V{S} specifies a string variable to be input.

The DMC-1500, stores all variables as 6 bytes of information. When a variable is specified as a number, the value of the variable is represented as 4 bytes of integer and 2 bytes of fraction. When a variable is specified as a string, the variable can hold up to 6 characters (each ASCII character is 1 byte). When using the IN command for string input, the first input character will be placed in the top byte of the variable and the last character will be placed in the lowest significant byte of the fraction. The characters can be individually separated by using bit-wise operations, see section *Bit-Wise Operators*.

Output of Data (Numeric and String)

Numerical and string data can be output from the controller using several methods. The message command, MG, can output string and numerical data. Also, the controller can be commanded to return the values of variables and arrays, as well as other information using the interrogation commands (the interrogation commands are described in chapter 5).

Sending Messages

Messages may be sent out of the serial ports using the message command, MG. This command sends specified text and numerical or string data in ASCII format

Text strings are specified in quotes and variable or array data is designated by the name of the variable or array. For example:

```
MG "The Final Value is", RESULT
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In addition to variables, functions and commands, responses can be used in the message command. For example:

```
MG "Analog input is", @AN[1]
MG "The Gain of X is", _GNX
```

Specifying the Serial Port for Messages:

By default, messages will be sent through port 1, the main serial port. The serial port can be specified with the specifier, {P1} for the main serial port and {P2} for auxiliary serial port.

Example - Sending Message Out of Auxiliary Port:

```
MG {P2} "Hello World"
```

Formatting Messages

String variables can be formatted using the specifier, {Sn} where n is the number of characters, 1 through 6. For example:

```
MG STR {S3}
```

This statement returns 3 characters of the string variable named STR.

Numeric data may be formatted using the {Fn.m} expression following the completed MG statement. {$n.m} formats data in HEX instead of decimal. The actual numerical value will be formatted with n characters to the left of the decimal and m characters to the right of the decimal. Leading zeros will be used to display specified format.

For example::

```
MG "The Final Value is", RESULT {F5.2}
```

If the value of the variable RESULT is equal to 4.1, this statement returns the following:

```
The Final Value is 00004.10
```

If the value of the variable RESULT is equal to 999999.999, the above message statement returns the following:

```
The Final Value is 99999.99
```

The message command normally sends a carriage return and line feed following the statement. The carriage return and the line feed may be suppressed by sending {N} at the end of the statement. This is useful when a text string needs to surround a numeric value.

Example:

```
#A
JG 50000;BGX;ASX
MG "The Speed is", _TVX {F5.1} {N}
MG "counts/sec"
EN
```

When #A is executed, the above example will appear on the screen as:

```
The speed is 50000 counts/sec
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the MG Command to Configure Terminals

The MG command can be used to configure a terminal. Any ASCII character can be sent by using the format {^n} where n is any integer between 1 and 255.

Example:

```
MG {^07} {^255}
```

sends the ASCII characters represented by 7 and 255 to the bus.

Summary of Message Functions:

Function	Description
" "	Surrounds text string
{Fn.m}	Formats numeric values in decimal n digits to the right of the decimal point and m digits to the left
{N}	Suppresses carriage return/line feed
{P1} or {P2}	Send message to Main Port or Auxiliary Port
{Sn}	Sends the first n characters of a string variable, where n is 1 through 6.
{$n.m}	Formats numeric values in hexadecimal
{^n}	Sends ASCII character specified by integer n

Displaying Variables and Arrays

Variables and arrays may be sent to the screen using the format, VARIABLE= **or** ARRAY[X]=. For example, V1= , returns the value of V1. These values may also be displayed using the message command, MG. If a variable was not previously defined, using the command, VARIABLE=, will cause the variable to be defined and the controller will not return an error. If the MG command is used to display a variable which has not been defined, the controller will return an error.

Example - Printing a Variable and an Array element

Instruction	Interpretation
#DISPLAY	Label
DM POSX[7]	Define Array POSX with 7 entries
PR 1000	Position Command
BGX	Begin
AMX	After Motion
V1=_TPX	Assign Variable V1
POSX[1]=_TPX	Assign the first entry
V1=	Print V1

Interrogation Commands

The DMC-1700 has a set of commands that directly interrogate the controller. When these command are entered, the requested data is returned in decimal format on the next line followed by a carriage return and line feed. The format of the returned data can be changed using the Position Format (PF), and Leading Zeros (LZ) command. For a complete description of interrogation commands, see chapter 5.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the PF Command to Format Response from Interrogation Commands

The command, PF, can change format of the values returned by theses interrogation commands:

BL ?	LE ?
DE ?	PA ?
DP ?	PR ?
EM ?	TN ?
FL ?	VE ?
IP ?	TE
TP	

The numeric values may be formatted in decimal or hexadecimal* with a specified number of digits to the right and left of the decimal point using the PF command.

Position Format is specified by:

PF m.n

where m is the number of digits to the left of the decimal point (0 thru 10) and n is the number of digits to the right of the decimal point (0 thru 4) A negative sign for m specifies hexadecimal format.

Hex values are returned preceded by a $ and in 2's complement. Hex values should be input as signed 2's complement, where negative numbers have a negative sign. The default format is PF 10.0.

If the number of decimal places specified by PF is less than the actual value, a nine appears in all the decimal places.

Examples:

:DP21	Define position
:TPX	Tell position
0000000021	Default format
:PF4	Change format to 4 places
:TPX	Tell position
0021	New format
:PF-4	Change to hexadecimal format
:TPX	Tell Position
$0015	Hexadecimal value
:PF2	Format 2 places
:TPX	Tell Position
99	Returns 99 if position greater than 99

Removing Leading Zeros from Response to Interrogation Response

The leading zeros on data returned as a response to interrogation commands can be removed by the use of the command, LZ.

Example - Using the LZ command

LZ0	Disables the LZ function
TP	Tell Position Interrogation Command
-0000000009, 0000000005, 0000000000, 0000000007	Response from Interrogation Command (With Leading Zeros)
LZ1	Enables the LZ function

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TP	Tell Position Interrogation Command
-9, 5, 0, 7	Response from Interrogation Command (Without Leading Zeros)

Local Formatting of Response of Interrogation Commands

The response of interrogation commands may be formatted locally. To format locally, use the command, {Fn.m} or {$n.m} on the same line as the interrogation command. The symbol F specifies that the response should be returned in decimal format and $ specifies hexadecimal. n is the number of digits to the left of the decimal, and m is the number of digits to the right of the decimal. For example:

Examples:

TP {F2.2}	Tell Position in decimal format 2.2
-05.00, 05.00, 00.00, 07.00	Response from Interrogation Command
TP {$4.2}	Tell Position in hexadecimal format 4.2
FFFB.00,$0005.00,$0000.00,$0007.00	Response from Interrogation Command

Formatting Variables and Array Elements

The Variable Format (VF) command is used to format variables and array elements. The VF command is specified by:

VF m.n

where m is the number of digits to the left of the decimal point (0 through 10) and n is the number of digits to the right of the decimal point (0 through 4).

A negative sign for m specifies hexadecimal format. The default format for VF is VF 10.4 Hex values are returned preceded by a $ and in 2's complement.

Instruction	Interpretation
V1=10	Assign V1
V1=	Return V1
0000000010.0000	Response from controller with default format
VF2.2	Change format
V1=	Return V1
10.00	Response from controller with new format
VF-2.2	Specify hex format
V1=	Return V1
$0A.00	Response from controller in hexadecimal format
VF1	Change format
V1=	Return V1
9	Response from controller - returns 9 if value greater than 9

Local Formatting of Variables

PF and VF commands are global format commands that effect the format of all relevant returned values and variables. Variables may also be formatted locally. To format locally, use the command, {Fn.m} or {$n.m} following the variable name and the '=' symbol. F specifies decimal and $ specifies hexadecimal. n is the number of digits to the left of the decimal, and m is the number of digits to the right of the decimal. For example:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Instruction	Interpretation
V1=10	Assign V1
V1=	Return V1
0000000010.0000	Response from controller with default format
V1={F4.2}	Specify local format
0010.00	Response from controller with new format
V1={$4.2}	Specify hex format
$000A.00	Response from controller in hexadecimal format
V1="ALPHA"	Assign string "ALPHA" to V1
V1={S4}	Specify string format first 4 characters
ALPH	Response from controller in string format

The local format is also used with the MG command.

Converting to User Units

Variables and arithmetic operations make it easy to input data in desired user units such as inches or RPM.

The DMC-1500 position parameters such as PR, PA and VP have units of quadrature counts. Speed parameters such as SP, JG and VS have units of counts/sec. Acceleration parameters such as AC, DC, VA and VD have units of counts/sec^2. The controller interprets time in milliseconds.

All input parameters must be converted into these units. For example, an operator can be prompted to input a number in revolutions. A program could be used such that the input number is converted into counts by multiplying it by the number of counts/revolution.

Example - Converting to User Units

Instruction	Interpretation
#RUN	Label
IN "ENTER # OF REVOLUTIONS",N1	Prompt for revs
PR N1*2000	Convert to counts
IN "ENTER SPEED IN RPM",S1	Prompt for RPMs
SP S1*2000/60	Convert to counts/sec
IN "ENTER ACCEL IN RAD/SEC2",A1	Prompt for ACCEL
AC A1*2000/(2*3.14)	Convert to counts/sec2
BG	Begin motion
EN	End program

Programmable Hardware I/O

Digital Outputs

The DMC-1500 has an 8-bit uncommitted output port for controlling external events. The DMC-1580 has an additional eight output bits available at JD5 pins 10-17. Each bit on the output port may be set and cleared with the software instructions SB (Set Bit) and CB(Clear Bit), or OB (define output bit).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example - Using Set Bit and Clear Bit Commands (SB, CB)

Instruction	Interpretation
SB6	Sets bit 6 of output port
CB4	Clears bit 4 of output port
CB9	Clear bit 9 of output port on DMC-1580

The Output Bit (OB) instruction is useful for setting or clearing outputs depending on the value of a variable, array, input or expression. Any non-zero value results in a set bit.

Example - Using the Output Bit Command (OB)

Instruction	Interpretation
OB1, POS	Set Output 1 if the variable POS is non-zero. Clear Output 1 if POS equals 0.
OB 2, @IN [1]	Set Output 2 if Input 1 is high. If Input 1 is low, clear Output 2.
OB 3, @IN [1]&@IN [2]	Set Output 3 only if Input 1 and Input 2 are high.
OB 4, COUNT [1]	Set Output 4 if element 1 in the array COUNT is non-zero.

The output port can be set by specifying an 8-bit word using the instruction OP (Output Port). This instruction allows a single command to define the state of the entire 8-bit output port, where 2^0 is output 1, 2^1 is output 2 and so on. A 1 designates that the output is on.

Example - Using the Output Port Command (OP)

Instruction	Interpretation
OP6	Sets outputs 2 and 3 of output port to high. All other bits are 0. ($2^1 + 2^2 = 6$)
OP0	Clears all bits of output port to zero
OP 255	Sets all bits of output port to one. ($2^2 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 + 2^7$)

Example - Using OP to Turn on Output After Move

Instruction	Interpretation
#OUTPUT	Label
PR 2000	Position Command
BG	Begin
AM	After move
SB1	Set Output 1
WT 1000	Wait 1000 msec
CB1	Clear Output 1
EN	End

Digital Inputs

The DMC-1500 has eight digital inputs for controlling motion by local switches. The @IN[n] function returns the logic level of the specified input 1 through 8. For example, a Jump on Condition instruction can be used to execute a sequence if a high condition is noted on an input 3. To halt program execution, the After Input (AI) instruction waits until the specified input has occurred.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example - Using Inputs to control program flow:

Instruction	Interpretation
JP #A,@IN[1]=0	Jump to A if input 1 is low
JP #B,@IN[2]=1	Jump to B if input 2 is high
AI 7	Wait until input 7 is high
AI -6	Wait until input 6 is low

Example - Start Motion on Switch

Motor X must turn at 4000 counts/sec when the user flips a panel switch to on. When panel switch is turned to off position, motor X must stop turning.

Solution: Connect panel switch to input 1 of DMC-1500. High on input 1 means switch is in on position.

Instruction	Interpretation
#S;JG 4000	Set speed
AI 1;BGX	Begin after input 1 goes high
AI -1;STX	Stop after input 1 goes low
AMX;JP #S	After motion, repeat
EN;	

Input Interrupt Function

The DMC-1500 provides an input interrupt function which causes the program to automatically execute the instructions following the #ININT label. This function is enabled using the II m,n,o command. The m specifies the beginning input and n specifies the final input in the range. The parameter o is an interrupt mask. If m and n are unused, o contains a number with the mask. A 1 designates that input to be enabled for an interrupt, where 2^0 is bit 1, 2^1 is bit 2 and so on. For example, II,,5 enables inputs 1 and 3 ($2^0 + 2^2 = 5$).

A low input on any of the specified inputs will cause automatic execution of the #ININT subroutine. The Return from Interrupt (RI) command is used to return from this subroutine to the place in the program where the interrupt had occurred. If it is desired to return to somewhere else in the program after the execution of the #ININT subroutine, the Zero Stack (ZS) command is used followed by unconditional jump statements.

IMPORTANT: Use the RI instruction (not EN) to return from the #ININT subroutine.

Examples - Input Interrupt

Instruction	Interpretation
#A	Label #A
II 1	Enable input 1 for interrupt function
JG 30000,-20000	Set speeds on X and Y axes
BG XY	Begin motion on X and Y axes
#B	Label #B
TP XY	Report X and Y axes positions
WT 1000	Wait 1000 milliseconds
JP #B	Jump to #B
EN	End of program
#ININT	Interrupt subroutine

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MG "Interrupt occurred"	Display message
ST XY	Stops motion on X and Y axes
#LOOP	Loop until Interrupt cleared
JP #LOOP,@IN[1]=0	
JG 15000,10000	Specify new speeds
WT 300	Wait 300 milliseconds
BG XY	Begin motion on X and Y axes
RI	Return from Interrupt subroutine

Analog Inputs

The DMC-1500 provides seven analog inputs. The value of these inputs in volts may be read using the @AN[n] function where n is the analog input 1 through 7. The resolution of the Analog-to-Digital conversion is 12 bits. This resolution can be increased to 16 bits by specifying the -16 option when purchasing the DMC-1500 controller. Analog inputs are useful for reading special sensors such as temperature, tension or pressure.

The following examples show programs which cause the motor to follow an analog signal. The first example is a point-to-point move. The second example shows a continuous move.

Example - Position Follower (Point-to-Point)

Objective - The motor must follow an analog signal. When the analog signal varies by 10V, motor must move 10000 counts.

Method: Read the analog input and command X to move to that point.

Instruction	Interpretation
#Points	Label
SP 7000	Speed
AC 80000;DC 80000	Acceleration
#Loop	
VP=@AN[1]*1000	Read and analog input, compute position
PA VP	Command position
BGX	Start motion
AMX	After completion
JP #Loop	Repeat
EN	End

Example - Position Follower (Continuous Move)

Method: Read the analog input, compute the commanded position and the position error. Command the motor to run at a speed in proportions to the position error.

Instruction	Interpretation
#Cont	Label
AC 80000;DC 80000	Acceleration rate
JG 0	Start job mode
BGX	Start motion
#Loop	
VP=@AN[1]*1000	Compute desired position

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VE=VP-_TPX	Find position error
VEL=VE*20	Compute velocity
JG VEL	Change velocity
JP #Loop	Change velocity
EN	End

Example Applications

Wire Cutter

An operator activates a start switch. This causes a motor to advance the wire a distance of 10". When the motion stops, the controller generates an output signal which activates the cutter. Allowing 100 ms for the cutting completes the cycle.

Suppose that the motor drives the wire by a roller with a 2" diameter. Also assume that the encoder resolution is 1000 lines per revolution. Since the circumference of the roller equals 2π inches, and it corresponds to 4000 quadrature, one inch of travel equals:

$$4000/2\pi = 637 \text{ count/inch}$$

This implies that a distance of 10 inches equals 6370 counts, and a slew speed of 5 inches per second, for example, equals 3185 count/sec.

The input signal may be applied to input 1, for example, and the output signal is chosen as output 1. The motor velocity profile and the related input and output signals are shown in Fig. 7.1.

The program starts at a state that we define as #A. Here the controller waits for the input pulse on I1. As soon as the pulse is given, the controller starts the forward motion.

Upon completion of the forward move, the controller outputs a pulse for 20 ms and then waits an additional 80 ms before returning to #A for a new cycle.

Instruction	Function
#A	Label
AI1	Wait for input 1
PR 6370	Distance
SP 3185	Speed
BGX	Start Motion
AMX	After motion is complete
SB1	Set output bit 1
WT 20	Wait 20 ms
CB1	Clear output bit 1
WT 80	Wait 80 ms
JP #A	Repeat the process

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

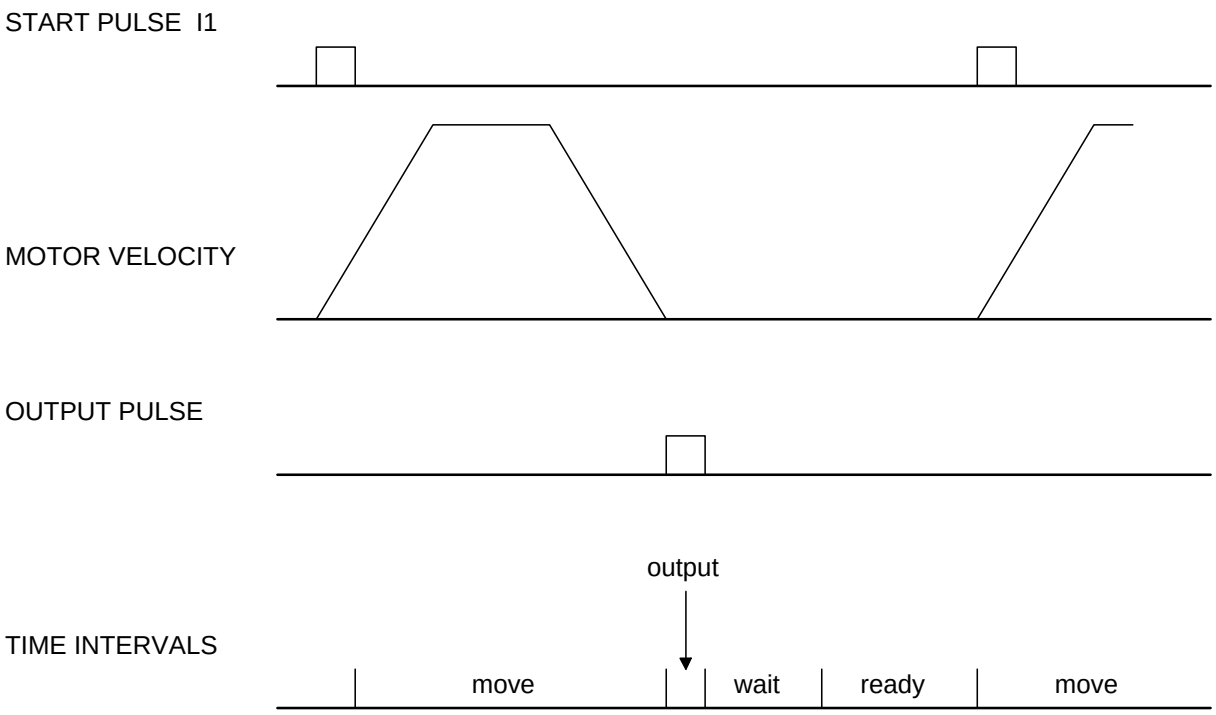


Figure 7.1 - Motor Velocity and the Associated input/output signals

X-Y Table Controller

An X-Y-Z system must cut the pattern shown in Fig. 7.2. The X-Y table moves the plate while the Z-axis raises and lowers the cutting tool.

The solid curves in Fig. 7.2 indicate sections where cutting takes place. Those must be performed at a feedrate of 1 inch per second. The dashed line corresponds to non-cutting moves and should be performed at 5 inch per second. The acceleration rate is 0.1 g.

The motion starts at point A, with the Z-axis raised. An X-Y motion to point B is followed by lowering the Z-axis and performing a cut along the circle. Once the circular motion is completed, the Z-axis is raised and the motion continues to point C, etc.

Assume that all of the 3 axes are driven by lead screws with 10 turns-per-inch pitch. Also assume encoder resolution of 1000 lines per revolution. This results in the relationship:

1 inch = 40,000 counts

and the speeds of

1 in/sec = 40,000 count/sec

5 in/sec = 200,000 count/sec

an acceleration rate of 0.1g equals

0.1g = 38.6 in/s2 = 1,544,000 count/s^2

Note that the circular path has a radius of 2" or 80000 counts, and the motion starts at the angle of 270° and traverses 360° in the CW (negative direction). Such a path is specified with the instruction

CR 80000,270,-360

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Further assume that the Z must move 2" at a linear speed of 2" per second. The required motion is performed by the following instructions:

Instruction	Interpretation
#A	Label
VM XY	Circular interpolation for XY
VP 160000,160000	Positions
VE	End Vector Motion
VS 200000	Vector Speed
VA 1544000	Vector Acceleration
BGS	Start Motion
AMS	When motion is complete
PR,,-80000	Move Z down
SP,,80000	Z speed
BGZ	Start Z motion
AMZ	Wait for completion of Z motion
CR 80000,270,-360	Circle
VE	
VS 40000	Feedrate
BGS	Start circular move
AMS	Wait for completion
PR,,80000	Move Z up
BGZ	Start Z move
AMZ	Wait for Z completion
PR -21600	Move X
SP 20000	Speed X
BGX	Start X
AMX	Wait for X completion
PR,,-80000	Lower Z
BGZ	
AMZ	
CR 80000,270,-360	Z second circle move
VE	
VS 40000	
BGS	
AMS	
PR,,80000	Raise Z
BGZ	
AMZ	
VP -37600,-16000	Return XY to start
VE	
VS 200000	
BGS	
AMS	
EN	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

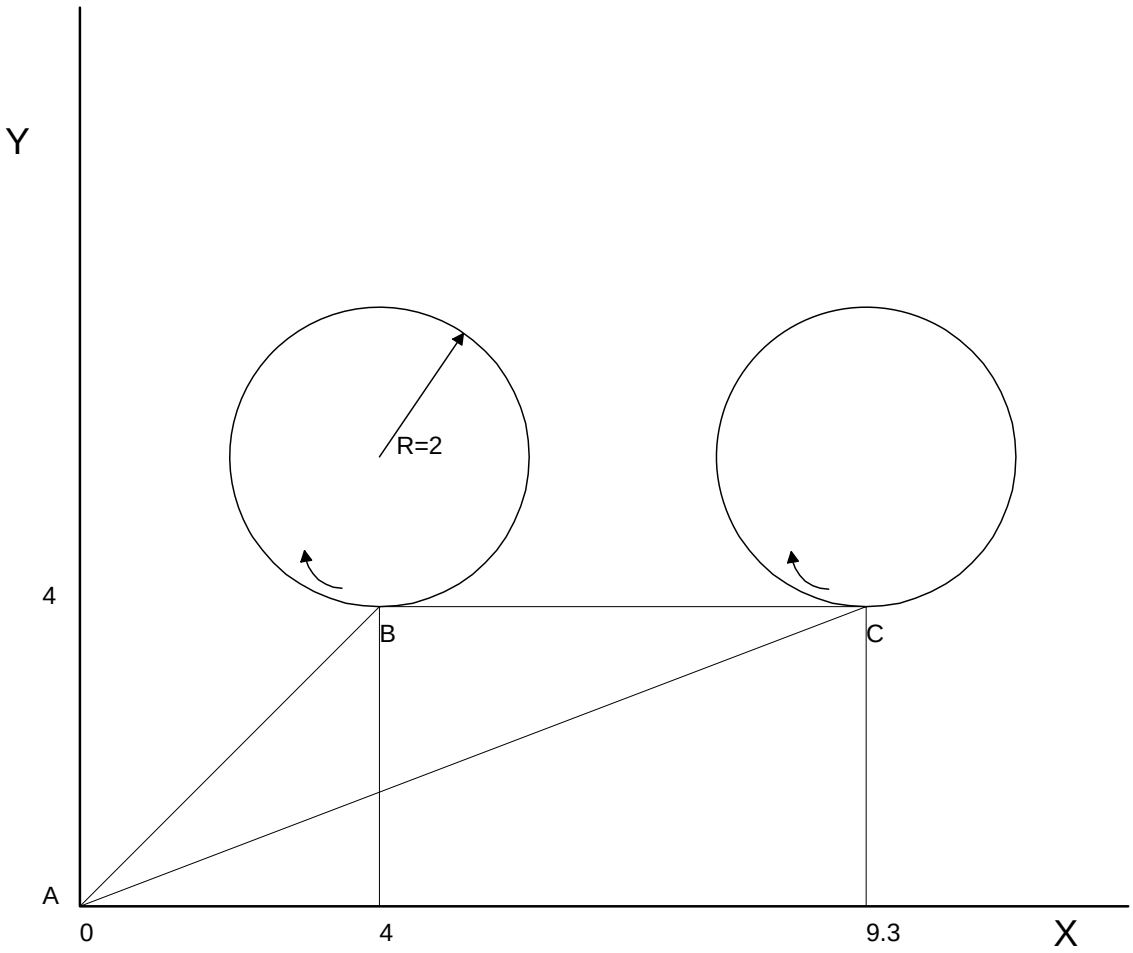


Figure 7.2 - Motor Velocity and the Associated input/output signals

Speed Control by Joystick

The speed of a motor is controlled by a joystick. The joystick produces a signal in the range between -10V and +10V. The objective is to drive the motor at a speed proportional to the input voltage.

Assume that a full voltage of 10 Volts must produce a motor speed of 3000 rpm with an encoder resolution of 1000 lines or 4000 count/rev. This speed equals:

3000 rpm = 50 rev/sec = 200000 count/sec

The program reads the input voltage periodically and assigns its value to the variable VIN. To get a speed of 200,000 cts/sec for 10 volts, we select the speed as

Speed = 20000 x VIN

The corresponding velocity for the motor is assigned to the VEL variable.

Instruction	Interpretation
#A	Label
JG0	Set jog speed of zero
BGX	Begin jogging (at speed zero)
#B	Label
VIN=@AN[1]	Set variable, VIN, to value of analog input 1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VEL=VIN*20000	Set variable, VEL to multiple of variable of VIN
JG VEL	Update jog speed to value of variable VEL
JP #B	Loop back to label, #B
EN	End

Position Control by Joystick

This system requires the position of the motor to be proportional to the joystick angle. Furthermore, the ratio between the two positions must be programmable. For example, if the control ratio is 5:1, it implies that when the joystick voltage is 5 Volts, corresponding to 1028 counts, the required motor position must be 5120 counts. The variable V3 changes the position ratio.

Instruction	Interpretation
#A	Label
V3=5	Initial position ratio
DP0	Define the starting position
JG0	Set motor in jog mode as zero
BGX	Start
#B	
V1=@AN[1]	Read analog input
V2=V1*V3	Compute the desired position
V4=V2-_TPX-_TEX	Find the following error
V5=V4*20	Compute a proportional speed
JG V5	Change the speed
JP #B	Repeat the process
EN	End

Backlash Compensation by Sampled Dual-Loop

The continuous dual loop, enabled by the DV1 function is an effective way to compensate for backlash. In some cases, however, when the backlash magnitude is large, it may be difficult to stabilize the system. In those cases, it may be easier to use the sampled dual loop method described below.

This design example addresses the basic problems of backlash in motion control systems. The objective is to control the position of a linear slide precisely. The slide is to be controlled by a rotary motor, which is coupled to the slide by a leadscrew. Such a leadscrew has a backlash of 4 micron, and the required position accuracy is for 0.5 micron.

The basic dilemma is where to mount the sensor. If you use a rotary sensor, you get a 4 micron backlash error. On the other hand, if you use a linear encoder, the backlash in the feedback loop will cause oscillations due to instability.

An alternative approach is the dual-loop, where we use two sensors, rotary and linear. The rotary sensor assures stability (because the position loop is closed before the backlash) whereas the linear sensor provides accurate load position information. The operation principle is to drive the motor to a given rotary position near the final point. Once there, the load position is read to find the position error and the controller commands the motor to move to a new rotary position which eliminates the position error.

Since the required accuracy is 0.5 micron, the resolution of the linear sensor should preferably be twice finer. A linear sensor with a resolution of 0.25 micron allows a position error of +/-2 counts.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The dual-loop approach requires the resolution of the rotary sensor to be equal or better than that of the linear system. Assuming that the pitch of the lead screw is 2.5mm (approximately 10 turns per inch), a rotary encoder of 2500 lines per turn or 10,000 count per revolution results in a rotary resolution of 0.25 micron. This results in equal resolution on both linear and rotary sensors.

To illustrate the control method, assume that the rotary encoder is used as a feedback for the X-axis, and that the linear sensor is read and stored in the variable LINPOS. Further assume that at the start, both the position of X and the value of LINPOS are equal to zero. Now assume that the objective is to move the linear load to the position of 1000.

The first step is to command the X motor to move to the rotary position of 1000. Once it arrives we check the position of the load. If, for example, the load position is 980 counts, it implies that a correction of 20 counts must be made. However, when the X-axis is commanded to be at the position of 1000, suppose that the actual position is only 995, implying that X has a position error of 5 counts, which will be eliminated once the motor settles. This implies that the correction needs to be only 15 counts, since 5 counts out of the 20 would be corrected by the X-axis. Accordingly, the motion correction should be:

Correction = Load Position Error - Rotary Position Error

The correction can be performed a few times until the error drops below +/-2 counts. Often, this is performed in one correction cycle.

Example - Backlash Compensation by Sampled Dual Loop

Instruction	Interpretation
#A	Label
DP0	Define starting positions as zero
LINPOS=0	
PR 1000	Required distance
BGX	Start motion
#B	
AMX	Wait for completion
WT 50	Wait 50 msec
LIN POS = _DEX	Read linear position
ER=1000-LINPOS-_TEX	Find the correction
JP #C,@ABS[ER]<2	Exit if error is small
PR ER	Command correction
BGX	Begin motion on X axis
JP #B	Repeat the process
#C	Label
EN	End program

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE LEFT BLANK INTENTIONALLY

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 8 Hardware & Software Protection

Introduction

The DMC-1500 provides several hardware and software features to check for error conditions and to inhibit the motor on error. These features help protect the various system components from damage.

WARNING: Machinery in motion can be dangerous! It is the responsibility of the user to design effective error handling and safety protection as part of the machine. Since the DMC-1500 is an integral part of the machine, the engineer should design his overall system with protection against a possible component failure on the DMC-1500. Galil shall not be liable or responsible for any incidental or consequential damages.

Hardware Protection

The DMC-1500 includes hardware input and output protection lines for various error and mechanical limit conditions. These include:

Output Protection Lines

Amp Enable - This signal goes low when the motor off command is given, when the position error exceeds the value specified by the Error Limit (ER) command, or when off-on-error condition is enabled (OE1) and the abort command is given. Each axis amplifier has separate amplifier enable lines. This signal also goes low when the watch-dog timer is activated, or upon reset. *Note: The standard configuration of the AEN signal is TTL active high. Both the polarity and the amplitude can be changed if you are using the ICM-1100 interface board. To make these changes, see section entitled 'Amplifier Interface' pg. 3-25.*

Input Protection Lines

Abort - A low input stops commanded motion instantly without a controlled deceleration. For any axis in which the Off-On-Error function is enabled, the amplifiers will be disabled. This could cause the motor to 'coast' to a stop. If the Off-On-Error function is not enabled, the motor will instantaneously stop and servo at the current position. The Off-On-Error function is further discussed in this chapter.

Forward Limit Switch - Low input inhibits motion in forward direction. If the motor is moving in the forward direction when the limit switch is activated, the motion will decelerate and stop. In addition, if the motor is moving in the forward direction, the controller will automatically jump to the limit switch subroutine, #LIMSWI (if such a routine has been written by the user). The CN command can be used to change the polarity of the limit switches.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Reverse Limit Switch - Low input inhibits motion in reverse direction. If the motor is moving in the reverse direction when the limit switch is activated, the motion will decelerate and stop. In addition, if the motor is moving in the reverse direction, the controller will automatically jump to the limit switch subroutine, #LIMSWI (if such a routine has been written by the user). The CN command can be used to change the polarity of the limit switches.

Software Protection

The DMC-1500 provides a programmable error limit for servo operation. The error limit can be set for any number between 1 and 32767 using the ER n command. The default value for ER is 16384.

Example:

ER 200,300,400,500	Set X-axis error limit for 200, Y-axis error limit to 300, Z-axis error limit to 400 counts, W-axis error limit to 500 counts
ER,1,,10	Set Y-axis error limit to 1 count, set W-axis error limit to 10 counts.

The units of the error limit are quadrature counts. The error is the difference between the command position and actual encoder position. If the absolute value of the error exceeds the value specified by ER, the DMC-1500 will generate several signals to warn the host system of the error condition. These signals include:

Signal or Function	Indication of Error
# POSERR	Jumps to automatic excess position error subroutine
Error Light	Turns on when position error exceeds error limit
OE Function	Shuts motor off by setting AEN output line low if OE1.

The position error of X,Y,Z and W can be monitored during execution using the TE command.

Programmable Position Limits

The DMC-1500 provides programmable forward and reverse position limits. These are set by the BL and FL software commands. Once a position limit is specified, the DMC-1500 will not accept position commands beyond the limit. Motion beyond the limit is also prevented.

Example - Using Position Limits

Instruction	Interpretation
DP0,0,0	Define Position
BL -2000,-4000,-8000	Set Reverse position limit
FL 2000,4000,8000	Set Forward position limit
JG 2000,2000,2000	Jog
BG XYZ	Begin

(motion stops at forward limits)

Off-On-Error

The DMC-1500 controller has a built in function which can turn off the motors under certain error conditions. This function is know as ‘Off-On-Error”. To activate the OE function for each axis, specify 1 for X,Y,Z and W axis. To disable this function, specify 0 for the axes. When this function is enabled, the specified motor will be disabled under the following 3 conditions:

The position error for the specified axis exceeds the limit set with the command, ER

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The abort command is given

The abort input is activated with a low signal.

Note: If the motors are disabled while they are moving, they may 'coast' to a stop because they are no longer under servo control.

To re-enable the system, use the Reset (RS) or Servo Here (SH) command.

Examples - Using Off-On-Error

OE 1,1,1,1	Enable off-on-error for X,Y,Z and W
OE 0,1,0,1	Enable off-on-error for Y and W axes and disable off-on-error for W and Z axes

Automatic Error Routine

The #POSERR label causes the statements following to be automatically executed if error on any axis exceeds the error limit specified by ER. The error routine must be closed with the RE command. The RE command returns from the error subroutine to the main program.

NOTE: The Error Subroutine will be entered again unless the error condition is gone.

Example - using automatic error subroutine

Instruction	Interpretation
#A;JP #A;EN	"Dummy" program
#POSERR	Start error routine on error
MG "error"	Send message
SB 1	Fire relay
STX	Stop motor
AMX	After motor stops
SHX	Servo motor here to clear error
RE	Return to main program

NOTE: An applications program must be executing for the #POSERR routine to function.

Limit Switch Routine

The DMC-1500 provides forward and reverse limit switches which inhibit motion in the respective direction. There is also a special label for automatic execution of a limit switch subroutine. The #LIMSWI label specifies the start of the limit switch subroutine. This label causes the statements following to be automatically executed if any limit switch is activated and that axis motor is moving in that direction. The RE command ends the subroutine.

The state of the forward and reverse limit switches may also be tested during the jump-on-condition statement. The _LR condition specifies the reverse limit and _LF specifies the forward limit. X,Y,Z, or W following LR or LF specifies the axis. The CN command can be used to configure the polarity of the limit switches.

Example - Using Limit Switch Subroutine

Instruction	Interpretation
#A;JP #A;EN	Dummy Program
#LIMSWI	Limit Switch Utility
V1=_LFX	Check if forward limit

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
V2=_LRX                 Check if reverse limit
JP#LF,V1=0              Jump to #LF if forward
JP#LR,V2=0              Jump to #LR if reverse
JP#END                  Jump to end
#LF                     #LF
MG "FORWARD LIMIT"      Send message
STX;AMX                 Stop motion
PR-1000;BGX;AMX         Move in reverse
JP#END                  End
#LR                     #LR
MG "REVERSE LIMIT"      Send message
STX;AMX                 Stop motion
PR1000;BGX;AMX          Move forward
#END                    End
RE                      Return to main program
```

NOTE: An applications program must be executing for #LIMSWI to function.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 9 Troubleshooting

Overview

The following discussion may help you get your system to work.

Potential problems have been divided into groups as follows:

1. Installation
2. Communication
3. Stability and Compensation
4. Operation

The various symptoms along with the cause and the remedy are described in the following tables.

Installation

Symptom	Cause	Remedy
Motor runs away when connected to amplifier with no additional inputs.	Amplifier offset too large.	Adjust amplifier offset
Same as above, but offset adjustment does not stop the motor.	Damaged amplifier.	Replace amplifier.
Controller does not read changes in encoder position.	Wrong encoder connections.	Check encoder wiring.
Same as above	Bad encoder	Check the encoder signals. Replace encoder if necessary.
Same as above	Bad controller	Connect the encoder to different axis input. If it works, controller failure. Repair or replace.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Communication

Symptom	Cause	Remedy
Using COMDISK and TALK2DMC, cannot communicate with controller.	Baud Rate not correctly configured	Check Baud rate switch positions and registry settings for software setup.
	Hardware handshaking is not enabled (Must be enabled to use Galil Software)	Set Hardware HSHK DIP switch
	Serial Cable is not correct	Use serial cable supplied by Galil or Check pinouts for cable (see appendix)
	Incorrect Com Port Specified	Change registry setting for com port

Stability

Symptom	Cause	Remedy
Motor runs away when the loop is closed.	Wrong feedback polarity.	Invert the polarity of the loop by inverting the motor leads (brush type) or the encoder.
Motor oscillates.	Too high gain or too little damping.	Decrease KI and KP. Increase KD.

Operation

Symptom	Cause	Remedy
Controller rejects command. Responded with a ?	Invalid Command	Interrogate the cause with TC or TC1.
Motor does not complete move.	Noise on limit switches stops the motor.	To verify cause, check the stop code (SC). If caused by limit switch noise, reduce noise.
During a periodic operation, motor drifts slowly.	Encoder noise	Interrogate the position periodically. If controller states that the position is the same at different locations it implies encoder noise. Reduce noise. Use differential encoder inputs.
Same as above.	Programming error.	Avoid resetting position error at end of move with SH command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 10 Theory of Operation

Overview

The following discussion covers the operation of motion control systems. A typical servo control system consists of the elements shown in Fig 10.1.

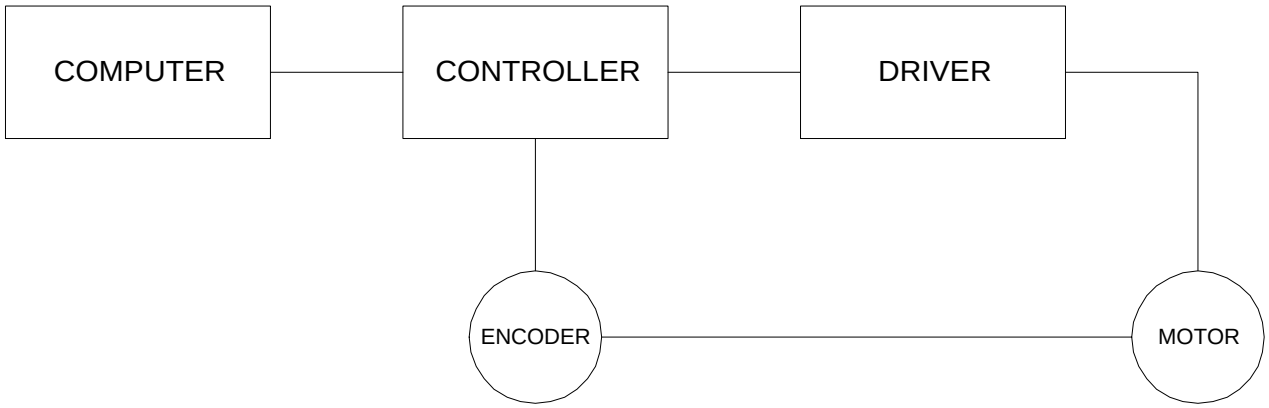


Figure 10.1 - Elements of Servo Systems

The operation of such a system can be divided into three levels, as illustrated in Fig. 10.2. The levels are:

1. Closing the Loop
2. Motion Profiling
3. Motion Programming

The first level, the closing of the loop, assures that the motor follows the commanded position. This is done by closing the position loop using a sensor. The operation at the basic level of closing the loop involves the subjects of modeling, analysis, and design. These subjects will be covered in the following discussions.

The motion profiling is the generation of the desired position function. This function, R(t), describes where the motor should be at every sampling period. Note that the profiling and the closing of the loop are independent functions. The profiling function determines where the motor should be and the closing of the loop forces the motor to follow the commanded position

The highest level of control is the motion program. This can be stored in the host computer or in the controller. This program describes the tasks in terms of the motors that need to be controlled, the distances and the speed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

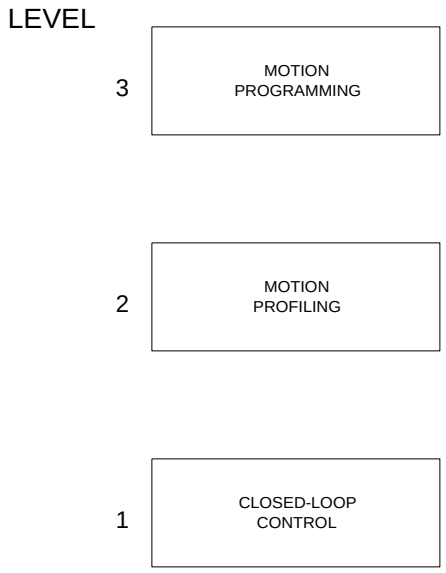


Figure 10.2 - Levels of Control Functions

The three levels of control may be viewed as different levels of management. The top manager, the motion program, may specify the following instruction, for example.

```
PR 6000,4000
SP 20000,20000
AC 200000,00000
BG X
AD 2000
BG Y
EN
```

This program corresponds to the velocity profiles shown in Fig. 10.3. Note that the profiled positions show where the motors must be at any instant of time.

Finally, it remains up to the servo system to verify that the motor follows the profiled position by closing the servo loop.

The following section explains the operation of the servo system. First, it is explained qualitatively, and then the explanation is repeated using analytical tools for those who are more theoretically inclined.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

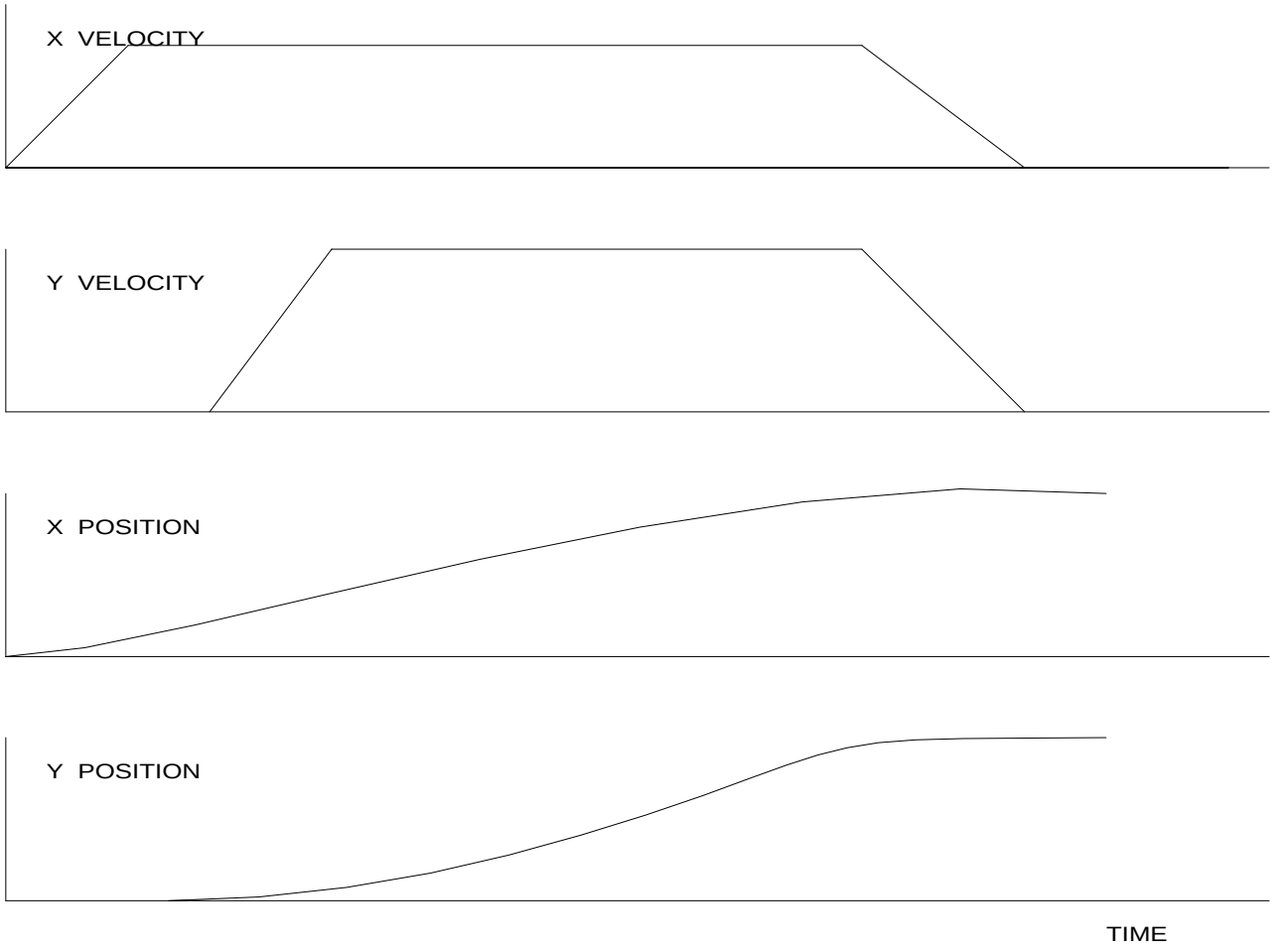


Figure 10.3 - Velocity and Position Profiles

Operation of Closed-Loop Systems

To understand the operation of a servo system, we may compare it to a familiar closed-loop operation, adjusting the water temperature in the shower. One control objective is to keep the temperature at a comfortable level, say 90 degrees F. To achieve that, our skin serves as a temperature sensor and reports to the brain (controller). The brain compares the actual temperature, which is called the feedback signal, with the desired level of 90 degrees F. The difference between the two levels is called the error signal. If the feedback temperature is too low, the error is positive, and it triggers an action which raises the water temperature until the temperature error is reduced sufficiently.

The closing of the servo loop is very similar. Suppose that we want the motor position to be at 90 degrees. The motor position is measured by a position sensor, often an encoder, and the position feedback is sent to the controller. Like the brain, the controller determines the position error, which is the difference between the commanded position of 90 degrees and the position feedback. The controller then outputs a signal that is proportional to the position error. This signal produces a proportional current in the motor, which causes a motion until the error is reduced. Once the error becomes small, the resulting current will be too small to overcome the friction, causing the motor to stop.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The analogy between adjusting the water temperature and closing the position loop carries further. We have all learned the hard way, that the hot water faucet should be turned at the "right" rate. If you turn it too slowly, the temperature response will be slow, causing discomfort. Such a slow reaction is called overdamped response.

The results may be worse if we turn the faucet too fast. The overreaction results in temperature oscillations. When the response of the system oscillates, we say that the system is unstable. Clearly, unstable responses are bad when we want a constant level.

What causes the oscillations? The basic cause for the instability is a combination of delayed reaction and high gain. In the case of the temperature control, the delay is due to the water flowing in the pipes. When the human reaction is too strong, the response becomes unstable.

Servo systems also become unstable if their gain is too high. The delay in servo systems is between the application of the current and its effect on the position. Note that the current must be applied long enough to cause a significant effect on the velocity, and the velocity change must last long enough to cause a position change. This delay, when coupled with high gain, causes instability.

This motion controller includes a special filter which is designed to help the stability and accuracy. Typically, such a filter produces, in addition to the proportional gain, damping and integrator. The combination of the three functions is referred to as a PID filter.

The filter parameters are represented by the three constants KP, KI and KD, which correspond to the proportional, integral and derivative term respectively.

The damping element of the filter acts as a predictor, thereby reducing the delay associated with the motor response.

The integrator function, represented by the parameter KI, improves the system accuracy. With the KI parameter, the motor does not stop until it reaches the desired position exactly, regardless of the level of friction or opposing torque.

The integrator also reduces the system stability. Therefore, it can be used only when the loop is stable and has a high gain.

The output of the filter is applied to a digital-to-analog converter (DAC). The resulting output signal in the range between +10 and -10 Volts is then applied to the amplifier and the motor.

The motor position, whether rotary or linear is measured by a sensor. The resulting signal, called position feedback, is returned to the controller for closing the loop.

The following section describes the operation in a detailed mathematical form, including modeling, analysis and design.

System Modeling

The elements of a servo system include the motor, driver, encoder and the controller. These elements are shown in Fig. 10.4. The mathematical model of the various components is given below.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

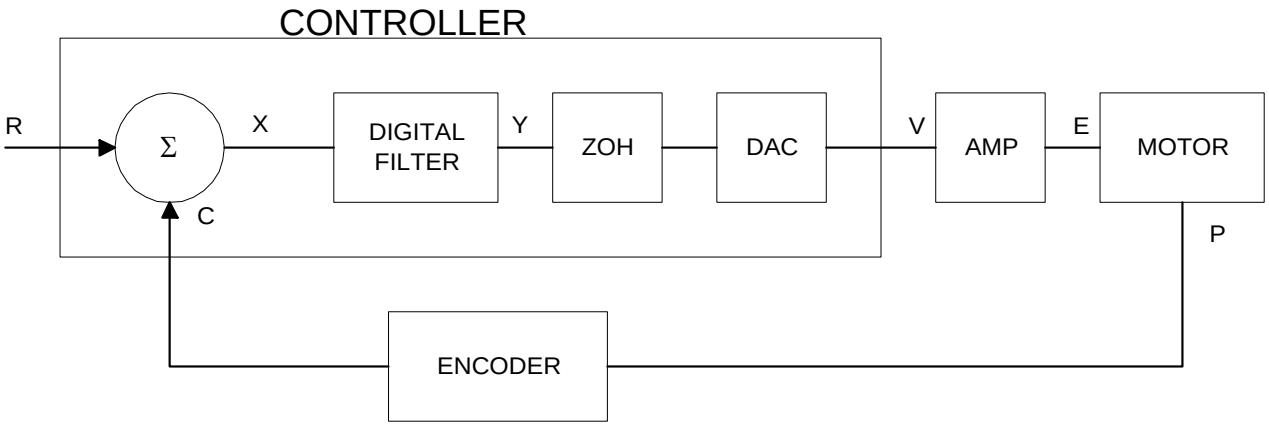


Figure 10.4 - Functional Elements of a Servo Control System

Motor-Amplifier

The motor amplifier may be configured in three modes:

1. Voltage Drive
2. Current Drive
3. Velocity Loop

The operation and modeling in the three modes is as follows:

Voltage Drive

The amplifier is a voltage source with a gain of Kv [V/V]. The transfer function relating the input voltage, V, to the motor position, P, is

$$P/V = K_V / \left[K_t S (ST_m + 1)(ST_e + 1) \right]$$

where

$$T_m = RJ / K_t^2 \quad [s]$$

and

$$T_e = L/R \quad [s]$$

and the motor parameters and units are

K_t	Torque constant [Nm/A]
R	Armature Resistance Ω
J	Combined inertia of motor and load [kg.m^2]
L	Armature Inductance [H]

When the motor parameters are given in English units, it is necessary to convert the quantities to MKS units. For example, consider a motor with the parameters:

K_t = 14.16 oz - in/A = 0.1 Nm/A

R = 2 Ω

J = 0.0283 oz-in-s^2 = 2.10^{-4} kg . m^2

L = 0.004H

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Then the corresponding time constants are

$$T_m = 0.04 \text{ sec}$$

and

$$T_e = 0.002 \text{ sec}$$

Assuming that the amplifier gain is Kv = 4, the resulting transfer function is

$$P/V = 40/[s(0.04s+1)(0.002s+1)]$$

Current Drive

The current drive generates a current I, which is proportional to the input voltage, V, with a gain of Ka. The resulting transfer function in this case is

$$P/V = K_a K_t / Js^2$$

where Kt and J are as defined previously. For example, a current amplifier with K_a = 2 A/V with the motor described by the previous example will have the transfer function:

$$P/V = 1000/s^2 \qquad \text{[rad/V]}$$

If the motor is a DC brushless motor, it is driven by an amplifier that performs the commutation. The combined transfer function of motor amplifier combination is the same as that of a similar brush motor, as described by the previous equations.

Velocity Loop

The motor driver system may include a velocity loop where the motor velocity is sensed by a tachometer and is fed back to the amplifier. Such a system is illustrated in Fig. 10.5. Note that the transfer function between the input voltage V and the velocity ω is:

$$\omega /V = [K_a K_t/Js]/[1+K_a K_t K_g/Js] = 1/[K_g(sT_1+1)]$$

where the velocity time constant, T1, equals

$$T1 = J/K_a K_t K_g$$

This leads to the transfer function

$$P/V = 1/[K_g s(sT1+1)]$$

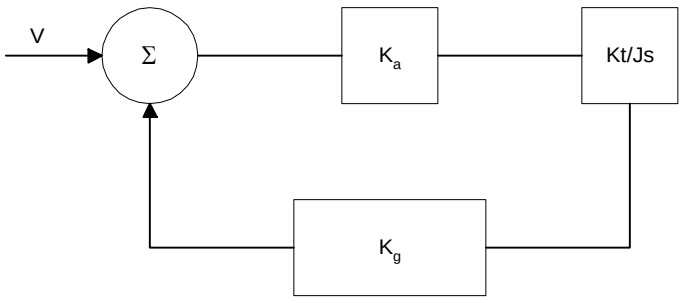
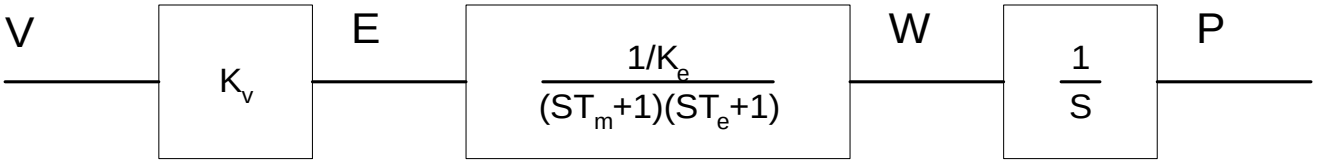


Figure 10.5 - Elements of velocity loops

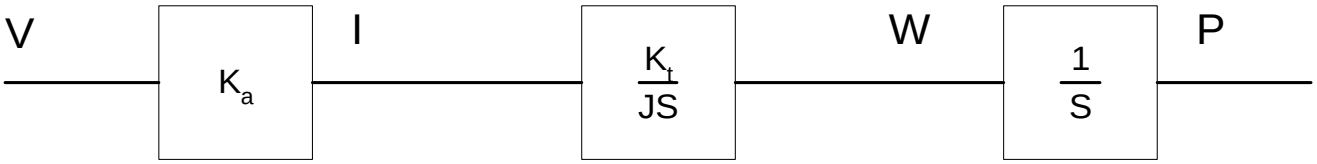
The resulting functions derived above are illustrated by the block diagram of Fig. 10.6.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VOLTAGE SOURCE



CURRENT SOURCE



VELOCITY LOOP

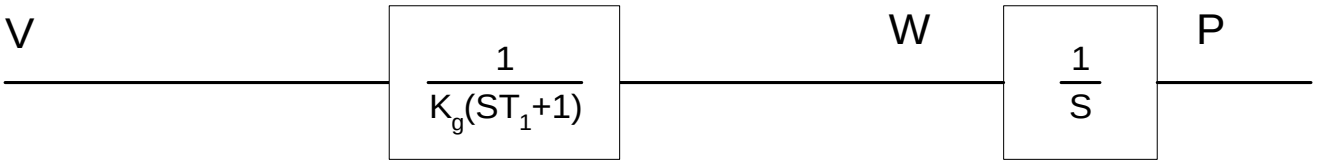


Figure 10.6 - Mathematical model of the motor and amplifier in three operational modes

Encoder

The encoder generates N pulses per revolution. It outputs two signals, Channel A and B, which are in quadrature. Due to the quadrature relationship between the encoder channels, the position resolution is increased to 4N quadrature counts/rev.

The model of the encoder can be represented by a gain of

$K_f = 4N/2\pi$ [count/rad]

For example, a 1000 lines/rev encoder is modeled as

$K_f = 638$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DAC

The DAC or D-to-A converter converts a 16-bit number to an analog voltage. The input range of the numbers is 65536 and the output voltage range is +/-10V or 20V. Therefore, the effective gain of the DAC is

K= 20/65536 = 0.0003 [V/count]

Digital Filter

The digital filter has a transfer function of D(z) = K(z-A)/z + Cz/z-1 and a sampling time of T.

The filter parameters, K, A and C are selected by the instructions KP, KD, KI or by GN, ZR and KI, respectively. The relationship between the filter coefficients and the instructions are:

K = (KP + KD) • 4 or K = GN • 4

A = KD/(KP + KD) or A = ZR

C = KI/2

This filter includes a lead compensation and an integrator. It is equivalent to a continuous PID filter with a transfer function G(s).

G(s) = P + sD + I/s

P = K(1-A) = 4 • KP

D = T • K • A = 4 • T • KD

I = C/T = KI/2T

For example, if the filter parameters of the DMC-1500 are

KP = 4

KD = 36

KI = 2

T = 0.001 s

the digital filter coefficients are

K = 160

A = 0.9

C = 1

and the equivalent continuous filter, G(s), is

G(s) = 16 + 0.144s + 1000/s

ZOH

The ZOH, or zero-order-hold, represents the effect of the sampling process, where the motor command is updated once per sampling period. The effect of the ZOH can be modeled by the transfer function

H(s) = 1/(1+sT/2)

If the sampling period is T = 0.001, for example, H(s) becomes:

H(s) = 2000/(s+2000)

However, in most applications, H(s) may be approximated as one.

This completes the modeling of the system elements. Next, we discuss the system analysis.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

System Analysis

To analyze the system, we start with a block diagram model of the system elements. The analysis procedure is illustrated in terms of the following example.

Consider a position control system with the DMC-1500 controller and the following parameters:

$K_t = 0.1$	Nm/A	Torque constant
$J = 2.10^{-4}$	$kg.m^2$	System moment of inertia
$R = 2$	Ω	Motor resistance
$K_a = 4$	Amp/Volt	Current amplifier gain
KP = 12.5		Digital filter gain
KD = 245		Digital filter zero
KI = 0		No integrator
N = 500	Counts/rev	Encoder line density
T = 1	ms	Sample period

The transfer function of the system elements are:

Motor

$$M(s) = P/I = Kt/Js2 = 500/s^2 \ [rad/A]$$

Amp

$$K_a = 4 \ [Amp/V]$$

DAC

$$K_d = 0.0003 \ [V/count]$$

Encoder

$$K_f = 4N/2\pi = 318 \ [count/rad]$$

ZOH

$$2000/(s+2000)$$

Digital Filter

$$KP = 12.5, \ KD = 245, \ T = 0.001$$

Therefore,

$$D(z) = 50 + 980 \ (1-z-1)$$

Accordingly, the coefficients of the continuous filter are:

$$P = 50$$

$$D = 0.98$$

The filter equation may be written in the continuous equivalent form:

$$G(s) = 50 + 0.98s$$

The system elements are shown in Fig. 10.7.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

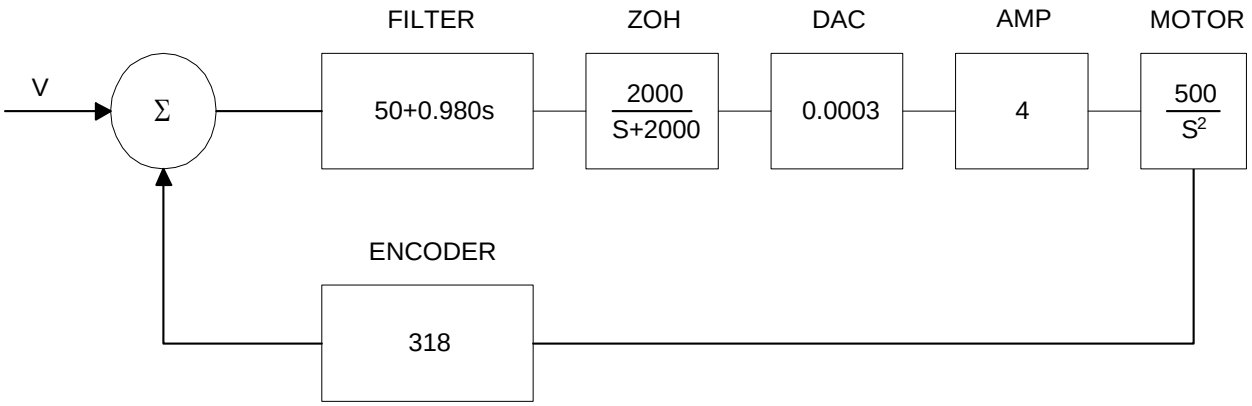


Figure 10.7 - Mathematical model of the control system

The open loop transfer function, A(s), is the product of all the elements in the loop.

$$A = 390{,}000\ (s+51)/[s^2(s+2000)]$$

To analyze the system stability, determine the crossover frequency, ω_C at which A(j ω_C) equals one. This can be done by the Bode plot of A(j ω_C), as shown in Fig. 10.8.

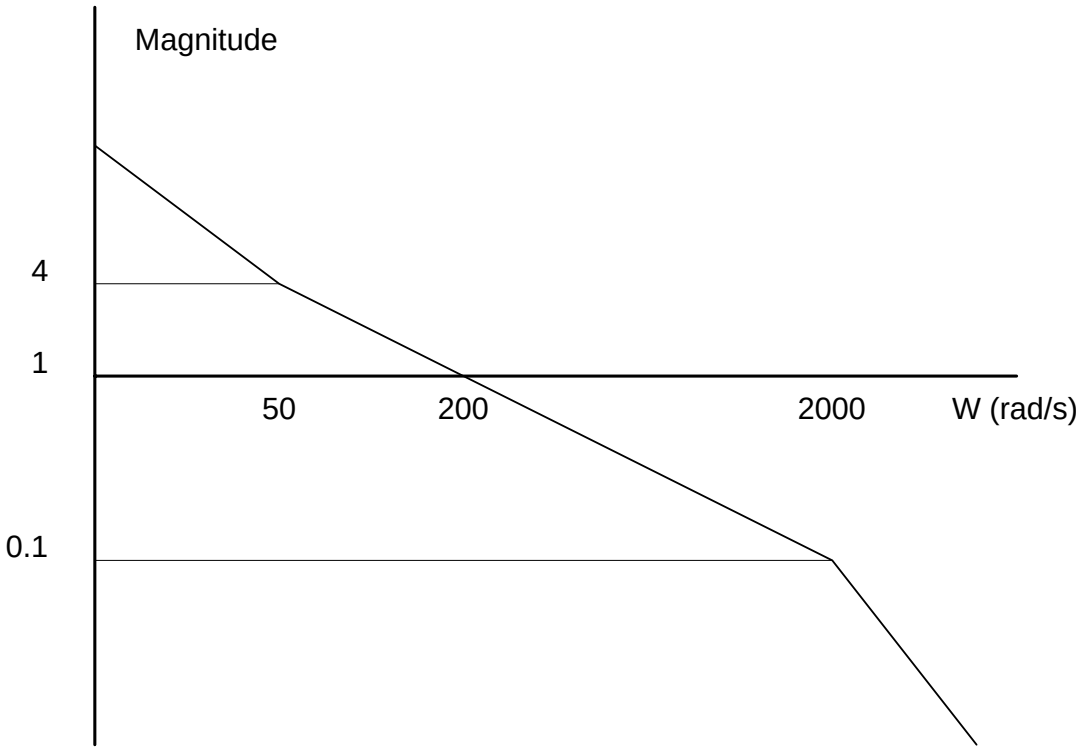


Figure 10.8 - Bode plot of the open loop transfer function

For the given example, the crossover frequency was computed numerically resulting in 200 rad/s.

Next, we determine the phase of A(s) at the crossover frequency.

$$A(j200) = 390{,}000\ (j200+51)/[(j200)^2\ .\ (j200 + 2000)]$$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

α = Arg[A(j200)] = $\tan^{-1}$(200/51)-180° -$\tan^{-1}$(200/2000)

α = 76° - 180° - 6° = -110°

Finally, the phase margin, PM, equals

PM = 180° + α = 70°

As long as PM is positive, the system is stable. However, for a well damped system, PM should be between 30 degrees and 45 degrees. The phase margin of 70 degrees given above indicated overdamped response.

Next, we discuss the design of control systems.

System Design and Compensation

The closed-loop control system can be stabilized by a digital filter, which is preprogrammed in the DMC-1500 controller. The filter parameters can be selected by the user for the best compensation. The following discussion presents an analytical design method.

The Analytical Method

The analytical design method is aimed at closing the loop at a crossover frequency, ω_c, with a phase margin PM. The system parameters are assumed known. The design procedure is best illustrated by a design example.

Consider a system with the following parameters:

K_t = 0.83	Nm/A	Torque constant
J = 2.10^{-4}	$kg.m^2$	System moment of inertia
R = 2	Ω	Motor resistance
K_a = 2	Amp/Volt	Current amplifier gain
N = 1000	Counts/rev	Encoder line density

The DAC of the DMC-1500 outputs +/-10V for a 16-bit command of +/- 32,768 counts.

The design objective is to select the filter parameters in order to close a position loop with a crossover frequency of ω_c = 500 rad/s and a phase margin of 45 degrees.

The first step is to develop a mathematical model of the system, as discussed in the previous system.

Motor

$M(s) = P/I = K_t/Js^2 = 4150/s^2$

Amp

K_a = 2 [Amp/V]

DAC

K_d = 20/65536 = .0003

Encoder

$K_f = 4N/2\pi$ = 636

ZOH

H(s) = 2000/(s+2000)

Compensation Filter

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

$$G(s) = P + sD$$

The next step is to combine all the system elements, with the exception of G(s), into one function, L(s).

$$L(s) = M(s)\, K_a\, K_d\, K_f\, H(s) = 0.3175*10^7/[s^2(s+2000)]$$

Then the open loop transfer function, A(s), is

$$A(s) = L(s)\, G(s)$$

Now, determine the magnitude and phase of L(s) at the frequency $\omega_c = 500$.

$$L(j500) = 0.3175*10^7/[(j500)^2\ (j500+2000)]$$

This function has a magnitude of

$$|L(j500)| = 0.00625$$

and a phase

$$Arg[L(j500)] = -180° - \tan^{-1}(500/2000) = -194°$$

G(s) is selected so that A(s) has a crossover frequency of 500 rad/s and a phase margin of 45 degrees. This requires that

$$|A(j500)| = 1$$

$$Arg\ [A(j500)] = -135°$$

However, since

$$A(s) = L(s)\, G(s)$$

then it follows that G(s) must have magnitude of

$$|G(j500)| = |A(j500)/L(j500)| = 160$$

and a phase

$$Arg\ [G(j500)] = Arg\ [A(j500)] - Arg\ [L(j500)] = -135° + 194° = 59°$$

In other words, we need to select a filter function G(s) of the form

$$G(s) = P + sD$$

so that at the frequency ω_c =500, the function would have a magnitude of 160 and a phase lead of 59 degrees.

These requirements may be expressed as:

$$|G(j500)| = |P + (j500D)| = 160$$

and

$$Arg\ [G(j500)] = \tan^{-1}[500D/P] = 59°$$

The solution of these equations leads to:

$$P = 40\cos 59° = 82.4$$

$$500D = 40\sin 59° = 137.2$$

Therefore,

$$D = 0.2744$$

and

$$G = 82.4 + 0.2744s$$

The function G is equivalent to a digital filter of the form:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

$$D(z) = 4 \bullet KP + 4 \bullet KD(1-z^{-1})$$

where

$$KP = P/4$$

and

$$KD = D/ (4 \bullet T)$$

Assuming a sampling period of T=1ms, the parameters of the digital filter are:

KP = 20.6

KD = 68.6

The DMC-1500 can be programmed with the instruction:

KP 20.6

KD 68.6

In a similar manner, other filters can be programmed. The procedure is simplified by the following table, which summarizes the relationship between the various filters.

Equivalent Filter Form

	DMC - 1000 -18
Digital	$D(z) = K(z-A/z) + Cz/(z-1)$
Digital	$D(z) = 4\ KP + 4\ KD(1-z^{-1}) + KI/2(1-z^{-1})$
KP, KD, KI	$K = (KP + KD) \cdot 4$
	$A = KD/(KP+KD)$
	$C = KI/2$
Digital	$D(z) = 4\ GN(z-ZR)/z + KI\ z/2(z-1)$
GN, ZR, KI	$K = 4\ GN$
	$A = ZR$
	$C = KI/2$
Continuous	$G(s) = P + Ds + I/s$
PID, T	$P = 4\ KP$
	$D = 4\ T*KD$
	$I = KI/2T$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE LEFT BLANK INTENTIONALLY

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendices

Electrical Specifications

Servo Control

ACMD Amplifier Command:	+/-10 Volts analog signal. Resolution 16-bit DAC or .0003 Volts. 3 mA maximum
A+,A-,B+,B-,IDX+,IDX- Encoder and Auxiliary	TTL compatible, but can accept up to +/-12 Volts. Quadrature phase on CHA,CHB. Can accept single-ended (A+,B+ only) or differential (A+,A-,B+,B-). Maximum A,B edge rate: 8 MHz. Minimum IDX pulse width: 120 nsec.

Stepper Control

Pulse	TTL (0-5 Volts) level at 50% duty cycle. 2,000,000 pulses/sec maximum frequency
Direction	TTL (0-5 Volts)

Input/Output

Uncommitted Inputs, Limits, Home Abort Inputs:	2.2K ohm in series with optoisolator. Requires at least 1 mA to activate. Can accept up to 28 Volts without additional series resistor. Above 28 Volts requires additional resistor.
AN[1] thru AN[7] Analog Inputs:	Standard configuration is +/-10 Volt. 12-Bit Analog-to-Digital converter.
OUT[1] thru OUT[8] Outputs:	TTL.
OUT[9] through OUT [16] Outputs	TTL (only available on controllers with 4 or more axes)
IN[17] through IN[24] Inputs	TTL (only available on controllers with 4 or more axes)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Power

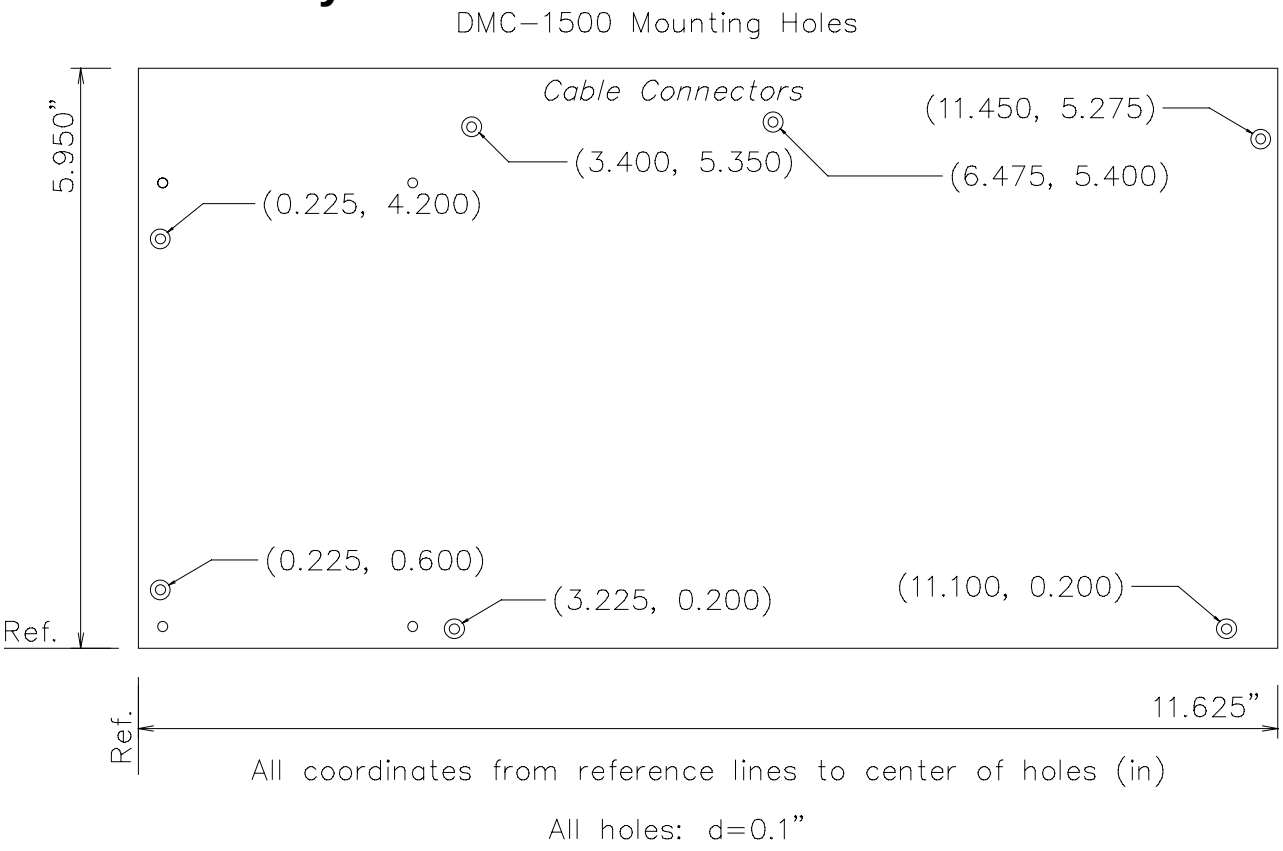
+5V	750 mA
+12V	40 mA
-12V	40mA

Performance Specifications

Minimum Servo Loop Update Time:	DMC-1510 -- 250 µsec
	DMC-1520 -- 375 µsec
	DMC-1530 -- 500 µsec
	DMC-1540 – 500 µsec
	DMC-1550 – 625 µsec
	DMC-1560 – 750 µsec
	DMC-1570 – 875 µsec
	DMC-1580 – 875 µsec
Position Accuracy:	+/-1 quadrature count
Velocity Accuracy:	
Long Term	Phase-locked, better than .005%
Short Term	System dependent
Position Range:	+/-2147483647 counts per move
Velocity Range:	Up to 8,000,000 counts/sec
Velocity Resolution:	2 counts/sec
Motor Command Resolution:	16 Bits or 0.0003 V
Variable Range:	+/-2 billion
Variable Resolution:	$1 \cdot 10^{-4}$
Array Size:	8000 elements
Program Size:	1000 lines x 80 characters

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Card Level Layout



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Connectors for DMC-1500 Main Board

J2 - Main (60 pin IDC)

1	Ground	2	5 Volts
3	Error	4	Reset
5	Limit Common (LSCOM)	6	Forward Limit - X
7	Reverse Limit - X	8	Home - X
9	Forward Limit - Y	10	Reverse Limit - Y
11	Home - Y	12	Forward Limit - Z
13	Reverse Limit - Z	14	Home - Z
15	Forward Limit - W	16	Reverse Limit - W
17	Home - W	18	Output 1
19	Input Common (INCOM)	20	Latch X INPUT 1
21	Latch Y INPUT 2	22	Latch Z
23	Latch W INPUT 4	24	Abort input
25	Motor Command X	26	Amp enable X
27	Motor Command Y	28	Amp enable Y
29	Motor Command Z	30	Amp enable Z
31	Motor Command W	32	Amp enable W
33	A+X	34	A-X
35	B+X	36	B-X
37	I+X	38	I-X
39	A+Y	40	A-Y
41	B+Y	42	B-Y
43	I+Y	44	I-Y
45	A+Z	46	A-Z
47	B+Z	48	B-Z
49	I+Z	50	I-Z
51	A+W	52	A-W
53	B+W	54	B-W
55	I+W	56	I-W
57	+12V	58	-12V
59	5V	60	Ground

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

J5 - General I/O (26 pin IDC)

1 Analog 1	2 Analog 2
3 Analog 3	4 Analog 4
5 Analog 5	6 Analog 6
7 Analog 7	8 Ground
9 5 Volts	10 Output 1
11 Output 2	12 Output 3
13 Output 4	14 Output 5
15 Output 6	16 Output 7
17 Output 8	18 Input 8
19 Input 7	20 Input 6
21 Input 5	22 Input 4 (Latch W)
23 Input 3 (latch Z)	24 Input 2 (Latch Y)
25 Input 1 (latch X)	26 Input Common (INCOM)

J3 - Aux Encoder (20 pin IDC)

1 Sample clock	2 Synch
3 B-Aux W	4 B+Aux W
5 A-Aux W	6 A+Aux W
7 B-Aux Z	8 B+Aux Z
9 A-Aux Z	10 A+Aux Z
11 B-Aux Y	12 B+Aux Y
13 A-Aux Y	14 A+Aux Y
15 B-Aux X	16 B+Aux X
17 A-Aux X	18 A+Aux X
19 5 Volt	20 Ground

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

J4 - Driver (20 pin IDC)

1 Motor Command X	2 Amp enable X
3 PWM X/STEP X	4 Sign X/DIR X
5 NC	6 Motor Command Y
7 Amp enable Y	8 PWM Y/STEP Y
9 Sign Y/DIR Y	10 NC
11 Motor command Z	12 Amp enable Z
13 PWM Z/STEP Z	14 Sign Z/DIR Z
15 5 Volt	16 Motor command W
17 Amp enable W	18 PWM W/STEP W
19 Sign W/DIR W	20 Ground

J6 - Daughter Board Connector (60 pin)

For use only with a Galil daughter board.

J7 - 10 pin

For test only.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Connectors for Auxiliary Board (Axes E,F,G,H)

JD2 - Main (60 pin IDC)

1 Ground	2 5 Volts
3 N.C.	4 N.C.
5 Limit Common	6 Forward Limit - E
7 Reverse Limit - E	8 Home - E
9 Forward Limit - F	10 Reverse Limit - F
11 Home F	12 Forward Limit - G
13 Reverse Limit - G	14 Home - G
15 Forward Limit - H	16 Reverse Limit - H
17 Home H	18 Output 9
19 Input Common	20 Latch E
21 Latch F	22 Latch G
23 Latch H	24 Input 24
25 Motor Command E	26 Amp enable E
27 Motor Command F	28 Amp enable F
29 Motor Command G	30 Amp enable G
31 Motor Command H	32 Amp enable H
33 Channel A+ E	34 Channel A- E
35 Channel B+ E	36 Channel B- E
37 Channel I+ E	38 Channel I- E
39 Channel A+ F	40 Channel A- F
41 Channel B+ F	42 Channel B- F
43 Channel I+ F	44 Channel I- F
45 Channel A+ G	46 Channel A- G
47 Channel B+ G	48 Channel B- G
49 Channel I+ G	50 Channel I- G
51 Channel A+ H	52 Channel A- H
53 Channel B+ H	54 Channel B- H
55 Channel I+ H	56 Channel I- H
57 +12V	58 -12V
59 5V	60 Ground

NOTE: The ABCD axes and other I/O are located on the main DMC-1500 card

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

JD5 - I/O (26 pin IDC)

1 Input 17 (TTL)	2 Input 18 (TTL)
3 Input 19 (TTL)	4 Input 20 (TTL)
5 Input 21 (TTL)	6 Input 22 (TTL)
7 Input 23 (TTL)	8 Ground
9 5 Volts	10 Output 9
11 Output 10	12 Output 11
13 Output 12	14 Output 13
15 Output 14	16 Output 15
17 Output 16	18 Input 16
19 Input 15	20 Input 14
21 Input 13	22 Input 12 (Latch H)
23 Input 11 (Latch G)	24 Input 10 (Latch F)
25 Input 9 (Latch E)	26 Input Common (Isolated 5 Volts)

JD3 - 20 pin IDC - Auxiliary Encoders

1 N.C.	2 N.C.
3 Aux. B- H	4 Aux. B+ H
5 Aux. A- H	6 Aux. A+ H
7 Aux. B- G	8 Aux. B+ G
9 Aux. A- G	10 Aux. A+ G
11 Aux. B- F	12 Aux. B+ F
13 Aux. A- F	14 Aux. A+ F
15 Aux. B- E	16 Aux. B+ E
17 Aux. A- E	18 Aux. A+ E
19 5 Volt	20 Ground

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

JD4 - 20 pin IDC - Amplifiers

1 Motor Command E	2 Amp enable E
3 PWM E/Step E	4 Sign E/Dir E
5 NC	6 Motor Command F
7 Amp enable F	8 PWM F/Step F
9 Sign F/Dir F	10 NC
11 Motor Command G	12 Amp enable G
13 PWM G/Step G	14 Sign G/Dir G
15 5 Volt	16 Motor Command H
17 Amp enable H	18 PWM H/Step H
19 Sign H/Dir H	20 Ground H

JD6 - Daughterboard Connector (60 pin)

Connects to DMC-1500 Main Board, connector J6

Cable Connections for DMC-1500

The DMC-1500 requires the transmit, receive, and ground for slow communication rates. (ie 1200 baud) For faster rates the handshake lines are required. The connection tables below contain the handshake lines. These descriptions and tables are for RS-232 only. RS-422 is available on request.

Standard RS-232 Specifications

25 pin Serial Connector

Table describes the pinout for standard serial ports found on most computers.

PIN NUMBER	FUNCTION
1	NC
2	Transmitted Data
3	Received Data
4	Request to Send
5	Clear to Send
6	Data Set Ready
7	Signal Ground
8	Carrier Detect
9	+Transmit Current Loop Return
10	NC
11	-Transmit Current Loop Data
12	NC
13	NC

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

14	NC
15	NC
16	NC
17	NC
18	+Receive Current Loop Data
19	NC
20	Data Terminal Ready
21	NC
22	Ring Indicator
23	NC
24	NC
25	-Receive Current Loop Return

9 Pin Serial Connector

Table describes the pinout for standard serial ports found on most computers.

PIN NUMBER	FUNCTION
1	Carrier Detect
2	Receive Data
3	Transmit Data
4	Data Terminal Ready
5	Signal Ground
6	Data Set Ready
7	Request to Send
8	Clear to Send
9	Ring Indicator

DMC-1500 Serial Cable Specifications

Computer 25 pin to DMC-1500 Main Port Cable (9 pin)

25 PIN (MALE - COMPUTER)	9 PIN (FEMALE - CONTROLLER)
8 (Carrier Detect)	1
3 (Receive Data)	2
2 (Transmit Data)	3
20 (Data Terminal Ready)	4
7 (Signal Ground)	5
Controller Ground	9

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Computer 9 pin to DMC-1500 Main Port Cable (9 pin)

9 PIN (FEMALE - COMPUTER)	9 PIN (FEMALE - CONTROLLER)
1 (Carrier Detect)	1
2 (Receive Data)	2
3 (Transmit Data)	3
4 (Data Terminal Ready)	4
5 (Signal Ground)	5
Controller Ground	9

Computer 25 pin to DMC-1500 Auxiliary Port Cable (9 pin)

25 PIN (MALE - TERMINAL)	9 PIN (MALE - CONTROLLER)
20 (Data Terminal Ready)	1
2 (Transmit Data)	2
3 (Receive Data)	3
8 (Carrier Detect)	4
7 (Signal Ground)	5
Controller +5V	9

Computer 9 pin to DMC-1500 Auxiliary Port Cable (9 pin)

9 PIN (FEMALE - TERMINAL)	9 PIN (MALE – CONTROLLER)
4 (Data Terminal Ready)	1
3 (Transmit Data)	2
2 (Receive Data)	3
1 (Carrier Detect)	4
5 (Signal Ground)	5
Controller +5V	9

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Pin-Out Description for DMC-1500

Outputs	
Analog Motor Command	+/- 10 Volt range signal for driving amplifier. In servo mode, motor command output is updated at the controller sample rate. In the motor off mode, this output is held at the OF command level.
Amp Enable	Signal to disable and enable an amplifier. Amp Enable goes low on Abort and OE1.
PWM/STEP OUT	PWM/STEP OUT is used for directly driving power bridges for DC servo motors or for driving step motor amplifiers. For servo motors: If you are using a conventional amplifier that accepts a +/-10 Volt analog signal, this pin is not used and should be left open. The switching frequency is 16.7 KHZ. The PWM output is available in two formats: Inverter and Sign Magnitude. In the Inverter mode, the PWM signal is .2% duty cycle for full negative voltage, 50% for 0 Voltage and 99.8% for full positive voltage. In the Sign Magnitude Mode (Jumper SM), the PWM signal is 0% for 0 Voltage, 99.6% for full voltage and the sign of the Motor Command is available at the sign output.
PWM/STEP OUT	For step motors: The STEP OUT pin produces a series of pulses for input to a step motor driver. The pulses may either be low or high. The pulse width is 50%. Upon Reset, the output will be low if the SM jumper is on. If the SM jumper is not on, the output will be Tri-state.
Sign/Direction	Used with PWM signal to give the sign of the motor command for servo amplifiers or direction for step motors.
Error	The signal goes low when the position error on any axis exceeds the value specified by the error limit command, ER.
Output 1-Output 8 Output 9-Output 16 (DMC-1580 only)	These 8 TTL outputs are uncommitted and may be designated by the user to toggle relays and trigger external events. The output lines are toggled by Set Bit, SB, and Clear Bit, CB, instructions. The OP instruction is used to define the state of all the bits of the Output port.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Inputs

Encoder, A+, B+	Position feedback from incremental encoder with two channels in quadrature, CHA and CHB. The encoder may be analog or TTL. Any resolution encoder may be used as long as the maximum frequency does not exceed 8,000,000 quadrature states/sec. The controller performs quadrature decoding of the encoder signals resulting in a resolution of quadrature counts (4 x encoder cycles). Note: Encoders that produce outputs in the format of pulses and direction may also be used by inputting the pulses into CHA and direction into Channel B and using the CE command to configure this mode.
Encoder Index, I+	Once-Per-Revolution encoder pulse. Used in Homing sequence or Find Index command to define home on an encoder index.
Encoder, A-, B-, I-	Differential inputs from encoder. May be input along with CHA, CHB for noise immunity of encoder signals. The CHA- and CHB- inputs are optional.
Auxiliary Encoder, Aux A+, Aux B+, Aux I+, Aux A-, Aux B-, Aux I-	Inputs for additional encoder. Used when an encoder on both the motor and the load is required.
Abort	A low input stops commanded motion instantly without a controlled deceleration. Also aborts motion program.
Reset	A low input resets the state of the processor to its power-on condition. The previously saved state of the controller, along with parameter values, and saved sequences are restored.
Forward Limit Switch	When active, inhibits motion in forward direction. Also causes execution of limit switch subroutine, #LIMSWI. The polarity of the limit switch may be set with the CN command.
Reverse Limit Switch	When active, inhibits motion in reverse direction. Also causes execution of limit switch subroutine, #LIMSWI. The polarity of the limit switch may be set with the CN command.
Home Switch	Input for Homing (HM) and Find Edge (FE) instructions. Upon BG following HM or FE, the motor accelerates to slew speed. A transition on this input will cause the motor to decelerate to a stop. The polarity of the Home Switch may be set with the CN command.
Input 1 - Input 8 - Isolated *Input 9 - Input 16 - Isolated* *(DMC-1580 only)* *Input 17 - Input 23 - TTL*	Uncommitted inputs. May be defined by the user to trigger events. Inputs are checked with the Conditional Jump instruction and After Input instruction or Input Interrupt. Input 1 is latch X, Input 2 is latch Y, Input 3 is latch Z and Input 4 is latch W if the high speed position latch function is enabled.
Latch	High speed position latch to capture axis position within 20 nano seconds on occurrence of latch signal. AL command arms latch. Input 1 is latch X, Input 2 is latch Y, Input 3 is latch Z and Input 4 is latch W. *Input 9 is latch E, Input 10 is latch F, Input 11 is latch G, Input 12 is latch H.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Configuration Description for DMC-1500

Jumpers

LABEL	FUNCTION (IF JUMPERED)
SMX	For each axis, the SM jumper selects the SM
SMY	magnitude mode for servo motors or selects stepper
SMZ	motors. If you are using a stepper motor for an axis,
SMW	you must jumper SM for that axis. The analog motor
SME	command is not valid with SM jumpered.
SMF	
SMG	
SMH	
OPT	Reserved
JP30, JP31	Used to configure RS422 - main
JP31	Used to configure RS422 - auxiliary
MRST	Master Reset enable. Returns controller to factory default settings and erases EEPROM. Depress reset to activate.

Address Configuration Jumpers

ADD4	ADDR2	ADDR1	ADDRESS
OFF	OFF	OFF	0
OFF	OFF	ON	1
OFF	ON	OFF	2
OFF	ON	ON	3
ON	OFF	OFF	4
ON	OFF	ON	5
ON	ON	OFF	6
ON	ON	ON	7

Front Panel Baud Rate Switches

1200	9600	19.2	BAUD
ON	ON	OFF	300
ON	OFF	OFF	1200
ON	OFF	ON	4800
OFF	ON	OFF	9600
OFF	OFF	ON	19,200
OFF	ON	ON	38,400
ON	ON	ON	Self Test

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Adjustment Pots

ADJUSTMENT POTS	
X offset	Used to null ACMD offset for X axis
Y offset	Used to null ACMD offset for Y axis
Z offset	Used to null ACMD offset for Z axis
W offset	Used to null ACMD offset for W axis

Dip Switch Settings

A2-A8 Seven Dip Switches for Address Selection.
(Please follow silkscreen; not switch labels)

Offset Adjustments for DMC-1500

X offset	Used to null ACMD offset for X axis
Y offset	Used to null ACMD offset for Y axis
Z offset	Used to null ACMD offset for Z axis
W offset	Used to null ACMD offset for W axis

Note: These adjustments are made at the Galil factory and should need adjustment under most applications.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Accessories and Options

DMC-1510	Single Axis Controller
DMC-1520	Two-Axis Controller
DMC-1530	Three-Axis Controller
DMC-1540	Four-Axis Controller
DMC-1550	Five-Axis Controller
DMC-1560	Six-Axis Controller
DMC-1570	Seven-Axis Controller
DMC-1580	Eight-Axis Controller
-16	16 Bit ADC Option
ICM-1100*	Interface board
AMP-1110	Single axis amplifier
AMP-1120	Two-axis amplifier
AMP-1130	Three-axis amplifier
AMP-1140	Four-axis amplifier
N23-54-1000	Servo motor; NEMA 23; 54 oz-in continuous
N34-150-1000	Servo motor; NEMA 34; 150 oz-in, continuous
COMM-Disk	DOS and Windows (16 or 32 bit) Terminal Emulator and Software Sources
SDK-1500	DOS Servo Design Software
WSDK16	16 Bit Windows Servo Design Kit for Windows 3.1, Windows 3.11 and Windows 95
WSDK32	32 Bit Windows Servo Design Kit for Windows 95 and Windows NT
CAD-to-DMC	Autocad to DMC Translator
Visual Basic Toolkit	Visual Basic VBX and OCX Extensions

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ICM-1100 Interconnect Module

The ICM-1100 Interconnect Module provides easy connections between the DMC-1500 series controllers and other system elements, such as amplifiers, encoders, and external switches. The ICM-1100 accepts each DMC-1500 ribbon cable (for J2, J3, J4 and J5) and breaks them into screw-type terminals. Each screw terminal is labeled for quick connection of system elements.

The ICM-1100 is packaged as a circuit board mounted to a metal enclosure. A version of the ICM-1100 is also available with servo amplifiers (see AMP-11X0).

Features

- Breaks out all DMC-1500 ribbon cables into individual screw-type terminals.
- Clearly identifies all terminals
- Provides jumper for connecting limit and input supplies to 5 volt supply from PC.
- Available with on-board servo drives (see AMP-1100).
- 10-pin IDC connectors for encoders.

Specifications

Dimensions	5.7" x 13.4" x 2.4"
Weight	2.2 pounds

AMP/ICM-1100 Connections

Rev A+B boards (orange) and Rev C boards (black) have the pinouts listed below:

Rev A + B Terminal #	Rev C Terminal #	Label	Description
	2	GND	Ground
1	1	GND	Ground
2	4	ACMDX	X input to servo amp
3	3	AENX	X amp enable
4	6	PULSX	X pulse input for stepper
5	5	DIRX	X direction input for stepper
6	8	ACMDY	Y amp input
7	7	AENY	Y amp enable
8	10	PULSY	Y pulse for stepper
9	9	DIRY	Y direction for stepper
10	12	ACMDZ	Z amp input
11	11	AENZ	Z amp enable
12	14	PULSZ	Z pulse for stepper
13	13	DIRZ	Z direction for stepper
14	16	ACMDW	W amp input
15	15	AENW	W amp enable
16	18	PULSW	W pulse for stepper

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Rev A + B Terminal #	Rev C Terminal #	Label	Description
17	17	DIRW	W direction for stepper
18	20	AN1	Analog Input 1
19	19	AN2	Analog Input 2
20	22	AN3	Analog Input 3
21	21	AN4	Analog Input 4
22	24	AN5	Analog Input 5
23	23	AN6	Analog Input 6
24	26	AN7	Analog Input 7
25	25	GND	Ground
26	28	OUT1	Digital Output 1
27	27	OUT2	Digital Output 2
28	30	OUT3	Digital Output 3
29	29	OUT4	Digital Output 4
30	32	OUT5	Digital Output 5
31	31	OUT6	Digital Output 6
32	34	OUT7	Digital Output 7
33	33	OUT8	Digital Output 8
34	36	INP8	Uncommitted Input 8
35	35	INP7	Uncommitted Input 7
36	38	INP6	Uncommitted Input 6
37	37	INP5	Uncommitted Input 5
38	40	INP4/LW	Uncommitted Input 4
39	39	INP3/LZ	Uncommitted Input 3
40	42	INP2/LY	Uncommitted Input 2
41	41	INP1/LX	Uncommitted Input 1
42	44	INCOM	Input common
43	43	GND	Ground
44	46	WAB-	W Auxiliary encoder B-
45	45	WAB+	W Auxiliary encoder B+
46	48	WAA-	W Auxiliary encoder A-
47	47	WAA+	W Auxiliary encoder A+
48	50	ZAB-	Z Auxiliary encoder B-
49	49	ZAB+	Z Auxiliary encoder B+
50	52	ZAA-	Z Auxiliary encoder A-
51	51	ZAA+	Z Auxiliary encoder A+
52	54	YAB-	Y Auxiliary encoder B-
53	53	YAB+	Y Auxiliary encoder B+
54	56	YAA-	Y Auxiliary encoder A-
55	55	YAA+	Y Auxiliary encoder A+
56	58	XAB-	X Auxiliary encoder B-

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

57	57	XAB+	X Auxiliary encoder B+
Rev A + B Terminal #	Rev C Terminal #	Label	Description
58	60	XAA-	X Auxiliary encoder A-
59	59	XAA+	X Auxiliary encoder A+
60	62	GND	Ground
61	61	5V	5 Volts
62	64	LSCOM	Limit common
63	63	FLSX	X Forward limit
64	66	RLSX	X Reverse limit
65	65	HOMEX	X Home Input
66	68	FLSY	Y Forward limit
67	67	RLSY	Y Reverse limit
68	70	HOMEY	Y Home
69	69	FLSZ	Z Forward limit
70	72	RLSZ	Z Reverse limit
71	71	HOMEZ	Z Home
72	74	FLSW	W Forward limit
73	73	RLSW	W Reverse limit
74	76	HOMEW	W Home
75	75	GND	Ground
76	78	ABORT	Abort input
77	77	XA+	X Main encoder A+
78	80	XA-	X Main encoder A-
79	79	XB+	X Main encoder B+
80	82	XB-	X Main encoder B-
81	81	XI+	X Main encoder I+
82	84	XI-	X Main encoder I-
83	83	YA+	Y Main encoder A+
84	86	YA-	Y Main encoder A-
85	85	YB+	Y Main encoder B+
86	88	YB-	Y Main encoder B-
87	87	YI+	Y Main encoder I+
88	90	YI-	Y Main encoder I-
89	89	ZA+	Z Main encoder A+
90	92	ZA-	Z Main encoder A-
91	91	ZB+	Z Main encoder B+
92	94	ZB-	Z Main encoder B-
93	93	ZI+	Z Main encoder I+
94	96	ZI-	Z Main encoder I-
95	95	WA+	W Main encoder A+
96	98	WA-	W Main encoder A-

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

97	97	WB+	W Main encoder B+
98	100	WB-	W Main encoder B-
Rev A + B Terminal #	Rev C Terminal #	Label	Description
99	99	WI+	W Main encoder I+
100	102	WI-	W Main encoder I-
101	101	+12V	12V supply
102	104	-12V	-12V supply
103	103	5V	5V supply
104		GND	Ground

J2 - Main (60 pin IDC)

J3 - Aux Encoder (20 pin IDC)

J4 - Driver (20 pin IDC)

J5 - General I/O (26 pin IDC)

Connectors are the same as described in section entitled

"Connectors for DMC-1500 Main Board". see pg. 149

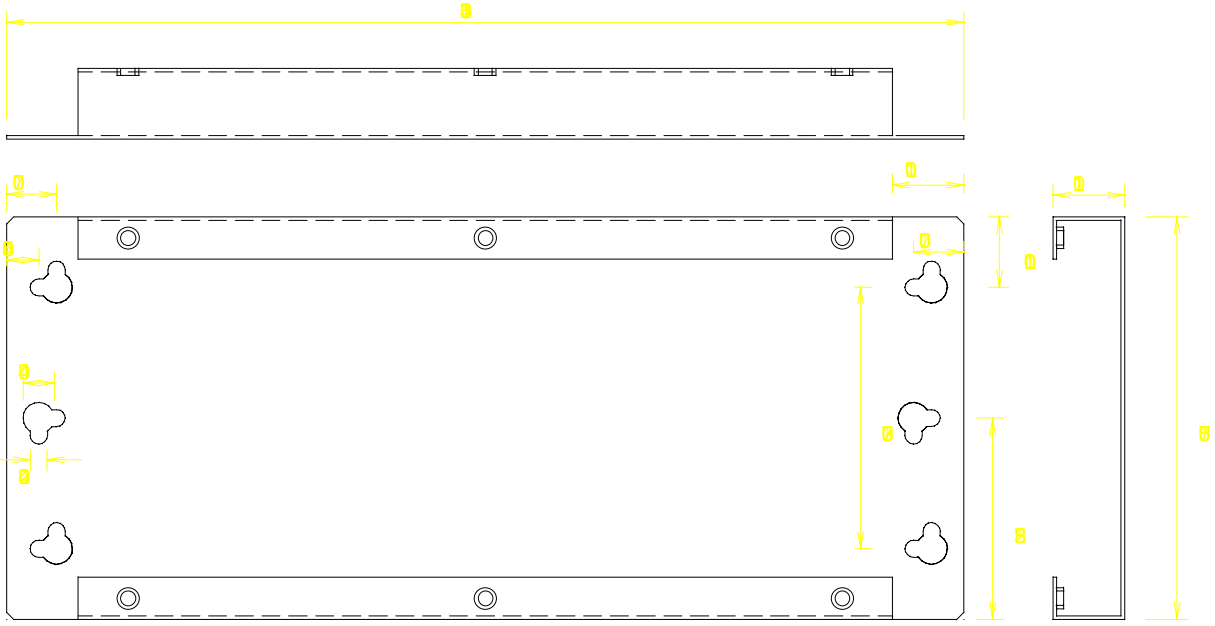
JX6, JY6, JZ6, JW6 - Encoder Input (10 pin IDC)

1 CHA	2 +VCC
3 GND	4 No Connection
5 CHA -	6 CHA
7 CHB -	8 CHB
9 INDEX -	10 INDEX

***CAUTION:** The ICM-1100 10-pin connectors are designed for the N23 and N34 encoders from Galil. If you are using Galil's Motor-5-500, Motor-50-1000 or Motor-500-1000, you must cut encoder wires 5, 6, 7 and 9.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ICM-1100 Drawing



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

AMP-11x0 Mating Power Amplifiers

The AMP-11X0 series are mating, brush-type servo amplifiers for the DMC-1500. The AMP-1110 contains one amplifier; the AMP-1120, two amplifiers; the AMP-1130, three; and the AMP-1140, four. Each amplifier is rated for 7 amps continuous, 10 amps peak at up to 80 volts. The gain of the AMP-11X0 is 1 amp/volt. The AMP-11X0 requires an external DC supply. The AMP-11X0 connects directly to the DMC-1500 ribbon connectors, and screw-type terminals are provided for connection to motors, encoders and external switches.

Features

- 6 amps continuous, 10 amps peak; 20 to 80 volts.
- Available with 1, 2, 3, or 4 amplifiers.
- Connects directly to DMC-1500 series controllers via ribbon cables.
- Screw-type terminals for easy connection to motors, encoders and switches.
- Steel mounting plate with 1/4" keyholes.

Specifications

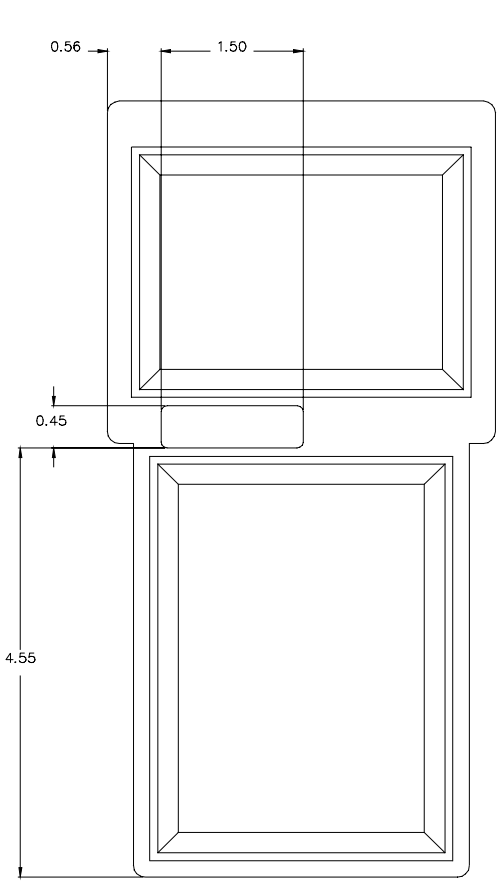
Minimum motor inductance:	1 mH
PWM frequency	30 KHz
Ambient operating temperature	0-70° C
Dimensions	5.7" x 13.4" x 2.5"
Weight	4 pounds
Mounting	Keyholes - 1/4Φ
Gain	1 amp/volt

TERM-1500 Operator Terminal

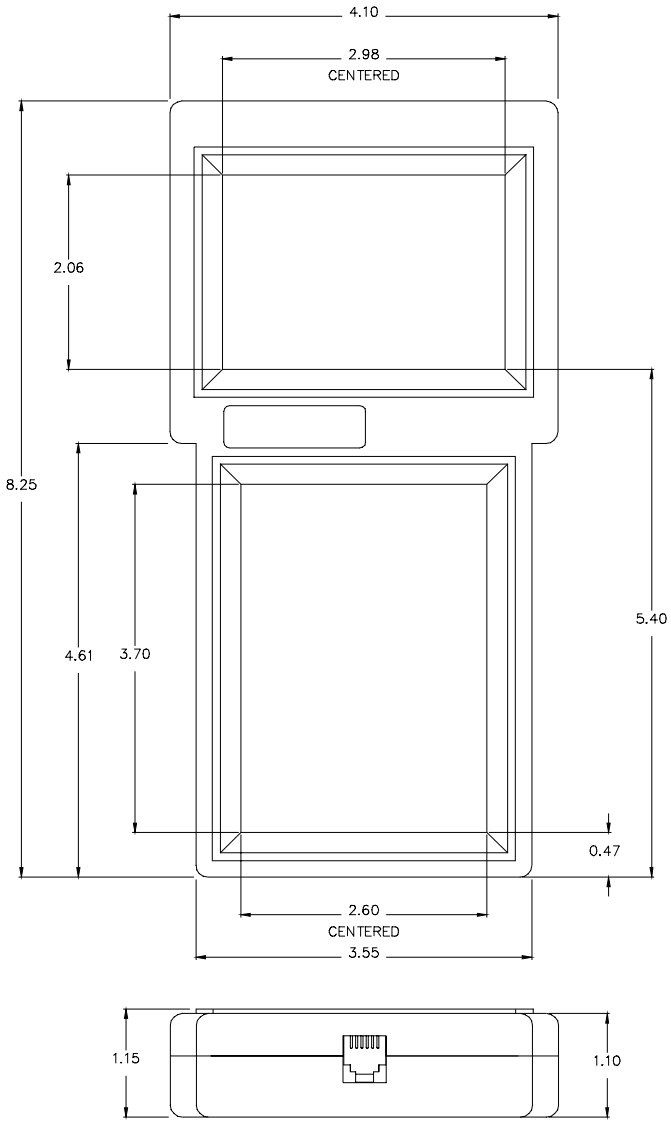
Two types of terminals are offered from Galil; the handheld unit and the panel mount unit. Both have the same programming characteristics.

Hand held unit is shown in the in the following:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

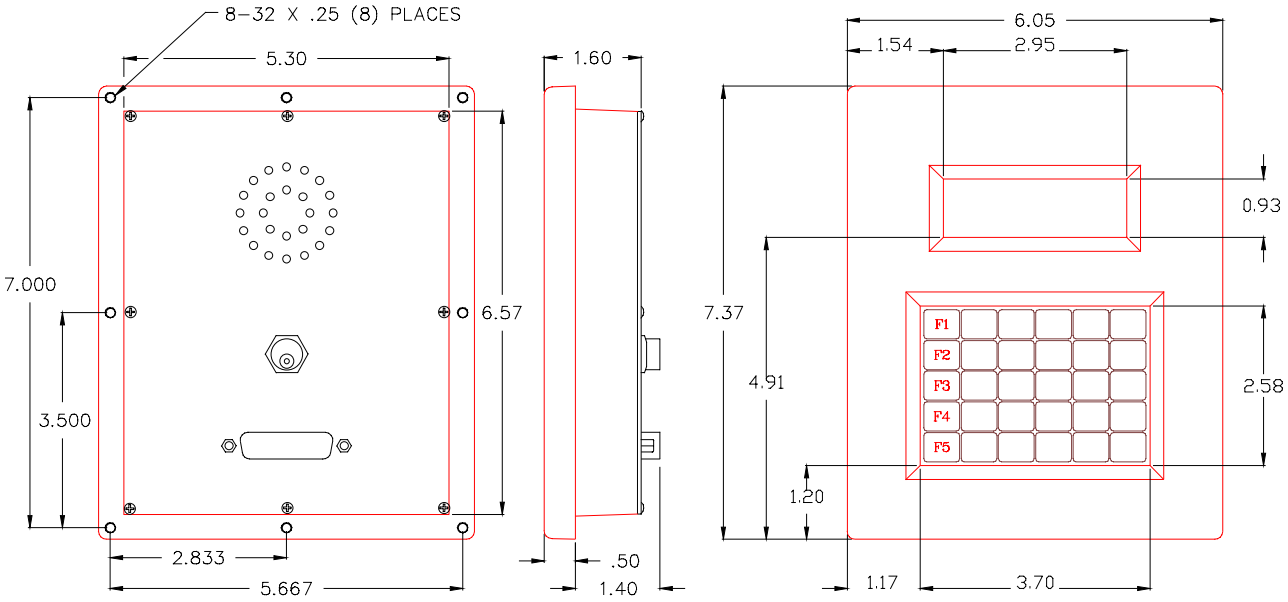


LOGO TAG AREA DIMENSION



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The panel mount terminal is shown following:



Features

- For easy data entry to DMC-1500 motion controller
- 4 line x 20 character Liquid Crystal Display
- Full numeric keypad
- Five programmable function keys
- Available in Hand-held or Panel Mount
- No external power supply required
- Connects directly to RS232 port P2 via coiled cable

Description

The TERM-1500 is a compact ASCII terminal for use with the DMC-1500 motion controller. Its numeric keypad allows easy data entry from an operator. The TERM-1500 is available with a male adapter for connection to P2 (Dataset).

NOTE: Since the TERM-1500 requires +5V on pin 9 of RS-232, it can only work with port 2 of the DMC-1500.

Specifications - Hand-Held

Keypad	Key Tactile 4 row x 5 character
Display	LCD with 5 by 7 character font
Power	5 Volts, 30mA (from DMC-1500)

Specifications - Panel Mount

Keypad	30-Key; 5 rows x 6 columns; 5x7 font
Display	4 row x 20 character LCD

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Keypad Maps - Hand-Held

30 Keys: 5 keys across, 6 down

Single Key Output

6	F1 (22)	F2 (23)	F3 (24)	F4 (25)	F5 (26)
5		1	2	3	
4		4	5	6	
3		7	8	9	
2			0		
1	CTRL	SHIFT	SPACE	BKSPC	ENTER

Shift Key Output

6	A	B	C	D	E
5	F	G	H	I	J
4	K	L	M	N	O
3	P	Q	R	S	T
2	U	V	W	X	Y
1	CTRL	SHIFT	Z	,	?

CTRL Key Output

6	(18)	(16)	(9)	(4)	(17)
5	(19)	(2)	!	"	%
4	*	+	/	$	;
3	<	>	\	[	]
2	^	-	@	{	}
1	CTRL	SHIFT	ESC	=	#

Note: Values in parentheses are ASCII decimal values. Key locations are represented by [m,n] where m is element column, n is element row.

Example:

U is <Shift>[1,2]

is <Cntrl>[5,1]

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Keypad Map - Panel Mount -- 6 columns x 5 rows

Single Key Output

5	F1		1	2	3	
4	F2		4	5	6	
3	F3		7	8	9	
2	F4		-	0	.	
1	F5	CTRL	SHIFT	SPACE	BKSPC	ENTER

Shift Key Output

5	A	F	G	H	I	J
4	B	K	L	M	N	O
3	C	P	Q	R	S	T
2	D	U	V	W	X	Y
1	E	CTRL	SHIFT	Z	,	?

CTRL Key Output

5	(18)	(19)	(2)	!	"	%
4	(16)	*	+	/	$	;
3	(9)	<	>	\	[	]
2	(4)	^	-	@	{	}
1	(17)	CTRL	SHIFT	ESC	=	#

Note: Values in parentheses are ASCII decimal values. Key locations are represented by [m,n] where m is element column, n is element row.

Escape Commands

Escape codes can be used to control the TERM-1500 display, cursor style, and position, and sound settings.

Note: The escape character (hex 1B) can be sent through port 2 of the DMC-1500 with special syntax {^27}:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Example: MG {P2}{^27},"H" Sends escape H to the terminal from port 2

Cursor Movement Commands

ESC A	Cursor Up
ESC B	Cursor Down
ESC C	Cursor Right
ESC D	Cursor Left

Erasing Display

ESC E	Clear Display and Home
ESC I	Clear Display
ESC J	Cursor to End of Display
ESC K	Cursor to End of Line
ESC M	Line Containing Cursor

Sounds

ESC T	Short Bell
ESC L	Long Bell
ESC P	Click
ESC Q	Alert

Cursor Style

ESC F	Underscore Cursor On
ESC G	Underscore Cursor Off
ESC R	Blinking Cursor On
ESC S	Blinking Cursor Off

Key Clicks (audible sounds from terminal)

ESC U	Key Click Enable
ESC V	Key Click Disable

Identify (sends "TT!" then terminal firmware version)

ESC Z	Send Terminal ID

Cursor Position

ESC Y	Pr Pc

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In the above sequence, Pr is the row number and Pc is the column number of the target cursor location. These parameters are formed by adding hexadecimal 1F to the row and column numbers. Row and column numbers are absolute, with row 1, column 1 (Pr = H20, Pc = H20) representing the upper left corner of the display.

Configuration

<CNTRL><SHIFT>F1 Allows user to configure terminal; Follow prompts on display to change configuration

Default Configuration:

Baude Rate 9600

Data bits

Parity

Display

Repeat

Echo

Handshake Disabled

Self Test

Key Click - Disabled <Ctrl>Space <Shift> [2,2]

Key Click - Enabled <Ctrl>Space <Shift> [1,2]

Clear Display and Home <Ctrl>Space <Shift> [5,6]

Function Keys

<CNTRL><SHIFT>F3 Allows function keys to be configured; Follow prompts on display to change function keys

Default Function Keys

F1 22 decimal

F2 23 decimal

F3 24 decimal

F4 25 decimal

F5 26 decimal

Input/Output of Data -- DMC-1500 Commands

Refer to Chapter 7 in this manual for Data Communication commands.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When using Port 2, use CC command to configure P2. Use male adapter.

Example:

CC 9600,0,0,1	Configures P2
MG{P2} "Hello There", V1{F2.1}	Send message to P2
IN{P2} "Enter Value", NUM	Prompts operator for value

Example:

#A

CI 0,2,1;CC 9600,0,0,1	#A Interrupt on any key; Configure P2
MG {P2} "press F1 to start X"	Print Message to P2
MG {P2} "Press F2 to start Y"	Print Message to P2
#B; JP#B;EN	End Program
#COMINT	Interrupt Routine
JS #XMOVE,P2CH=F1	Jump to X move if F1
JS #YMOVE,P2CH=F2	Jump to Y move if F2
EN1,1	End, Re-enable comm interrupt & restore trippoint
#XMOVE;PR1000;BGX;EN	Move X routine
#YMOVE;PR,1000;BGY;EN	Move Y routine

Note: F1 through F5 are used as dedicated keywords for testing function keys. Do not use these as variables.

Pin-Out

6-Pin Modular Connector

1 +5 Volts
2 Handshake in
3 Handshake out
4 Data in
5 Data out
6 Ground

9-Pin D Adapter - Male (For P2)

1. CTS input
2. Transmit Data - input
3. Receive Data - output
4. RTS - output

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5. Ground
6. CTS - input
7. RTS - output
8. CTS - input
9. 5V or no connect or sample clock with jumpers

NOTE: Out and in are referenced to the terminal.

Ordering Information

TERM-1500H-P2 Hand-held, female adapter

TERM-1500P-P2 Panel Mount, female adapter

DB-15072 OPTO-22 Expansion Option

The DB-15072 is a version of the DMC-1500 that provides an interface to up to 72 OPTO 22 I/O modules. The first 24 I/O points can be configured through software. I/O points 25 through 48 can be configured as inputs or outputs (in groups of 8) and the I/O configuration options are shown below. I/O points 49 through 96 are always inputs. Each group of 24 I/O points is brought out via a separate 50 pin cable which plugs directly into an H-style OPTO 22 rack.

Configuring the I/O for the DB-15072

The I/O on the DMC-1500 is organized in 8 bit increments known as blocks. The extended I/O is configured as blocks 3-9. Blocks 3-6 can be configured as inputs or outputs, using the command, CO. The command has one field:

CO n

where n is a 3-bit number represented in decimal. A 3 bit number ranges, in decimal, between 0 and 7. Each bit in the 3-bit number represents one of the 8-bit I/O blocks. If the representative bit is one, the corresponding I/O block will be configured as an output.

8-Bit I/O Block	Block	Bit	Binary Representation	Decimal Value of Bit
25-32	3	3	2^2	4
33-40	4	2	2^1	2
41-48	5	1	2^0	1

The simplest method for determining the proper value for n is to do the following: 1. Choose which 8-bit I/O blocks that should be configured as outputs. 2. From the table, determine the decimal value for each I/O block to be set as an output. 3. Add up all of the values determined in step 2. This is the value to be used for n.

For example, if blocks 4 and 5 are to be outputs, then n is 3 and the command, CO3, should be issued.

This parameter, and the state of the outputs, can be stored in the EEPROM with the BN command. If no value has been set, the default of CO 0 is used (all blocks are inputs).

When configured as an output, each I/O point may be defined with the SBn and CBn commands (where n=9 through 56). OBn can also be used with n=9 through 56.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Accessing the I/O of the DB-15072

The command, OP, may be used to set the state of output bits. The OP command has 2 parameters. The first parameter sets the values of the main output port of the controller (On a DB-15072 controller, this parameter would address outputs 1-8). The second parameter sets the value of the daughter board outputs (Bits 25-48 can be configured as outputs). The command syntax for the command is the following:

OP m,n

where m is the decimal representation of the bits 1-8 (values from 0 to 255) and n is the decimal representation of the bits 25-48 (values from 0 to 65535).

When accessing I/O blocks configured as inputs, use the TIn command. The argument 'n' refers to the block to be read (n=1 to 9). Individual bits can be queried using the @IN[n] command (where n=9 to 80).

If the command below is issued

MG @IN[17]

the response is the least significant bit of block 2 (assuming block 2 is configured as input).

The outputs of the extended I/O can be monitored with the operand _OP1. The value of the operand is a decimal which represents the bits 25-96.

Connector Description of the DB-15072

Three cables connect the DB-15072 to OPTO-22 products. Pinouts are described as follows:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

JD1 Pinout

Pin	Block	Bit No.	SBn,@IN[n]	Pin	Signal
1	5	7	48	2	Ground
3	5	6	47	4	Ground
5	5	5	46	6	Ground
7	5	4	45	8	Ground
9	5	3	44	10	Ground
11	5	2	43	12	Ground
13	5	1	42	14	Ground
15	5	0	41	16	Ground
17	4	7	40	18	Ground
19	4	6	39	20	Ground
21	4	5	38	22	Ground
23	4	4	37	24	Ground
25	4	3	36	26	Ground
27	4	2	35	28	Ground
29	4	1	34	30	Ground
31	4	0	33	32	Ground
33	3	7	32	34	Ground
35	3	6	31	36	Ground
37	3	5	30	38	Ground
39	3	4	29	40	Ground
41	3	3	28	42	Ground
43	3	2	27	44	Ground
45	3	1	26	46	Ground
47	3	0	25	48	Ground
49	5 volts			50	Ground

JD2 Pinout (Note: All points are inputs on this cable)

Pin	Block	Bit No.	SBn,@IN[n]	Pin	
1	8	7	72	2	Ground
3	8	6	71	4	Ground
5	8	5	70	6	Ground
7	8	4	69	8	Ground
9	8	3	68	10	Ground
11	8	2	67	12	Ground
13	8	1	66	14	Ground
15	8	0	65	16	Ground
17	7	7	64	18	Ground
19	7	6	63	20	Ground
21	7	5	62	22	Ground
23	7	4	61	24	Ground
25	7	3	60	26	Ground

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Pin	Block	Bit No.	SBn,@IN[n]	Pin	
27	7	2	59	28	Ground
29	7	1	58	30	Ground
31	7	0	57	32	Ground
33	6	7	56	34	Ground
35	6	6	55	36	Ground
37	6	5	54	38	Ground
39	6	4	53	40	Ground
41	6	3	52	42	Ground
43	6	2	51	44	Ground
45	6	1	50	46	Ground
47	6	0	49	48	Ground
49	5 volts			50	Ground

JD3 Pinout (Note: All points are inputs on this cable):

Pin	Block	Bit No.	@IN[n]	Pin	
1	11	7	96	2	Ground
3	11	6	95	4	Ground
5	11	5	94	6	Ground
7	11	4	93	8	Ground
9	11	3	92	10	Ground
11	11	2	91	12	Ground
13	11	1	90	14	Ground
15	11	0	89	16	Ground
17	10	7	88	18	Ground
19	10	6	87	20	Ground
21	10	5	86	22	Ground
23	10	4	85	24	Ground
25	10	3	84	26	Ground
27	10	2	83	28	Ground
29	10	1	82	30	Ground
31	10	0	81	32	Ground
33	9	7	80	34	Ground
35	9	6	79	36	Ground
37	9	5	78	38	Ground
39	9	4	77	40	Ground
41	9	3	76	42	Ground
43	9	2	75	44	Ground
45	9	1	74	46	Ground
47	9	0	73	48	Ground
49	5 volts			50	Ground

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Coordinated Motion - Mathematical Analysis

The terms of coordinated motion are best explained in terms of the vector motion. The vector velocity, Vs, which is also known as the feed rate, is the vector sum of the velocities along the X and Y axes, Vx and Vy.

$$Vs = \sqrt{Vx^2 + Vy^2}$$

The vector distance is the integral of Vs, or the total distance traveled along the path. To illustrate this further, suppose that a string was placed along the path in the X-Y plane. The length of that string represents the distance traveled by the vector motion.

The vector velocity is specified independently of the path to allow continuous motion. The path is specified as a collection of segments. For the purpose of specifying the path, define a special X-Y coordinate system whose origin is the starting point of the sequence. Each linear segment is specified by the X-Y coordinate of the final point expressed in units of resolution, and each circular arc is defined by the arc radius, the starting angle, and the angular width of the arc. The zero angle corresponds to the positive direction of the X-axis and the CCW direction of rotation is positive. Angles are expressed in degrees, and the resolution is 1/256th of a degree. For example, the path shown in Fig. 12.2 is specified by the instructions:

VP	0,10000
CR	10000, 180, -90
VP	20000, 20000

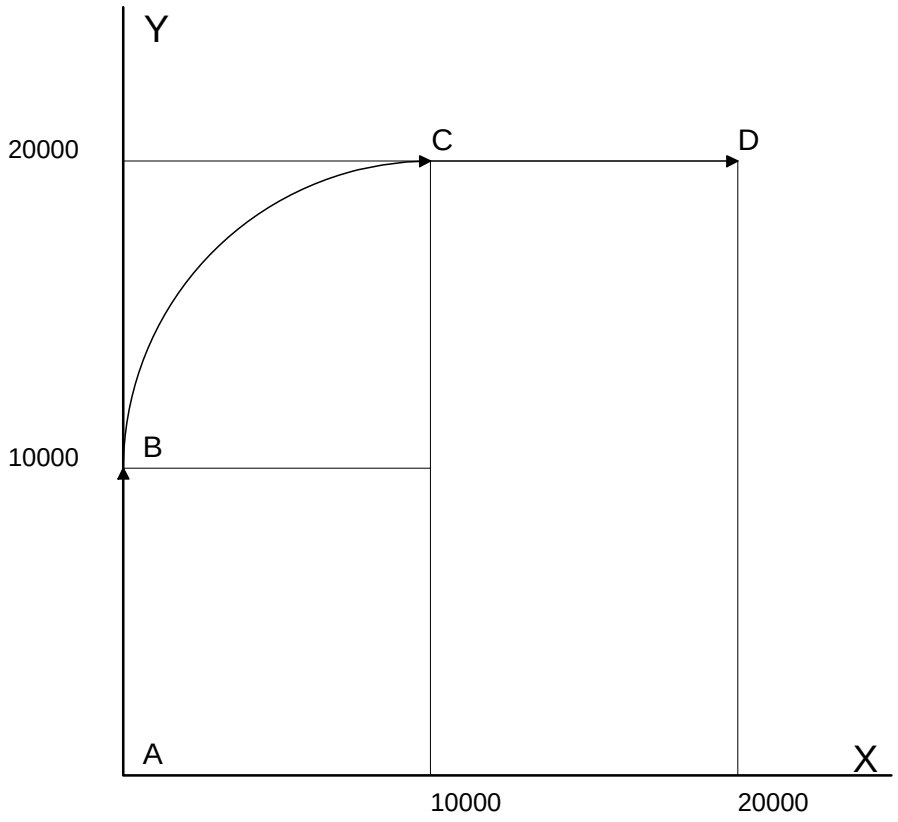


Figure 12.2 - X-Y Motion Path

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The first line describes the straight line vector segment between points A and B. The next segment is a circular arc, which starts at an angle of 180° and traverses -90°. Finally, the third line describes the linear segment between points C and D. Note that the total length of the motion consists of the segments:

A-B	Linear	10000 units
B-C	Circular	$\frac{R\lvert\Delta\theta\rvert 2\pi}{360}$ = 15708
C-D	Linear	10000
	Total	35708 counts

In general, the length of each linear segment is

$$L_k = \sqrt{Xk^2 + Yk^2}$$

Where Xk and Yk are the changes in X and Y positions along the linear segment. The length of the circular arc is

$$L_k = R_k\lvert\Delta\Theta_k\rvert 2\pi/360$$

The total travel distance is given by

$$D = \sum_{k=1}^{n} L_k$$

The velocity profile may be specified independently in terms of the vector velocity and acceleration.

For example, the velocity profile corresponding to the path of Fig. 12.2 may be specified in terms of the vector speed and acceleration.

VS	100000
VA	2000000

The resulting vector velocity is shown in Fig. 12.3.

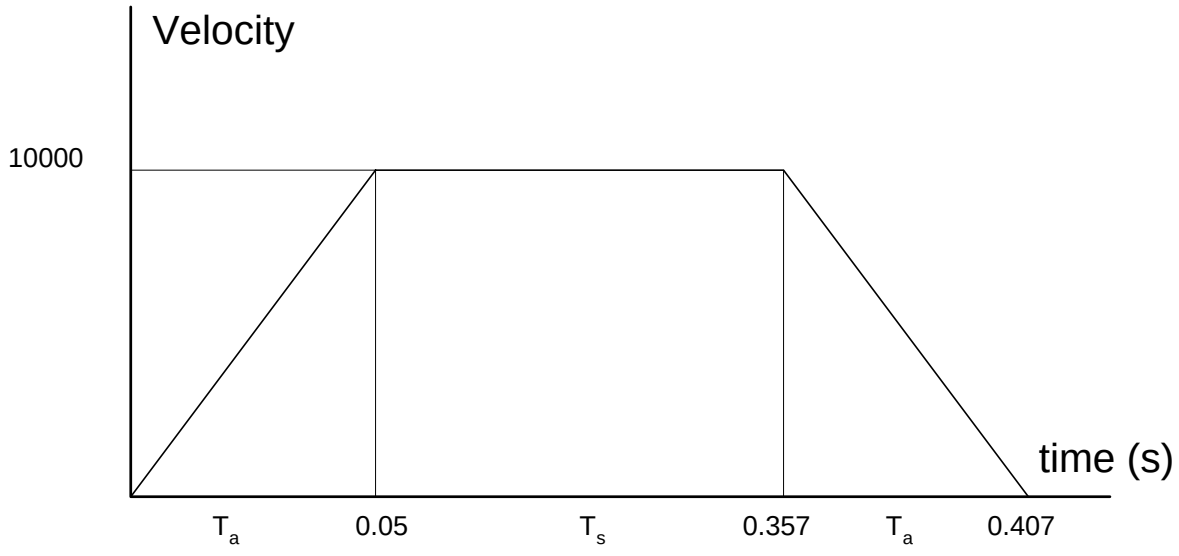


Figure 12.3 - Vector Velocity Profile

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The acceleration time, T_a, is given by

$$T_a = \frac{VS}{VA} = \frac{100000}{2000000} = 0.05s$$

The slew time, Ts, is given by

$$T_s = \frac{D}{VS} - T_a = \frac{35708}{100000} - 0.05 = 0.307s$$

The total motion time, Tt, is given by

$$T_t = \frac{D}{VS} + T_a = 0.407s$$

The velocities along the X and Y axes are such that the direction of motion follows the specified path, yet the vector velocity fits the vector speed and acceleration requirements.

For example, the velocities along the X and Y axes for the path shown in Fig. 12.2 are given in Fig. 12.4.

Fig. 12.4a shows the vector velocity. It also indicates the position point along the path starting at A and ending at D. Between the points A and B, the motion is along the Y axis. Therefore,

Vy = Vs

and

Vx = 0

Between the points B and C, the velocities vary gradually and finally, between the points C and D, the motion is in the X direction.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

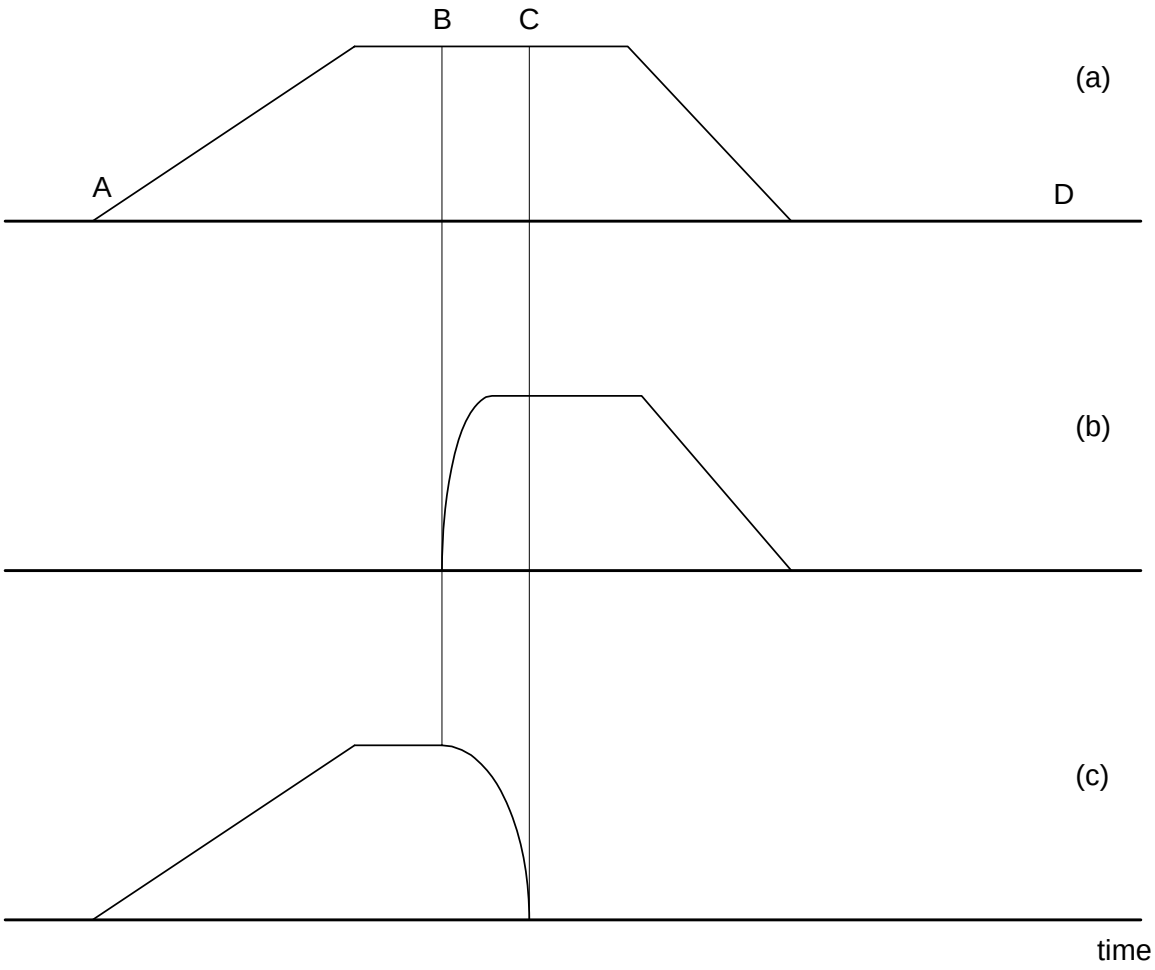


Figure 12.4 - Vector and Axes Velocities

DMC-700/DMC-1500 Comparison

Modes of Motion	DMC-700	DMC-1500
Relative positioning	Yes	Yes
Absolute positioning	Yes	Yes
Velocity control	Yes	Yes
Linear interpolation	Up to 4 axes	Up to 8 axes
Circular interpolation	Any 2 axes plus 3rd tangent	Any 2 axes plus 3rd tangent
Maximum number of segments in motion path	255	Infinite, continuous vector feed
Contouring	Yes	Yes
Electronic gearing	Yes	Yes
S-curve profiling	Yes	Yes
Programmable acceleration rate	Yes	Yes
Programmable deceleration rate	No	Yes

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Specifications	DMC-700	DMC-1500
Maximum encoder frequency	2 x 10^6 counts/s	8 x 10^6 counts/s
DAC resolution	12-bits	14-bits or 16-bits
Maximum move length	8 x 10^6	2 x 10^9
Sample time	1 msec	0.5 msec (4 axes)
Program memory	500 lines, 32 char	1000 lines, 80 char
EEPROM memory for parameter storage	Yes	Yes
Number of variables	126; symbolic up to 8 chars.	254; symbolic up to 8 chars.
Number of array elements	1600 (up to 14 arrays)	8000 (up to 30 arrays)
Digital filter type	GN,ZR,KI	KP,KI,KD with velocity and acceleration feedforward and integrator limit

Hardware	DMC-700	DMC-1500
Maximum # of axes/card	4	4 (8 for DMC-1580)
Analog inputs	8	7 standard
Digital inputs	8 opto-isolated	8 opto-isolated (24 for DMC-1580)
Digital outputs	8 opto-isolated	8 TTL (16 for DMC-1580)
High speed position latch	Yes	Yes
Dual encoder inputs	Yes	Yes
Motor command output	+/- 10V	+/- 10V and step/direction
Dimensions	14” x 4” x 10”	13” x 2.5” x 6.6”

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

List of Other Publications

"Step by Step Design of Motion Control Systems"

by Dr. Jacob Tal

"Motion Control Applications"

by Dr. Jacob Tal

"Motion Control by Microprocessors"

by Dr. Jacob Tal

Contacting Us

Galil Motion Control

270 Technology Way

Rocklin, CA 95765

Phone: 916-626-0101

Fax: 916-626-0102

Internet address: support@galilmc.com

URL: www.galilmc.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

WARRANTY

All products manufactured by Galil Motion Control are warranted against defects in materials and workmanship. The warranty period for controller boards is 1 year. The warranty period for all other products is 180 days.

In the event of any defects in materials or workmanship, Galil Motion Control will, at its sole option, repair or replace the defective product covered by this warranty without charge. To obtain warranty service, the defective product must be returned within 30 days of the expiration of the applicable warranty period to Galil Motion Control, properly packaged and with transportation and insurance prepaid. We will reship at our expense only to destinations in the United States.

Any defect in materials or workmanship determined by Galil Motion Control to be attributable to customer alteration, modification, negligence or misuse is not covered by this warranty.

EXCEPT AS SET FORTH ABOVE, GALIL MOTION CONTROL WILL MAKE NO WARRANTIES EITHER EXPRESSED OR IMPLIED, WITH RESPECT TO SUCH PRODUCTS, AND SHALL NOT BE LIABLE OR RESPONSIBLE FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES.

COPYRIGHT (10-94)

The software code contained in this Galil product is protected by copyright and must not be reproduced or disassembled in any form without prior written consent of Galil Motion Control, Inc.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

THIS PAGE LEFT BLANK INTENTIONALLY

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Index

A

Abort 25–27, 30, 45, 51, 129, 131, 151, 154, 162–63, 169
  Off-On-Error 12, 27, 30, 129, 131
  Stop Motion 45, 51, 99, 132
Absolute Position 20, 41–42, 91–92, 96
Absolute Value 96, 103, 130
Acceleration 185–86
Accessories 166
Address 107–9, 165, 166–67, 188
  Jumpers 33
AMP-1100 15, 167
Amplifier Enable 31–32, 129
Amplifier Gain 4, 142, 145, 147
Amplifiers 4
Analog Input 25, 31, 103–5, 106, 114, 121–22, 126, 151, 167
Analysis
  SDK 81
Arithmetic Functions 81, 95, 102, 104, 118
Arm Latch 78
Array 49, 65–67, 81, 87, 88, 95, 102, 106–17, 119, 152
Autocad 166
Automatic Subroutine 84, 98
  CMDERR 84, 98, 100
  LIMSWI 25, 84, 97–98, 130–32
  MCTIME 84, 91, 98, 100
  POSERR 84, 97–99, 130–31
Auxiliary Board 157
Auxiliary Encoder 25, 55, 70–71, 70–71, 158, 163, 168
  Dual Encoder 71, 108

B

Backlash 70–71, 127
Backlash Compensation
  Dual Loop 70–71, 70–71, 127
Baud Rate 10, 34
Bit-Wise 95, 101
Bypassing Optoisolation 29

C

Capture Data
  Record 64, 67, 106, 109, 110
Circle 124
Circular Interpolation 1, 24, 50–53, 55, 108, 124
Clear Bit 119
Clear Sequence 45, 47, 51, 53
Clock 106
CMDERR 84, 98, 100
Command
  Syntax 37–38
Command Summary 40, 106, 108
Commanded Position 42–43, 55–56, 100, 108, 122, 137–39
Commdisk 7, 9
Communication 1, 111
  Baud Rate 10, 34
  FIFO 36
  Handshake 34
  Serial Ports 3, 10
Compensation
  Backlash 70–71, 127
Conditional jump 22, 27, 81, 92–97, 121
Configuration
  Jumper 29
Connector 25, 28, 32
Contour Mode 62–67
Control Filter
  Damping 134, 140
  Integrator 140, 144–45
  Proportional Gain 140
Coordinated Motion 1, 38, 50–53
  Circular 24, 50–53, 55, 108, 124
  Contour Mode 62–67
  Ecam 58, 61
  Electronic Cam 57, 59
  Electronic Gearing 54–57
  Gearing 54–57
  Linear Interpolation 23, 44–47, 44–47, 49, 55, 63
Cosine 102–3, 107
Cycle Time
  Clock 106

D

DAC 140, 144–45, 147
Damping 134, 140
Data Capture 107–9
Data Output
  Set Bit 119
Debugging 86, 88
Differential Encoder 12, 14, 134

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Digital Filter 144–45, 147–49
  Tuning 1
Digital Input 25, 27, 103, 120
Digital Output 103, 119
  Clear Bit 119
Dip Switch
  Address 165, 166–67, 188
Download 81, 108
Dual Encoder 71, 108
  Backlash 70–71, 127
  Dual Loop 70–71, 70–71, 127
Dual Loop 70–71, 70–71, 127
  Backlash 70–71, 127

E

Ecam 58, 61
  Electronic Cam 57, 59
Echo 34–36
Edit Mode 21, 81–82, 89
Editor 21–22, 81–82
EEPROM 9
Electronic Cam 57, 59
Electronic Gearing 1, 54–57
  Gearing 1
Ellipse Scale 53
Enable
  Amplifer Enable 31–32, 129
Encoder 39
  Auxiliary Encoder 25, 55, 70–71, 70–71, 158, 163, 168
  Differential 12, 14, 134
  Dual Encoder 71, 108
  Index Pulse 12, 26, 75
  Quadrature 4, 118, 122, 130, 143
Encoders
  Auxiliary Encoders 1
  Quadrature 3
Error
  Handling 1
Error Handling 25, 84, 97–98, 130–32
Error Limit 12, 13, 18, 30, 98, 129–31
  Off-On-Error 12, 27, 30, 129, 131
Example
  Wire Cutter 122
Excessive Error 1
Execute Program 22–23

F

Feedrate 47, 52, 53, 93, 123–24
FIFO 36
Filter Parameter
  Damping 134, 140
  Integrator 140, 144–45
  PID 14, 140, 144, 149
  Proportional Gain 140
  Stability 71, 127, 133–34
Find Edge 26, 75
Formatting 114, 115–18
Frequency 5, 146–48
Function 26–27, 101–7, 124, 126
Functions
  Arithmetic 81, 95, 102, 104, 118

G

Gain
  Proportional 140
Gear Ratio 54–56
Gearing 1, 54–57

H

Halt 46, 51, 85–91, 92–94, 120
  Abort 25–27, 30, 45, 51, 129, 131, 151, 154, 162–63, 169
  Off-On-Error 12, 27, 30, 129, 131
  Stop Motion 45, 51, 99, 132
Hardware 119, 129
  Address 107–9, 165, 166–67, 188
  Amplifier Enable 31–32, 129
  Clear Bit 119
  Jumper 29
  Offset Adjustment 32, 133
  Output of Data 113
  Set Bit 119
  TTL 5, 25, 27, 31–32, 129
Home Input 26, 75, 106
Homing 26, 75
  Find Edge 26, 75

I

I/O
  Amplifier Enable 31–32, 129
  Analog Input 103–5, 106, 114, 121–22, 126
  Clear Bit 119
  Digital Input 25, 27, 103, 120
  Digital Output 103, 119
  Home Input 26, 75, 106
  Output of Data 113
  Set Bit 119
  TTL 5, 25, 27, 31–32, 129
ICM-1100 7–8, 12, 25, 29, 30, 129
Independent Motion
  Jog 20, 92–93, 99–100, 105, 126, 130
Index Pulse 12, 26, 75
ININT 84, 98–99, 120–21
Input

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Analog 103–5, 106, 114, 121–22, 126
Digital 103, 120
Input Interrupt 84, 92, 98–99, 120–21
ININT 84, 98–99, 120–21
Input of Data 110
Inputs 1
Analog 25, 31, 151, 167
Analog Inputs 1
Encoder Input 1
Interconnect Module 9
Optoisolated Inputs 1, 3
Installation 8–14, 133
Integrator 140, 144–45
Interconnect Module 9
AMP-1100 167
ICM-1100 12, 25, 29, 30, 129
Interface
Terminal 81, 104, 115
Internal Variable 23, 95, 104, 105
Interrogation 19, 21, 39–40, 113, 115
Interrupt 84–85, 92, 97–99, 120–21
Invert Loop Polarity 134

J

Jog 20, 92–93, 99–100, 105, 126, 130
Joystick 105, 126
Jumper 29
Jumpers 1, 33

K

Keyword 95, 102, 104, 106–7
TIME 106–7

L

Label 29, 81–89, 91–99, 105–6, 110, 115, 119–22, 124, 126–28, 131
LIMSWI 130–32
POSERR 130–31
Special Label 84, 131
Latch 30, 78
Arm Latch 78
Data Capture 107–9
Position Capture 78
Record 64, 67, 106, 109, 110
Teach 67
Limit
Torque Limit 14, 21
Limit Switch 25–27, 30, 84–85, 98, 106, 130–32, 134
LIMSWI 25, 84, 97–98, 130–32
Linear Interpolation 23, 44–47, 44–47, 49, 55, 63
Clear Sequence 45, 47, 51, 53
Logical Operator 97
Logical Operators 111

M

Masking
Bit-Wise 95, 101
Math Function
Absolute Value 96, 103, 130
Bit-Wise 95, 101
Cosine 102–3, 107
Logical Operator 97
Sine 103
Math Functions
Logical Operators 111
Mathematical Expression 95, 101, 103
MCTIME 84, 91, 98, 100
Memory 1, 21, 81, 87, 88, 97, 98, 106, 108
Array 49, 65–67, 81, 87, 88, 95, 102, 106–17, 119, 152
Download 81, 108
Upload 81
Message 87, 88, 98–100, 102, 109–14, 121, 131–32
Messages 34
Modeling 137, 140–41, 145
Motion Complete
MCTIME 84, 91, 98, 100
Motion Smoothing 1, 72, 74
S-Curve 46, 51
Motor Command 1, 14, 21, 32, 144
Moving
Acceleration 185–86
Circular 24, 50–53, 55, 108, 124
Circular Interpolation 1
Coordinated Motion 1
Slew Speed 1
Multitasking 1, 85
Execute Program 22–23
Halt 46, 51, 85–91, 92–94, 120

N

Non-volatile Memory 1–3

O

OE
Off-On-Error 129, 131
Off-On-Error 12, 27, 30, 129, 131
Offset Adjustment 32, 133
Operand
Internal Variable 23, 95, 104, 105
Operators
Bit-Wise 95, 101
Optoisolated Inputs 1, 3
Optoisolation 25, 27–29, 30

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Home Input 26, 75, 106
Output
Amplifier Enable 31–32, 129
ICM-1100 12, 25, 29, 30
Interconnect Module 7–8
Motor Command 14, 21, 32, 144
Output of Data 113
Clear Bit 119
Set Bit 119
Outputs 1, 33
Interconnect Module 9
Motor Command 1

P

PID 1, 14, 140, 144, 149
Play Back 109
POSERR 84, 97–99, 130–31
Position Error 13, 20
Position Capture 78
Latch 30, 78
Teach 67
Position Error 12, 13, 20, 30, 84, 98–99, 105, 108, 122, 127, 129–31, 134, 139
Position Follow 121–22
Position Limit 130
Program Flow 83, 89
Interrupt 84–85, 92, 97–99, 120–21
Stack 97, 100, 121
Programmable 119, 126, 130
Programming
Halt 85–91, 92–94, 120
Proportional Gain 140
Protection
Error Limit 12, 13, 18, 30, 98, 129–31
Torque Limit 14, 21
PWM 4

Q

Quadrature 3, 4, 118, 122, 130, 143
Quit
Abort 25–27, 30, 45, 51, 129, 131, 151, 154, 162–63, 169
Stop Motion 45, 51, 99, 132

R

Record 64, 67, 106, 109, 110
Latch 30, 78
Position Capture 78
Teach 67
Register 105
Reset 25, 31, 94, 129, 131

S

SB
Set Bit 119
Scaling
Ellipse Scale 53
S-Curve
Motion Smoothing 1, 74
SDK 81
Selecting Address 107–9, 165, 166–67, 188
Serial Port 3, 10
Servo Design Kit
SDK 81
Set Bit 119
Sine 103
Single-Ended 4, 12, 14
Slew 91, 93, 122
Slew Speed 1
Smoothing 1, 46, 48, 51, 53, 72–74
Software
Autocad 166
Commdisk 7, 9
SDK 81
Terminal 81, 104, 115
Special Label 84, 131
Stability 71, 127, 133–34, 140, 146
Stack 97, 100, 121
Zero Stack 100, 121
Status 48, 88–90, 105, 109
Interrogation 19, 21, 39–40, 54, 113, 115
Stop Code 109, 134
Step Motor 7, 74–75
KS, Smoothing 1, 46, 48, 51, 53, 72–74
Step Motors 1
Stop
Abort 25–27, 30, 45, 51, 129, 131, 151, 154, 162–63, 169
Stop Code 109, 134
Stop Motion 45, 51, 99, 132
Subroutine 25, 84, 94–99, 121, 130–31
Automatic Subroutine 84, 98
Synchronization 4, 33, 57
Syntax 37–38

T

Tangent 50, 52–53
Teach 67
Data Capture 107–9
Latch 30, 78
Play-Back 109
Position Capture 78
Record 64, 67, 106, 109, 110
Tell Error 89
Position Error 13, 20

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tell Position 36, 92, 105–7
Term-1500 33
Terminal 26, 29, 81, 104, 115
Theory 137
  Damping 134, 140
  Digital Filter 144–45, 147–49
  Modeling 137, 140–41, 145
  PID 14, 140, 144, 149
  Stability 71, 127, 133–34, 140, 146
TIME 106–7
  Clock 106
Time Interval 63–64, 67, 108
Timeout 10, 84, 91, 98, 99
  MCTIME 84, 91, 98, 100
Torque Limit 14, 21
Trigger 81, 89, 91–94
Trippoint 91, 97
Troubleshooting 133
TTL 5, 25, 27, 31–32, 129
Tuning 1
  SDK 81
  Stability 71, 127, 133–34, 140, 146

U

Upload 81
User Unit 118

V

Variable
  Internal 23, 95, 104, 105
Vector Acceleration 24, 47–48, 53, 124
Vector Deceleration 24, 47–48, 53
Vector Mode
  Circle 124
  Circular Interpolation 24, 50–53, 55, 108, 124
  Clear Sequence 45, 47, 51, 53
  Ellipse Scale 53
  Feedrate 47, 52, 53, 93, 123–24
  Tangent 50, 52–53
Vector Speed 23–24, 45–51, 53, 93, 124

W

Wire Cutter 122

X

XQ
  Execute Program 22–23

Z

Zero Stack 100, 121

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP