Command Descriptions

Each executable instruction is listed in this chapter in alphabetical order.

The two-letter Opcode for each instruction is placed in the upper left corner. Below the opcode is a description of the command and required arguments. As arguments, some commands require actual values to be specified following the instruction. These commands are followed by lower case x,y,z,w. Values may be specified for any axis separately or any combination of axes. Axis values are separated by commas. Examples of valid x,y,z,w syntax are listed below.

VALID X,Y,Z,W SYNTAX

INSTRUCTION	INTERPRETATION
AC x	Specify x only
AC x,y	Specify x and y only
AC x,,z	Specify x and z only
AC x,y,z,w	Specify x,y,z,w
AC ,y	Specify y only
AC ,y,z	Specify y and z
AC ,,z	Specify z only
AC ,,,w	Specify w only
AC x,,,w	Specify x and w only

Where x,y,z and w are replaced by actual values.

An alternative method for specifying data is to set data for individual axes using an axis designator followed by an equals sign. The * symbol defines data for all axes to be the same. For example:

INSTRUCTION	INTERPRETATION
PRY=1000	Sets Y axis data at 1000
PR*=1000	Sets all axes to 1000

A ? returns the specified value for that axis. For example, AC ?,?,?,?, returns the acceleration of the X,Y,Z and W axes. The syntax format is the same as x,y,z,w except ? replaces the values.

Other commands require action on the X,Y,Z or W axis to be specified. These commands are followed by uppercase X,Y,Z or W. Action for a particular axis or any combination is specified by writing X,Y,Z or W. No commas are needed. Valid XYZW syntax is listed below.

COMMAND DESCRIPTIONS

VALID XYZW SYNTAX

INSTRUCTION	INTERPRETATION
SH X	Servo Here, X only
SH XYW	Servo Here, X,Y and W axes
SH XZW	Servo Here, X,Z and W axes
SH XYZW	Servo Here, X,Y,Z and W axes
SH Y	Servo Here, Y only
SH YZW	Servo Here, Y,Z and W axes
SH Z	Servo Here, Z only
SH	Servo Here, X,Y,Z and W axes
SH W	Servo Here, W only
SH ZW	Servo Here, Z and W axes
	Where X,Y,Z and W specify axis.

The Usage description specifies the restrictions on allowable execution. “While Moving” states whether or not the command is valid while the controller is performing a previously defined motion. “In a program” states whether the command may be used as part of a user-defined program. “Command Line” states whether the command may be used other than in a user-defined program.

“Can be Interrogated” states whether or not the command can be interrogated with sending ? to return the specified value. “Used as an Operand” states whether a command can be used to generate a value for another command or variable (i.e. V=_GNX). “Default Format” defines the format of the value with number of digits before and after the decimal point. Finally, “Default Value” defines the values the instruction’s parameters will have after a Master Reset.

The Operand Usage description describes proper syntax and the value contained by the operand.

When downloading commands to the SSC, do not insert a space prior to any command. For example, STX; AMX is invalid because there is a space after the semicolon.

Two-Letter Command Summary

SERVO CONTROL FILTER SETTINGS

DV	Dual loop operation
FA	Acceleration Feedforward
FV	Velocity Feedforward
GN	Gain
IL	Integrator Limit
IT	Independent Smoothing Constant
KD	Derivative Constant
KI	Integrator Constant
KP	Proportional Constant
MO	Motor Off
OF	Offset
SH	Servo Here
TL	Torque Limit
TM	Sample Time
VT	Smoothing Time Constant — Vector
ZR	Zero

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 Decoder
TE	Tell Error
TI	Tell Input
TP	Tell Position
TR	Trace
TS	Tell Switches
TT	Tell Torque
TV	Tell Velocity

HOMING COMMANDS

FE	Find Edge Routine
FI	Find Index Routine
HM	Homing Routine

SYSTEM CONFIGURATION

BN	Burn EEPROM
CE	Configure Encoder Type
CN	Configure Switches
CO	Configure I/O points (DB–10072)
CW	Data Adjustment Bit
DE	Define Dual Encoder Position
DP	Define Position
ED	Edit Mode
EI	Enable Interrupts
EO	Echo Off
LS	List
MO	Motor Off
MT	Motor Type
OB	Define Output Bit
PF	Position Format
RS	Reset
^R^S	Master Reset
^R^V	Firmware Revision Information
SC	Stop Code — Status Information

DUAL ENCODER OPERATION

CE	Configure Encoder
DE	Defines dual encoder position
GA	Maser Axis for Gearing[1]
GR	Gear Ratio for Gearing[1]
TD	Current position of dual encoder

EDITOR COMMANDS

ED	Edit Mode
<return>	Save line
^P	Previous line
^I	Insert line
^D	Delete line
^Q	Quit Editor

COMMUNICATION COMMANDS

CC	Configure Communications
CI	Communication Interrupt

I/O COMMANDS

AL	Arm Latch
CB	Clear Bit
HM	Homing Routine
CO	Configure I/O points (DB–10072)
IN	Input Variable
II	Input Interrupt
OB	Define Output Bit
OP	Output Port
SB	Set Bit

ARITHMETIC FUNCTIONS

@SIN[x]	Sine of ‘x’
@COS[x]	Cosine of ‘x’
@COM[n]	1’s Compliment of n
@ABS[x]	Absolute value of ‘x’
@FRAC[x]	Fraction portion of ‘x’
@INT[x]	Integer portion of ‘x’
@RND[x]	Round of ‘x’
@SQR[x]	Square root of ‘x’
@IN[x]	Return state of digital input ‘x’
@OUT[x]	Return state of digital output ‘x’
@AN[x]	Return value of analog input ‘x’
+	Add
-	Minus
*	Multiply
/	Divide
&	And
|	Or
()	Parentheses

STEPPER MOTOR COMMANDS

DE	Define Main Encoder Position
DP	Define Reference Position
KS	Stepper Motor Smoothing
MT	Motor Type (2 or -2 for Stepper)
RP	Report Commanded Position (Counts)
TD	Number of Stepper Counts Output
TP	Tell Position of Main Encoder

[1] Gearing can use dual encoder as master.

INDEPENDENT MOTION COMMANDS

AB	Aborts motion and/or program
AC	Sets Acceleration Rate
AM	After Motion Trippoint
AD	After Distance Trippoint
AP	After Position Trippoint
AR	After Relative Distance Trippoint
AS	At Speed Trippoint
AT	After Time Trippoint
BG	Begin Motion Command
DC	Sets Deceleration Rate
GA	Master Axis for Gearing[1]
GR	Gear Ratio for Gearing[1]
IP	Increment Position Movement
JG	Jog Mode
MC	Motion Complete Trippoint
MF	Trippoint — After Motion (Forward)
MR	Trippoint — After Motion Reverse
PA	Position Absolute
PR	Position Relative
SC	Stop Code — Status Information
SP	Set Speed
ST	Stop Motion or Program

TRIPPOINT COMMANDS

AD	After Distance
AI	After Input
AM	After Motion Complete
AP	After Absolute Position
AR	After Relative Distance
AS	At Speed
AT	After Time
AV	After Vector Distance
MC	After motion is in position
MF	After motion — forward direction
MR	After motion — reverse direction
WC	Wait for Contour Data

CONTOUR MODE COMMANDS

CD	Contour Data
CM	Contour Mode
CS	Clear Motion Sequence
DT	Contour Time Interval
SC	Stop Code — Status Information
WC	Wait for Contour Data

LINEAR INTERPOLATION COMMANDS

AMS	After Motion Trippoint (for segment)
AS	At Speed Trippoint
AT	After Time Trippoint
AV	After Vector Trippoint
GA	Master Axis for Gearing[1]
GR	Gear Ratio for Gearing[1]
LE	Linear Interpolation End
LI	Linear Interpolation Distance
LM	Linear Interpolation Mode
SC	Stop Code — Status Information
ST	Stop
TN	Tangent Motion for Vector Mode
VA	Set Vector acceleration
VD	Set Vector Deceleration
VR	Set Vector speed ratio
VS	Set Vector Speed
VT	Smoothing Time Constant — Vector

VECTOR MODE COMMANDS

AMS	After Motion Trippoint (for segment)
AV	After Vector Trippoint
CR	Circular Interpolation Move
CS	Clear Motion Sequence
ES	Ellipse Scaling
GA	Maser Axis for Gearing[1]
GR	Gear Ratio for Gearing[1]
ST	Stop Motion
SC	Stop Code — Provides Status Information
VA	Vector acceleration
VD	Vector Deceleration
VE	Vector Sequence End
VM	Coordinated Motion Mode
VP	Vector Position
VR	Vector speed ratio
VS	Vector Speed
VT	Smoothing Time Constant — Vector

ELECTRONIC GEARING COMMANDS

GA	Master Axis for Gearing
GR	Gear Ratio for Gearing

ECAM COMMANDS

EA	Choose ECAM master
EB	Enable CAM
EG	Engage ECAM
EM	CAM cycle command
EN	End Program
EP	CAM interval and starting point
EQ	Disengage ECAM
ET	ECAM table entry
SC	Stop Code — Status Information

[1]Independent motion, Vector motion, and Linear Interpolation can be used with gearing.

AB

ABORT

DESCRIPTION:

AB (Abort) stops a motion instantly without a controlled deceleration. If there is a program operating, AB also aborts the program unless a 1 argument is specified. The command, AB, will shut off the motors for any axis in which the off-on-error function is enabled (see page 354).

ARGUMENTS: AB n

where n = no argument or 1
1 aborts motion without aborting program, 0 aborts motion and program
AB aborts motion on all axes in motion and cannot stop individual axes.

USAGE:

While Moving	Yes	Default Value —-
In a Program	Yes	Default Format —-
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"SH" on page 12-141	Turns servos back on if they were shut-off by Abort and OE1.

EXAMPLES:

INSTRUCTION	INTERPRETATION
AB	Stops motion
OE 1,1,1,1	Enable off-on-error
AB	Shuts off motor command and stops motion
#A	Label - Start of program
JG 20000	Specify jog speed on X-axis
BGX	Begin jog on X-axis
WT 5000	Wait 5000 msec
AB1	Stop motion without aborting program
WT 5000	Wait 5000 milliseconds
SH	Servo Here
JP #A	Jump to Label A
EN	End of the routine

Hint: Remember to use the parameter 1 following AB if you only want the motion to be aborted. Otherwise, your application program will also be aborted.

AC

ACCELERATION

DESCRIPTION:

The Acceleration (AC) command sets the linear acceleration rate of the motors for independent moves, such as PR, PA and JG moves. The parameters input will be rounded down to the nearest factor of 1024. The units of the parameters are counts per second squared. The acceleration rate may be changed during motion. The DC command is used to specify the deceleration rate.

ARGUMENTS: AC x,y,z,w

where x,y,z,w are unsigned numbers in the range in the range 1024 to 67107840

AC ?,?,?,? returns the value

USAGE:

While Moving	Yes	Default Value 25600
In a Program	Yes	Default Format 8.0
Command Line	Yes	
Can be Interrogated	Yes	AC ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_ACx contains the value of acceleration for the specified axis, 'x'..

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"DC" on page 12-41	Specifies deceleration rate.
"FA" on page 12-67	Feedforward Acceleration
"IT" on page 12-87	Smoothing constant - S-curve

EXAMPLES:

INSTRUCTION	INTERPRETATION
AC 150000,200000,300000,400000	Set X-axis acceleration to 150000, Y-axis to 200000 counts/sec^2, the Z-axis to 300000 counts/sec^2, and the W-axis to 400000 count/sec^2.
AC ?,?,?,?	Request the Acceleration
0149504,0199680,0299008,0399360	Return Acceleration (resolution, 1024)
V=_ACY	Assigns the Y acceleration to the variable V

Hint: Specify realistic acceleration rates based on your physical system such as motor torque rating, loads, and amplifier current rating. Specifying an excessive acceleration will cause large following error during acceleration and the motor will not follow the commanded profile. The acceleration feedforward command FA will help minimize the error.

AD

AFTER DISTANCE

DESCRIPTION:

The After Distance (AD) command is a trippoint used to control the timing of events. This command will hold up the execution of the following command until the position command has reached the specified relative distance from the start of the move. The units of the command are quadrature counts. Only one axis may be specified at a time.

ARGUMENTS: AD x or AD,y or AD,,z or AD,,,w ADX=x

where x,y,z,w are unsigned integers in the range 0 to 2147483647 decimal
Only one axis may be specified at a time

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"AR" on page 12-17	After distance for repetitive triggering
"AV" on page 12-20	After distance for vector moves

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A;DP0,0,0,0	Begin Program
PR 10000,20000,30000,40000	Specify positions
BG	Begin motion
AD 5000	After X reaches 5000
MG "Halfway to X";TPX	Send message
AD ,10000	After Y reaches 10000
MG "Halfway to Y";TPY	Send message
AD ,,15000	After Z reaches 15000
MG "Halfway to Z";TPZ	Send message
AD ,,,20000	After W reaches 20000
MG "Halfway to W";TPW	Send message
EN	End Program

Hint: The AD command is accurate to the number of counts that occur in 2 msec. Multiply your speed by 2 msec to obtain the maximum position error in counts. Remember AD measures incremental distance from start of move on one axis.

AF

ANALOG FEEDBACK

DESCRIPTION:

The Analog Feedback (AF) command is used to set an axis with analog feedback instead of digital feedback (quadrature/pulse dir). As the analog feedback is decoded by a 12-bit A/D converter, an input voltage of 10 volts is decoded as a position of 2047 counts and a voltage of -10 volts corresponds to a position of -2048 counts. When using AF use analog input 1 for X axis, input 2 for Y axis, input 3 for Z axis etc.

ARGUMENTS: AF x,y,z,w AFX=x

where x,y,z,w are integers

1 = Enables analog feedback

0 = Disables analog feedback and switches to digital feedback

USAGE:

While Moving	No	Default Value 0,0,0,0
In a Program	Yes	
Not in a Program	Yes	
Can be Interrogated	Yes	
Used in an Operand	Yes	

RELATED COMMANDS:

INSTRUCTION
"MT" on page 12-112
"CE" on page 12-31

EXAMPLES:

INSTRUCTION	INTERPRETATION
AF 1,0,0,1	Analog feedback on X and W axis
V1 = _AFX	Assign feedback type to variable
AF ?,?,?	Interrogate feedback type

AI

AFTER INPUT

DESCRIPTION:

The AI command is used in motion programs to wait until after the specified input has occurred. If n is positive, it waits for the input to go high. If n is negative, it waits for n to go low.

ARGUMENTS: AI +/-n

where n is an integer in the range 1 to 8 decimal

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
@IN[n]	Function to read input 1 through 8
"II" on page 12-80	Input interrupt
#ININT	Label for input interrupt

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A	Begin Program
AI 8	Wait until input 8 is high
SP 10000	Speed is 10000 counts/sec
AC 20000	Acceleration is 20000 counts/sec2
PR 400	Specify position
BG X	Begin motion
EN	End Program

Hint: The AI command actually halts execution until specified input is at desired logic level. Use the conditional Jump command (JP) or input interrupt (II) if you do not want the program sequence to halt.

AL

ARM LATCH

DESCRIPTION:

The AL command enables the latching function (high speed position capture) of the controller. When the ALXYZW command is used to arm the position latches, the encoder position will be captured upon a low going signal on Input 1 (X axis), Input 2 (Y axis), Input 3 (Z axis), Input 4 (W axis), The command RL returns the captured position for the specified axes. When interrogated or used in an operand the AL command will return a 1 if the latch for that axis is armed or a zero after the latch has occurred. The CN command will change the polarity of the latch.

ARGUMENTS: AL XYZW

where X,Y,Z,W specifies the X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_ALx contains the state of the specified latch. 0 = not armed, 1 = armed.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"RL" on page 12-134	Report Latch

EXAMPLES:

INSTRUCTION	INTERPRETATION
#START	Start program
ALY	Arm Y-axis latch
JG,50000	Set up jog at 50000 counts/sec
BGY	Begin the move
#LOOP	Loop until latch has occurred
JP #LOOP,_ALY=1	
RLY	Transmit the latched position
EN	End of program

AM

AFTER MOVE

DESCRIPTION:

The AM command is a trippoint used to control the timing of events. This command will hold up execution of the following commands until the current move on the specified axis or axes is completed. Any combination of axes or a motion sequence may be specified with the AM command. For example, AM XY waits for motion on both the X and Y axis to be complete. AM with no parameter specifies that motion on all axes is complete.

ARGUMENTS: AM XYZWS

where X,Y,Z,S,W specifies X,Y,Z or W axis or sequence. No argument specifies that motion on all axes is complete.

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"BG" on page 12-21	_BG contains a 0 if motion complete

EXAMPLES:

INSTRUCTION	INTERPRETATION
#MOVE	Program MOVE
PR 5000,5000,5000,5000	Position relative moves
BG X	Start the X-axis
AM X	After the move is complete on X,
BG Y	Start the Y-axis
AM Y	After the move is complete on Y,
BG Z	Start the Z-axis
AM Z	After the move is complete on Z
BG W	Start the W-axis
AM W	After the move is complete on W
EN	End of Program

AM

Hint: AM is a very important command for controlling the timing between multiple move sequences. For example, if the X-axis is in the middle of a position relative move (PR) you cannot make a position absolute move (PAX, BGX) until the first move is complete. Use AMX to halt the program sequences until the first motion is complete. AM tests for profile completion. The actual motor may still be moving. Another method for testing motion complete is to check for the internal variable, _BG, being equal to zero.

AP

AFTER ABSOLUTE POSITION

DESCRIPTION:

The AP command is a trippoint used to control the timing of events. This command will hold up the execution of the following command until one of the following three conditions are met:

1. The actual motor position crosses position specified with the AP command
2. The motion on the axis is complete.
3. The motion on the axis is in the direction which moves away from the trippoint.

the absolute actual position of the motor crosses the position specified. The units of the command are in quadrature counts. Only one axis may be specified at a time. This trippoint will also be cleared by the completion of the move.

ARGUMENTS: APx or AP,y or AP,,z or AP,,,w APX=X

where x,y,z,w are signed integers in the range -2147483648 to 2147483647 decimal

USAGE:

While Moving	Yes	Default Value —-
In a Program	Yes	Default Format —-
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"AD" on page 12-8	trippoint for relative distances
"MF" on page 12-107	trippoint for forward motion

EXAMPLES:

INSTRUCTION	INTERPRETATION
#TEST	Program B
DP0	Define zero
JG 1000	Jog mode (speed of 1000 counts/sec)
BG X	Begin move
AP 2000	After passing the position 2000

AP

```
V1=_TPX                    Assign V1 X position
MG "Position is", V1=      Print Message
ST                         Stop
EN                         End of Program
```

Hint: The accuracy of the AP command is the number of counts that occur in 2 msec. Multiply the speed by 2 msec to obtain the maximum error. AP tests for absolute position. Use the AD command to measure incremental distances.

AR

AFTER RELATIVE DISTANCE

DESCRIPTION:

The After Relative Distance command is a trippoint used to control the timing of events. This command will hold up execution of the following command until the specified relative distance has reached from either last AR or AD command, or from the start of the move. Only one axis may be specified at a time. The units of the command are quadrature counts.

ARGUMENTS: ARx or AR,y or AR,,z or AR,,,w ARX=x

where x,y,z,w are unsigned integers in the range 0 to 2147483647 decimal
Only one axis may be specified at a time

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“AV” on page 12-20	After vector position for coordinated moves
“AP” on page 12-15	After absolute position

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A;DP 0,0,0,0	Begin Program
JG 50000,,,7000	Specify speeds
BG XW	Begin motion
#B	Label
AR 25000	After passing 25000 counts of relative distance on X-axis
MG “Passed_X”;TPX	Send message on X-axis
JP #B	Jump to Label #B
EN	End Program

Hint: AR is used to specify incremental distance from last AR or AD command. Use AR if multiple position trippoints are needed in a single motion sequence.

AS

AT SPEED

DESCRIPTION:

The AS command is a trippoint that occurs when the generated motion profile has reached the specified speed. This command will hold up execution of the following command until the speed is reached. The AS command will operate after either accelerating or decelerating. If the speed is not reached, the trippoint will be triggered after the motion is stopped (after deceleration).

ARGUMENTS: AS X or AS Y or AS Z or AS W or AS S

where XYZWS specifies X,Y,Z,W axis or sequence

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

INSTRUCTION	INTERPRETATION
#SPEED	Program SPEED
PR 100000	Specify position
SP 10000	Specify speed
BG X	Begin X
ASX	After speed is reached
MG “At Speed”	Print Message
EN	End of Program

WARNING: The AS command applies to a trapezoidal velocity profile only with linear acceleration. AS used with S-curve profiling will be inaccurate.

AT

AT TIME

DESCRIPTION:

The AT command is a trippoint which is used to hold up execution of the next command until after the specified time has elapsed. The time is measured with respect to a defined reference time. AT 0 establishes the initial reference. AT n specifies n msec from the reference. AT -n specifies n msec from the reference and establishes a new reference after the elapsed time period.

ARGUMENTS: AT N

where n is a signed integer in the range 0 to 2 Billion

n = 0 defines a reference time at current time

positive n waits n msec from reference

negative n waits n msec from reference and sets new reference after elapsed time period (AT -n is equivalent to AT n; AT 0)

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

The following commands are sent sequentially

INSTRUCTION	INTERPRETATION
AT 0	Establishes reference time 0 as current time
AT 50	Waits 50 msec from reference 0
AT 100	Waits 100 msec from reference 0
AT -150	Waits 150 msec from reference 0 and sets new reference at 150
AT 80	Waits 80 msec from new reference (total elapsed time is 230 msec)

AV

AFTER VECTOR DISTANCE

DESCRIPTION:

The AV command is a trippoint which is used to hold up execution of the next command during coordinated moves such as VP, CR or LI. This trippoint occurs when the path distance of a sequence reaches the specified value. The distance is measured from the start of a coordinated move sequence or from the last AV command. The units of the command are quadrature counts.

ARGUMENTS: AV n

where n is an unsigned integer in the range 0 to 2147483647 decimal

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_AV contains the vector distance from the start of the sequence. _AV is valid in the linear mode, LM and in the vector mode, VM.

EXAMPLES:

INSTRUCTION	INTERPRETATION
#MOVE;DP 0,0	Label
LMXY	Linear move for X,Y
LI 1000,2000	Specify distance
LI 2000,3000	Specify distance
LE	
BGS	Begin
AV 500	After path distance = 500,
MG “Path>500”;TPXY	Print Message
EN	End Program

Hint: Remember AV is the vector distance along the path where $AV^2=X^2+Y^2+Z^2+W^2$.

BG

BEGIN

DESCRIPTION:

The BG command starts a motion on the specified axis or sequence. When Used as an Operand, the BG command will return a 1 if the controller is performing a move of that axis.

ARGUMENTS: BG XYZWS

where XYZW are X,Y,Z,W axes and S is coordinated sequence

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_BG contains a '0' if motion complete on the specified axis, otherwise contains a '1'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"AM" on page 12-13	After motion complete
"ST" on page 12-144	Stop motion

EXAMPLES:

INSTRUCTION	INTERPRETATION
PR 2000,3000,,5000	Set up for a relative move
BG XYW	Start the X,Y and W motors moving
HM	Set up for the homing
BGX	Start only the X-axis moving
JG 1000,4000	Set up for jog
BGY	Start only the Y-axis moving
YSTATE=_BGY	Assign a 1 to YSTATE if the Y-axis is performing a move
VP 1000,2000	Specify vector position
VS 20000	Specify vector velocity
BGS	Begin coordinated sequence
VMXY	Vector Mode

BG

VP 4000,-1000	Specify vector position
VE	Vector End
PR ,,8000,5000	Specify Z and W position
BGSZW	Begin sequence and Z,W motion
MG _BGS	Displays a 1 if coordinated sequence move is running

Hint: You cannot give another BG command until current BG motion has been completed. Use the AM trippoint to wait for motion complete between moves. Another method for checking motion complete is to test for _BG being equal to 0.

BL

REVERSE SOFTWARE LIMIT

DESCRIPTION:

The BL command sets the reverse Software limit. If this limit is exceeding during motion, motion on that axis will decelerate to a stop. Reverse motion beyond this limit is not permitted. The reverse limit is activated at X-1, Y-1, Z-1, W-1. To disable the reverse limit, set X,Y,Z,W to -2147483648. The units are in quadrature counts.

ARGUMENTS: BL x,y,z,w BLX=x

where x,y,z,w are signed integers in the range -2147483648 to 2147483647.

-214783648 turns off the reverse limit.

USAGE:

While Moving	Yes	Default Value -214783648
In a Program	Yes	Default Format Position format
Command Line	Yes	
Can be Interrogated	Yes	BL ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_BLx contains the value of the reverse Software limit for the specified axis, 'x'..

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"FL" on page 12-71	Forward Limit

EXAMPLES:

INSTRUCTION	INTERPRETATION
#TEST	Test Program
AC 1000000	Acceleration Rate
DC 1000000	Deceleration Rate
BL -15000	Set Reverse Limit
JG -5000	Jog Reverse
BGX	Begin Motion
AMX	After Motion (limit occurred)
TPX	Tell Position
EN	End Program

Hint: The SSC also provides hardware limits.

BN

BURN SYSTEM PARAMETERS

DESCRIPTION:

The BN command saves the certain controller parameters shown below in non-volatile EEPROM memory. Programs are not saved and must be downloaded from the host computer upon power-up. This command typically takes 1 second to execute and must not be interrupted. The controller returns a : when the Burn is complete.

PARAMETERS SAVED DURING BURN:

AC	ER	OP
BL	FL	PF
CB	GA	SB
CE	GR	SP
CN	IL	TL
CO	KD (ZR converted to KD)	TM
CW	KI	VA
DV	KP (GN converted to KP)	VD
DC	MO (MOTOR OFF or ON)	VF
EO	MT	VS
	OE	VT

ARGUMENTS: None

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"BP" on page 12-26	Burn Program
"BV" on page 12-27	Burn Variable

EXAMPLES:

INSTRUCTION	INTERPRETATION
KD 100	Set damping term for X axis
KP 10	Set proportional gain term for X axis
KI 1	Set integral gain term for X axis
AC 200000	Set acceleration
DC 150000	Set deceleration rate
SP 10000	Set speed
MT -1	Set motor type for X axis to be type '-1', reversed polarity servo motor
MO	Turn motor off
BN	Burn parameters; may take up to 15 seconds

BP

BURN PROGRAM

DESCRIPTION:

The BP command saves the application program in non-volatile EEPROM memory. This command typically takes up to 10 seconds to execute and must not be interrupted. The controller returns a : when the Burn is complete.

ARGUMENTS: None

USAGE:

While Moving	No	Default Value —-
In a Program	No	
Not in a Program	Yes	
Can be Interrogated	No	
Used in an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"BN" on page 12-24	Burn Parameters
"BV" on page 12-27	Burn Variable

Note: Does not burn arrays.

BV

BURN VARIABLES

DESCRIPTION:

The BV command saves the controller variables in non-volatile EEPROM memory. This command typically takes up to 2 seconds to execute and must not be interrupted. The controller returns a : when the Burn is complete.

ARGUMENTS: None

USAGE:

While Moving	No	Default Value —-
In a Program	Yes	
Not in a Program	Yes	
Can be Interrogated	No	
Used in an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"BN" on page 12-24	Burn Parameters
"BP" on page 12-26	Burn Program

Note: BV does not save any arrays unless the 'Save Array' option was requested on the SSC.

CB

CLEAR BIT

DESCRIPTION:

The CB command clears one of eight bits on the output port. The CB and SB (Set Bit) instructions can be used to control the state of output lines.

ARGUMENTS: CB n,

where n is an integer in the range 1 to 8

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"SB" on page 12-139	Set Bit
"OP" on page 12-119	Define all outputs

EXAMPLES:

INSTRUCTION	INTERPRETATION
CB 0	Clear output bit 0
CB 7	Clear output bit 7

CC

CONFIGURE COMMUNICATIONS PORT 2

DESCRIPTION:

The CC command configures baud rate, handshake, mode, and echo for the AUX SERIAL PORT, referred to as Port 2. This command must be given before using the MG, IN, or CI commands with Port 2.

ARGUMENTS: CC m,n,r,p

m - Baud rate	300,1200,4800,9600,19200, or 38400
n - Handshake	0 for handshake off, 1 for handshake on
r - Mode	0 for daisy chain off, 1 for daisy chain on
p - Echo	0 for echo off, 1 for echo on

Note: echo only active when daisy chain feature is off

USAGE:

While Moving	Yes	Default Value 0,0,0
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"CI" on page 12-32	Command Interrupt

EXAMPLES:

INSTRUCTION	INTERPRETATION
CC 9600,0,0,1	9600 baud, no handshake, daisy chain off, echo on. Typical setting with TERM-1500.
CC 19200,1,1,0	19,200 baud, handshake on, daisy chain on, echo off.

CD

CONTOUR DATA

DESCRIPTION:

The CD command specifies the incremental position on X,Y,Z and W axes. The units of the command are in quadrature counts. This command is used only in the Contour Mode (CM).

ARGUMENTS: CD x,y,z,w CDX=x

where x,y,z,w are integers in the range of +/-32762

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"CM" on page 12-34	Contour Mode
"WC" on page 12-178	Wait for Contour
"DT" on page 12-49	Time Increment
"CS" on page 12-38	_CS is the Segment Counter

EXAMPLES:

INSTRUCTION	INTERPRETATION
CM XYZW	Specify Contour Mode
DT 4	Specify time increment for contour
CD 200,350,-150,500	Specify incremental positions on X,Y,Z and W axes X-axis moves 200 counts Y-axis moves 350 counts Z-axis moves -150 counts W-axis moves 500 counts
WC	Wait for complete
CD 100,200,300,400	New position data
WC	Wait for complete
DT0	Stop Contour
CD 0,0,0,0	Exit Mode

CE

CONFIGURE ENCODER

DESCRIPTION:

The CE command configures the encoder to the quadrature type or the pulse and direction type. It also allows inverting the polarity of the encoders. The configuration applies independently to the four main axes encoders and the four auxiliary encoders.

ARGUMENTS: CE x,y,z,w CEX=x

Where x,y,z,w are integers in the range of 0 to 15. Each integer is the sum of two integers n and m which configure the main and the auxiliary encoders.

The values of m and n are

m =	Main encoder type	n =	Auxiliary encoder type
0	Normal quadrature	0	Normal quadrature
1	Normal pulse and direction	4	Normal pulse and direction
2	Reversed quadrature	8	Reversed quadrature
3	Reversed pulse and direction	12	Reversed pulse and direction

For example: x = 10 implies m = 2 and n = 8, both encoders are reversed quadrature.

USAGE:

While Moving	Yes	Default Value O
In a Program	Yes	Default Format 2.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_CEx contains the value of encoder type for the axis specified by ‘x’.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“MT” on page 12-112	Specify motor type

EXAMPLES:

INSTRUCTION	INTERPRETATION
CE 0, 3, 6, 2	Configure encoders
CE ?,?,?,?	Interrogate configuration
V = _CEX	Assign configuration to a variable

CI

COMMUNICATION INTERRUPT

DESCRIPTION:

The CI command configures a program interrupt based on characters received on either Port 1, the MAIN serial port, or Port 2, the AUX serial port. An interrupt causes program flow to jump to the #COMINT subroutine label. If multiple program threads are used, the #COMINT subroutine runs in thread 0 and threads 1, 2, and 3 continue to run in the background without interruption. The characters received on the serial port is stored in internal variables such as P2CH. See chapter 9 for more detailed information on the communications interrupt.

ARGUMENTS: CI m,n,o

PARAMETER	EXPLANATION
m = 0	Do not interrupt Port 1
m = 1	Interrupt on carriage return character on Port 1
m = 2	Interrupt on any character Port 1
m = -1	Clear interrupt data buffer
n = 0	Do not interrupt Port 2
n = 1	Interrupt on carriage return character on Port 2
n = 2	Interrupt on any character Port 2
n = -1	Clear interrupt data buffer
o = 0	Disable live data mode for Port 1
o = 1	Enable live data mode for Port 1

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	No	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

INSTRUCTION	INTERPRETATION
CI 0,1,0	Interrupt on <enter> received on Port 2
CI 0,2,0	Interrupt on a single character received on Port 2
CI 1,2,1	Interrupt on <enter> received on Port 1, interrupt on any character received on Port 2

Note: The third field of the CI command enables or disables live data mode on Port 1. If live data mode is enabled, then the controller will not respond to commands sent to the main serial port. This setting is necessary to use the communications interrupt on the main serial port.

CM

CONTOURING MODE

DESCRIPTION:

The Contour Mode is initiated by the instruction CM. This mode allows the generation of an arbitrary motion trajectory with any of the axes. The CD command specified the position increment, and the DT command specifies the time interval.

The command, CM?, can be used to check the status of the Contour Buffer. A value of 1 returned from the command CM? indicates that the Contour Buffer is full. A value of 0 indicates that the Contour Buffer is empty.

ARGUMENTS: CM XYZW

where XYZW specify the X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 2.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_CM contains a '0' if the contour buffer is empty, otherwise contains a '1'

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"CD" on page 12-30	Contour Data
"WC" on page 12-178	Wait for Contour
"DT" on page 12-49	Time Increment

EXAMPLES:

INSTRUCTION	INTERPRETATION
V=_CM;V=	Return contour buffer status
CM?	Return contour buffer status
CM XZ	Specify X,Z axes for Contour Mode

CN

CONFIGURE

DESCRIPTION:

The CN command configures the polarity of the limit switches, the home switch and the latch input.

ARGUMENTS: CN m,n,o

where m,n,o are integers with values 1 or -1.

PARAMETER		EXPLANATION
m =	1	Limit switches active high
	-1	Limit switches active low
n =	1	Home switch is active high
	-1	Home switch is active low
o =	1*	Latch input is active high
	-1	Latch input is active low

*Note: The latch function will occur within 25usec only when used in active low mode.

USAGE:

While Moving	Yes	Default Value -1.-1.-1.-1
In a Program	Yes	Default Format 2.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"AL" on page 12-12	Arm latch

EXAMPLES:

INSTRUCTION	INTERPRETATION
CN 1,1	Sets limit and home switches to active high
CN,, -1	Sets input latch active low

CR

CIRCLE

DESCRIPTION:

The CR command specifies a 2-dimensional arc segment of radius, r, starting at angle, θ, and traversing over angle Δθ. A positive Δθ denotes counterclockwise traverse, negative Δθ denotes clockwise. The VE command must be used to denote the end of the motion sequence after all CR and VP segments are specified. The BG (Begin Sequence) command is used to start the motion sequence. All parameters, r, θ, Δθ, must be specified. Radius units are in quadrature counts. θ and Δθ have units of degrees. The parameter n is optional and describes the vector speed that is attached to the motion segment.

ARGUMENTS: CR r,θ,Δθ < n

r is an unsigned real number in the range 10 to 6000000 decimal

θ is a signed number in the range 0 to +/-32000 decimal

Δθ is a signed real number in the range 0.0001 to +/-32000 decimal

n is an unsigned even number between 0 and 8,000,000

The product r * Δθ must be limited to +/-4.5 108

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VP" on page 12-172	Vector Position
"VS" on page 12-176	Vector Speed
"VD" on page 12-166	Vector Deceleration
"VA" on page 12-164	Vector Acceleration
"VM" on page 12-170	Vector Mode
"VE" on page 12-168	End Vector
"BG" on page 12-21	Begin Sequence

EXAMPLES:

INSTRUCTION	INTERPRETATION
VMXY	Specify vector motion in the X and Y plane
VS 10000	Specify vector speed
CR 1000,0,360	Generate circle with radius of 1000 counts, start at 0 degrees and complete one circle in counterclockwise direction.
CR 1000,0,360 < 40000	Generate circle with radius of 1000 counts, start at 0 degrees and complete one circle in counterclockwise direction and use a vector speed of 40000.
VE	End Sequence
BGS	Start motion

CS

CLEAR SEQUENCE

DESCRIPTION:

The CS command will remove VP, CR or LI commands stored in a motion sequence. Note, after a sequence has been run, the CS command is not necessary to put in a new sequence. This command is useful when you have incorrectly specified VP, CR or LI commands.

Note: This command is not valid for the SSC10.

ARGUMENTS: None

USAGE:

While Moving	Yes	Default Value —-
In a Program	Yes	Default Format —-
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

When used as an operand, _CS contains the number of the segment in the sequence,starting at zero. The instruction _CS is valid in the Linear mode, LM, Vector mode, VM, and contour mode, CM.

EXAMPLES:

INSTRUCTION	INTERPRETATION
#CLEAR	Label
VP 1000,2000	Vector position
VP 4000,8000	Vector position
CS	Clear vectors
VP 1000,5000	New vector
VP 8000,9000	New vector
VE	End Sequence
BGS	Begin sequence
EN	End of Program

DA

DEALLOCATE THE VARIABLES & ARRAYS

DESCRIPTION:

The DA command frees the array and/or variable memory space. In this command, more than one array or variable can be specified for deallocation of memories. Different arrays and variables are separated by comma when specified in one command. The only limit is the 40 character per line. The * argument deallocates all the variables, and *[0] deallocates all the arrays.

ARGUMENTS: DA c[0],variable-name

where

c[0] = Defined array name
variable-name = Defined variable name
* - Deallocates all the variables
*[0] - Deallocates all the arrays

USAGE:

While Moving	Yes*	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	Yes	DA ?
Used as an Operand	Yes	

OPERAND USAGE:

_DA contains the total number of arrays available. For example, before any arrays have been defined, the operand _DA on a standard SSC will return 30. If an array is defined, the operand _DA will return 29.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"DM" on page 12-46	Dimension Array

EXAMPLES: 'Cars' and 'Sales' are arrays and 'Total' is a variable.

INSTRUCTION	INTERPRETATION
DM Cars[400],Sales[50]	Dimension 2 arrays
Total=70	Assign 70 to the variable Total
DA Cars[0],Sales[0],Total	Deallocate the 2 arrays & variables
DA*[0]	Deallocate all arrays
DA *,*[0]	Deallocate all variables and all arrays

DA

Note: Since this command deallocates the spaces and compacts the array spaces in the memory, it is possible that execution of this command may take longer time than 2 ms.

DC

DECELERATION

DESCRIPTION:

The Deceleration command (DC) sets the linear deceleration rate of the motors for independent moves such as PR, PA and JG moves. The parameters will be rounded down to the nearest factor of 1024 and have units of counts per second squared.

ARGUMENTS: DC x,y,z,w DCX=x

where x,y,z,w are unsigned numbers in the range 1024 to 67107840

USAGE:

While Moving	Yes*	Default Value 256000
In a Program	Yes	Default Format 8.0
Command Line	Yes	
Can be Interrogated	Yes	DC ?,?,?,?
Used as an Operand	Yes	

*When moving, the DC command can only be specified while in the jog mode.

OPERAND USAGE:

_DCx contains the deceleration rate for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"AC" on page 12-6	Acceleration
"PR" on page 12-124	Position Relative
"PA" on page 12-120	Position Absolute
"SP" on page 12-142	Speed
"JG" on page 12-88	Jog
"BG" on page 12-21	Begin
"IT" on page 12-87	Smoothing

DC

EXAMPLES:

INSTRUCTION	INTERPRETATION
PR 10000	Specify position
AC 2000000	Specify acceleration rate
DC 1000000	Specify deceleration rate
SP 5000	Specify slew speed
BG	Begin motion

Note: The DC command may be changed during the move in JG move, but not in PR or PA move.

DE

DUAL (AUXILIARY) ENCODER POSITION

DESCRIPTION:

The DE command defines the position of the auxiliary encoders. The auxiliary encoders may be used for dual-loop applications. DE ? returns the position of the auxiliary encoders.

The DE command defines the current motor position when used with stepper motors. DE ? returns the commanded reference position of the motor. The units are in steps.

ARGUMENTS: DE x,y,z,w DEX=x

where x,y,z,w are signed integers in the range -2147483647 to 2147483648 decimal

USAGE:

While Moving	Yes	Default Value 0,0,0,0
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	DC ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_DEx contains the current position of the specified auxiliary encoder.

EXAMPLES:

INSTRUCTION	INTERPRETATION
DE 0,100,200,400	Set the current auxiliary encoder position to 0,100,200,400 on X,Y,Z and W axes
DE?,?,?,?	Return auxiliary encoder positions
DUALX=_DEX	Assign auxiliary encoder position of X-axis to the variable DUALX

Hint: Dual encoders are useful when you need an encoder on the motor and on the load. The encoder on the load is typically the auxiliary encoder and is used to verify the true load position. Any error in load position is used to correct the motor position.

DL

DOWNLOAD

DESCRIPTION:

The DL command transfers a data file from the host computer to the SSC. Instructions in the file will be accepted as a datastream without line numbers. The file is terminated using <control> Z, <control> Q, <control> D, or \.

If no parameter is specified, downloading a data file will clear all programs in the SSC RAM. The data is entered beginning at line 0. If there are too many lines or too many characters per line, the SSC will return a ?. To begin the download following a label, that label may be specified following DL. Or, the # argument may be used with DL to append a file at the end of the SSC program in RAM. DO NOT insert any spaces before each command.

ARGUMENTS: DL n

n = no argument	Downloads program beginning at line 0. Erases programs in RAM.
n = #Label	Begins download at line following #Label where label may be any valid program label.
n = #	Begins download at end of program in RAM.

USAGE:

While Moving	Yes	Default Value —-
In a Program	No	Default Format —-
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

When used as an operand, _DL gives the number of available labels. The total number of labels is 254.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“UL” on page 12-163	Upload

EXAMPLES:

INSTRUCTION	INTERPRETATION
DL;	Begin download
#A;PR 4000;BGX	Data
AMX;MG DONE	Data
EN	Data
<control> Z	End download

DM

DIMENSION

DESCRIPTION:

The DM command defines a single dimensional array with a name and n total elements. The maximum number of arrays which the user can define is 30 and the maximum number of elements is 8000. If the user needs only one array, then the maximum number of elements is still 8000. The first element of the defined array starts with element number 0 and the last element is at n-1.

ARGUMENTS: DM c[n]

where c is a name of up to eight characters, starting with an uppercase alphabetic character. n is the number of entries from 1 to 8000.

USAGE:

While Moving	Yes	Default Value —-
In a Program	Yes	Default Format —-
Command Line	Yes	
Can be Interrogated	No	DM ?
Used as an Operand	Yes	

OPERAND USAGE:

_DM contains the available array space. For example, before any arrays have been defined, the operand _DM on a standard SSC will return 8000. If an array of 100 elements is defined, the operand _DM will return 7900.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“DA” on page 12-39	Deallocate Array

EXAMPLES:

INSTRUCTION	INTERPRETATION
DM Pets[5],Dogs[2],Cats[3]	Define dimension of arrays, pets with 5 elements; Dogs with 2 elements; Cats with 3 elements
DM Tests[1600]	Define dimension of array Tests with 1600 elements

DP

DEFINE POSITION

DESCRIPTION:

The DP command sets the current motor position and current command positions to a user specified value. If a ? is used, it returns the current position of the motor. The units are in quadrature counts.

The DP command sets the commanded reference position for axes configured as steppers. DP ? returns the current position of the motor. The units are in steps.

ARGUMENTS: DP x,y,z,w DPX=x

where x,y,z,w are signed integers in the range -2147483648 to 2147483647 decimal

USAGE:

While Moving	No	Default Value 0,0,0,0
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	DP ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_DPx contains the current position of the specified axis.

EXAMPLES:

INSTRUCTION	INTERPRETATION
DP 0,100,200,400	Sets the current position of the X-axis to 0, the Y axis to 100, the Z-axis to 200, and the W-axis to 400
DP ,-50000	Sets the current position of Y-axis to -50000. The Y,Z and W axes remain unchanged.
DP ?,?,?,?	Interrogate the position of X,Y,Z and W axis.
0000000,-0050000, 0000200,0000400	Returns all the motor positions
DP ?	Interrogate the position of X axis
0000000	Returns the X-axis motor position

DP

Hint: The DP command is useful to redefine the absolute position. For example, you can manually position the motor by hand using the Motor Off command, MO. Turn the servo motors back on with SH and then use DP0 to redefine the new position as your absolute zero.

DT

DELTA TIME

DESCRIPTION:

The DT command sets the time interval for Contouring Mode. Sending the DT command once will set the time interval for all following contour data until a new DT command is sent. 2^n milliseconds is the time interval. Sending DT0 followed by CD0 command terminates the Contour Mode.

ARGUMENTS: DT n

where n is an integer in the range 0 to 8. 0 terminates the Contour Mode. n=1 thru 8 specifies the time interval of 2^n samples.
The default time interval is n=1 or 2 msec for a sample period of 1 msec.

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	Yes	DT ?
Used as an Operand	Yes	

OPERAND USAGE:

_DT contains the value for the time interval for Contour Mode

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"CM" on page 12-34	Contour Mode
"WC" on page 12-178	Contour Data
"CD" on page 12-30	Wait for next data

EXAMPLES:

INSTRUCTION	INTERPRETATION
DT 4	Specifies time interval to be 16 msec
DT 7	Specifies time interval to be 128 msec
#CONTOUR	Begin
CMXY	Enter Contour Mode
DT 4	Set time interval
CD 1000,2000	Specify data
WC	Wait for contour
CD 2000,4000	New data

DT

WC	Wait
DT0	Stop contour
CD0	Exit Contour Mode
EN	End

DV

DUAL VELOCITY (DUAL LOOP)

DESCRIPTION:

The DV function changes the operation of the filter. It causes the KD (derivative) term to operate on the dual encoder instead of the main encoder. This results in improved stability in the cases where there is a backlash between the motor and the main encoder, and where the dual encoder is mounted on the motor.

ARGUMENTS: DV x,y,z,w

where x,y,z,w may be 0 or 1. 0 disables the function. 1 enables the dual loop.

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_DVx contains the state of dual velocity mode for specified axis.
0 = disabled, 1 = enabled.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"KD" on page 12-92	Damping constant
"FV" on page 12-72	Velocity feedforward

EXAMPLES:

INSTRUCTION	INTERPRETATION
DV 1,1,1,1	Enables dual loop on all axes
DV 0	Disables DV on X axis
DV,,1,1	Enables dual loop on Z axis and W axis. Other axes remain unchanged.
DV 1,0,1,0	Enables dual loop on X and Z axis. Disables dual loop on Y and W axis.

Hint: The DV command is useful in backlash and resonance compensation.

EA

CHOOSE ECAM MASTER

DESCRIPTION:

The EA command selects the master axis for the electronic cam mode. Any axis may be chosen.

ARGUMENTS: EA p

where p is XYZW

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"EB" on page 12-53	Enable Ecam
"EG" on page 12-56	Engage Ecam
"EM" on page 12-57	Specify Ecam Cycle
"EP" on page 12-61	CAM table intervals & starting point
"EQ" on page 12-62	Ecam quit
"ET" on page 12-66	Electronic CAM table

EXAMPLES:

INSTRUCTION	INTERPRETATION
EAY	Select Y as a master for ECAM

EB

ENABLE ECAM

DESCRIPTION:

The EB function enables or disables the cam mode. In this mode, the starting position of the master axis is specified within the cycle. When the EB command is given, the master axis is modularized.

ARGUMENTS: EB n

where n = 1 starts cam mode and n = 0 stops cam mode.

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_EB contains the state of Ecam mode. 0 = disabled, 1 = enabled

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"EA" on page 12-52	Choose Ecam master
"EG" on page 12-56	Engage Ecam
"EM" on page 12-57	Specify Ecam Cycle
"EP" on page 12-61	CAM table intervals & starting point
"EQ" on page 12-62	Ecam quit
"ET" on page 12-66	Electronic CAM table

EXAMPLES:

INSTRUCTION	INTERPRETATION
EB1	Starts ECAM mode
EB0	Stops ECAM mode
B = _EB	Return status of cam mode

ED

EDIT

DESCRIPTION:

The ED command puts the controller into the Edit subsystem. In the Edit subsystem, programs can be created, changed, or destroyed. The commands in the Edit subsystem are:

<cntrl>D	Deletes a line
<cntrl>I	Inserts a line before the current one
<cntrl>P	Displays the previous line
<cntrl>Q	Exits the Edit subsystem
<return>	Saves a line

ARGUMENTS: ED n

where n specifies the line number to begin editing. The default line number is the last line of program space with commands.

USAGE:

While Moving	No
In a Program	No
Command Line	Yes
Can be Interrogated	No
Used as an Operand	Yes

OPERAND USAGE:

_ED contains the line number of the last line to have an error.

EXAMPLES:

INSTRUCTION	INTERPRETATION
ED	
000 #START	
001 PR 2000	
002 BGX	
003 SLKJ	Bad line
004 EN	
005 #CMDERR	Routine which occurs upon a command error
006 V=_ED	
007 MG	“An error has occurred” {n}
008 MG	“In line”, V{F3.0}
009 ST	

```
010 ZS0
011 EN
```

Hint: Remember to quit the Edit Mode prior to executing or listing a program.

EG

ECAM GO (ENGAGE)

DESCRIPTION:

The EG command engages an ECAM slave axis at a specified position of the master. If a value is specified outside of the master's range, the slave will engage immediately. Once a slave motor is engaged, its position is redefined to fit within the cycle.

ARGUMENTS: EG x,y,z,w

where x,y,z,w are the master positions at which the X,Y,Z,W axis must be engaged.

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_EGx contains ecam status for specified axis. 0 = axis is not engaged, 1 = axis is engaged.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"EA" on page 12-52	Choose Ecam master
"EB" on page 12-53	Enable Ecam
"EM" on page 12-57	Specify Ecam Cycle
"EP" on page 12-61	CAM table intervals & starting point
"EQ" on page 12-62	Ecam quit
"ET" on page 12-66	Electronic CAM table

EXAMPLES:

INSTRUCTION	INTERPRETATION
EG 700,1300	Engages the X and Y axes at the master position 700 and 1300 respectively.
B = _EPY	Return the status of Y axis, 1 if engaged

Note: This command is not a trippoint. This command will not hold the execution of the program flow. If the execution needs to be held until master position is reached, use MF or MR command.

EM

CAM CYCLES

DESCRIPTION:

The EM command is part of the ECAM mode. It is used to define the change in position over one complete cycle of the master. The field for the master axis is the cycle of the master position. For the slaves, the field defines the net change in one cycle. If a slave will return to its original position at the end of the cycle, the change is zero. If the change is negative, specify the absolute value.

ARGUMENTS: EM x,y,z,w

where the parameters are positive integers in the range between 1 and 8,388,607 for the master axis and between 1 and 2,147,483,647 for a slave axis.

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_EMx contains the cycle of the specified axis

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"EA" on page 12-52	Choose Ecam master
"EB" on page 12-53	Enable Ecam
"EG" on page 12-56	Engage Ecam
"EP" on page 12-61	CAM table intervals & starting point
"EQ" on page 12-62	Ecam quit
"ET" on page 12-66	Electronic CAM table

EXAMPLES:

INSTRUCTION	INTERPRETATION
EAZ	Select Z axis as master for ECAM.
EM 0,3000,2000	Define the changes in X and Y to be 0 and 3000 respectively. Define master cycle as 2000.
V = _EMX	Return cycle of X

EN

END

DESCRIPTION:

The EN command is used to designate the end of a program or subroutine. If a subroutine was called by the JS command, the EN command ends the subroutine and returns program flow to the point just after the JS command. When the EN command is used to terminate the #COMINT communications interrupt subroutine, there are two arguments available to control trippoints and to reset the interrupt.

ARGUMENTS: EN m, n

m=0	Return from subroutine without restoring trippoint
m=1	Return from subroutine and restore trippoint
n=0	Return from #COMINT without restoring interrupt
n=1	Return from communications interrupt #COMINT and restore interrupt

Note 1: Not giving an argument is the same as using 0 for the argument, for example EN,1 and EN0,1 have the same effect.

Note 2: Trippoints such as the after motion command AM cause the program to pause, the AM for example waits for motion on all axes to complete. If one of the program interrupts occurs, there are two options: return from the interrupt handling subroutine and continue with the trippoint as if nothing happened, or clear the trippoint and continue the program with the command just after the trippoint. The #COMINT subroutine uses the arguments of the EN command for trippoint control and to restore the interrupt.

Note 3: Use the RE command to return from the interrupt handling subroutines #CMDERR, #LIMSWI, #MCTIME, and #POSERR. Use the RI command to return from the #ININT subroutine.

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	No	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"RE" on page 12-132	Return from error subroutine
"RI" on page 12-133	Return from interrupt subroutine

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A	Program A
PR 500	Move X axis forward 500 counts
BGX	Pause the program until the X axis completes the motion
AMX	Move X axis forward 1000 counts
PR 1000	Set another Position Relative move
BGX	Begin motion
EN	End of Program

Note: Instead of EN, use the RE command to end the error subroutine and limit subroutine. Use the RI command to end the input interrupt (ININT) subroutine.

EO

ECHO

DESCRIPTION:

The EO command turns the echo on or off. If the echo is off, characters input over the bus will not be echoed back.

ARGUMENTS: EO n

where n=0 or 1. 0 turns echo off, 1 turns echo on.

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

INSTRUCTION	INTERPRETATION
EO 0	Turns echo off
EO 1	Turns echo on

EP

CAM TABLE INTERVALS AND STARTING POINT

DESCRIPTION:

The EP command defines the ECAM table intervals and offset. The offset is the master position of the first ECAM table entry. The interval is the difference of the master position between 2 consecutive table entries. This command effectively defines the size of the ECAM table. The parameter m is the interval and n is the starting point. Up to 257 points may be specified.

ARGUMENTS: EP m,n

where m is a positive integer in the range between 1 and 32, 767 and n is an integer between -2,147,483,648 and 2,147,483,647.

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes (m only)	

OPERAND USAGE:

_EP contains the value of the interval m.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“EB” on page 12-53	Choose Ecam master
“EA” on page 12-52	Enable Ecam
“EG” on page 12-56	Engage Ecam
“EM” on page 12-57	Specify Ecam Cycle
“EQ” on page 12-62	Ecam quit
“ET” on page 12-66	Electronic CAM table

EXAMPLES:

INSTRUCTION	INTERPRETATION
EP 20,100	Sets the cam master points to 100,120,140 . . .
D = _EP	Returns interval (m)

EQ

ECAM QUIT (DISENGAGE)

DESCRIPTION:

The EQ command disengages an electronic cam slave axis at the specified master position. Separate points can be specified for each axis. If a value is specified outside of the master's range, the slave will disengage immediately.

ARGUMENTS: EQ x,y,z,w

where x,y,z,w are the master positions at which the XYZW axes are to be disengaged.

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_EQX contains 1 if axis is waiting to start, 2 if axis is waiting to stop, 3 if both waiting to start and stop and 0 if ECAM engaged or already stopped.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"EA" on page 12-52	Choose Ecam master
"EB" on page 12-53	Enable Ecam
"EG" on page 12-56	Engage Ecam
"EM" on page 12-57	Specify Ecam Cycle
"EP" on page 12-61	CAM table intervals & starting point
"ET" on page 12-66	Electronic CAM table

EXAMPLES:

INSTRUCTION	INTERPRETATION
EQ 300,700	Disengages the X and Y motors at master positions 300 and 700 respectively.

Note: This command is not a trippoint. This command will not hold the execution of the program flow. If the execution needs to be held until master position is reached, use MF or MR command.

ER

ERROR LIMIT

DESCRIPTION:

The ER command sets the magnitude of the X,Y,Z and W-axis position errors that will trigger an error condition. When the limit is exceeded, the Error output will go low (true). If the Off On Error (OE1) command is active, the motors will be disabled. The units of ER are quadrature counts.

ARGUMENTS: ERX = x

x,y,z,w are unsigned numbers in the range 1 to 32767

USAGE:

While Moving	Yes	Default Value 16384
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	ER ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_ERx contains the value of the Error limit for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"OP" on page 12-119	Off-On Error
#POSERR	Automatic Error Subroutine

EXAMPLES:

INSTRUCTION	INTERPRETATION
ER 200,300,400,600	Set the X-axis error limit to 200, the Y-axis error limit to 300, the Z-axis error limit to 400, and the W-axis error limit to 600.
ER ,1000	Sets the Y-axis error limit to 1000, leave the X-axis error limit unchanged.
ER ?,?,?,?	Return X,Y,Z and W values
00200,00100,00400,00600	
ER ?	Return X value
00200	
V1=_ERX	Assigns V1 value of ERX
V1=	Returns V1
00200	

ER

Hint: The error limit specified by ER should be high enough as not to be reached during normal operation. Examples of exceeding the error limit would be a mechanical jam, or a fault in a system component such as encoder or amplifier.

ES

ELLIPSE SCALE

DESCRIPTION:

The ES command divides the resolution of one of the axes in a vector mode. This allows the generation of an ellipse instead of a circle.

The command has two parameters, m and n, (ES m,n), and it applies to the axes designated by the VM command (VMXY, for example). When m>n, the resolution of the first axis (X in the example), will be divided by the ratio m/n. When m<n, the resolution of the second axis (Y in the example), will be divided by n/m. The resolution change applies for the purpose of generating the VP and CR commands. Note that this command results in one axis moving a distance specified by the CR and VP commands while the other one moves a larger distance.

ARGUMENTS: ES m,n

where m and n are positive integers in the range between 1 and 65,535.

USAGE:

While Moving	Yes	Default Value 1,1
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VM" on page 12-170	Vector Mode
"CR" on page 12-36	Circle move
"VP" on page 12-172	Vector position

EXAMPLES:

INSTRUCTION	INTERPRETATION
VMXY;ES3,4	Divide Y resolution by 4/3
VMZX;ES2,3	Divide X resolution by 3/2

ET

ELECTRONIC CAM TABLE

DESCRIPTION:

The ET command sets the ECAM table entries for the slave axes.. The values of the master axes are not required. The slave entry (n) is the position of the slave axes when the master is at the point n * i + o, where i and o are the interval and offset as determined by the EP command.

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

where n is an integer between 0 and 256 and the parameters x,y,z,w are integers in the range between -2,147,438,648, and 2,147,438,647.

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"EA" on page 12-52	Choose Ecam master
"EB" on page 12-53	Enable Ecam
"EG" on page 12-56	Engage Ecam
"EM" on page 12-57	Specify Ecam Cycle
"EP" on page 12-61	Specify Ecam intervals and starting point
"EQ" on page 12-62	Ecam quit

EXAMPLES:

ET [7] = 1000,300,500 Specifies the position of the slave axes X, Y and Z that must be synchronized with the eighth increment of the master.

FA

ACCELERATION FEEDFORWARD

DESCRIPTION:

The FA command sets the acceleration feedforward coefficient, or returns the previously set value. This coefficient, when scaled by the acceleration, adds a torque bias voltage during the acceleration phase and subtracts the bias during the deceleration phase of a motion.

Acceleration Feedforward Bias = FA x AC x 1.5 x 10-7

Deceleration Feedforward Bias = FA x DC x 1.5 x 10-7

The Feedforward Bias product is limited to 10 Volts. FA will only be operational during independent moves.

ARGUMENTS: FA x,y,z,w

where x,y,z,w are unsigned numbers in the range 0 to 8191 decimal

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 4.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

_FAx contains the value of the feedforward acceleration coefficient for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"FV" on page 12-72	Velocity feedforward

EXAMPLES:

INSTRUCTION	INTERPRETATION
AC 500000,1000000	Set feedforward coefficient to 10 for the X-axis and 15 for the Y-axis.
FA 10,15	The effective bias will be 0.75V for X and 2.25V for Y.
FA ?,?	Return X and Y values
010,015	

Note: If the feedforward coefficient is changed during a move, then the change will not take effect until the next move.

FE

FIND EDGE

DESCRIPTION:

The FE command moves a motor until a transition is seen on the homing input for that axis. The direction of motion depends on the initial state of the homing input (use the CN command to configure the polarity of the home input). Once the transition is detected, the motor decelerates to a stop. This command is useful for creating your own homing sequences.

ARGUMENTS: FE XYZW

where X,Y,Z,W specify XYZ or W axis. No argument specifies all axes.

USAGE:

While Moving	No	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“FI” on page 12-70	Find Index
“HM” on page 12-77	Home
“BG” on page 12-21	Begin
“AC” on page 12-6	Acceleration Rate
“DC” on page 12-41	Deceleration Rate
“SP” on page 12-142	Speed for search

EXAMPLES:

INSTRUCTION	INTERPRETATION
FE	Set find edge mode
BG	Begin all axes
FEX	Only find edge on X
BGX	
FEY	Only find edge on Y
BGY	
FEZW	Find edge on Z and W
BGZW	

Hint: Find Edge only searches for a change in state on the Home Input. Use FI (Find Index) to search for the encoder index. Use HM (Home) to search for both the Home input and the Index. Remember to specify BG after each of these commands.

FI

FIND INDEX

DESCRIPTION:

The FI and BG commands move the motor until an encoder index pulse is detected. The controller looks for a transition from low to high. When the transition is detected, motion stops and the position is defined as zero. To improve accuracy, the speed during the search should be specified as 500 counts/s or less. The FI command is useful in custom homing sequences. The direction of motion is specified by the sign of the JG command.

ARGUMENTS: FI XYZW

where X,Y,Z,W specify XYZ or W axis. No argument specifies all axes.

USAGE:

While Moving	No	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"FE" on page 12-68	Find Edge
"HM" on page 12-77	Home
"BG" on page 12-21	Begin
"AC" on page 12-6	Acceleration Rate
"DC" on page 12-41	Deceleration Rate
"SP" on page 12-142	Search Speed

EXAMPLES:

INSTRUCTION	INTERPRETATION
#HOME	Home Routine
JG 500	Set speed and forward direction
FIX	Find index
BGX	Begin motion
AMX	After motion
MG "FOUND INDEX"	

Hint: Find Index only searches for a change in state on the Index. Use FE to search for the Home. Use HM (Home) to search for both the Home input and the Index. Remember to specify BG after each of these commands.

FL

FORWARD SOFTWARE LIMIT

DESCRIPTION:

The FL command sets the forward Software position limit. If this limit is exceeded during motion, motion on that axis will decelerate to a stop. Forward motion beyond this limit is not permitted. The forward limit is activated at X+1, Y+1, Z+1, W+1. The forward limit is disabled at 2147483647. The units are in counts.

ARGUMENTS: FL x,y,z,w FLX=x

where x,y,z,w are signed integers in the range -2147483648 to 2147483647
2147483647 turns off the forward limit

USAGE:

While Moving	Yes	Default Value 2147483647
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	FL ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_FLx contains the value of the forward limit switch for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"BL" on page 12-23	Reverse Limit

EXAMPLES:

INSTRUCTION	INTERPRETATION
FL 150000	Set forward limit to 150000 counts on the X-axis
#TEST	Test Program
AC 1000000	Acceleration Rate
DC 1000000	Deceleration Rate
FL 15000	Forward Limit
JG 5000	Jog Forward
BGX	Begin
AMX	After Limit
TPX	Tell Position
EN	End

Hint: The SSC also provides hardware limits.

FV

VELOCITY FEEDFORWARD

DESCRIPTION:

The FV command sets the velocity feedforward coefficient, or returns the previously set value. This coefficient, generates an output bias signal in proportion to the commanded velocity.Velocity feedforward bias = 1.22 x 10-6 x FV x Velocity [in ct/s].

For example, if FV=10 and the velocity is 200,000 count/s, the velocity feedforward bias equals 2.44 volts.

ARGUMENTS: FV x,y,z,w

where x,y,z,w are unsigned numbers in the range 0 to 8191 decimal

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 3.0
Command Line	Yes	
Can be Interrogated	Yes	FV ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_FVx contains the feedforward velocity for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"FA" on page 12-67	Acceleration feedforward

EXAMPLES:

INSTRUCTION	INTERPRETATION
FV 10,20	Set feedforward coefficients to 10 and 20 for x and y respectively.
JG 30000,80000	This produces 0.366 volts for x and 1.95 volts for y.
FV ?,?	Return the x and y values.
010,020	

GA

MASTER AXIS FOR GEARING

DESCRIPTION:

The GA command specifies the master axis for electronic gearing. Only one master may be specified. The master may be the main encoder input, auxiliary encoder input, or the commanded position of any axis. The master may also be the commanded vector move in a coordinated motion of LM or VM type. When the master is a simple axis, it may move in any direction and the slave follows. When the master is a commanded vector move, the vector move is considered positive and the slave will move forward if the gear ratio is positive, and backward if the gear ratio is negative. The slave axes and ratios are specified with the GR command and gearing is turned off by the command GR0.

ARGUMENTS: GA n

where n = X or Y or Z or W for main encoder as axis master

n = CX or CY or CZ or CW or CA,CB,CC,CD,CE,CF,CG,CH for command position as master axis

n = S for vector motion as master

n = DX or DY or DZ or DW for auxiliary encoder as master

USAGE:

While Moving	No	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"GR" on page 12-76	Gear Ratio

EXAMPLES:

INSTRUCTION	INTERPRETATION
#GEAR	Gear program
GAX	Specify X axis as master

GA

GR ,.5,-2.5	Specify Y and Z ratios
JG 5000	Specify master jog speed
BGX	Begin motion
WT 10000	Wait 10000 msec
STX	Stop

Hint: Using the command position as the master axis is useful for gantry applications. Using the vector motion as master is useful in generating Helical motion.

GN

GAIN

DESCRIPTION:

The GN command sets the gain of the control loop or returns the previously set value. It fits in the z-transform control equation as follows:

$$D(z) = GN(z-ZR)/z$$

ARGUMENTS: GN x,y,z,w GNX=x

where x,y,z,w are unsigned integers in the range 0 to 2047 decimal.

USAGE:

While Moving	Yes	Default Value 70
In a Program	Yes	Default Format 4
Command Line	Yes	
Can be Interrogated	Yes	GN ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_GNx contains the value of the gain for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"ZR" on page 12-181	Zero
"KI" on page 12-93	Integrator
"KP" on page 12-94	Proportional
"KD" on page 12-92	Derivative

EXAMPLES:

INSTRUCTION	INTERPRETATION
GN 12,14,15,20	Set X-axis gain to 12 Set Y-axis gain to 14
	Set Z-axis gain to 15 Set W-axis gain to 20
GN 6	Set X-axis gain to 6 Leave other gains unchanged
GN ,8	Set Y-axis gain to 8 Leave other gains unchanged
GN ?,?,?,?	Returns X,Y,Z,W gains
0006,0008,0015,0020	
GN ?	Returns X gain
0006	
GN ,?	Returns Y gain
0008	

GR

GEAR RATIO

DESCRIPTION:

GR specifies the Gear Ratios for the geared axes in the electronic gearing mode. The master axis is defined by the GAX or GAY or GAZ or GAW command. The gear ratio may be different for each geared axis and range between +/-127.9999. The slave axis will be geared to the actual position of the master. The master can go in both directions. GR 0,0,0,0 disables gearing for each axis. A limit switch also disables the gearing.

ARGUMENTS: GR x,y,z,w GRX=x

where x,y,z,w are signed numbers in the range +/-127, with a fractional resolution of .0001.

0 disables gearing

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 3.4
Command Line	Yes	
Can be Interrogated	Yes	GR ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_GRx contains the value of the gear ratio for the specified axis, 'x'..

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"GA" on page 12-73	Master Axis

EXAMPLES:

INSTRUCTION	INTERPRETATION
#GEAR	
MOY	Turn off servo to Y motor
GAY	Specify master axis as Y
GR .25,,-5	Specify X and Z gear ratios
EN	End program

Now when the Y motor is rotated by hand, the X will rotate at 1/4th the speed and Z will rotate 5 times the speed in the opposite direction.

HM

HOME

DESCRIPTION:
The HM command performs a three-stage homing sequence for servo systems and two stage sequence for stepper motor operation.

For servo motor operation:

The first stage consists of the motor moving at the user programmed speed until detecting a transition on the homing input for that axis. The direction for this first stage is determined by the initial state of the Homing Input. Once the homing input changes state, the motor decelerates to a stop. The state of the homing input can be configured using the CN command.

The second stage consists of the motor changing directions and slowly approaching the transition again. When the transition is detected, the motor is stopped instantaneously.

The third stage consists of the motor slowly moving forward until it detects an index pulse from the encoder. It stops at this point and defines it as position 0.

For stepper mode operation, the sequence consists of the first two stages. The frequency of the motion in stage 2 is 256 cts/ sec.

ARGUMENTS: NoneUSAGE:

While Moving	No	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:
_HMx contains the state of the home switch for the specified axis, ‘x’.

HM

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"CN" on page 12-35	Configure Home
"FI" on page 12-70	Find Index Only
"FE" on page 12-68	Find Home Only

EXAMPLES:

INSTRUCTION	INTERPRETATION
HM	Set Homing Mode for all axes
BG	Home all axes
BGX	Home only the X-axis
BGY	Home only the Y-axis
BGZ	Home only the Z-axis
BGW	Home only the W-axis

Hint: You can create your own custom homing sequence by using the FE (Find Home Sensor only) and FI (Find Index only) commands.

HX

HALT EXECUTION

DESCRIPTION:

The HX command halts the execution of any of the four programs that may be running independently in multitasking. The parameter n specifies the program to be halted.

ARGUMENTS: HXn

where n is an integer in the range of 0 to 3 to indicate the strand number.

USAGE:

While Moving	Yes	Default Value n = 0
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

When used as an operand, _HXn contains the running status of thread n with:

0 Thread not running
1 Thread is running
2 Thread has stopped at trippoint

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"XQ" on page 12-180	Execute program

EXAMPLES:

INSTRUCTION	INTERPRETATION
XQ #A	Execute program #A, thread zero
XQ #B,3	Execute program #B, thread three
HX0	Halt thread zero
HX3	Halt thread three

II

INPUT INTERRUPT

DESCRIPTION:

The II command enables the interrupt function for the specified inputs. m specifies the beginning input and n specifies the final input in the range. For example, II 2,4 specifies interrupts occurring for Input 2, Input 3 and Input 4. m=0 disables the Input Interrupts. If only the m parameter is given, only that input will generate an interrupt.

The parameter o is an interrupt mask for all eight inputs. If m and n are unused, o contains a number with the mask. A 1 designates that input to be enabled for an interrupt.

Example: II,,5 enables inputs 1 and 3

If any of the specified inputs go low during program execution, the program will jump to the subroutine with label #ININT. Any trippoints set by the program will be cleared. The RI command is used to return from the #ININT routine. The RI command also re-enables input interrupts. To avoid returning to the main program on an interrupt, use the command ZS to zero the subroutine stack.

ARGUMENTS: II m,n,o

where:
m is an integer in the range 0 to 8 decimal
n is an integer in the range 1 to 8 decimal
o is an integer in the range 0 to 255decimal

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 3.0 (mask only)
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"RI" on page 12-133	Return from Interrupt
#ININT	Interrupt Subroutine
"AI" on page 12-11	Trippoint for input

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A	Program A
II 1	Specify interrupt on input 1
JG 5000	Specify jog
BGX	Begin motion
#LOOP;JP #LOOP	Loop
EN	End Program
#ININT	Interrupt subroutine
STX;MG “INTERRUPT”	Stop X, print message
AMX	After stopped
#CLEAR;JP#CLEAR,@IN[1]=0	Check for interrupt clear
BGX	Begin motion
RI	Return to main program

IL

INTEGRATOR LIMIT

DESCRIPTION:

The IL command limits the effect of the integrator function in the filter to a certain voltage. For example, IL 2 limits the output of the integrator of the X-axis to the +/-2 Volt range.

A negative parameter also freezes the effect of the integrator during the move. For example, IL -3 limits the integrator output to +/-3V. If, at the start of the motion, the integrator output is 1.6 Volts, that level will be maintained through the move. Note, however, that the KD and KP terms remain active in any case.

ARGUMENTS: IL x,y,z,w ILX=x

where x,y,z,w are numbers in the range -9.9988 to 9.9988 Volts with a resolution of 0.0003.

USAGE:

While Moving	Yes	Default Value 9.9988
In a Program	Yes	Default Format 1.4
Command Line	Yes	
Can be Interrogated	Yes	IL ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_ILx contains the value of the integrator limit for the specified axis, ‘x’.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“KI” on page 12-93	Integrator

EXAMPLES:

INSTRUCTION	INTERPRETATION
KI 2,3,5,8	Integrator constants
IL 3,2,7,2	Integrator limits
IL ?	Returns the X-axis limit
3.0000	

IN

INPUT VARIABLE

DESCRIPTION:

The IN command allows a variable to be input from a keyboard. When the IN command is executed in a program, the prompt message is displayed. The operator then enters the variable value followed by a carriage return. The entered value is assigned to the specified variable name.

The IN command holds up execution of following commands in a program until a carriage return or semicolon is detected. If no value is given prior to a semicolon or carriage return, the previous variable value is kept. Input Interrupts, Error Interrupts and Limit Switch Interrupts will still be active.

ARGUMENTS: IN{P1 or P2} "m" , n {So}

where:
"m" is the prompt message. May be letters, numbers, or symbols up to 35 characters and must be placed in quotations
n is the name of variable to hold value returned from input
{P1} specifies Port1, the MAIN serial port (optional - Main port by default)
{P2} specifies Port2, the AUX serial port (optional - Main port by default)
{So} specifies string data and o is the number of characters from 1 to 6

Note 1: The IN command defaults to {P1}.

Note 2: Configure Port 2 communications with the CC command before using IN command with Port 2.

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format Position Format
Command Line	No	
Can be Interrogated	No	
Used as an Operand	No	

IN

EXAMPLES:

Operator specifies length of material to be cut in inches and speed in inches/sec (2 pitch lead screw, 2000 counts/rev encoder).

INSTRUCTION	INTERPRETATION
#A	Program A
CI -1	Clear Input Buffer*
IN “Enter Speed(in/sec)”,V1	Prompt operator for speed
IN “Enter Length(in)”,V2	Prompt for length
V3=V1*4000	Convert units to counts/sec
V4=V2*4000	Convert units to counts
SP V3	Speed command
PR V4	Position command
BGX	Begin motion
AMX	Wait for motion complete
MG “MOVE DONE”	Print Message
EN	End Program

*Note: It is a good practice to clear the input buffer before executing the IN command.

IP

INCREMENT POSITION

DESCRIPTION:

The IP command allows for a change in the command position while the motor is moving. This command does not require a BG. The command has three effects depending on the motion being executed. The units of this are quadrature.

Case 1: Motor is standing still

An IP x,y,z,w command is equivalent to a PR x,y,z,w and BG command. The motor will move to the specified position at the requested slew speed and acceleration.

Case 2: Motor is moving towards specified position

An IP x,y,z,w command will cause the motor to move to a new position target, which is the old target plus x,y,z,w. x,y,z,w must be in the same direction as the existing motion.

Case 3: Motor is in the Jog Mode

An IP x,y,z,w command will cause the motor to instantly try to servo to a position x,y,z,w from the present instantaneous position. The SP and AC parameters have no effect. This command is useful when synchronizing 2 axes in which one of the axis' speed is indeterminant due to a variable diameter pulley.

An IP used as an interrogation command will return the current position of the axis.

ARGUMENTS: IP x,y,z,w IPX = x

x,y,z,w are signed numbers in the range -2147483648 to 2147483647 decimal.

IP

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format 7.0
Command Line	Yes	
Can be Interrogated	Yes	IP ?,?,?,?
Used as an Operand	No	

EXAMPLES:

INSTRUCTION	INTERPRETATION
IP 50	50 counts with set acceleration and speed
#CORRECT	Label
AC 100000	Set acceleration
JG 10000;BGX	Jog at 10000 counts/sec rate
WT 1000	Wait 1000 msec
IP 10	Move the motor 10 counts instantaneously
STX	Stop Motion

IT

INDEPENDENT TIME CONSTANT SMOOTHING FUNCTION

DESCRIPTION:

The IT command filters the acceleration and deceleration functions in independent moves of JG, PR, PA type to produce a smooth velocity profile. The resulting profile, known as S-curve, has continuous acceleration and results in reduced mechanical vibrations. IT sets the bandwidth of the filter where 1 means no filtering and 0.004 means maximum filtering. Note that the filtering results in longer motion time.

ARGUMENTS: IT x,y,z,w ITX=x

where x,y,z,w are positive numbers in the range between 0.004 and 1.0 with a resolution of 1/256

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format 7.0
Command Line	Yes	
Can be Interrogated	Yes	IT ?,?,?,?
Used as an Operand	No	

OPERAND USAGE:

_ITx will return the value of the independent time constant for the specified 'x' axis

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VT" on page 12-177	Vector Time Constant for smoothing vector moves

EXAMPLES:

IT 0.8, 0.6, 0.9, 0.1	Set independent time constants for x,y,z,w axes
IT ?	Return independent time constant for X-axis
0.8	

JG

JOG

DESCRIPTION:

The JG command sets the jog mode. The parameters following the JG set the slew speed of the axes. Use of the question mark returns the previously entered value or default value. The units of this are counts/second.

ARGUMENTS: JG x,y,z,w JGX=x

where: x,y,z,w are signed numbers in the range 0 to +/-8,000,000 decimal

USAGE:

While Moving	Yes	Default Value
While Moving	Yes	Default Value 16385
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"BG" on page 12-21	Begin
"ST" on page 12-144	Stop
"AC" on page 12-6	Acceleration
"DC" on page 12-41	Deceleration
"IP" on page 12-85	Increment Position
"TV" on page 12-161	Tell Velocity

EXAMPLES:

INSTRUCTION	INTERPRETATION
JG 100,500,2000,5000	Set for jog mode with a slew speed of 100 counts/sec for the X-axis, 500 counts/sec for the Y-axis, 2000 counts/sec for the Z-axis, and 5000 counts/sec for W-axis.
BG	Begin Motion
JG ,,-2000	Change the Z-axis to slew in the negative direction at -2000 counts/sec.

JP

JUMP TO PROGRAM LOCATION

DESCRIPTION:

The JP command causes a jump to a program location on a specified condition. The program location may be any program line number or label. The condition is a conditional statement which uses a logical operator such as equal to or less than. A jump is taken if the specified condition is true.

ARGUMENTS: JP location,condition

where: location is a program line number or label
condition is a conditional statement using a logical operator

The logical operators are:

< less than
> greater than
= equal to
<= less than or equal to
>= greater than or equal to
<> not equal to

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	No	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

INSTRUCTION	INTERPRETATION
JP #POS1,V1<5	Jump to label #POS1 if variable V1 is less than 5
JP #A,V7*V8=0	Jump to #A if V7 times V8 equals 0
JP #B	Jump to #B (no condition)

Hint: JP is similar to an IF, THEN command. Text to the right of the comma is the condition that must be met for a jump to occur. The destination is the specified label before the comma.

JS

JUMP TO SUBROUTINE

DESCRIPTION:

The JS command will change the sequential order of execution of commands in a program. If the jump is taken, program execution will continue at the line specified by the destination parameter, which can be either a line number or label. The line number of the JS command is saved and after the next EN command is encountered (End of subroutine), program execution will continue with the instruction following the JS command. There can be a JS command within a subroutine. These can be nested 16 deep.

A jump is taken if the specified condition is true. Conditions are tested with logical operators. The logical operators are:

< less than or equal to

> greater than

= equal to

<= less than or equal to

>= greater than or equal to

<> not equal

ARGUMENTS: JS destination,condition

Where, destination is a line number or label condition is a conditional statement using a logical operator

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	No	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"JP" on page 12-89	Jump to line or label
"ZS" on page 12-182	Zero subroutine stack

EXAMPLES:

INSTRUCTION	INTERPRETATION
JS #SQUARE,V1<5	Jump to subroutine #SQUARE if V1 is less than 5
JS #LOOP,V1<>0	Jump to #LOOP if V1 is not equal to 0
JS #A	Jump to subroutine #A (no condition)

KD

DERIVATIVE CONSTANT

DESCRIPTION:

KD designates the derivative constant in the controller filter. The filter transfer function is

$$D(z) = 4\zeta^*KP + 4\zeta^*KD(z-1)/z + KIz/2\,(z-1)$$

ARGUMENTS: KD x,y,z,w KDX=x

where x,y,z,w are unsigned numbers in the range 0 to 4095.875 with a resolution of 1/32.

USAGE:

While Moving	Yes	Default Value 64
In a Program	Yes	Default Format 4.2
Command Line	Yes	
Can be Interrogated	Yes	KD ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_KDx contains the value of the derivative constant for the specified axis, 'x'..

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"ZR" on page 12-181	Zero
"KI" on page 12-93	Integrator
"KP" on page 12-94	Proportional

EXAMPLES:

INSTRUCTION	INTERPRETATION
KD 100,200,300,400.25	Specify KD
KD ?,?,?,?	Return KD
0100.00,0200.00,0300.00,0400.25	

KI

INTEGRATOR

DESCRIPTION:

The KI command sets the integral gain of the control loop. It fits in the control equation as follows:

$D(z) = 4\zeta * KP + 4\zeta * KD(z-1)/z + KI\ z/2(z-1)$

The integrator term will reduce the position error at rest to zero.

ARGUMENTS: KI x,y,z,w KIX=x

where x,y,z,w are unsigned numbers in the range 0 to 2047.875 with a resolution of 1/32.

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 4.0
Command Line	Yes	
Can be Interrogated	Yes	KI ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_KIx contains the value of the derivative constant for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"GN" on page 12-75	Gain
"KP" on page 12-94	Proportional Constant
"ZR" on page 12-181	Zero
"KI" on page 12-93	Integrator
"IL" on page 12-82	Integrator Limit

EXAMPLES:

INSTRUCTION	INTERPRETATION
KI 12,14,16,20	Specify x,y,z,w-axis integral
KI 7	Specify x-axis only
KI ,,8	Specify z-axis only
KI ?,?,?,?	Return X,Y,Z,W
0007,0014,0008,0020	KI values

KP

PROPORTIONAL CONSTANT

DESCRIPTION:

KP designates the proportional constant in the controller filter. The filter transfer function is

$$D(z) = 4\zeta * KP + 4\zeta * KD(z-1)/z + KI\ z/2(z-1)$$

For further details see the section Theory of Operation.

ARGUMENTS: KP x,y,z,w KPX=x

where x,y,z,w are unsigned numbers in the range 0 to 1023.875 with a resolution of 1/32.

USAGE:

While Moving	Yes	Default Value 6
In a Program	Yes	Default Format 4.2
Command Line	Yes	
Can be Interrogated	Yes	KP?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_KPx contains the value of the proportional constant for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"GN" on page 12-75	Gain
"KP" on page 12-94	Proportional Constant
"ZR" on page 12-181	Zero
"KI" on page 12-93	Integrator
"IL" on page 12-82	Integrator Limit

KS

STEP MOTOR SMOOTHING

DESCRIPTION:

The KS parameter smooths the frequency of the step motor pulses. Larger values of KS provide greater smoothness. This parameter will also increase the motion time by 3KS sampling periods.

ARGUMENTS: KS x,y,z,w KSX=x

where x,y,z,w are positive integers in the range between 1 and 16 with a resolution of 1.

USAGE:

While Moving	Yes	Default Value 2
In a Program	Yes	Default Format 4.0
Command Line	Yes	
Can be Interrogated	Yes	KS ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_KSx contains the value of the derivative constant for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"MT" on page 12-112	Motor Type

EXAMPLES:

INSTRUCTION	INTERPRETATION
KS 2, 4 , 8	Specify x,y,z axes
KS 5	Specify x-axis only
KS ,,15	Specify z-axis only

Hint: KS is valid for step motor only.

LE

LINEAR INTERPOLATION END

DESCRIPTION:

LE signifies the end of a linear interpolation sequence. It follows the last LI specification in a linear sequence. After the LE specification, the controller issues commands to decelerate the motors to a stop. LE? returns the length of the vector in counts. The VE command is interchangeable with the LE command.

ARGUMENTS: None

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	Yes	LE ?
Used as an Operand	Yes	

OPERAND USAGE:

_LE contains the length of the vector in counts.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"LI" on page 12-98	Linear Distance
"BG" on page 12-21	BGS - Begin Sequence
"LM" on page 12-100	Linear Interpolation Mode
"VS" on page 12-176	Vector Speed
"VA" on page 12-164	Vector Acceleration
"VD" on page 12-166	Vector Deceleration

EXAMPLES:

INSTRUCTION	INTERPRETATION
LM ZW	Specify linear interpolation mode
LI ,,100,200	Specify linear distance
LE	End linear move
BGS	Begin motion

_LF*

FORWARD LIMIT SWITCH OPERAND (KEYWORD)

DESCRIPTION:

*The _LF operand contains the state of the forward limit switch for the specified axis, 'x'.

_LFx where x is the specified axis.

Note: This is not a command.

USAGE:

Used as an Operand	Yes

EXAMPLES:

INSTRUCTION	INTERPRETATION
MG _LF X	Display the status of the X axis forward limit switch

LI

LINEAR INTERPOLATION DISTANCE

DESCRIPTION:

The LI x,y,z,w command specifies the incremental distance of travel for each axis in the Linear Interpolation (LM) mode. LI parameters are relative distances given with respect to the current axis positions. Up to 511 LI specifications may be given ahead of the Begin Sequence (BGS) command. Additional LI commands may be sent during motion when the SSC sequence buffer frees additional spaces for new vector segments. The Linear End (LE) command must be given after the last LI specification in a sequence. This command tells the controller to decelerate to a stop at the last LI command. It is the responsibility of the user to keep enough LI segments in the SSC sequence buffer to ensure continuous motion.

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. It should be noted that the SSC 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 speed of these axes will be computed from VS2=XS2+YS2+ZS2 where XS, YS and ZS are the speed of the X,Y and Z axes. If the LI command specifies only X and Y, the speed of Z will still be used in the vector calculations. The controller always uses the axis specifications from LM, not LI, to compute the speed. The parameter n is optional and can be used to define the vector speed that is attached to the motion segment.

ARGUMENTS: LI x,y,z,w

x,y,z,w are signed integers in the range -8,388,607 to 8,388,607 and represent incremental move distance

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"LE" on page 12-96	Linear end
"BG" on page 12-21	BGS - Begin sequence
"LM" on page 12-100	Linear Interpolation Mode
"CS" on page 12-38	Clear Sequence
"VS" on page 12-176	Vector Speed
"VA" on page 12-164	Vector Acceleration
"VD" on page 12-166	Vector Deceleration

EXAMPLES:

INSTRUCTION	INTERPRETATION
LM XYZ	Specify linear interpolation mode
LI 1000,2000,3000	Specify distance
LE	Last segment
BGS	Begin sequence

LM

LINEAR INTERPOLATION MODE

DESCRIPTION:

The LM XYZW command specifies the linear interpolation mode where XYZW denote the axes for linear interpolation. Any set of 1,2,3 or 4 axes may be used for linear interpolation. LI x,y,z,w commands are used to specify the travel distances for linear interpolation. The LE command specifies the end of the linear interpolation sequence. Several LI commands may be given as long as the SSC sequence buffer has room for additional segments. LM? will return the number of spaces available in the sequence buffer for additional LI commands. Once the LM command has been given, it does not need to be given again unless the VM command has been used.

It should be noted that the SSC 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 speed of these axes will be computed from $VS^2=XS^2+YS^2+ZS^2$, where XS, YS and ZS are the speed of the X,Y and Z axes. If the LI command specifies only X and Y, the speed of Z will still be used in the vector calculations. The controller always uses the axis specifications from LM, not LI, to compute the speed.

ARGUMENTS: LM XYZW

XYZW denote X,Y,Z or W axes

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	Yes	LM ?
Used as an Operand	Yes	

OPERAND USAGE:

_LM will return the number of spaces available in the sequence buffer for additional LI commands.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“LE” on page 12-96	Linear end
“LM” on page 12-100	Linear Distance
“BG” on page 12-21	BGS - Begin sequence
“VA” on page 12-164	Vector acceleration
“VS” on page 12-176	Vector Speed
“VD” on page 12-166	Vector deceleration
“AV” on page 12-20	Vector distance
“CS” on page 12-38	_CS - Sequence counter

EXAMPLES:

INSTRUCTION	INTERPRETATION
LM XYZW	Specify linear interpolation mode
VS 10000; VA 100000;VD 1000000	Specify vector speed, acceleration and deceleration
LI 100,200,300,400	Specify linear distance
LI 200,300,400,500	Specify linear distance
LE; BGS	Last vector, then begin motion

_LR*

REVERSE LIMIT SWITCH OPERAND (KEYWORD)

DESCRIPTION:

*The _LR operand contains the state of the reverse limit switch for the specified axis, 'x'.

_LRx where x is the specified axis.

Note: This is not a command.

USAGE:

Used as an Operand	Yes

EXAMPLES:

INSTRUCTION	INTERPRETATION
MG _LR X	Display the status of the X axis reverse limit switch

LS

LIST

DESCRIPTION:

The LS command sends a listing of the program in memory to the main serial port. The listing will start with the line pointed to by the first parameter, which can be either a line number or a label. If no parameter is specified, it will start with line 0. The listing will end with the line pointed to by the second parameter—again either a line number or label. If no parameter is specified, the listing will go to the last line of the program.

ARGUMENTS: LS n,m

where n,m are valid numbers from 0 to 499, or labels. n is the first line to be listed, m is the last.

USAGE:

While Moving	Yes	Default Value 0, Last Line
In a Program	No	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

INSTRUCTION	INTERPRETATION
:LS #A,6	List program starting at #A through line 6
002 #A	
003 PR 500	
004 BGX	
005 AM	
006 WT 200	

Hint: Remember to quit the Edit Mode <cntrl> Q prior to giving the LS command.

LZ

LEADING ZEROS

DESCRIPTION:

The LZ command is used for formatting the values returned from interrogation commands or interrogation of variables and arrays. By enabling the LZ function, all leading zeros of returned values will be removed.

ARGUMENTS: LZ n

where n is

1 to remove leading zeros
0 to disable the leading zero removal.

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	Yes	LZ ?
Used as an Operand	Yes	

OPERAND USAGE:

_LZ contains the state of the LZ function. '0' is disabled and '1' is enabled.

EXAMPLES:

INSTRUCTION	INTERPRETATION
LZ 0	Disable the LZ function
TPX	Interrogate the controller for current position of X axis
0000021645.0000	Value returned by the controller
VAR1=	Request value of variable "VAR1" (previously set to 10)
0000000010.0000	Value of variable returned by controller
LZ1	Enable LZ function
TPX	Interrogate the controller for current position of X axis
21645.0000	Value returned by the controller
VAR1=	Request value of variable "VAR1" (previously set to 10)
10.0000	Value of variable returned by controller

MC

MOTION COMPLETE - "IN POSITION"

DESCRIPTION:

The MC command is a trippoint used to control the timing of events. This command will hold up execution of the following commands until the current move on the specified axis or axes is completed and the encoder reaches or passes the specified position. Any combination of axes or a motion sequence may be specified with the MC command. For example, MC XY waits for motion on both the X and Y axis to be complete. MC with no parameter specifies that motion on all axes is complete. TW x,y,z,w sets the timeout to declare an error if the encoder is not in position within the specified time. If a timeout occurs, the trippoint will clear and the stopcode will be set to 99. An application program will jump to the special label #MCTIME.

ARGUMENTS: MC XYZW

where X,Y,Z,W specifies X,Y,Z or W axis or sequence. No argument specifies that motion on all axes is complete.

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"BG" on page 12-21	(returns a 0 if motion complete)
"AM" on page 12-13	
"TW" on page 12-162	

EXAMPLES:

INSTRUCTION	INTERPRETATION
#MOVE	Program MOVE
PR 5000,5000,5000,5000	Position relative moves
BG X	Start the X-axis
MC X	After the move is complete on X,
BG Y	Start the Y-axis
MC Y	After the move is complete on Y,

MC

```
BG Z                        Start the Z-axis
MC Z                        After the move is complete on Z
BG W                        Start the W-axis
MC W                        After the move is complete on W
EN                          End of Program
#F;DP 0,0,0,0               Program F Position
PR 5000,6000,7000,8000      relative moves
BG                          Start X,Y,Z and W axes
MC                          After motion complete on all axes
MG "DONE"; TP               Print message
EN                          End of Program
```

Hint: MC can be used to verify that the actual motion has been completed.

MF

FORWARD MOTION TO POSITION

DESCRIPTION:

The MF command is a trippoint used to control the timing of events. This command will hold up the execution of the following command until the specified motor moves forward and crosses the position specified. The units of the command are in quadrature counts. Only one axis may be specified at a time. The MF command can also be used when the encoder is the master and not under servo control.

ARGUMENTS: MFx or MF,y or MF,,z or MF,,,w MFX=X

where x,y,z,w are signed integers in the range -2147483648 to 2147483647 decimal

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"AD" on page 12-8	for relative distances
"AP" on page 12-15	

EXAMPLES:

INSTRUCTION	INTERPRETATION
#TEST	Program B
DP0	Define zero
JG 1000	Jog mode (speed of 1000 counts/sec)
BG X	Begin move
MF 2000	After passing the position 2000
V1=_TPX	Assign V1 X position
MG "Position is", V1= ST	Print Message Stop
EN	End of Program

Hint: The accuracy of the MF command is the number of counts that occur in 2 msec. Multiply the speed by 2 msec to obtain the maximum error. MF tests for absolute position. The MF command can also be used when the specified motor is driven independently by an external device.

MG

MESSAGE

DESCRIPTION:

The MG command sends data out the bus. This can be used to alert an operator, send instructions or return a variable value.

ARGUMENTS: MG "m", {^n}, V {Fm.n or $m,n} {N} {Pn}

"m" is a text message including letters, numbers, symbols or <ctrl>G (up to 31 characters).

{^n} is an ASCII character specified by the value n

V is a variable name or array element where the following specifiers can be used for formatting:

{Fm.n} Display variable in decimal format with m digits to left of decimal, and n to the right.

{$m,n} Display variable in hexadecimal format with m digits to left of decimal, and n to the right.

{Sn} Display variable as a string of length n where n is 1 thru 6

{N} Suppress carriage return line feed.

{Pn} Specifies which serial port to send the message. 0 = main port, 1 = auxiliary port

Note: Multiple text, variables, and ASCII characters may be used, each must be separated by a comma.

Note: The order of arguments is not important.

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format Variable Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

Case 1: Message command displays ASCII strings

MG “Good Morning” Displays the string

Case 2: Message command displays variables or arrays

MG “The Answer is”, Total {F4.2} Displays the string with the content of variable TOTAL in local format of 4 digits before and 2 digits after the decimal point.

Case 3: Message command sends any ASCII characters to the port.

MG {^13}, {^30}, {^37}, {N} Sends carriage return, characters 0 and 7 followed by no carriage return line feed command to the port.

MO

MOTOR OFF

DESCRIPTION:

The MO command shuts off the servo control. The controller will continue to monitor the motor position. To turn the motor back on use the Servo Here command (SH).

ARGUMENTS: MO XYZW

where X,Y,Z,W are XYZW axes. MO? will return the state of the motor for the specified axis, 'x'.

USAGE:

While Moving	No	Default Value 0
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_MOx will return the state of the motor for the specified axis, 'x'. 0 = on and 1 = off.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"SH" on page 12-141	Servo Here

EXAMPLES:

INSTRUCTION	INTERPRETATION
MO	Turn off all motors
MOX	Turn off the X motor. Leave the other motors unchanged
MOY	Turn off the Y motor. Leave the other motors unchanged
MOZX	Turn off the Z and X motors. Leave the other motors unchanged
SH	Turn all motors on
Bob=_MOX	Sets Bob equal to the X-axis servo status
Bob=	Return value of Bob. If 1, in motor off mode, If 0, in servo mode

Hint: The MO command is useful for positioning the motors by hand. Turn them back on with the SH command.

MR

REVERSE MOTION TO POSITION

DESCRIPTION:

The MR command is a trippoint used to control the timing of events. This command will hold up the execution of the following command until the specified motor moves backward and crosses the position specified. The units of the command are in quadrature counts. Only one axis may be specified at a time. The MR command can also be used when the encoder is the master and not under servo control.

ARGUMENTS: MRx or MR,y or MR,,z or MR,,,w MRX=X

where x,y,z,w are signed integers in the range -2147483648 to 2147483647 decimal

USAGE:

While Moving	No	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"AD" on page 12-8	for relative distances
"AP" on page 12-15	

EXAMPLES:

INSTRUCTION	INTERPRETATION
#TEST	Program B
DP0	Define zero
JG 1000	Jog mode (speed of 1000 counts/sec)
BG X	Begin move
MR -3000	After passing the position -3000
V1=_TPX	Assign V1 X position
MG "Position is", V1= ST	Print Message Stop
EN	End of Program

Hint: The accuracy of the MR command is the number of counts that occur in 2 msec. Multiply the speed by 2 msec to obtain the maximum error. MR tests for absolute position. The MR command can also be used when the specified motor is driven independently by an external device.

MT

MOTOR TYPE

DESCRIPTION:

The MT command selects the type of the motor and the polarity of the drive signal. Motor types include standard servo motors which require a voltage in the range of +/- 10 Volts, and step motors which require pulse and direction signals. The polarity reversal inverts the analog signals for servo motors, and inverts logic level of the pulse train, for step motors.

ARGUMENTS: MT x,y,z,w MTX=x

where x,y,z,w are integers with

1 - Servo motor
-1 - Servo motor reversed polarity
2 - Step motor
-2 - Step motor - Active low pulses

MT? returns the value of the motor type for the specified axis, 'x'.

USAGE:

While Moving	No	Default Value 1,1,1,1
In a Program	Yes	Default Format 1
Command Line	Yes	
Can be Interrogated	Yes	MT ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_MTx contains the value of the motor type for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"CE" on page 12-31	Configure encoder

EXAMPLES:

INSTRUCTION	INTERPRETATION
MT 1,-1,2,2	Configure x as servo, y as reverse servo, z and w as steppers
MT ?,?	Interrogate motor type
V=_MTX	Assign motor type to variable

Hint: When using step motors, you must install the SM jumpers and use the J4 20-pin connector on the SSC to bring out the Pulse and Direction signals. Use the CN command to configure the pulse width.

NO

NO OPERATION

DESCRIPTION:

The NO command performs no action in a sequence, but can be used as a comment in a program. After the NO, up to 37 characters can be given to form a program comment. This helps to document a program.

ARGUMENTS: NO m,

where m is any group of letter, number, symbol or <cntrl>G

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A	Program A
NO	No Operation
NO This Program	No Operation
NO Does Absolutely	No Operation
NO Nothing	No Operation
EN	End of Program

OB

OUTPUT BIT

DESCRIPTION:

The OB n, logical expression command defines output bit n = 1 through 8 as either 0 or 1 depending on the result from the logical expression. Any non-zero value of the expression results in a one on the output.

ARGUMENTS: OB n, expression

where n is 1 through 8 denoting output bit

expression is any valid logical expression, variable or array element.

USAGE:

While Moving	Yes	Default Value
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

INSTRUCTION	INTERPRETATION
OB 1, POS1	If POS1 is non-zero, Bit 1 is high. If POS1 is zero, Bit 1 is low.
OB 2, @IN[1]&@IN[2]	If Input 1 and Input 2 are both high, then Output 2 is set high
OB 3, COUNT[1]	If the element 1 in the array is zero, clear bit 3
OB N, COUNT[1]	If element 1 in the array is zero, clear bit N

OE

OFF ON ERROR

DESCRIPTION:

The OE command enables / disables the 'Off-On-Error' function. When enabled, the SSC will shut off the motor command if a position error exceeds the error limit specified by the ER command. The OE1 command enables this function, while command OE0 disables it. When the OE command is enabled, an Abort either from the Abort input or the Abort command will shut off the motor.

If a position error is detected on an axis, and the axis is currently making an independent move, only that axis will be shut off. However, if the motion is a coordinated mode of the types VM, LM or CM, all the participating axes will be stopped.

ARGUMENTS: OE x,y,z,w

where x,y,z,w may be 0 or 1. 0 disables function. 1 enables off-on-error.

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_OEx contains the status of the off-on-error function for the specified axis, 'x'. 0 = off, 1 = on

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"AB" on page 12-5	Abort
"ER" on page 12-63	Error limit
"SH" on page 12-141	Servo Here
#POSERR	Error Subroutine

EXAMPLES:

INSTRUCTION	INTERPRETATION
OE 1,1,1,1	Enable OE on all axes
OE 0	Disable OE on X-axis other axes remain unchanged
OE ,,1,1	Enable OE on Z-axis and W-axis other axes remain unchanged
OE 1,0,1,0	Enable OE on X and Z-axis Disable OE on Y and W axis

Hint: The OE command is useful for preventing system damage on excessive error.

OF

OFFSET

DESCRIPTION:

The OF command sets a bias voltage in the motor command output or returns a previously set value. This can be used to counteract gravity or an offset in an amplifier.

ARGUMENTS: OF x,y,z,w OFX=x

where x,y,z,w are signed numbers in the range -9.998 to 9.998 volts with resolution of 0.0003.

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	Yes	OF ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_OFx contains the offset for the specified axis, 'x'.

EXAMPLES:

INSTRUCTION	INTERPRETATION
OF 1,-2,3,5	Set X-axis offset to 1, the Y-axis offset to -2, the Z-axis to 3, and the W-axis to 5
OF -3	Set X-axis offset to -3 Leave other axes unchanged
OF ,0	Set Y-axis offset to 0 Leave other axes unchanged
OF ?,?,?,?	Return offsets -3.0000,0.0000,3.0000,5.0000
OF ?	Return X offset
-3.0000	
OF ,?	Return Y offset
0.0000	

OP

OUTPUT PORT

DESCRIPTION:

The OP command sends 8 bits of data to the output port (the general use outputs 1-8) of the controller. You can use the output port to control external switches and relays. The n parameter is used to specify the number of bits effected starting with the LSB. The other bits are masked. For example, if n=2, only outputs 1 and 2 will be changed by OP m. If the n parameter is not specified, all bits will be changed.

ARGUMENTS: OP m,n

where m is an integer in the range 0 to 255 decimal, or $0 to FF hexadecimal

n is an integer in the range 1 to 8 decimal

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format 3.0
Command Line	Yes	
Can be Interrogated	Yes	OP ?
Used as an Operand	Yes	

OPERAND USAGE:

_OP contains the status of the outputs.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"SB" on page 12-139	Set output bit
"CB" on page 12-28	Clear output bit

EXAMPLES:

INSTRUCTION	INTERPRETATION
OP 0	Clear Output Port — all bits
OP $85	Set outputs 1,3,8; clear the others
OP $85,2	Set 1; clears 2. Other unchanged.
MG _OP	Message out the status of the outputs

PA

POSITION ABSOLUTE

DESCRIPTION:

The PA command will set the final destination of the next move. The position is referenced to the absolute zero. If a ? is used, then the current destination (current command position if not moving, destination if in a move) is returned. For each single move, the largest position move possible is +/- 2147483647. Units are in quadrature counts.

ARGUMENTS: PA x,y,z,w PAX=x

where x,y,z,w are signed integers in the range -2147483647 to 2147483648 decimal

USAGE:

While Moving	No	Default Value -
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	PA ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_PAx contains current destination (current command position if not moving, destination if in a move).

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“PR” on page 12-124	Position relative
“SP” on page 12-142	Speed
“AC” on page 12-6	Acceleration
“DC” on page 12-41	Deceleration
“BG” on page 12-21	Begin

EXAMPLES:

INSTRUCTION	INTERPRETATION
:PA 400,-600,500,200	X-axis will go to 400 counts Y-axis will go to -600 counts Z-axis will go to 500 counts W-axis will go to 200 counts
:PA ?,?,?,?	Returns the current commanded position 0000000,0000000,0000000,0000000
:BG	Start the move
:PA 700	X-axis will go to 700 on the next move while the
:BG	Y,Z and W-axis will travel the previously set relative distance if the preceding move was a PR move, or will not move if the preceding move was a PA move.

PF

POSITION FORMAT

DESCRIPTION:

The PF command allows the user to format the position numbers such as those returned by TP. The number of digits of integers and the number of digits of fractions can be selected with this command. An extra digit for sign and a digit for decimal point will be added to the total number of digits. If PF is minus, the format will be hexadecimal and a dollar sign will precede the characters. Hex numbers are displayed as 2's complement with the first bit used to signify the sign.

If a number exceeds the format, the number will be displayed as the maximum possible positive or negative number (i.e. 999.99, -999, $8000 or $7FF).

ARGUMENTS: PF m.n

where m is an integer between -8 and 10

n is an integer between 0 and 4

The negative sign for m specifies hexadecimal representation.

USAGE:

While Moving	Yes	Default Value 10.0
In a Program	Yes	Default Format 10.0
Command Line	Yes	
Can be Interrogated	Yes	PF ?
Used as an Operand	Yes	

OPERAND USAGE:

_PF contains the value of position format parameter.

EXAMPLES:

INSTRUCTION	INTERPRETATION
:TPX	Tell position of X
0000000000	Default format
:PF 5.2	Change format to 5 digits of integers and 2 of fractions
:TPX	Tell Position
00021.00	
PF-5.2	New format Change format to hexadecimal*
:TPX	Tell Position
$00015.00	Report in hex

PR

POSITION RELATIVE

DESCRIPTION:

The PR command sets the incremental distance and direction of the next move. The move is referenced with respect to the current position. If a ? is used, then the current incremental distance is returned (even if it was set by a PA command). Units are in quadrature counts.

ARGUMENTS: PR x,y,z,w

where x,y,z,w are signed integers in the range -2147483648 to 2147483647 decimal

USAGE:

While Moving	No	Default Value 0
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	PR ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

PRx will return the current incremental distance for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"PA" on page 12-120	Position Absolute
"BG" on page 12-21	Begin
"AC" on page 12-6	Acceleration
"DC" on page 12-41	Deceleration
"SP" on page 12-142	Speed
"IP" on page 12-85	Increment Position

EXAMPLES:

INSTRUCTION	INTERPRETATION
:PR 100,200,300,400	On the next move the X-axis will go 100 counts, the Y-axis will go to 200 counts
:BG	forward, Z-axis will go 300 counts and the W-axis will go 400 counts.
:PR ?,?,?	Return relative distances 0000000100,0000000200,0000000300
:PR 500	Set the relative distance for the X axis to 500
:BG	The X-axis will go 500 counts on the next move while the Y-axis will go its previously set relative distance.

QD

DOWNLOAD ARRAY

DESCRIPTION:

The QD command transfers array data from the host computer to the SSC. QD array[],start,end requires that the array name be specified along with the first element of the array and last element of the array. The downloaded array is terminated by a <control>Z, <control>Q, <control>D or \.

ARGUMENTS: QD array[],start,end

where array[] is valid array name start is first element of array (default=0) end is last element of array (default=last element)

USAGE:

While Moving	No	Default Value 0
In a Program	Yes	Default Format Position
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"QU" on page 12-126	Upload array

QU

UPLOAD ARRAY

DESCRIPTION:

The QU command transfers array data from the SSC to a host computer. QU array[],start,end,delim requires that the array name be specified along with the first element of the array and last element of the array. If delim is 1, then the array elements will be separated by a comma. Otherwise, the elements will be separated by a carriage return. The uploaded array will be followed by a <control>Z as an end of text marker.

ARGUMENTS: QU array[],start,end,delim

where array[] is a valid array name, start is the first element of the array (default=0), end is last element of array (default=last element) comma — if it is a 1, then elements are separated by a comma, else a carriage return

USAGE:

While Moving	No	Default Value 0
In a Program	Yes	Default Format Position
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"QD" on page 12-125	Download array

RA

RECORD ARRAY

DESCRIPTION:

The RA command selects one through four arrays for automatic data capture. The selected arrays must be dimensioned by the DM command. The data to be captured is specified by the RD command and time interval by the RC command.

ARGUMENTS: RA n [],m [],o [],p []

where n,m,o and p are dimensioned arrays as defined by DM command. The [] contain nothing.

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"DM" on page 12-46	Dimension Array
"RD" on page 12-130	Record Data
"RC" on page 12-128	Record Interval

EXAMPLES:

INSTRUCTION	INTERPRETATION
#Record	Label
DM POS[100]	Define array
RA POS[]	Specify Record Mode
RD _TPX	Specify data type for record
RC 1	Begin recording at 2 msec intervals
PR 1000;BG	Start motion
EN	End

Hint: The record array mode is useful for recording the real-time motor position during motion. The data is automatically captured in the background and does not interrupt the program sequencer. The record mode can also be used for a teach or learn of a motion path.

RC

RECORD

DESCRIPTION:
The RC command begins recording for the Automatic Record Array Mode (RA). RC 0 stops recording.

ARGUMENTS: RC n,m
where n is an integer 1 thru 8 and specifies 22 samples between records. RC 0 stops recording.

m is optional and specifies the number of records to be recorded. If m is not specified, the DM number will be used. A negative number for m causes circular recording over array addresses 0 to m-1. The address for the array element for the next recording can be interrogated with _RD.

RC? returns status of recording. '1' if recording, '0' if not recording.

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	Yes	RC ?
Used as an Operand	Yes	

OPERAND USAGE:
_RC contains status of recording. '1' if recording, '0' if not recording.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"DM" on page 12-46	Dimension Array
"RD" on page 12-130	Record Data
"RA" on page 12-127	Record Array Mode

EXAMPLES:

INSTRUCTION	INTERPRETATION
#RECORD	Record
DM Torque[1000]	Define Array
RA Torque[]	Specify Record Mode
RD _TTX	Specify Data Type
RC 2	Begin recording and set 4 msec between records
JG 1000;BG	Begin motion
#A;JP #A,_RC=1	Loop until done
MG “DONE RECORDING”	Print message
EN	End program

RD

RECORD DATA

DESCRIPTION:

The RD command specifies the data type to be captured for the Record Array (RA) mode. The command type includes:

_DEx	2nd encoder
_TPx	Position
_TEx	Position error
_SHx	Commanded position
_RLx	Latched position
_TI	Inputs
_OP	Outputs
_TSx	Switches, only 0-4 bits valid
_SCx	Stop code
_TTx	Tell torque

where 'x' is the axis specifier.

ARGUMENTS: RD _TI,_TPX,_SVZ,_TSY

where _TI specifies the data type to be captured. The order is important. Each of the four data types correspond with the array specified in the RA command.

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_RD contains the address for the next array element for recording.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"RA" on page 12-127	Record Array
"RC" on page 12-128	Record Interval
"DM" on page 12-46	Dimension Array

EXAMPLES:

INSTRUCTION	INTERPRETATION
DM ERRORX[50],ERRORY[50]	Define array
RA ERRORX[],ERRORY[ ]	Specify record mode
RD _TEX,_TEYS	Specify data type
RC1	Begin record
JG 1000;BG	Begin motion

RE

RETURN FROM ERROR ROUTINE

DESCRIPTION:

The RE command is used to end a position error handling subroutine or limit switch handling subroutine. The error handling subroutine begins with the #POSERR label. The limit switch handling subroutine begins with the #LIMSWI. An RE at the end of these routines causes a return to the main program. Care should be taken to be sure the error or limit switch conditions no longer occur to avoid re-entering the subroutines. If the program sequencer was waiting for a trippoint to occur, prior to the error interrupt, the trippoint condition is preserved on the return to the program if RE1 is used. RE0 clears the trippoint. To avoid returning to the main program on an interrupt, use the ZS command to zero the subroutine stack.

ARGUMENTS: RE n

where n = 0 or 1

0 clears the interrupted trippoint
1 restores state of trippoint

USAGE:

While Moving	No	Default Value -
In a Program	Yes	Default Format -
Command Line	No	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
#POSERR	Error Subroutine
#LIMSWI	Limit Subroutine

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A;JP #A;EN	Label for main program
#POSERR	Begin Error Handling Subroutine
MG “ERROR”	Print message
SB1	Set output bit 1
RE	Return to main program and clear trippoint

Hint: An applications program must be executing for the #LIMSWI and #POSERR subroutines to function.

RI

RETURN FROM INTERRUPT ROUTINE

DESCRIPTION:

The RI command is used to end the interrupt subroutine beginning with the label #ININT. An RI at the end of this routine causes a return to the main program. The RI command also re-enables input interrupts. If the program sequencer was interrupted while waiting for a trippoint, such as WT, RI1 restores the trippoint on the return to the program. RI0 clears the trippoint. To avoid returning to the main program on an interrupt, use the command ZS to zero the subroutine stack. This turns the jump subroutine into a jump only.

ARGUMENTS: RI n

where n = 0 or 1

0 clears interrupt trippoint
1 restores trippoint

USAGE:

While Moving	No	Default Value -
In a Program	Yes	Default Format -
Command Line	No	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
#ININT	Input interrupt subroutine
"II" on page 12-80	Enable input interrupts

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A;II1;JP #A;EN	Program label
#ININT	Begin interrupt subroutine
MG "INPUT INTERRUPT"	Print Message
SB 1	Set output line 1
RI 1	Return to the main program and restore trippoint

Hint: An applications program must be executing for the #ININT subroutine to function.

RL

REPORT LATCHED POSITION

DESCRIPTION:

The RL command will return the last position captured by the latch. The latch must first be armed by the AL command and then a 0 must occur on the appropriate input. (Input 1,2,3 and 4 for X,Y,Z and W, respectively). The armed state of the latch can be configured using the CN command.

ARGUMENTS: RL XYZW

where X,Y,Z,W are X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_RLx contains the latched position of the specified axis.

RELATED COMMAND:

INSTRUCTION	INTERPRETATION
“AL” on page 12-12	Arm Latch

EXAMPLES:

INSTRUCTION	INTERPRETATION
JG ,5000	Set up to jog the Y-axis
BGY	Begin jog
ALY	Arm the Y latch; assume that after about 2 seconds, input goes low
RLY	Report the latch
10000	

RP

REFERENCE POSITION

DESCRIPTION:
This command returns the commanded reference position of the motor(s).

ARGUMENTS: RP XYZW
where XYZW are X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	RP ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:
_RPx contains the commanded reference position for the specified axis, 'x'.

RELATED COMMAND:

INSTRUCTION	INTERPRETATION
"TP" on page 12-156	

Note: The relationship between RP, TP and TE is that the position error, _TEX, equals the difference between the reference position, _RPX and the actual position, _TPX.

EXAMPLES:
Assume that XYZ and W axes are commanded to be at the positions 200, -10, 0, -110 respectively. The returned units are in quadrature counts.

INSTRUCTION	INTERPRETATION
:PF 7	Position format of 7
:RP	
0000200,-0000010,0000000,-0000110	Return X,Y,Z,W reference positions
RPX	
0000200	Return the X motor reference position
RPY	
-0000010	Return the Y motor reference position
PF-6.0	Change to hex format

RP

```
RP
$0000C8,$FFFFF6,$000000,$FFFF93Return X,Y,Z,W in hex
Position=_RPX                    Assign the variable, Position, the value of RPX
```

The Hint: RP command is useful when operating step motors since it provides the commanded position in steps when operating in stepper mode.

RS

RESET

DESCRIPTION:

The RS command 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.

USAGE:

While Moving	Yes	Default Value 0
In a Program	No	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

<control>R<control>S

MASTER RESET

DESCRIPTION:

The Master Reset command resets the SSC to factory default settings and erases EEPROM. You can also perform a hardware master reset by installing a jumper on the SSC at the location labeled MRST and performing power-up. Remove the jumper after this procedure.

USAGE:

While Moving	Yes	Default Value -
In a Program	No	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

SB

SET BIT

DESCRIPTION:

The SB command sets one of eight bits on the output port.

ARGUMENTS: SB n

where n is an integer in the range 1 to 8 decimal.

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"CB" on page 12-28	Clear Bit

EXAMPLES:

INSTRUCTION	INTERPRETATION
SB 5	Set output line 5
SB 1	Set output line 1

SC

STOP CODE

DESCRIPTION:

The SC command allows the user to determine why a motor stops. The controller responds with the stop code as follows:

CODE	MEANING
0	Motors are running, independent mode
1	Motors stopped at commanded independent position
2	Decelerating or stopped by FWD limit switch or Software limit, FL
3	Decelerating or stopped by REV limit switch or Software limit, BL
4	Decelerating or stopped by Stop Command (ST)
6	Stopped by Abort input
7	Stopped by Abort command (AB)
8	Decelerating or stopped by Off-on-Error (OE1)
9	Stopped after Finding Edge (FE)
10	Stopped after Homing (HM)
50	Contour running
51	Contour Stop
99	MC timeout
100	Motors are running, vector sequence
101	Motors stopped at commanded vector

ARGUMENTS: SC XYZW SC

where XYZW or are the axes

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format 3.0
Command Line	Yes	
Can be Interrogated	Yes	SC ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_SCx contains the value of the stop code for the specified axis, 'x'.

EXAMPLES:

Tom=_SCW Assign the Stop Code of W to variable Tom

SH

SERVO HERE

DESCRIPTION:

The SH commands tells the controller to use the current motor position as the command position and to enable servo control here.

This command can be useful when the position of a motor has been manually adjusted following a motor off (MO) command.

ARGUMENTS: SH XYZW

where XYZW are X,Y,Z,W axes

USAGE:

While Moving	No	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"MO" on page 12-110	Motor-off

EXAMPLES:

INSTRUCTION	INTERPRETATION
SH	Servo X,Y,Z,W motors
SHX	Only servo the X motor, the Y,Z and W motors remain in its previous state.
SHY	Servo the Y motor; leave the X,Z and W motors unchanged
SHZ	Servo the Z motor; leave the X,Y and W motors unchanged
SHW	Servo the W motor; leave the X,Y and Z motors unchanged

Note: The SH command changes the coordinate system. Therefore, all position commands given prior to SH, must be repeated. Otherwise, the controller produces incorrect motion.

SP

SPEED

DESCRIPTION:

This command sets the slew speed of any or all axes for independent moves, or it will return the previously set value. The parameters input will be rounded down to the nearest factor of 2. The units of the parameter are in counts per second.

The maximum value for speed is 2,000,000 cts per second when using stepper motors

ARGUMENTS: SP x,y,z,w SPX=x

where x,y,z, are unsigned numbers in the range 0 to 8,000,000 for servo motors AND

0 TO 2,000,000 for stepper motor operation.

USAGE:

While Moving	Yes	Default Value 25000
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	SP ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_SPx contains the value of the speed for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"AC" on page 12-6	Acceleration
"DC" on page 12-41	Deceleration
"PR" on page 12-124	Position Relation
"PA" on page 12-120	Position Absolute
"BG" on page 12-21	Begin

EXAMPLES:

INSTRUCTION	INTERPRETATION
PR 2000,3000,4000,5000	Specify x,y,z,w parameter
SP 5000,6000,7000,8000	Specify x,y,z,w speeds
BG	Begin motion of all axes
AM Z	After Z motion is complete

Note: For vector moves, use the vector speed command (VS) to change the speed. SP is not a "mode" of motion like JOG (JG).

ST

STOP

DESCRIPTION:

The ST command stops motion on the specified axis. Motors will come to a decelerated stop. If ST is given without an axis specification, program execution will stop in addition to XYZW. XYZW specification will not halt program execution.

ARGUMENTS: ST XYZW

where XYZW are X,Y,Z,W axes. No parameters will stop motion on all axes and stop program.

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"BG" on page 12-21	Begin Motion
"AB" on page 12-5	Abort Motion
"AM" on page 12-13	Wait for motion end
"DC" on page 12-41	Deceleration rate

EXAMPLES:

INSTRUCTION	INTERPRETATION
ST X	Stop X-axis motion
ST S	Stop coordinated sequence
ST XYZW	Stop X,Y,Z,W motion
ST	Stop program and XYZW motion
ST SZW	Stop coordinated XY sequence, and Z and W motion

Hint: Use the after motion complete command, AM, to wait for motion to be stopped.

TB

TELL STATUS BYTE

DESCRIPTION:

The TB command returns status information from the controller. Each bit of the status byte denotes the following condition when the bit is set (high):

BIT	STATUS
Bit 7	Controller addressed
Bit 6	Executing program
Bit 5	Contouring
Bit 4	Executing error or limit switch routine
Bit 3	Input interrupt enabled
Bit 2	Executing input interrupt routine
Bit 1	0 (Reserved)
Bit 0	Echo on

ARGUMENTS: None

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_TB Contains the status byte

EXAMPLES:

INSTRUCTION	INTERPRETATION
TB	Tell status information from the controller
65	Executing program and Echo is on (26 + 20 = 64 + 1 = 65)

TC

TELL ERROR CODE

DESCRIPTION:

The TC command returns a number between 1 and 255. This number is a code that reflects why a command was not accepted by the controller. This command is useful when the controller halts execution of a program at a command or when the response to a command is a question mark. Entering the TC command will provide the user with a code as to the reason. After TC has been read, it is set to zero. TC 1 returns the text message as well as the numeric code.

ARGUMENTS: TC n

n=0 returns code only
n=1 returns code and message

CODE	EXPLANATION	CODE	SECOND ENCODER
1	Unrecognized command	50	Not enough fields
2	Command only valid if from program	51	Question mark not valid
3	Command not valid in program	52	Missing " or string too long
4	Operand error	53	Error { }
5	Input buffer full	54	Question mark part of string
6	Number out of range	55	Missing [or[ ]
7	Command not valid while running	56	Array index invalid or out of range
8	Command not valid when not running	57	Bad function or array
9	Variable error	58	Unrecognized command in a command response (i.e._ GNX)
10	Empty program line or undefined label	59	Mismatched parenthesis
11	Invalid label or line number	60	Download error - line too long or too many lines
12	Subroutine more than 16 deep	61	Duplicate or bad label
13	JG only valid when running in jog mode	62	Too many labels
14	EEPROM check sum error	65	IN command must have a comma
15	EEPROM write error	66	Array space full
16	IP is incorrect sign during position move or IP given during forced deceleration	67	Too many arrays or variables
17	ED, BN, and DL not valid while program running	71	IN only valid in #0
18	Command not valid when contouring	80	Record mode already running
19	Application strand already executing	81	No array or source specified
20	Begin not valid with motor off	82	Undefined array
21	Begin not valid when running	83	Not a valid number
22	Begin not possible due to limit switch	84	Too many elements
24	Begin not valid because no sequence defined	90	Only X Y Z W valid operand

CODE	EXPLANATION	CODE	SECOND ENCODER
25	Variable not given in IN command	96	SM jumper needs to be installed for stepper motor operation
28	S operand not valid	100	Not vlaid when running ECAM
29	Not validuring coordinatted move	101	Improper index into ET (must be 0-256)
30	Sequence segment too short	102	No master axis defined fro ECAM
31	Total move distance in a sequence >2 billion]	103	No master axis modulus greater than 256*EP value
32	More than 511 segments	104	Not valid when performing ECAM
41	Contour record range error	105	EB1 command must be given first
42	Contour data being sent too slowly	118	Controller GL1600 not GL 1800
46	Gear axis both master and follower		

USAGE:

While Moving	Yes	Default Value —-
Command Line	Yes	Default Format 3.0
Not in a Program	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_TC contains the value of the error code.

EXAMPLES:

INSTRUCTION	INTERPRETATION
:GF32	Bad command
:TC	Tell error code
001	Unrecognized command

TD

TELL DUAL ENCODER

DESCRIPTION:

This command returns the current position of the dual (auxiliary) encoder(s). Auxiliary encoders are not available for stepper axes or for the axis where output compare is used.

When operating with stepper motors, the TD command returns the number of counts that have been output by the controller.

ARGUMENTS: TD XYZW

where XYZW are X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	
Used as an Operand	Yes	

OPERAND USAGE:

_TDx contains the dual encoder position for the specified axis, ‘x’.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
“TE” on page 12-149	Dual Encoder

EXAMPLES:

INSTRUCTION	INTERPRETATION
:PF 7	Position format of 7
:TD	Return X,Y,Z,W Dual encoders
0000200,-0000010,0000000,-0000110	
TDX	Return the X motor Dual encoder
0000200	
DUAL=_TDX	Assign the variable, DUAL, the value of TDX

TE

TELL ERROR

DESCRIPTION:

This command returns the current position error of the motor(s). The range of possible error is 2147483647. The Tell Error command is not valid for step motors since they operate open-loop.

ARGUMENTS: TE XYZW

where XYZW are X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value 0
In a Program	Yes	Default Format Position
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"OE" on page 12-116	Off On Error
"ER" on page 12-63	Error Limit
#POSERR	Error Subroutine

EXAMPLES:

INSTRUCTION	INTERPRETATION
TE	Return all position errors
00005,-00002,00000,00006	
TEX	Return the X motor position error
00005	
TEY	Return the Y motor position error
-00002	
Error=_TEX	Sets the variable, Error, with the X-axis position error

Hint: Under normal operating conditions with servo control, the position error should be small. The position error is typically largest during acceleration.

TI

TELL INPUTS

DESCRIPTION:

This command returns the state of the general inputs. TI or TI0 return inputs I1 through I8.

	TI
MSB Bit 7	Input 8
LSB Bit 6	Input 7
LSB Bit 5	Input 6
LSB Bit 4	Input 5
LSB Bit 3	Input 4
LSB Bit 2	Input 3
LSB Bit 1	Input 2
LSB Bit 0	Input 1

ARGUMENTS: TIn

where n equals 0, 1 or 2

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_TIn contains the status byte of the input block specified by 'n'. Note that the operand can be masked to return only specified bit information - see section on Bitwise operations.

EXAMPLES:

INSTRUCTION	INTERPRETATION
TI	
08	Input 4 is high, others low
TI	
00	All inputs low
Input=_TI	Sets the variable, Input, with the TI value
TI	
255	All inputs high

TIME*

TIME OPERAND (KEYWORD)

DESCRIPTION:

*The TIME operand contains the value of the internal free running, real time clock. The returned value represents the number of servo loop updates and is based on the TM command. The default value for the TM command is 1000. With this update rate, the operand TIME will increase by 1 count every update of approximately 1000usec. Note that a value of 1000 for the update rate (TM command) will actually set an update rate of 1/1024 seconds. Thus the value returned by the TIME operand will be off by 2.4% of the actual time.

The clock is reset to 0 with a standard reset or a master reset.

The keyword, TIME, does not require an underscore (_) as does the other SSC operands.

USAGE:

Used as an Operand	Yes	Format TIME

EXAMPLES:

INSTRUCTION	INTERPRETATION
MG TIME	Display the value of the internal clock

TL

TORQUE LIMIT

DESCRIPTION:

The TL command sets the limit on the motor command output. For example, TL of 5 limits the motor command output to 5 volts. Maximum output of the motor command is 9.998 volts.

ARGUMENTS: TL x,y,z,w TLX=x

where x,y,z,w are unsigned numbers in the range 0 to 9.998 volts with resolution of 0.0003 volts

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	Yes	TL ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_TLx contains the value of the torque limit for the specified axis, 'x'.

EXAMPLES:

INSTRUCTION	INTERPRETATION
TL 1,5,9,7.5	Limit X-axis to 1volt Limit Y-axis to 5 volts Limit Z-axis to 9 volts Limit W-axis to 7.5 volts
TL ?,?,?,? 1.0000,5.0000,9.0000, 7.5000	Return limits
TL ? 1.0000	Return X-axis limit

TM

TIME

DESCRIPTION:

The TM command sets the sampling period of the control loop. Changing the sampling period will uncalibrate the speed and acceleration parameters. A negative number turns off the internal clock allowing for an external source to be used as the time base. The units of this command are microsecond (µsec.).

ARGUMENTS: TM n

where n is an integer in the range 250 to 20000 decimal with resolution of 125 microseconds. The minimum sample time for the SSC10 is 250 µsec.; 375 µsec. for the SSC20 and 500 µsec for the SSC30 and SSC40.

Note:

SSC10	250 µsec.
SSC20	375 µsec.
SSC30	500 µsec.
SSC40	500 µsec.

USAGE:

While Moving	Yes	Default Value -1000
In a Program	Yes	Default Format 1.0
Command Line	Yes	
Can be Interrogated	Yes	TM ?
Used as an Operand	Yes	

OPERAND USAGE:

_TM contains the value of the sample time.

EXAMPLES:

INSTRUCTION	INTERPRETATION
TM -1000	Turn off internal clock
TM 2000	Set sample rate to 2000 [usec]
	(This will cut all speeds in half and all acceleration in fourths)
TM 1000	Return to default sample rate

TN

TANGENT

DESCRIPTION:

The TN m,n command describes the tangent axis to the coordinated motion path. m is the scale factor in counts/degree of the tangent axis. n is the absolute position of the tangent axis, at which the resulting angle of the tangent axis equals zero in the coordinated motion plane. The tangent axis is specified with the VM n,m,p command where p is the tangent axis. The tangent function is useful for cutting applications where a cutting tool must remain tangent to the part.

ARGUMENTS: TN m,n

where m is the scale factor in counts/degree, in the range between -127 and 127 with a fractional resolution of 0.004

n is the absolute position at which the tangent angle is zero, in the range between +/-2ᶻ109

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format —
Command Line	Yes	
Can be Interrogated	Yes	TN ?
Used as an Operand	Yes	

OPERAND USAGE:

_TN contains the first position value for the tangent axis. This allows the user to correctly position the tangent axis before the motion begins.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VM" on page 12-170	Vector mode

EXAMPLES:

INSTRUCTION	INTERPRETATION
VM X,Y,Z	Specify coordinated mode for X and Y-axis; Z-axis is tangent to the motion path
TN 100,50	Specify scale factor as 100 counts/degree and 50 counts at which tangent angle is zero
VP 1000,2000	Specify vector position X,Y
VE	End Vector
BGS	Begin coordinated motion with tangent axis

TP

TELL POSITION

DESCRIPTION:
This command returns the current position of the motor(s).

ARGUMENTS: TP XYZW
where XYZW are X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format —
Command Line	Yes	
Can be Interrogated	Yes	
Used as an Operand	Yes	

OPERAND USAGE:
_TPx contains the current position value for the specified axis, 'x'.

EXAMPLES:
Assume the X-axis is at the position 200 (decimal), the Y-axis is at the position -10 (decimal), the Z-axis is at position 0, and the W-axis is at -110 (decimal). The returned parameter units are in quadrature counts.

INSTRUCTION	INTERPRETATION
:PF 7	Position format of 7
:TP	Return X,Y,Z,W positions
0000200,-0000010,0000000,-0000110	
TPX	Return the X motor position
0000200	
TPY	Return the Y motor position
-0000010	
PF-6.0	Change to hex format
TP	Return X,Y,Z,W in hex
$0000C8,$FFFFF6,$000000,$FFFF93	
Position=_TPX	Assign the variable, Position, the value of TPX

TR

TRACE

DESCRIPTION:

The TR command causes each instruction in a program to be sent out the communications port prior to execution. TR1 enables this function and TR0 disables it. The trace command is useful in debugging programs.

ARGUMENTS: TR n

n=0	disable trace function
n=1	enable trace function

USAGE:

While Moving	Yes	Default Value TR0
In a Program	Yes	Default Format —
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

TS

TELL SWITCHES

DESCRIPTION:
TS returns the state of the Home, Forward Limit and Reverse Limit switches, the On/Off state of the motor, the error state of the axis, and the motion status of the axis. The status value is returned as a decimal value corresponding to the following status byte:

BIT	STATUS
Bit 7	Axis in motion if high
Bit 6	Error limit exceeded if high
Bit 5	Motor off if high
Bit 4	Undefined
Bit 3	Forward limit inactive if high
Bit 2	Reverse limit inactive if high
Bit 1	State of home switch
Bit 0	Latch not armed if high

ARGUMENTS: TS XYZW
where XYZW designate X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format 3.0
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:
_TS contains the current status of the switches.

EXAMPLES:

INSTRUCTION	INTERPRETATION
V1=_TSY	Assigns value of TSY to the variable V1
V1=	Interrogate value of variable V1
015 (returned value)	Decimal value corresponding to bit pattern 00001111 Y axis not in motion (bit 7 - value of 0) Y axis error limit not exceeded (bit 6 value of 0) Y axis motor is on (bit 5 value of 0) Y axis forward limit is inactive (bit 3 value of 1) Y axis reverse limit is inactive (bit 2 value of 1) Y axis home switch is high (bit 1 value of 1) Y axis latch is not armed (bit 0 value of 1)

TT

TELL TORQUE

DESCRIPTION:
The TT command reports the value of the analog output signal, which is a number between -9.998 and 9.998 volts.

ARGUMENTS: TT XYZW
where XYZW specify X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format 1.4
Command Line	Yes	
Can be Interrogated	Yes	TT ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:
_TTx contains the value of the torque for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"TL" on page 12-152	Torque Limit

EXAMPLES:

INSTRUCTION	INTERPRETATION
V1=_TTX	Assigns value of TTX to variable, V1
TTX	Report torque on X
-0.2843	Torque is -.2843 volts

TV

TELL VELOCITY

DESCRIPTION:
The TV command returns the actual velocity of the axes in units of quadrature count/s. The value returned includes the sign.

ARGUMENTS: TV XYZW
where XYZW specifies X,Y,Z,W axes

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format 7.0
Command Line	Yes	
Can be Interrogated	Yes	TV ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:
_TVx contains the value of the velocity for the specified axis, 'x'.

EXAMPLES:

INSTRUCTION	INTERPRETATION
VELX=_TVX	Assigns value of X-axis velocity to the variable VELX
TVX	Returns the Y-axis velocity
0003420	

Note: The TV command is computed using a special averaging filter (over approximately .25 sec). Therefore, TV will return average velocity, not instaneous velocity.

TW

TIMEOUT FOR IN-POSITION (MC)

DESCRIPTION:

The TW x,y,z,w command sets the timeout in msec to declare an error if the MC command is active and the motor is not at or beyond the actual position within n msec after the completion of the motion profile. If a timeout occurs, then the MC trippoint will clear and the stopcode will be set to 99. An application program will jump to the special label #MCTIME. The RE command should be used to return from the #MCTIME subroutine.

ARGUMENTS: TW x,y,z,w TWX=x

where x,y,z,w specifies timeout in msec range 0 to 32767 msec -1 disables the timeout

USAGE:

While Moving	Yes	Default Value 32766
In a Program	Yes	Default Format
Command Line	Yes	
Can be Interrogated	Yes	TW ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_TWx contains the timeout in msec for the MC command for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"MC" on page 12-105	Motion Complete trippoint

UL

UPLOAD

DESCRIPTION:

The UL command transfers data from the SSC to a host computer through port 1. Programs are sent without line numbers. The Uploaded program will be followed by a <control>Z or a \ as an end of text marker.

ARGUMENTS: None

USAGE:

While Moving	Yes	Default Value 0
In a Program	No	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

When used as an operand, _UL gives the number of available variables. The total number of variables is 254.

RELATED COMMAND:

INSTRUCTION	INTERPRETATION
"DL" on page 12-44	Download

EXAMPLES:

INSTRUCTION	INTERPRETATION
UL;	Begin upload
#A	Line 0
NO This is an Example	Line 1
NO Program	Line 2
EN	Line 3
<cntrl>Z	Terminator

VA

VECTOR ACCELERATION

DESCRIPTION:

This command sets the acceleration rate of the vector in a coordinated motion sequence. The parameter input will be rounded down to the nearest factor of 1024. The units of the parameter is counts per second squared.

ARGUMENTS: VA n

where n is an unsigned number in the range 1024 to 68,431,360 decimal.

USAGE:

While Moving	Yes	Default Value 262144
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	VA ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_VAx contains the value of the vector acceleration for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VS" on page 12-176	Vector Speed
"VP" on page 12-172	Vector Position
"VE" on page 12-168	End Vector
"CR" on page 12-36	Circle
"VM" on page 12-170	Vector Mode
"BG" on page 12-21	Begin Sequence
"VD" on page 12-166	Vector Deceleration
"VT" on page 12-177	Vector smoothing constant - S-curve

EXAMPLES:

INSTRUCTION	INTERPRETATION
VA 1024	Set vector acceleration to 1024 counts/sec2
VA ?	Return vector acceleration
00001024	
VA 20000	Set vector acceleration
VA ?	
0019456	Return vector acceleration
ACCEL=_VA	Assign variable, ACCEL, the value of VA

Note: This command is not valid in the SSC10.

VD

VECTOR DECELERATION

DESCRIPTION:

This command sets the deceleration rate of the vector in a coordinated motion sequence. The parameter input will be rounded down to the nearest factor of 1024. The units of the parameter is counts per second squared.

ARGUMENTS: VD n

where n is an unsigned number in the range 1024 to 68,431,360 decimal.

USAGE:

While Moving	No	Default Value 262144
In a Program	Yes	Default Format Position Format
Command Line	Yes	
Can be Interrogated	Yes	VD ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_VDx contains the value of the vector deceleration for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VA" on page 12-164	Vector Acceleration
"VS" on page 12-176	Vector Speed
"VP" on page 12-172	Vector Position
"CR" on page 12-36	Circle
"VE" on page 12-168	Vector End
"VM" on page 12-170	Vector Mode
"VT" on page 12-177	Smoothing constant - S-curv

EXAMPLES:

INSTRUCTION	INTERPRETATION
#VECTOR	Vector Program Label
VMXY	Specify plane of motion
VA1000000	Vector Acceleration
VD 5000000	Vector Deceleration
VS 2000	Vector Speed
VP 10000, 20000	Vector Position
VE	End Vector
BGS	Begin Sequence

Note: This command is not valid in the SSC10.

VE

VECTOR SEQUENCE END

DESCRIPTION:

VE is required to specify the end segment of a coordinated move sequence. VE would follow the final VP or CR command in a sequence. VE ? returns the length of the vector in counts. VE is equivalent to the LE command.

ARGUMENTS: None

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	Yes	VE ?
Used as an Operand	Yes	

OPERAND USAGE:

_VE contains the length of the vector in counts.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VM" on page 12-170	Vector Mode
"VS" on page 12-176	Vector Speed
"VA" on page 12-164	Vector Acceleration
"VD" on page 12-166	Vector Deceleration
"CR" on page 12-36	Circle
"VP" on page 12-172	Vector Position
"BG" on page 12-21	Begin Sequence
"CS" on page 12-38	Clear Sequence

EXAMPLES:

INSTRUCTION	INTERPRETATION
VM XY	Vector move in XY
VP 1000,2000	Linear segment
CR 0,90,180	Arc segment
VP 0,0	Linear segment
VE	End sequence
BGS	Begin motion

VF

VARIABLE FORMAT

DESCRIPTION:

The VF command allows the variables and arrays to be formatted for number of digits before and after the decimal point. When displayed, the value m represents the number of digits before the decimal point, and the value n represents the number of digits after the decimal point. When in hexadecimal, the string will be preceded by a $. Hex numbers are displayed as 2's complement with the first bit used to signify the sign.

If a number exceeds the format, the number will be displayed as the maximum possible positive or negative number (i.e. 999.99, -999, $8000 or $7FF).

ARGUMENTS: VF m.n

where m and n are unsigned numbers in the range 0<m<10 and 0<n<4.
A negative m specifies hexadecimal format

USAGE:

While Moving	Yes	Default Value 10.4
In a Program	Yes	Default Format 2.1
Command Line	Yes	
Can be Interrogated	Yes	VF ?
Used as an Operand	Yes	

OPERAND USAGE:

_VF contains the value of the format for variables and arrays.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"PF" on page 12-122	Vector Position

EXAMPLES:

INSTRUCTION	INTERPRETATION
VF 5.3	Sets 5 digits of integers and 3 digits after the decimal point
VF 8.0	Sets 8 digits of integers and no fractions
VF -4.0	Specify hexadecimal format with 4 bytes to the left of the decimal

VM

COORDINATED MOTION MODE

DESCRIPTION:

The VM command specifies the coordinated motion mode and the plane of motion. This mode may be specified for motion on any set of two axes.

The motion is specified by the instructions VP and CR, which specify linear and circular segments. Up to 511 segments may be given before the Begin Sequence (BGS) command. Additional segments may be given during the motion when the SSC buffer frees additional spaces for new segments.

The Vector End (VE) command must be given after the last segment. This tells the controller to decelerate to a stop during the last segment.

It is the responsibility of the user to keep enough motion segments in the buffer to ensure continuous motion. VM ? returns the available spaces for motion segments that can be sent to the buffer.

511 returns means that the buffer is empty and 511 segments may be sent. A zero means that the buffer is full and no additional segments may be sent.

ARGUMENTS: VM nmp

where nm is the plane of motion of any two axes X,Y,Z,W

p is the tangent axis X,Y,Z,W. N turns off tangent.

USAGE:

While Moving	No	Default Value X,Y
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VP" on page 12-172	Vector Position
"VS" on page 12-176	Vector Speed
"VA" on page 12-164	Vector Acceleration
"VD" on page 12-166	Vector Deceleration
"CR" on page 12-36	Circle
"VE" on page 12-168	End Vector Sequence
"BG" on page 12-21	Begin Sequence
"CS" on page 12-38	Clear Sequence
"CS" on page 12-38	_CS - Segment counter
"VT" on page 12-177	Vector smoothing constant — S-curve
"AV" on page 12-20	Vector distance

EXAMPLES:

INSTRUCTION	INTERPRETATION
VM X,Y	Specify coordinated mode for X,Y
CR 500,0,180	Specify arc segment
VP 100,200	Specify linear segment
VE	End vector
BGS	Begin sequence

VP

VECTOR POSITION

DESCRIPTION:

The VP command defines the target coordinates of a straight line segment in a 2 axis motion sequence. The axes are chosen by the VM command. The motion starts with the Begin sequence command. The units are in quadrature counts, and are a function of the vector scale factor. For three or four axis linear interpolation, use the LI command.

ARGUMENTS: VP n,m < N

where n,m are signed integers in the range -2147483648 to 2147483647 The length of each segment must be limited to 8 x 106. N is an unsigned even integer between 0 and 8,000,000

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_VPx contains the absolute coordinate of the axes at the last intersection along the sequence. For example, during the first motion segment, this instruction contains the coordinate at the start of the sequence. The use as an operand is valid in the linear mode, LM, and in the Vector mode, VM.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"CR" on page 12-36	Circle
"VM" on page 12-170	Vector Mode
"VA" on page 12-164	Vector Acceleration
"VD" on page 12-166	Vector Deceleration
"VE" on page 12-168	Vector End
"VS" on page 12-176	Vector Speed
"BG" on page 12-21	Begin Sequence
"VT" on page 12-177	Vector smoothing

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A	Program A
VM X,Y	Specify motion plane
VP 1000,2000	Specify vector position X,Y
CR 1000,0,360	Specify arc
VE	Vector end
VS 2000	Specify vector speed
VA 400000	Specify vector acceleration
BGS	Begin motion sequence
EN	End Program

Note: This command is not valid in the single axis SSC.

Hint: The first vector in a coordinated motion sequence defines the origin for that sequence. All other vectors in the sequence are defined by their endpoints with respect to the start of the move sequence.

VR

VECTOR SPEED RATIO

DESCRIPTION:

The VR r command multiplies the vector speed specifications given by VS or < by the value specified by r. r is between 0 and 10 with a resolution of .0001. VR takes effect immediately and will ratio all the following VS commands and any <n specifications used on VP, CR or LI segments. VR doesn't ratio the accelerations, however, the change in speed is accomplished by accelerating or decelerating at the rate specified by VA and VD.

ARGUMENTS: VR r

where r is between 0 and 10 with a resolution of .0001

USAGE:

While Moving	Yes	Default Value 1
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	Yes	VR ?
Used as an Operand	Yes	

OPERAND USAGE:

_VR contains the vector speed ratio

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VS" on page 12-176	Vector Speed

EXAMPLES:

INSTRUCTION	INTERPRETATION
#A	Vector Program
VMXY	Vector Mode
VP 1000,2000	Vector Position
CR 1000,0,360	Specify Arc
VE	End Sequence
VS 2000	Vector Speed
BGS	Begin Sequence
AMS	After Motion
JP#A	Repeat Move
#SPEED	Speed Override

```
VR@AN[1]*.1                 Read analog input compute ratio
JP#SPEED                    Loop
XQ#A,0; XQ#SPEED,1          Execute task 0 and 1 simultaneously
```

Note: VR is useful for feedrate override, particularly when specifying the speed of individual segments using the operator '<'.

VS

VECTOR SPEED

DESCRIPTION:

The VS command specifies the speed of the vector in a coordinated motion sequence in either the LM or VM modes. The parameter input is rounded down to the nearest factor of 2. The units are counts per second. VS may be changed during motion.

Vector Speed can be calculated by taking the square root of the sum of the squared values of speed for each axis specified for vector or linear interpolated motion.

ARGUMENTS: VS n

where n is an unsigned number in the range 2 to 8000000 decimal

USAGE:

While Moving	Yes	Default Value 8192
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	Yes	VS ?
Used as an Operand	Yes	

OPERAND USAGE:

_VS contains the vector speed.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"VA" on page 12-164	Vector Acceleration
"VP" on page 12-172	Vector Position
"CR" on page 12-36	Circle
"LI" on page 12-98	Linear Interpolation
"VM" on page 12-170	Vector Mode
"BG" on page 12-21	Begin Sequence
"VE" on page 12-168	Vector End

EXAMPLES:

INSTRUCTION	INTERPRETATION
VS 2000	Define vector speed as 2000 counts/sec
VS ?	Return vector speed
002000	

VT

VECTOR TIME CONSTANT - S CURVE

DESCRIPTION:

The VT command filters the acceleration and deceleration functions in vector moves of VM, LM type to produce a smooth velocity profile. The resulting profile, known as S-curve, has continuous acceleration and results in reduced mechanical vibrations. VT sets the bandwidth of the filter, where 1 means no filtering and 0.004 means maximum filtering. Note that the filtering results in longer motion time.

ARGUMENTS: VT n

where n is a positive number in the range between 0.004 and 1.0, with a resolution of 1/256

USAGE:

While Moving	Yes	Default Value 1.0
In a Program	Yes	Default Format 1.4
Command Line	Yes	
Can be Interrogated	Yes	VT ?
Used as an Operand	Yes	

OPERAND USAGE:

_VT contains the vector time constant.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"IT" on page 12-87	Independent Time Constant for smoothing independent moves

EXAMPLES:

INSTRUCTION	INTERPRETATION
VT 0.8	Set vector time constant
VT ?	Return vector time constant
0.8	

WC

WAIT FOR CONTOUR DATA

DESCRIPTION:

The WC command acts as a flag in the Contour Mode. After this command is executed, the controller does not receive any new data until the internal contour data buffer is ready to accept new commands. This command prevents the contour data from overwriting on itself in the contour data buffer.

USAGE:

While Moving	Yes	Default Value 1.0
In a Program	Yes	Default Format 1.4
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"CM" on page 12-34	Contour Mode
"CD" on page 12-30	Contour Data
"DT" on page 12-49	Contour Time

EXAMPLES:

INSTRUCTION	INTERPRETATION
CM XYZW	Specify contour mode
DT 4	Specify time increment for contour
CD 200,350,-150,500	Specify incremental position on X,Y,Z and W
	X-axis moves 200 counts
	Y-axis moves 300 counts
	Z-axis moves -150 counts
	W axis moves 500 counts
WC	Wait for contour data to complete
CD 100,200,300,400	
WC	Wait for contour data to complete
DT 0	Stop contour
CD 0,0,0,0	Exit mode

WT

WAIT

DESCRIPTION:

The WT command is a trippoint used to time events. After this command is executed, the controller will wait for the number of samples specified before executing the next command. If the TM command has not been used to change the sample rate from 1 msec, then the units of the Wait command are milliseconds.

ARGUMENTS: WT n

where n is an integer in the range 0 to 2 Billion decimal

USAGE:

While Moving	Yes	Default Value -
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	No	

EXAMPLES:

Assume that 10 seconds after a move is over a relay must be closed.

INSTRUCTION	INTERPRETATION
#A	Program A
PR 50000	Position relative move
BGX	Begin the move
AMX	After the move is over
WT 10000	Wait 10 seconds
SB 0	Turn on relay
EN	End Program

Hint: To achieve longer wait intervals, just stack multiple WT commands.

XQ

EXECUTE PROGRAM

DESCRIPTION:

The XQ command starts a previously entered program. Execution will start at the label or line number specified. Up to four programs may be executed simultaneously to perform multitasking.

ARGUMENTS: XQ #A,n XQm,n

where A is a program name of up to seven characters
where m is a line number
where n is the thread number (0,1,2 or 3) for multitasking

Note: If the XQ command is given with no parameters, the first program in memory will be executed in thread 0.

USAGE:

While Moving	Yes	Default Value n=0
In a Program	Yes	Default Format -
Command Line	Yes	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_XQn contains the current line number of execution for thread n, and -1 if thread n is not running.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"HX" on page 12-79	Halt execution

EXAMPLES:

INSTRUCTION	INTERPRETATION
XQ #Apple,0	Start execution at label Apple, thread zero
XQ #data,2	Start execution at label data, thread two
XQ 0	Start execution at line 0

Hint: Don't forget to quit the edit mode first before executing a program!

ZR

ZERO

DESCRIPTION:

The ZR command sets the compensating zero in the control loop or returns the previously set value. It fits in the control equation as follows:

$D(z) = GN(z-ZR/z)$

ARGUMENTS: ZR x,y,z,w ZRX=x

where x,y,z,w are unsigned numbers in the range 0 to 1 decimal with a resolution of 1/256

USAGE:

While Moving	Yes	Default Value .9143
In a Program	Yes	Default Format 3.0
Command Line	Yes	
Can be Interrogated	Yes	ZR ?,?,?,?
Used as an Operand	Yes	

OPERAND USAGE:

_ZRx contains the value of the compensating zero for the specified axis, 'x'.

RELATED COMMANDS:

INSTRUCTION	INTERPRETATION
"GN" on page 12-75	Gain
"KD" on page 12-92	Derivative
"KP" on page 12-94	Proportional
"KI" on page 12-93	Integral Gain

EXAMPLES:

INSTRUCTION	INTERPRETATION
ZR .95,.9,.8,.822	Set X-axis zero to 0.95, Y-axis to 0.9, Z-axis to 0.8, W-axis zero to 0.822
ZR ?,?,?,?	Return all zeroes
0.9527,0.8997,0.7994,0.8244	
ZR ?	Return X zero only
0.9527	
ZR ,?	Return Y zero only
0.8997	

ZS

ZERO SUBROUTINE STACK

DESCRIPTION:

The ZS command is only valid in an application program and is used to avoid returning from an interrupt (either input or error). ZS alone returns the stack to its original condition. ZS1 adjusts the stack to eliminate one return. This turns the jump to subroutine into a jump. Do not use RI (Return from Interrupt) when using ZS. To re-enable interrupts, you must use II command again.

The status of the stack can be interrogated with the operand _ZSx - see operand usage below.

ARGUMENTS: ZS n

where 0 returns stack to original condition

1 eliminates one return on stack

USAGE:

While Moving	Yes	Default Value .9143
In a Program	Yes	Default Format 3.0
Command Line	No	
Can be Interrogated	No	
Used as an Operand	Yes	

OPERAND USAGE:

_ZSn contains the stack level for the specified thread where n = 0,1,2 or 3.

Note: n can also be specified using X (thread 0), Y(thread1), Z(thread2) or W(thread3).

EXAMPLES:

INSTRUCTION	INTERPRETATION
II1	Input Interrupt on 1
#A;JP #A;EN	Main program
#ININT	Input Interrupt
MG “INTERRUPT”	Print message
S=_ZS	Interrogate stack
S=	Print stack
ZS	Zero stack
S=_ZS	Interrogate stack
S=	Print stack
EN	End

Notes: