OMS

A Pro-Dex Company

"The Company in Motion"

COMMAND REFERENCE MANUAL

MAX FAMILY

INTELLIGENT MOTION CONTROLLERS

Pro-Dex, Inc. – Oregon Micro Systems
15201 NW Greenbrier Parkway, B-1 Ridgeview
Beaverton, Oregon 97006
1-800-707-8111
http://www.omsmotion.com

COPYRIGHT NOTICE

TRADEMARKS

DISCLAIMER

Pro-Dex, Inc. - Oregon Micro Systems makes no representations or warranties regarding the contents of this document. We reserve the right to revise this document, or make changes to the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or change.

3301-0000001

Rev. H

TABLE OF CONTENTS

1. INTRODUCTION

An extensive command structure is built into the MAX family of intelligent motor controllers. The commands consist of two or three ASCII characters and may be in upper or lower case. Some of the commands expect a numerical operand to follow. These commands are identified with a “#” after the command. The operand must be terminated by a space, carriage return, or semi-colon to indicate the end of the operand list. No terminator is required on the other commands, but it is strongly recommended it be included to improve readability and to insure compatibility with future firmware releases. The operand must immediately follow the command with no space or separation character. The “#” indicates a signed integer input parameter or a signed fixed point number of the format “##.#” when user units are enabled. With user units enabled, distance, velocity, and acceleration parameters may be entered in units such as inches, revolutions, etc.

Most commands are usable in both single-axis and multi-axis modes. Those that require a numeric parameter in single-axis mode require multiple numeric parameters of the same type in multi-axis modes. For example, the MR(Move Relative) command takes a distance as a numeric parameter and is formatted as “MR#;” in single-axis modes. Multi-axis modes have a parameter position for each axis and must be formatted as “MR#,#,#,#,#,#,#,#;” in an 8-axis system. (Note: Use of commas “,” between axes.) Any “#” parameter may be omitted for any axis which is not to be affected by the command and the command may be terminated prematurely with a semicolon. For example, to move only the Y and Z axes, enter the command as “MR,#,#;”.

Some commands that are usable in both single-axis and multi-axis modes do not take a parameter in single-axis mode. These commands require numeric parameters in multi-axis modes, and the parameters indicate whether or not to take action for each axis. If a parameter exists for an axis, then the command affects that axis and if the parameter does not exist for that axis, then the command has no affect on that axis. For example, the single-axis format of IP (Interrupt when in position) is simply "IP" without any parameters of any kind. The multi-axis format of IP is "IPb,b,b,b,b,b,b,b,b;" for 8-axis systems where 'b' represents the parameter for the corresponding axis. Like other multi-axis commands, a 'b' parameter may be omitted if that axis is to remain unchanged and command may be prematurely terminated with a semicolon. Each 'b' position, if used, can be any numeric value. For example, to enable the Y and Z axes limit switches and leave the X and T axes unchanged, send the command "IP,1,100;". The 1 and 100 parameters could be any numeric value whatsoever, and the effect of the command would be the same. For example, the following commands are equivalent:

```
"IP,1,1;"
"IP,0,0;"
"IP,50,99;"
```

For other commands in multi-axis modes, the parameters may have different meanings. See command descriptions for details

Multiple axes can be commanded to start motion simultaneously by entering the AA or AM command. These commands perform a context switch which allows entering motion commands in the format MRx#,y#,z#,t#,u#,v#,r#,s#,w#,k#;. Numbers are entered for each axis which is to be commanded to move. An axis may be skipped by entering a comma with no parameter. The command may be prematurely terminated with a “;” i.e. a move requiring only the X and Y axes would use the command MRx#,y#; followed by the GO command. Each axis programmed to move will start together upon executing the GO command. The MAX can be switched back to the single axis mode by entering the desired single axis command such as AX.

The AM command is provided for complex applications where the host manages multiple motion processes by a multitasking operating system. This mode shares the same instructions as the AA mode, but allows starting a task while some other task involving one or more axes is active. For example, the X and Y axes could be doing linear interpolation while the Z axis is making an unrelated move simultaneously.

Continuous variable velocity contouring can be programmed in AA and AM mode with VP commands for linear interpolation on any combination of axes and VC commands for circular interpolation on any two axes, VV commands control the vector velocity and VA commands control the vector acceleration. VO command or an analog input controls the velocity override or analog input.

1.1. QUEUES

The input characters are placed in a character buffer on input then removed and interpreted. The commands are then placed in separate command and argument queues for each axis. The command queues contain the commands, while the argument queues contain the operands for the commands. For example, in the string “AX; MR100; GO;” the X axis command queues would get “AX; MR; GO;” and the X axis argument queue would get “100”. The argument queue is used by the controller to pass other internal information besides command operands input by user. As they are executed the space is reclaimed allowing the host to pass commands ahead of the moves actually being processed. Most of the commands are placed in the appropriate command and argument queues for execution, while others are executed immediately allowing return of status information in a timely way rather than when encountered in the command stream. This information is provided in a table for each command which shows the queue requirements, if any, and indicates immediate in those cases where the command is not queued. The queue requirements shown in the tables are typical. Depending on the circumstances in which the command is issued, the actual queue requirement may vary slightly. The single axis cases are indicated by the mode reference indicating the appropriate axis. The synchronized mode is indicated by the mode identifier AA/AM. The RQC command may be used to determine the actual queue space available at any time. The queues operate independently allowing each axis to perform separate processes concurrently. The synchronized modes (AA) insert special wait opcodes which allow the axes to be synchronized in this mode. When the commands are nested within loops, the queue space is not reclaimed until after the loop has been executed the programmed number of times. For loops larger than the queue space, the loop may never be completed since it cannot reclaim the queue space and cannot accept the loop terminator. Therefore, loops are effectively limited in size by the size of the command queue. The current axis command queue size for MAX is 2559. Note that if either queue is overrun the MAX will stall and the communication link will be lost.

Some commands are valid only for stepper axes, others for stepper axes with encoder feedback, and still others for servo axes. Most are valid for all three types or some combination of types. A set of symbols to the right of each command identifies which motor types with which each command may be used. The symbols' meanings are as follows:

Stepper motor without an encoder (open loop)

Stepper motor with an encoder (closed loop)

Servo motor

If a command is usable with one of these motor types, the symbol will appear in black. If the command is not usable with a motor type, that motor symbol will be displayed in gray

This command is not usable with servo motors

Indicates an example.

This page is intentionally left blank.

2. COMMAND SUMMARY

The following commands are included in the MAX family of motor controllers. The '#' indicates a signed integer input parameter or a signed fixed point number of the format ##.# when user units are enabled. With User Units enabled, distances, velocity and acceleration parameters may be input in inches, revolutions, etc. Note that numeric parameters must be within the range of a 32-bit signed integer minus one (2147483646 to -2147483646). For fixed point numeric parameters, the value without the decimal point must be within the range of a signed 32-bit integer minus one. Entering parameter values outside the range of a signed 32-bit integer minus one will cause a Command Error.

COMMANDS ALPHABETICALLY			

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

AA	39	C	All axes mode
A*	40	C	Axis selection
AB	41	C	Auxiliary control
AC	42	C	Set acceleration maximum
AD	43	C	Set auxiliary default
#AE	44	Q	Report absolute encoder capability
AEL	45	C	Load auxiliary Encoder Position
AER	46	Q	Report auxiliary Encoder Position
AI	47	C	Report an analog input port value
AJ	48	C	Custom S-curve Definition
AM	51	C	Axes multitasking mode
AO	52	C	Set analog input zero offset
APB	53	C	Archive current parameters in back up archive
APP	54	C	Archive current parameter as power up defaults
#BB	55	C	Select the big ASCII command buffer
BC	56	C	Set backlash compensation
BD	57	C	Set the direction of the general purpose I/O bits
#BF	58	C	Flush the big command buffer
BH	59	C	User defined output bit high
BL	60	C	User defined output bit low
#BN	61	C	Select the normal ASCII command buffer
#BQ	62	C	Report free characters in the big command buffer
BR	63	C	Output reset state
BS	64	C	User defined output bits set
BW	65	C	Wait for input to go low
BX	66	Q	Report user defined I/O bit states in hex
CA	67	C	Clear axis done flag
CC	68	Q	Report update cycle counter value
#CC	69	C	Data Capture Table Clear
#CD	70	C	Data Capture Table Device Assignment

COMMANDS ALPHABETICALLY			

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

#CE	72	C	Data Capture Valid Entries Report
#CF	73	C	Data Capture Free Entries Report
#CI	74	C	Enable Data Capture Imediate
CL	75	C	Closed loop control
#CM	77	C	Enable Data Capture at Go
#CN	78	C	Data Capture Next Entries Report
#CR	80	C	Data Capture Table Report
#CS	82	C	Data Capture Stop
CW	83	C	Clear while
?DA	84	Q	Print a custom acceleration ramp
DAB	85	C	Define custom acceleration ramp breakpoint
DAE	86	C	End custom acceleration ramp definition
DAR	87	C	Begin custom acceleration ramp definition
DBI	88	C	Invert step direction bit
DBN	89	C	Normalize step direction bit
DC	90	C	Set deceleration rate
?DE	91	Q	Report a custom ramp table entry
DL	92	C	Dual Loop Enable/Disable
DLI	93	C	Dual Loop Integral Encoder
DLP	94	C	Dual Loop Proportional Encoder
DLR	95	C	Dual Loop Encoder Ratio
DOV	96	C	Output DAC Voltage
DOZ	98	C	Set DAC Zero Offset Voltage
?DS	100	Q	Report the size of a custom ramp table
EA	101	Q	Encoder status
EC?	102	Q	Report encoder configuration
ECA	103	C	Absolute encoder configuration
ECB	105	C	BiSS Absolute Encoder Configuration
ECI	107	C	Incremental encoder configuration
EDI	108	C	Invert encoder direction
EDN	109	C	Normalize encoder direction
EG	110	C	Engage electronic gearing
EGF	112	C	Turn off electronic gearing
EGM	113	Q	Report electronic gearing map
EH	114	C	Defining encoder home
EHD	115	C	Encoder home delay
ER	116	C	Encoder ratio
#ER	117	C	Report illegal command
ES	118	C	Encoder slip tolerance
ET	120	C	Encoder tracking

COMMANDS ALPHABETICALLY

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

FL	121	C	Flush queue
FT	122	C	Set the servo PID filter type
FV	121	Q	Report instantaneous fractional velocity
GD	124	C	Go and reset done
GN	126	C	Go and notify when done
GO	127	C	Go
GP	128	C	Go to position
GS	129	C	Go and monitor slip trigger
HD	130	C	Set stepper hold deadband
HG	132	C	Set stepper hold gain
HI	133	C	Home index
HM	134	C	Home in positive direction
HR	136	C	Home reverse
HT	137	C	Home true state
HV	138	C	Set stepper hold velocity
IC	139	C	Interrupt clear
ID	140	C	Interrupt when done
II	141	C	Interrupt independent
IN	142	C	Interrupt nearly done
IO	143	C	Set I/O bit direction
IOE	144	C	Define error outputs
IOE,C	146	C	Clear defined error outputs
IOE,R	147	C	Reset error outputs
IOE,X	148	C	Define error outputs (hex)
IOK	149	C	Define kill inputs
IOK,C	150	C	Clear defined kill inputs
IOK,X	151	C	Define kill inputs (hex)
IP	152	C	Interrupt when in position
IS	153	C	Interrupt on slip
JF	154	C	Jog fractional velocities
JG	155	C	Jog
KA	156	C	Set PID acceleration feed-forward gain coeffcient
KB	157	C	Set PID maximum output
KD	158	C	Set PID derivative gain coefficient
KF	159	C	Set servo axis PID friction coefficient
KI	160	C	Set PID integral gain coefficient
KL	161	C	Kill
KM	162	C	Home and kill
KO	163	C	Set PID offset coefficient
KOD	164	C	Set PID output dead band compensation offset

COMMANDS ALPHABETICALLY			

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

KP	165	C	Set PID proportional gain coefficient
KR	166	C	Home reverse and kill
KS	167	C	Kill selected axes
KU	168	C	Set PID integration sum upper limit
KV	169	C	Set PID velocity feedforward
LE	170	C	Loop end
LM	171	C	Limit mode
LO	172	C	Load motor position
LP	173	C	Load motor/encoder position
LPE	174	C	Load encoder position
LS	175	C	Loop start
LT	177	C	Set limit true state
MA	178	C	Move absolute
MD	180	C	Temporary macro define
ML	181	C	Move linear
MM	182	C	Set negative direction
MO	183	C	Move one step
MP	184	C	Set positive direction
MR	185	C	Move relative
MT	187	C	Move to
MX	188	C	Macro execute
#NI	189	C	Network IP address
#NL	190	C	Set Ethernet Communication Mode
#NM?	191	Q	Network Mac address
#NP	192	C	Network Port address
#NU	193	C	Network update interval
PA	194	C	Power automatic
PAA	196	C	Auxiliary automatic
PAV	197	C	Power automatic at velocity
PC?	198	Q	Position Capture Enable Report
PCA	199	C	Position Capture Assignment
PCB	201	C	Position Capture Boundary Assignment
PCC	202	C	Position Capture Table Clear
PCE	203	Q	Position Capture Entry Report
PCF	204	C	Position Capture Off
PCO	205	C	Position Capture On
PCQ	206	Q	Positon Capture Table Query
PCR	207	C	Position Capture Remove Trigger
PCS	208	Q	Position Capture Table Size Report
PE	209	Q	Report encoder positions

COMMANDS ALPHABETICALLY			

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

PM	210	C	Print macro
PP	211	Q	Report motor positions
PS	212	C	Configure current axis
PT	213	C	Preserve a temporary macro
QA	214	Q	Query axis status
QI	215	Q	Query interrupt status
QL	216	Q	Query all limit sensors
RA	217	Q	Report axis status
RC	218	Q	Report instantaneous acceleration
RDB	219	C	Restore default parameter in back-up
RDF	220	C	Restore factory default values
RDP	222	C	Restore power-up default values
RE	223	Q	Report encoder position
RI	224	C	Report axes status
RL	225	Q	Report slip status
RM	226	Q	Reset modulo position
RMC	227	C	Reset modulo position continuous
RP	229	Q	Report position
RQC	230	Q	Report axis command queue
RR	231	Q	Report absolute encoder reference position
#RR	232	Q	Report Raw BiSS Absolute Encoder
RT	233	C	Ramp type control
RU	234	Q	Report position in user units
RV	235	Q	Report instantaneous velocity
SA	236	C	Stop all
SB	237	C	Set baud rate
SD	238	C	Stop and reset done
SE	239	C	Aux power automatic settling time
SEA	240	C	Synchronize encoder position with absolute encoder position
SI	241	C	Stop individual
SK	242	C	Turn on slip kill mode
SME	244	C	Synchronize motor position to encoder position
SMEA	245	C	Synchronize motor and encoder position with absolute encoder position
SO	246	C	Stop by ramping from distance
SR	247	C	Select custom ramp
ST	248	C	Stop
SV	249	C	Servo voltage control

COMMANDS ALPHABETICALLY			

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

SW	250	C	Sync wait for user-defined input high
SXC	252	C	Clear macro links
SXK	253	C	Define kill links
SXL	255	C	Define macro link
SXM	257	C	Enables/Disables stand-alone mode
TL	258	C	Set software overtravel limits
TM	259	C	Timed jog
TP	260	C	Track position
TPS	262	C	Track position stop
TPX	263	C	Track position exit
#UC	264	C	User custom configuration
UF	265	C	User units off
#UR	266	C	Set update rate
UU	267	C	Set user units
VA	268	C	Set vector acceleration
VB	269	C	Set velocity base
VC	270	C	Vector circular interpolation
VCP	271	C	Set vector circular plane
VE	272	C	Vector end
VG	273	C	Vector go (resume)
VH	274	C	Vector halt
VID	275	C	Interrupt when vector done
VIO	276	C	Vector input/output control
VIP	277	C	Interrupt when vector in position
VL	278	C	Set velocity maximum
VO	279	C	Velocity override
VP	281	C	Vector position – linear interpolation
VRC	283	Q	Report vector instantaneous acceleration
VRV	284	Q	Report vector instantaneous velocity
VSD	285	C	Set vector scan distance
VV	286	C	Set vector velocity
VZ	287	C	Set vector stop mode
WA	289	C	Wait for axes to synchronize
#WC	290	C	Watchdog timeout counter clear
WD	291	C	While end of WS loop
WG	292	C	While end of WH loop
WH	293	C	While
WQ	295	C	Wait for queue to empty
#WS	296	Q	Report watchdog timeout counter status
WS	297	C	While user-defined input is high

COMMANDS ALPHABETICALLY			

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

WT	298	C	Wait
WTC	299	C	Wait for update cycle count value
WY	298	Q	Who are you
ZM	302	C	Set piezo motor servo DAC minimum
ZD	301	C	Set piezo motor direction change delay

2.1. SYSTEM STATUS AND CONTROL COMMANDS

2.1.1. IDENTIFICATION COMMANDS

These commands allow the host to request the status of various axis and system parameters and operating modes

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

#ER	117	C	Report illegal command
WY	298	Q	Who are you

2.1.2. POWER-UP DEFAULTS

The MAX can store most user-settable parameters and reload them when the board powers-up or is reset. The following commands can be used to store these parameters, return the board to factory default, reload the stored parameters, and reset the board to reload the stored parameters. The following list of parameters can have their values saved to flash memory: AC, AD, AO, BC, BD/IO, BR, DBI/DBN, ECA/ECI, EDI/EDN, EH, EHD, ER, ES, FT, HD, HG, HT, HV, KA, KB, KD, KF, KI, KO, KOD, KP, KU, KV, LM, LT, #NI, #NP, #NU, PA, PAA, PAV, PS, RT, SB, SC/SS/AJ, SE, SK, SR, SV, SX, TL, #UR, UU, VB, VL, ZD, ZM.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

APB	53	C	Archive current parameters in back up of flash
APP	54	C	Archive current parameters in back up archive
RDB	219	C	Restore default parameter in back-up
RDF	220	C	Restore factory default values
RDP	222	C	Restore power-up default values

2.1.3. QUEUE SELECTION COMMANDS

The following commands set the context to direct the commands which follow to the appropriate axis. They remain in effect until superseded by another command of the same type, specifying a different axis.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

AA	39	C	All axes mode
AM	51	C	Axes multitasking mode
A*	40	C	Axis selection

2.1.4. QUEUE STATUS COMMANDS

Commands sent to the MAX are either queued or immediate. As the type names imply, queued commands are stored in first-in-first-out buffers to be executed in the order they were sent while immediate commands are executed the moment they are received. There are several internal queues in the MAX corresponding to the various axis and command modes and each of these queues has a limited amount of storage space. For example, the X axis command queue can hold 2559 “units”. Each command requires some number of storage units. The amount of storage required is listed in a table with each command. The following commands provide control and monitoring capability for the queues.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

FL	121	C	Flush
RQC	230	Q	Report axis command queue

2.1.5. USER UNIT COMMANDS

The following commands allow specification of move parameters in user defined units. In the UU mode, the controller will automatically convert all move parameters to these units once they have been initialized

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

UF	265	C	User units off
UU	267	C	Set user units

2.1.6. AXIS STATUS COMMANDS

The MAX monitors the various inputs and conditions that can affect motor movement and system status. This information is frequently needed by host applications so that proper motion decisions can be made and appropriate actions taken. The following commands provide this status feedback to the host.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

EA	101	Q	Encoder status
QA	214	Q	Query axis status
QI	215	Q	Query interrupt status
RA	217	Q	Report axis status
RI	224	Q	Report axes status

2.1.7. MACROS

In applications that must perform frequent, repetitive tasks, macros can be used to minimize communication bandwidth consumption and speed up initial task execution. Macros are storage areas in the MAX of which 5 are “temporary”; i.e. not saved at power-off, and 20 are “permanent”; i.e. stored in non-volatile flash RAM.

Macros can be defined, stored, reported back to the host, and executed using the following commands.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

MD	180	C	Temporary macro define
MX	188	C	Macro execute
PM	210	C	Print macro
PT	213	C	Preserve a temporary macro

2.1.8. ASCII BUFFER COMMANDS

Firmware Revision Required: V1.28 and greater

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

#BB	55	C	Select the big ASCII command buffer
#BF	58	C	Flush the big command buffer
#BN	61	C	Select the normal ASCII command buffer
#BQ	62	C	Report free characters in the big command buffer

2.2. I/O CONTROL COMMANDS

2.2.1. AUXILIARY CONTROL COMMANDS

Each axis of the MAX has an associated auxiliary output line. Though this line can be used as a general purpose output, it also has a special purpose: Power-Automatic Mode. In power-automatic mode, the auxiliary line will invert at the beginning of every motion and return to normal at the end. The "normal" state of this line is user-controllable as is the amount of time to delay, allowing the motor to settle, before returning the line to normal at the end of a move. The following commands provide this control as well as feedback regarding the state and function of each auxiliary line.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

AB	41	C	Set auxiliary state
AD	43	C	Set auxiliary default
PA	194	C	Power automatic
PAA	196	C	Auxiliary automatic
PAV	197	C	Power automatic at velocity
SE	239	C	Aux power automatic settling time

2.2.2. GENERAL PURPOSE I/O CONTROL

The MAX has 16 configurable general purpose I/O bits. From the factory they are configured as 8 inputs and 8 outputs. The following commands can be used to set outputs high or low individually or as a group and to read inputs as a group.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

BD	57	C	Set the direction of the general purpose I/O bits
BH	59	C	User-defined output bit high
BL	60	C	User-defined output bit low
BR	63	C	Output reset state
BS	64	C	User-defined output bit set
BX	66	Q	Report user-defined I/O bit states in hex
IO	143	C	I/O bit direction selection
IOE	144	C	Define error outputs
IOK	149	C	Define kill inputs

2.2.3. LIMIT CONTROL COMMANDS

Limit conditions are treated as critical errors in the MAX. When a limit is encountered, the axis involved will cease motion and flush any pending motion commands for that axis. However, since needs vary from application to application, the following commands will allow limit behavior customization to fit almost any system.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

LM	171	C	Limit mode
LT	177	C	Set limit true state
QL	216	Q	Query all limit sensors
TL	258	C	Set software overtravel limits

2.2.4. ANALOG I/O COMMANDS

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

AI	47	C	Report an analog input port value
AO	52	C	Set analog input zero offset
DOV	96	C	Output DAC Voltage
DOZ	98	C	Set DAC Zero Offset Voltage

2.3. SERVO CONTROL COMMANDS

The following commands are valid only for servo axes and should never be executed while the specific axis is in motion.

2.3.1. SERVO VOLTAGE CONTROL COMMANDS

Different servo amplifiers have different requirements for their control inputs. Some simply behave differently despite similar input requirements. To enable the use of a wide range of amplifiers, the MAX will accept the following commands for use in configuring servo outputs.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

KB	157	C	Set PID upper bound limit coefficient
KF	159	C	Set PID friction coefficient
KO	163	C	Offset PID offset coefficient
KOD	164	C	Set PID output dead band compensation offset
SV	249	C	Servo voltage control
ZM	302	C	Set piezo motor minimum DAC value

2.3.2. PID COMMANDS

The MAX uses a PID filter for servo position maintenance. The following commands provide user-control over the filter parameters and feedback of the same.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

CL	75	C	Closed loop control
FT	122	C	Set PID filter type
KA	156	C	Set PID acceleration feed-forward gain coeffcient
KD	158	C	Derivative PID derivative gain coefficient
KI	160	C	Set PID integral gain coefficient
KP	165	C	Set PID proportional gain coefficient
KU	168	C	Set PID integration sum upper limit
KV	169	C	Set PID velocity feedforward
ZD	301	C	Set piezo motor direction change delay

2.4. STEP ENCODER CONTROL COMMANDS

2.4.1. STEP ENCODER CONTROL COMMANDS

Stepper systems, like servo systems, use encoder for position feedback. However, stepper systems do not use PID filters due to operating constraints in the stepper motors themselves. Instead, the MAX uses the following commands to perform position maintenance for stepper axes.

It is important to note that stepper motor position cannot be maintained over the course of a move but rather at the end of the move. Once the axis has initially stopped, the axis will begin moving again to correct for any error encountered during the course of the full move. This process will continue until the encoder position is within the dead band of the motor's target position.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

CL	75	C	Closed loop control
ER	116	C	Encoder ratio
HD	130	C	Hold deadband
HG	132	C	Hold gain
HV	138	C	Set stepper hold velocity

2.4.2. STEP ENCODER SLIP COMMANDS

In applications that require notification when a stepper motor slips beyond a given tolerance, the following commands will be of assistance. These commands do nothing to maintain position. Instead, they tell the MAX to react to a slip condition by notifying the host or ceasing motion.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

ES	118	C	Encoder slip tolerance
IS	153	C	Interrupt on slip
RL	225	Q	Report slip status
SK	242	C	Turn on slip kill mode

2.4.3. ENCODER SLAVE MODES

Encoder tracking modes connect a motor to an axis at a given ratio. For each turn of the encoder, the motor will move proportionately.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

ET	120	C	Encoder tracking

2.5. HOMING COMMANDS

Homing commands detail the commands available for customizing and initiating homing operations.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

EH	114	C	Defining encoder home
EHD	115	C	Encoder home delay
HM	134	C	Home in positive direction
HR	136	C	Home reverse
KM	162	C	Home and kill
KR	166	C	Home reverse and kill

2.6. POSITION COUNTERS

Applications frequently need to know the actual positions of motors and encoders as opposed to the assumed positions the applications keep track of. The following commands are available for retrieving that information as well as forcibly setting those positions. This can be useful for setting "floating zero" positions.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

LO	172	C	Load motor position
LP	173	C	Load motor/encoder position
LPE	174	C	Load encoder position
PE	209	Q	Report encoder positions
PP	211	Q	Report motor positions
RE	223	Q	Report encoder position
RM	226	Q	Reset modulo position
RMC	227	C	Reset modulo position continuous
RP	229	Q	Report position
RU	234	Q	Return position in user units
SEA	240	C	Synchronize encoder position with absolute encoder position
SME	244	C	Synchronize motor position to encoder position
SMEA	245	C	Synchronize motor and encoder position with absolute encoder position

2.7. PROFILE CONTROL COMMANDS

2.7.1. VELOCITY COMMANDS

Part of configuring any system involves defining velocity limits. The commands below provide control over setting these limits and reporting them back to the host.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

FV	121	Q	Report instantaneous fractional velocity
RV	235	Q	Report instantaneous velocity
VB	269	C	Set velocity base
VL	278	C	Set velocity maximum
VV	286	C	Set vector velocity

2.7.2. ACCELERATION COMMANDS

Along with velocity limits, acceleration limits are also critical to most systems. The following commands allow customization of these parameters.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

AC	42	C	Set acceleration maximum
DC	90	C	Set deceleration rate
RC	218	Q	Report instantaneous acceleration
VA	268	C	Vector acceleration

2.7.3. PROFILE COMMANDS

Often, the default linear acceleration profile is not optimum for a given system. To meet the needs of those systems, the MAX has a number of commands that allow partial or even complete customization of the profile. The commands below allow the use of parabolic, cosine, and even custom ramps.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

AJ	48	C	Custom S-curve Definition
RT	233	C	Ramp type control
SR	247	C	Select custom ramp

2.7.4. CUSTOM PROFILE COMMANDS

When linear, parabolic, and cosine acceleration ramps are insufficient, custom ramps can be defined to meet virtually any profiling need. The following commands provide the capability to define a custom ramp. For even more control over the custom ramp's definition, it is recommended to use the S-curve commands. (See AJ and RT)

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

?DA	84	C	Print a custom acceleration ramp
DAB	85	C	Define custom acceleration ramp breakpoint
DAE	86	C	End custom acceleration ramp definition
DAR	87	C	Begin custom acceleration ramp definition
?DE	91	Q	Report a custom ramp table entry
?DS	100	Q	Report the size of a custom ramp table

2.8. MOTION GENERATION COMMANDS

2.8.1. JOGGING COMMANDS

When an application requires a motor to move without stopping or, perhaps, to move until told to stop, the jogging commands that follow will be useful. These commands will start motion on an axis, ramping up to the specified jog velocity, and continue indefinitely, stopping only when told to stop, a limit is reached, or a timeout occurs.

The JG command is very useful when first setting up and testing a system because it generates a continuous stream of step pulse that can easily be tracked.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

JF	154	C	Jog fractional velocities
JG	155	C	Jog
MO	183	C	Move one step
TM	259	C	Timed jog

2.8.2. PROFILE MOVE SPECIFICATION COMMANDS

The following commands define motions on one or more axes that terminate at specified positions. Full profiles are generated that guarantee position achievement either on a per axis basis or in a coordinated fashion.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

MA	178	C	Move absolute
ML	181	C	Move linear
MR	185	C	Move relative
MT	187	C	Move to

2.8.3. MOVE EXECUTION COMMANDS

The following commands initiate moves defined by commands in section Move Specification Commands. A number of different commands are available, tailored to various application needs.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

GD	124	C	Go and reset done
GN	126	C	Go and notify when done
GO	127	C	Go
GS	129	C	Go and monitor slip trigger
VC	270	C	Vector circular interpolation
VP	281	C	Vector position – linear interpolation

2.8.4. MOVE TERMINATION COMMANDS

The following commands allow termination of move sequences in process. When things go wrong or a motion simply needs to be commanded to stop prematurely, the commands below will be useful. These commands can be used to stop motors gracefully or abruptly, depending on the needs of the application.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

KL	161	C	Kill
KS	167	C	Kill selected axes
SA	236	C	Stop all
SD	238	C	Stop and reset done
SI	241	C	Stop individual
SO	246	C	Stop by ramping from distance
ST	248	C	Stop
VE	272	C	Vector end
VH	274	C	Vector halt
VZ	287	C	Set vector stop mode

2.8.5. MOVE COMPLETION NOTIFICATION COMMANDS

These commands allow the synchronization of moves with external events or multiple axis sequences. If an application needs to know when a move or series of commands has completed and fully processed, the following commands can be used to generate notifications. Commands are available to generate a simple notification or perform a somewhat more complex analysis to decide when and how to notify the host.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

CA	67	C	Clear axis done flag
IC	139	C	Interrupt clear
ID	140	C	Interrupt when done
II	141	C	Interrupt independent
IN	142	C	Interrupt nearly done
IP	152	C	Interrupt when in position
VID	275	C	Interrupt when vector done
VIP	277	C	Interrupt when vector in position

2.8.6. VARIABLE VELOCITY CONTOURING COMMANDS

Firmware Revision Required: V1.30 and greater

These commands provide the ability to run one or more axes at a constant, or variable velocity, along a course of many points without stopping at the successive command endpoints. Either linear or two-axis circular interpolation can be used between the endpoints. The velocity can be varied dynamically along the multi-axis path with the velocity override feature. User specified values for vector acceleration and deceleration are used to accelerate or decelerate along the multi-axis path, as required, when the vector velocity is changed, either by velocity override or the vector velocity commands. The controller will accelerate to the vector velocity through multiple points and decelerate through multiple points, if necessary, in order to avoid violating the user specified acceleration and deceleration maximums.

See the Application Note on Variable Velocity Contouring for more details on how to use these commands in contouring applications.

Advantages of these commands are:

The ability to dynamically override the velocity at any time.

The ability to specify a starting and ending velocity and acceleration and deceleration for each linear or circular segment.

The ability to multitask, meaning that two or more distinct multi-axis contours can be running simultaneously, each with it’s own independent velocity, acceleration, velocity override, etc.

The ability to suspend and resume motion. Motion can be suspended, with or without deceleration, at any time during the segment. A mode is provided for stopping at the end of each segment to facilitate the development of contours.

A contour can be unlimited length rather than having to fit into the contour queue as constant velocity contour commands.

Motion can begin with the first command rather than having to completely define the contour before motion begins as is the case with constant velocity contour.

Power auxiliary mode can be used on any or all axes involved to control auxiliary outputs.

Done and in-position interrupts can be provided for any segment.

Stepper hold mode can be active for all axes involved in segment.

Step outputs can be provided at user specified distances along a contour path.

General purpose outputs can be controlled at the segment endpoints.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

VA	268	C	Vector acceleration
VC	270	C	Vector circular interpolation
VCP	271	C	Set vector circular plane
VE	272	C	Vector end
VG	273	C	Vector go (resume)
VH	274	C	Vector halt
VID	275	C	Interrupt when vector done
VIO	276	C	Vector input/output control
VIP	277	C	Interrupt when vector in position
VO	279	C	Vector override
VP	281	C	Vector position – linear interpolation
VRC	283	Q	Report vector instantaneous acceleration
VRV	284	Q	Report vector instantaneous velocity
VSD	285	C	Set vector scan distance
VV	286	C	Set vector velocity
VZ	287	C	Set vector stop mode

2.8.7. VARIABLE VELOCITY POSITIONING COMMANDS

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

GP	128	C	Go to position
TP	260	C	Track position
TPS	262	C	Track position stop
TPX	263	Q	Track position exit

2.9. SYNCHRONIZATION COMMANDS

2.9.1. WAITING COMMANDS

The commands below provide several methods of command and move synchronization. By forcing the MAX to wait a specified amount of time or wait until a set of axes has stopped moving before processing the next command in the queue, the host gains fine-grained control over the motion process.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

BW	65	C	Wait for input to go low
CC	68	Q	Report update cycle count value
SW	250	C	Sync wait for user-defined input high
WA	289	C	Wait for axes to synchronize
WQ	295	C	Wait for queue to empty
WT	298	C	Wait
WTC	299	C	Wait for update cycle count

2.10. LOOPING COMMANDS

2.10.1. LOOPING COMMANDS

Often, applications have need of the ability to repeat a sequence of commands until some event occurs. The commands in this section will allow looping over a given series of commands until a timeout or an I/O event such as a rising or falling edge occurs or simply to loop a specific number of times. It should be noted that only queued commands can be looped; immediate commands will be executed immediately and will not stay in the queue to be part of a loop.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

CW	83	C	Clear while
LE	170	C	Loop end
LS	175	C	Loop start
WD	291	C	While end of WS loop
WG	292	C	While end of WH loop
WH	293	C	While
WS	297	C	While user-defined input is high

2.11. STAND-ALONE COMMANDS

The stand-alone mode allows a MAX Motion Controller to run in a completely independent operation mode when powered by a separate +5 VDC power supply and ±12 VDC (for servos). This mode has several commands that can establish links to macros. When set up properly in this mode the controller can scan for a predefined I/O Input bit, until it changes to the specified state Upon sensing that this condition has been met, it will execute the permanent Macro from Non-volatile flash memory that had been previously associated or linked with this I/O bit and its state.

All of these selections are temporary. They can be made permanent by executing the APP command, which assigns the current parameter values as the Power Up defaults.

Note: The APP command should be used sparingly as it causes a write to the on board Flash Memory and there is a finite amount of times that it can be re-written to (i.e. less than 1,000,000 times, typical).

Application Overview:

The setup of the stand-alone mode is performed through the communication interface by the use of the commands. The user would define the required motion and processes and store them in a macro. Then, with the use of the commands below, the execution of the specific macros would be defined. Once all of the setup is completed the controller would be put in the stand-alone mode (SXM1;) and the execution of the macros controlled by the defined input bits.

There are no queue requirements for these commands.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

SXC	252	C	Clear macro links
SXL	255	C	Define macro link
SXM	257	C	Enables/Disables stand-alone mode

2.12. SPECIAL COMMANDS

2.12.1. S-CURVE ACCELERATION

An S-curve acceleration profile is an alternative to the traditional trapezoidal profile. A trapezoidal profile has a constant rate of acceleration, or zero jerk, on both the acceleration and deceleration sides of the profile. S-curve provides a more controlled and efficient mode of operation for those situations that require optimal speed and smooth starting and stopping of motion. The S-curve acceleration is a means for softening the jerk, controlling shifting materials such as liquids, and to prevent overshoot of high inertial loads.

Functional Description: An S-curve acceleration profile is one that starts with increasing jerk, then transitions to constant jerk, then transitions to decreasing jerk until zero acceleration is reached at the desired velocity. When it is time to start decelerating, the S-curve profile starts increasing negative jerk and then transitions to constant negative jerk, and finally transitions to decreasing negative jerk until zero velocity is reached.

The S-curve profile is not truncated in moves that do not reach full velocity. This happens when the total distance of the move is less than the distance required to accelerate to full velocity plus the distance to decelerate from the full velocity to a stop. In this case, the controller calculates the entire profile and selects the velocity that will preserve selected S-curve profile, by limiting the velocity on short moves to a velocity that is sufficiently small to allow the entire profile to be preserved. Typical S-curve Acceleration Profile (Symmetrical)

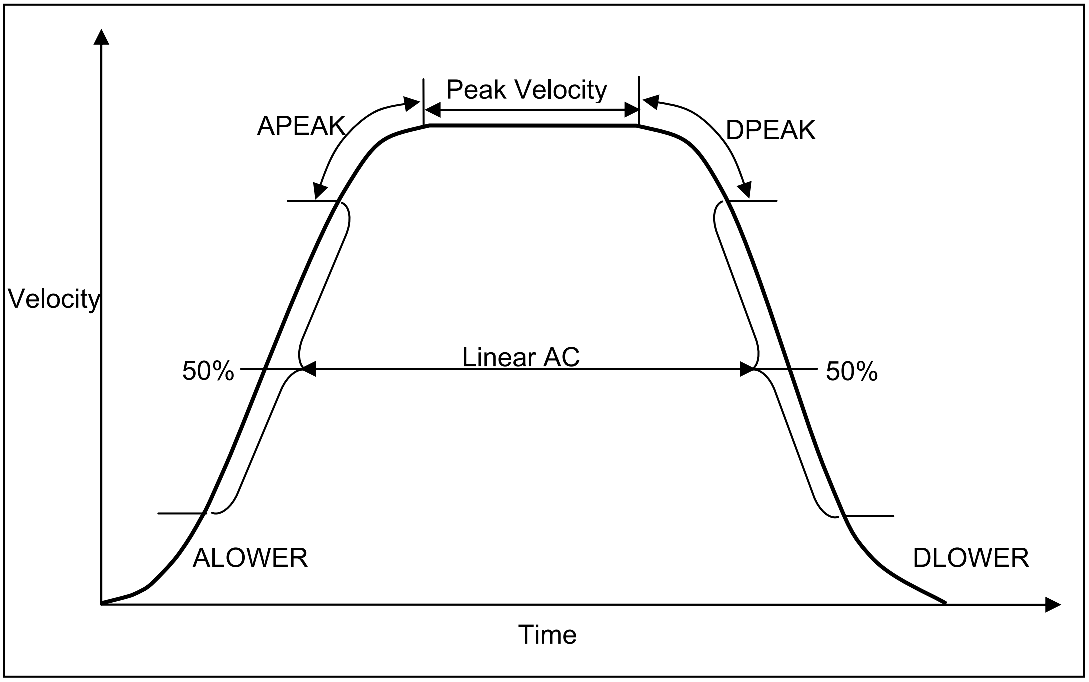

FIGURE 2-1 SYMMETRICAL S-CURVE

2.12.2. BACKLASH COMPENSATION

Some motion systems require adjustments for backlash inherent in gearing systems when the direction of the drive motion is changed. This type of adjustment is particularly necessary when the gear train wears due to use and it may be necessary to adjust the number of steps compensated as the gear train ages.

The MAX family of controllers includes the backlash compensation capability and is implemented for both stepper and servo axes. Note that the implementation is slightly different between the stepper and servos to accommodate the inherent differences.

The BC <step count> command is used to provide backlash compensation for an axis, and it can be saved as a parameter with the APP command. The command BC0; (zero) negates the backlash compensations for the selected axis.

2.12.2.1. STEPPER MOTOR AXIS

If the referenced axis controls a stepper motor, additional motor steps (specified by the number of steps in <step count> are generated), whenever a move command causes the motor to reverse direction to compensate for backlash. The backlash compensation count is output to the motor during the first motor update cycle following a direction change. This permits the compensation to be used with constant velocity contours. Note: it is assumed that the stepper motor is able to develop sufficient torque to accommodate this burst of motor steps. The backlash compensation for stepper motors is limited to a range of from 0 to 50 steps.

Example: For a stepper system, provide for 10 steps Backlash Compensation, i.e., cause the controller to output and additional 10 motor steps whenever a move causes the motor direction to be reversed.

Enter:
```
AX;
BC10;
```

This command line will set the backlash compensation for the X-Axis to 10 steps adding 10 steps on a change of direction for the selected axis.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
BC#;	Single Axis	1	1
-	AA-AM	Not Valid	

2.12.2.2. SERVO MOTOR AXIS

If the referenced axis is a servo motor then offsetting the motor’s position “set point” from the commanded position will compensate for backlash.

If the motor’s “home position” has been set by moving in the positive direction, then the servo set point will automatically include the compensation for backlash in the positive direction. In this case the backlash compensation should be entered as a negative number of steps.

The backlash compensation for a servo motor occurs in the direction of the sign of the number in the BC command, and is limited to a range of from minus 50 to plus 50 encoder counts.

Example: Provide a 10 step backlash compensation for a servo motor on the X-Axis that has been homed in the positive direction.

Enter:

```
(1)
  AX;
  BC-10;
  MA1000;
  GO;
(2)
  AX;
  MA-1000;
  GO;
```

This would cause the controller to go to a position set point of 1000. Note the RP command would report the motor’s commanded position of 1000. The RE command would report the motor’s actual position of 1000.

However, in the second command you would be changing directions so that the motor will go 2010 steps, ending at -1000, and the encoder will read –1010.

Note: The recommended method of using the backlash compensation is to assign backlash compensation for each axis and archive the command, so that the compensation will occur automatically, for each appropriate axis, For example,AX;BC-10;AY;BC5;AZ;BC7; APP;

COMMAND	**PAGE**	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

AJ	48	C	Custom S-curve Definition
BC	56	C	Set backlash compensation
#UR	266	C	Set update rate

Related Commands: APP

2.12.3. ELECTRONIC GEARING COMMANDS

Firmware Revision Required: V1.28 and greater

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

EG	110	C	Electronic gearing
EGF	112	C	Turn off electronic gearing
EGM	113	Q	Report electronic gearing map

2.12.4. CONTROLLER SPECIFIC COMMANDS

MAXv			

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

AEL	45	C	Load auxiliary encoder position
AER	46	Q	Report auxiliary encoder position

MAXnet			

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

#NI	189	C	Network IP address
#NL	190	C	Set Ethernet Communication Mode
#NM?	191	Q	Network Mac address
#NP	192	C	Network Port address
#NU	193	C	Network update interval
SB	237	C	Set baud rate

2.12.5. DUAL LOOP COMMANDS

Firmware Revision Required: V1.42 and greater

A dual loop PID is used in systems where there is mechanical backlash between the motor and the load. To resolve the mechanical backlash, an encoder is used on both the motor and the load. The load encoder can be assigned to either the proportional term, the integral term, or both terms, of the dual loop PID. The derivative term of the dual loop PID always uses the motor encoder. The DLI and/or the DLP commands are used to assign an encoder on another axis as the load encoder for the current axis (motor encoder). There can be a gearing difference between the motor encoder and the load encoder; therefore, the DLR command is used to define the gearing ratio between these encoders. The DL command is used to enable/disable dual loop PID. In order to have the dual loop PID enabled, either the DLI or the DLP command must have assigned an axis other that the current axis (motor encoder) as the load encoder, this load encoder axis must be configured for an encoder, and the current axis must be configured for a servo motor with an encoder. If both the DLI and DLP commands have configured a load axis, then they must specify the same axis. Plus, the current (motor encoder) axis, and the load encoder axis cannot be paired for electronic gearing. When the DL command is executed to enable dual loop, the configuration is checked to be valid before enabling the dual loop feature.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

DL	92	C	Dual Loop Enable/Disable
DLI	93	C	Dual Loop Integral Encoder
DLP	94	C	Dual Loop Proportional Encoder
DLR	95	C	Dual Loop Encoder Ratio

2.12.6. DATA CAPTURE COMMANDS

Firmware Revision Required: V1.42 and greater

The data capture commands control the recording and reporting of motor, encoder, and device data. The maximum size of the data capture table is 65536 entries, where each entry consists of 32 bits of motor position, 32 bits of encoder position, and 16 bits of device data. Data capture table commands allow the user to select which device is to be recorded, the rate the data will be collected at, and control the reporting of the data collected.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

#CC	69	C	Data Capture Table Clear
#CD	70	C	Data Capture Table Device Assignment
#CE	72	C	Data Capture Valid Entries Report
#CF	73	C	Data Capture Free Entries Report
#CI	74	C	Enable Data Capture Immediate
#CM	77	C	Enable Data Capture at Go
#CN	78	C	Data Capture Next Entries Report
#CR	80	C	Data Capture Table Report
#CS	82	C	Data Capture Stop

2.12.7. REAL-TIME POSITION CAPTURE

Firmware Revision Required: V1.42 and greater

The position capture commands control the real-time recording of axis position data and the management of the captured position data. The captured position data includes the axis, the positive edge I/O bits, the negative edge I/O bits, the home switch, encoder home events, and the encoder position of the axis. The position data is captured when the conditions specified for the input bit are met. The capture conditions for the home switch and a general purpose input bit can be a rising/positive edge, a falling/negative edge, or the event can be both the rising/positive and the falling/negative edge so data is captured on any transition of the input bit. The edge event can be further qualified with a position range. The encoder home event bit is treated as a positive edge only bit. The capture data is stored in a table that is implemented as a ring buffer with 800 entries, and is shared by all axes. The ring buffer has input and output table entry indices. The entry input index is updated in the motor update cycle when a capture event occurs and an entry is filled. The output entry index is updated by the PCE command which displays filled table entries. The PCE command advances the output entry pointer by the number of entries displayed, and reduces the number of filled entries by the number of entries displayed. The real-time position capture feature is only available on an axis with incremental encoders.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

PC?	198	Q	Position Capture Enable Report
PCA	199	C	Position Capture Assignment
PCB	201	C	Position Capture Boundary Assignment
PCC	202	C	Position Capture Table Clear
PCE	203	Q	Position Capture Entry Report
PCF	204	C	Position Capture Off
PCO	205	C	Position Capture On
PCQ	206	Q	Positon Capture Table Query
PCR	207	C	Position Capture Remove Trigger
PCS	208	Q	Position Capture Table Size Report

2.13. COMMAND DESCRIPTIONS

AA ALL AXES MODE

The AA command performs a context switch to multi-axis mode. All commands entered after this one will be treated as “all axes” commands which must be formatted for multi-axis use rather than single-axis use. Each command will be executed in the order in which it is received. This is true even if the second command affects axes other than those affected by the first command. For example, if AA mode is entered followed by a move of the X axis and then a move of the Y axis, the Y axis move will not begin until the X axis move has completed.

In firmware V1.30 or higher, if the control is in multi-axis mode, then **A?** will report the correct multi-axis mode, either AA or AM.

Example: Perform an absolute move using the X and Y axes. When this move is complete, perform a relative move using the Y, Z, and T axes.

Enter:

```
AA;
MA12000,14000;
GO;
MR,5000,1500,100000;
GO;
```

Response: None

NOTE: This command changes the axis mode immediately, but has axis queue requirements because it places synchronization entries in all axis queues.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
AA;	Single Axis	1	1
AA;	AA-AM	1	1

Related commands: A*, AM

A* AXIS SELECTION

A*, where * is replaced with one of the following characters:

X – to select the X axis
Y – to select the Y axis
Z – to select the Z axis
T – to select the T axis
U – to select the U axis
V – to select the V axis
R – to select the R axis
S – to select the S axis
W – to select the W axis
K – to select the K axis
? – to report the current axis selected

All subsequent commands are directed to the selected axis. The X axis is the default mode at power up or reset. Commands sent in this mode will be executed in the order in which they are received (subject to queued versus immediate execution) and are expected to be formatted for single-axis use.

Example: Make the X axis step at a rate of 5,000 steps/second.

Enter: AX;
JG5000;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
A*;	Single Axis	Immediate	
A*;	AA-AM	Immediate	

Related commands: AA, AM

AB AUXILIARY CONTROL

Firmware Revision Required: V1.30 and greater

The AB controls the state of the auxiliary output of the selected axes. The AB command may be used to change power level on driver modules so equipped, trigger another board's input or as a user specified output.

AB? can be used to query the current setting.

A parameter must be supplied for the desired axes when used in the AA mode so that the other axes are not affected. Note this command will turn power automatic (PA) mode off.

Example: Turn on the Y axis auxiliary output in the single axis mode.

Enter:
AY;
ABH;

Response: None.

Example: Turn on the X and off the Z axis auxiliary outputs when in the AA command mode. The Y axis is unchanged in this example.

Enter:
AA;
ABH,,L;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
AB**[**H/L/?**]**;	Single Axis	1	0
ABb,b...,b;	AA-AM	1	0

Related commands: BH, BL, BS, PA, PAA

AC SET ACCELERATION MAXIMUM

The AC command sets the maximum acceleration/deceleration to the operand which follows the command. The parameter must be greater than zero (zero is not valid) and less than 8,000,000, and the unit is in steps per second per second. All the following move commands, except contouring commands, for the axis being programmed will accelerate and decelerate at this rate until another AC command is entered. See the APP command to preserve the AC settings as the power-up/reset values. The factory default value is 2,000,000.

AC? can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE: 1 ≤ AC ≤ 8000000

Example: In the single axis mode, set the Y axis acceleration to 200,000 counts per second per second.

Enter:
```
AY;
AC200000;
```

Response: None

Example: In the AA mode, set the acceleration of the X axis to 200,000 and the Z axis to 50,000 and leave the other axes with their previous values.

Enter:
```
AA ;
AC200000,,50000;
```

Response: None

QUEUE REQUIREMENTS				
FORMAT	**MODE**	**Axis Ramp Type**	**COMMAND**	**ARGUMENT**
AC#; or AC#,#...;	Single Axis or AA-AM	Linear (RTL)	3	3
AC#; or AC#,#...;	Single Axis or AA-AM	Custom Ramps (SR)	2	No. of ramp segments + 1
AC#; or AC#,#...;	Single Axis or AA-AM	S-curve (AJ)	5	45

Related commands: DC, RC, VB, VL

AD SET AUXILIARY DEFAULT

The AD command sets the default power up or reset state of the auxiliary output for the current axis. 'H' following the AD command sets the power-up default to high and 'L' sets it to low. This change is stored as a power up parameter in flash automatically and need not be stored via the APP command. Since this command writes to non-volatile memory it should be used only when necessary and not in repeatedly called functions.

AD? Can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

NOTE: This command will also archive all other parameter values as power up defaults.

Example: Set the power up state of the Z axis auxiliary output to high

Enter:
```
AZ;
ADH;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
AD[H/L/?;];	Single Axis	Immediate	
AD[H/L,H/L,…..;];	AA-AM	Immediate	

Related commands: PAA

#AE REPORT ABSOLUTE ENCODER CAPABILITY

The #AE command reports which axes contain absolute encoders and up to how many bits resolution there are on each axis.

Single mode or All Axis mode, the #AE command works the same way in providing a report of absolute encoder information.

Example: #AE reports 12,12,0,0,0,0,0,0 - This represents an 8 axis MAX product with the X and Y axes having absolute encoders for up to 12 bits resolution each. This is also the default that should be seen with no absolute encoders added-on for an 8 axis MAX product.

Example #AE reports 32,32,32,32,0 - This represents a 5 axis MAX product with X, Y, Z, T axes having absolute encoders for up to 32 bits resolution each. The U axis has no absolute encoder.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#AE	Single Axis	Immediate	
#AE	AA-AM	Immediate	

Related commands EC?, ECA, ECI, RR

MAXv only

AEL LOAD AUXILIARY ENCODER POSITION

This command sets the encoder position of the specified auxiliary encoder channel.

NOTE: In order to use this command., general purpose, I/O channels need to be configured as inputs as follows: For auxiliary encoder channel 0, inputs 0, 1, and 2 need to be configured as inputs with the BD or IO command. For auxiliary encoder channel 1, inputs 4, 5, and 6 need to be configured as inputs with the BD or IO command. A command error will result if the appropriate I/O lines are not configured as inputs.

Caution

Configure the appropriate general purpose I/O channels as inputs PRIOR to connecting encoder devices to these signals. Not doing so risks damage to the encoder device and/or the MAXv controller.

AEL#, value;

# specifies the auxiliary encoder channel to load

RANGE: 0 ≥ # ≥ 1

Value specifies the encoder position value to set

RANGE: +/- position range

Example: Set the encoder position to 0 for the auxiliary encoder channel number 1.

Enter: AEL1,0;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
AEL#,#;	Single Axis	Immediate	
AEL#,#;	AA-AM	Immediate	

Related commands: AER, BD, IO

MAXv only

AER REPORT AUXILIARY ENCODER POSITION

This command reports the encoder position of the specified auxiliary encoder channel.

NOTE: In order to use this command., general purpose, I/O channels need to be configured as inputs as follows: For auxiliary encoder channel 0, inputs 0, 1, and 2 need to be configured as inputs with the BD or IO command. For auxiliary encoder channel 1, inputs 4, 5, and 6 need to be configured as inputs with the BD or IO command. A command error will result if the appropriate I/O lines are not configured as inputs.

Caution

Configure the appropriate general purpose I/O channels as inputs PRIOR to connecting encoder devices to these signals. Not doing so risks damage to the encoder device and/or the MAXv controller.

AER#

# specifies the auxiliary encoder channel to report

RANGE: 0 ≥ # ≥ 1

Example: Examine the current encoder position of auxiliary channel 0.

Enter: AER0;

Response: 12345<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
AER#;	Single Axis	Immediate	
AER#;	AA-AM	Immediate	

Related commands: AEL, BD, IO

AI REPORT AN ANALOG INPUT PORT VALUE

The AI command reports the value of one of the specified analog input channel. The command takes a parameter between 0 and 3. The value is displayed in volts.

RANGE: 0 ≤ # ≤ 3 for MAXp

0 ≤ # ≤ 5 for MAXv

0 ≤ # ≤ 1 for MAXnet

Example: Read the value of the first analog input port.

Enter: AI0;

Response: AI0=1.797<LF>

NOTE: This command is not related to an axis and works in AA/AM mode.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
AI#;	Single Axis	Immediate	
AI#;	AA-AM	Immediate	

Related commands: AO

AJ CUSTOM S-CURVE DEFINITION

The AJ command defines the ramp up and ramp down portions of the S-curve. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

AJ? can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

It can accept from 2 to 7 parameters. The command parameters are as follows:

AJ#**[**,ALOWER**][**,APEAK**][**,ARADIUS**][**,DLOWER**][**,DPEAK**][**,DRADIUS**]**
Parameters in brackets **[]** are optional and if omitted will default to the values listed below.

# specifies the S-curve profile number.
Range: 1 ≤ # ≤ number of axes on board

ALOWER specifies the flat portion of the lower half of the ramp up in a percentage.
Range: 0 ≤ ALOWER ≤ 1.0
Default value: 0.0

APEAK specifies the flat portion of the upper half of the ramp up in a percentage.
Range: 0 ≤ APEAK ≤1.0
Default value: alower

ARADIUS specifies the stretch factor in the curved portions of the ramp up.
Range: 1.0 ≤ ARADIUS ≤ 10.0
Default value: 1.0

DLOWER specifies the flat portion of the lower half of the ramp down in a percentage
Range: 0 ≤ DLOWER ≤ 1.0

DPEAK specifies the flat portion of the upper half of the ramp down in a percentage.
Range: 0 ≤ DPEAK ≤ 1.0
Default value: DLOWER for ALOWER if DLOWER is not specified

DRADIUS specifies the stretch factor in the curved portions of the ramp down.
Range: 1.0 ≤ DRADIUS ≤10.0
Default value: 1.0

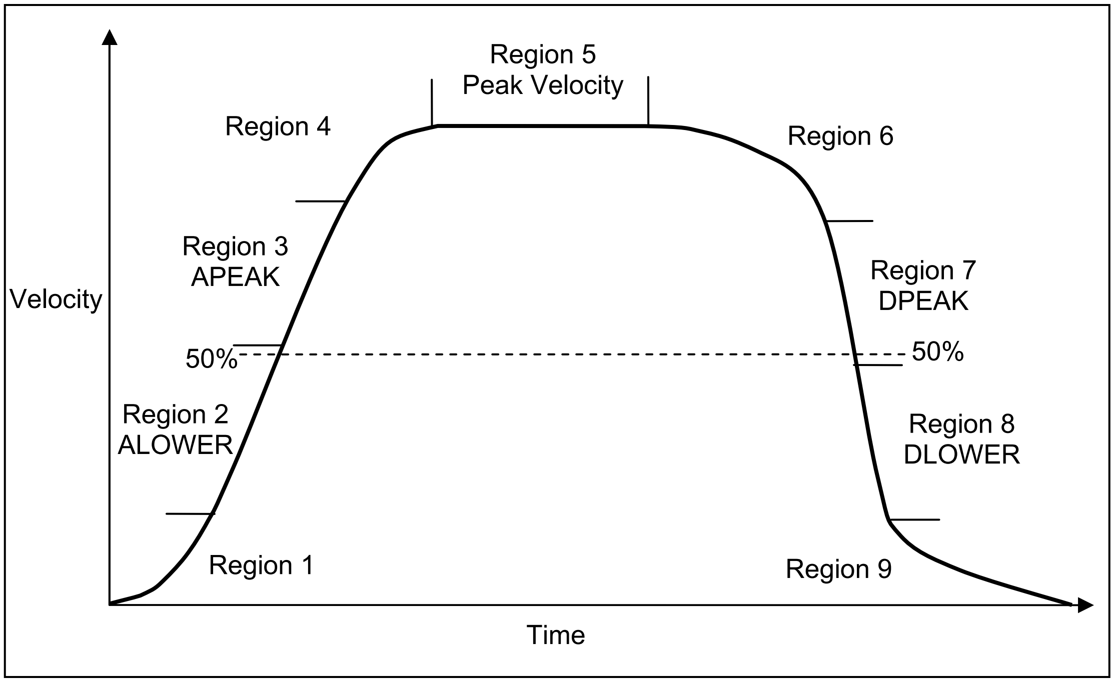

FIGURE 2-2

EXAMPLE OF A CUSTOM S-CURVE PROFILE WITH THE DEFINED REGIONS IDENTIFIED.

Region 1 = 1-ALOWER. It is the percentage of the lower half of the ramp up that S-curved.
Region 2 = ALOWER. It is the percentage of the lower half of the ramp up that is flat.
Region 3 = APEAK. It is the percentage of the upper half of the ramp up that is flat.
Region 4 = 1-APEAK. It is the percentage of the upper half of the ramp up that S-curved.
ARADIUS can “stretch” regions 1 and 4 in the time dimension.
Region 5 = Portion of profile running at maximum velocity.
Region 6 = 1-DPEAK. It is the percentage of the upper half of the ramp down that S-curved.
Region 7= DPEAK. It is the percentage of the upper half of the ramp down that is flat.
Region 8 = DLOWER. It is the percentage of the lower half of the ramp down that is flat.
Region 9 = 1-DLOWER. It is the percentage of the lower half of the ramp down that S-curved.
DRADIUS can “stretch” regions 6 and 9 in the time dimension. If the parameters are not given to define regions 6, 7, 8, and 9, they will be symmetrical with regions 4, 3, 2, and 1 respectively.

Example: Define a custom S-curve profile.

Enter:
```
AX;
AJ1,0.5,0.5,1.0;
*Defines a profile where ramp down is symmetrical with ramp up.
AZ;
AJ2,0.1,0.1,1.0,0.8,0.8,1.0;
*Defines an asymmetrical S-curve profile.
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AJ#,#,#,#,#,#,#,#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: RT

AM AXES MULTITASKING MODE

The AM mode allows several tasks to be managed simultaneously. This command changes the mode of all future commands to multi-tasking mode. In this mode, a task may be performing coordination motion on 2 axes, while a second task is performing unrelated but simultaneous motion on another axis. All commands sent in this mode must be formatted for multi-axis mode rather than single-axis mode.

In firmware V1.30 or higher, if the control is in multi-axis mode, then A? will report the correct multi-axis mode, either AA or AM.

Example: Perform a coordinated relative move on the X and Y axes, while moving the T axis as a separate move at the same time.

Enter:

```
AM;
ML2000,3000;
GO;
MA,,,10000;
GO;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AM;	Single Axis	Immediate	
AM;	AA-AM	Immediate	

Related commands: AA, A*

AO SET ANALOG INPUT ZERO OFFSET

This command sets the zero voltage offset of an analog input channel. The command provides a calibration value so the AI command will report a zero value when zero volts are applied to the analog input. The first parameter specifies the analog input channel. The second parameter specifies the offset in volts. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

RANGE: 0 ≤ parameter 1 ≤ 3 for MAXp

0 ≤ parameter 1 ≤ 5 for MAXv

0 ≤ parameter 1 ≤ 0 for MAXnet

-10.0 ≤ parameter 2 ≤ 10.0

Example Set the zero offset of analog input 2 to -0.05 volts.

Enter: AO2,-0.05;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
AO#,#;	Single Axis	Immediate	
AO#,#;	AA-AM	Immediate	

Related Commands: AI

APB ARCHIVE CURRENT PARAMETERS IN BACK-UP ARCHIVE

The APB command stores the current user parameters into static flash memory so they will be preserved as the back up archive. These parameters can be restored via the RDB command. Also, if an error is found in the power up default parameter set, the parameters in the back up archive will be used during power up. The following list of parameters can have their values saved to flash memory: AC, AD, AO, BC, BD/IO, BR, DBI/DBN, DOZ, ECA/ECI, EDI/EDN, EH, EHD, ER, ES, FT, HD, HG, HT, HV, KA, KB, KD, KF, KI, KO, KOD, KP, KU, KV, LM, LT, #NI, #NP, #NU, PA, PAA, PAV, PS, RT, SB, SC/SS/AJ, SE, SK, SR, SV, SX, TL, #UR, UU, VB, VL, ZD, and ZM.

Note: This command should not be issued when an axis is in motion and it should be used sparingly because the flash memory has a limited number of write cycles.

Example: Store currently used parameters in back up archive of flash memory.

Enter: APB;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
APB;	Single Axis	Immediate	
APB;	AA-AM	Immediate	

Related Commands: APP, RDB, RDF

APP ARCHIVE CURRENT PARAMETERS AS POWER-UP DEFAULTS

The APP command stores the current parameter set as the power-up default set of values. This is done by writing the current parameter set to flash memory. The following list of parameters can have their values saved to flash memory: AC, AD, AO, BC, BD/IO, BR, DBI/DBN, DOZ, ECA/ECI, EDI/EDN, EH, EHD, ER, ES, FT, HD, HG, HT, HV, KA, KB, KD, KF, KI, KO, KOD, KP, KU, KV, LM, LT, #NI, #NP, #NU, PA, PAA, PAV, PS, RT, SB, SC/SS/AJ, SE, SK, SR, SV, SX, TL, #UR, UU, VB, VL, ZD, and ZM.

Note: This command should not be issued when an axis is in motion and it should be used sparingly because the flash memory has a limited number of write cycles.

Example: Save the current parameter set to be the power up default set of values.

Enter: APP;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
APP;	Single Axis	Immediate	
APP;	AA-AM	Immediate	

Related commands: RDB, RDF, RDP

#BB SELECT THE BIG ASCII COMMAND BUFFER

Firmware Revision Required: V1.28 and greater

Allocates memory for an, ASCII, command buffer, in the controller's 512k byte shared memory region and selects it for use by the command interpreter. The command takes the form of:

#BB<*Buffer Size***>;**

where:
<*Buffer Size***>** specifies the size of the command buffer in characters.
Where 1024 ≤ Buffer Size ≤ 500000. The default buffer size is 100000.

Example: Allocate a 500000 character command buffer and select it for use by the controller's command interpreter.

Enter:
```
AX;
#BB500000;
```

Response: None

Example: Allocate a 2000 character command buffer and select it for use by the controller's command interpreter.

Enter:
```
AX;
#BB2000;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#BB<size>;	Single Axis	Immediate	
#BB<size>;	AA-AM	Immediate	

Related commands: #BF, #BN, #BQ

BC SET BACKLASH COMPENSATION

The BC command sets the backlash compensation factor for the currently active axis. This is a numeric value of the number of additional steps output when a direction reversal occurs. The additional steps output are intended to compensate for any mechanical play (known as backlash or lost motion) that occurs when an axis changes directions from plus to minus or minus to plus. See the APP and APB commands for archiving the controller’s parameters in non-volatile FLASH.

BC? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE: 0 ≤ backlash ≤ 1024 Steppers

-1024 ≤ backlash ≤ +1024 Servos

Example: Set backlash compensation factor of axis X to 12 counts

Enter: AX;
BC12;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
BC#;	Single Axis	1	1
BC#,#...;	AA-AM	1	1

Related commands: None

BD SET THE DIRECTION OF THE GENERAL PURPOSE I/O BITS

This command sets the direction of the 16 general purpose I/O bits. Bit directions are encoded into a hexadecimal number. A one in a bit position specifies an output bit. A zero in a bit position specifies an input. Note the bit direction selection may be preserved by using the APP or APB commands to archive the controller's parameters.

BD? Can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE: 0000 ≤ Bit Directions ≤ FFFF for MAXp, MAXv, MAXk and MAXnet (Axis cnt > 5)

00 ≤ Bit Directions ≤ FF for MAXnet (Axis cnt ≤ 5)

Example: To set bits 0 through 3 to outputs and bits 4 through 15 inputs.

Enter: BD000F;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
BD#;	Single Axis	Immediate	
BD#;	AA-AM	Immediate	

Related commands: BL, BS, BX, RB

#BF FLUSH THE BIG COMMAND BUFFER

Firmware Revision Required: V1.28 and greater

Flushes all characters from the big command buffer.

Example: Flush all characters from the big command buffer.

Enter: #BF;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#BF;	Single Axis	Immediate	
#BF;	AA-AM	Immediate	

Related commands: #BB, #BN, #BQ

BH USER DEFINED OUTPUT BIT HIGH

The BH command turns the selected general purpose output off (i.e. logic high). The default state of general purpose outputs is off at power up or reset, but can be changed by the BR command. The selected bit must be configured as an output in order for the command to be valid (see BD command.)

Note: Output bits should not be used as triggers for applying power to any device unless master power is applied separately and after the MAX is fully configured. The states of the outputs are unpredictable during power-up and reset and can toggle several times before settling at a high level.

Range: 0 ≤ bit number ≤ 15 for MAXp and MAXv

0 ≤ bit number ≤ 7 for MAXnet

Example: Set general purpose bits 4 and 5 to high.

Enter:

```
BH4;
BH5;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
BH#;	Single Axis	1	1
BH#;	AA-AM	2	1

Related commands: BD, BL, BS, BR, BX

BL USER DEFINED OUTPUT BIT LOW

The BL command turns the selected general purpose output line on (i.e. logic low). The default states of all output bits at power-up are logic high (off), but can be changed by the BR command. The BS command can be used to set all outputs to a known state at once. The selected bit must be configured as an output in order for the command to be valid (see BD command).

Note: Output bits should not be used as triggers for applying power to any device unless master power is applied separately and after the MAX is fully configured. The states of the outputs are unpredictable during power-up and reset and can toggle several times before settling at a high level.

RANGE: 0 ≤ bit number ≤ 15 for MAXp and MAXv

0 ≤ bit number ≤ 7 for MAXnet

Example: Turn on output bits 4 and 5 after a move. Note that this is only valid for output bits; input bits cannot be modified.

Enter:

```
AX;
MA1000;
GO;
BL4;
BL5;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
BL#;	Single Axis	1	1
BL#;	AA-AM	2	1

Related commands: BD, BH, BR, BS, BX

#BN SELECT THE NORMAL ASCII COMMAND BUFFER

Firmware Revision Required: V1.28 and greater

Select the normal command buffer, in the 64k byte memory region, for use by the controller's ASCII command interpreter.

Example: Tell the controller's command interpreter to begin using normal ASCII command buffer.

Enter: AX;
#BN;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#BN;	Single Axis	Immediate	
#BN;	AA-AM	Immediate	

Related commands: #BB, #BF, #BQ

#BQ REPORT FREE CHARACTERS IN THE BIG COMMAND BUFFER

Firmware Revision Required: V1.28 and greater

Report the number characters that can be placed in the big command buffer before it becomes full.

Example: Find out how many characters can be placed in the buffer before it becomes full (the command buffer was previously set with the command - #BB2000).

Enter: #BQ;

Response: =1999<LF> (*1999 = 2000-1*)

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#BQ;	Single Axis	Immediate	
#BQ;	AA-AM	Immediate	

Related commands: #BB, #BF, #BN

BR OUTPUT RESET STATE

The BR command allows the user to define the state of the general output bits at power-up and reset of a MAX family motion controller. The BR command is valid for single axis, AM and AA modes of operation. It is an immediate command that requires no queue space. It is expected that the BR command is done manually to set up a system. The BR command is automatically archived.

Format: BR*n,m* where the parameter "***n***" is the **Output Bit Number** and the parameter **"*m*"** is a binary one or zero depending on the desired state.

n is the output bit number from **0-15** or a question mark **('?')**. Note that an error is generated if ***n*** is a number not currently defined as an output.

If ***n*** is a '?' character, such as the command **BR?** , then the Output Reset State of all bits is reported in binary. For each bit that is defined as an output a 1 or 0 is displayed indicating the currently defined Output Reset State.
If the bit is defined as an input then '-' is displayed.

m is numeric or a question mark **(?)**
If ***m*** is numeric it represents the desired Output Reset State for that output bit – non-zero parameter defines the Output Reset State as HIGH and a zero parameter defines the Output Reset State as LOW.
If ***m*** is a **'?'** character, the currently defined Output Reset State of bit" ***n*** **"**is reported as 0 or 1. 0 represents LOW and 1 represents HIGH.

Defaults:
The factory default Output Reset State for all outputs is HIGH, which is represented as 1.
If an input is changed to an output, then that output is initialized to the state defined by the Output Reset State.
If an output is changed to an input, the Output Reset State of the former output remains defined, but has no effect.

Example: (assume default bit direction of BD=FF00, bits 8-15 outputs)
BR? - reports the Output Reset State of all output bits, displays 1 for HIGH and 0 for LOW.

BR15,0; – set I/O bit 15 to a LOW output state at reset.

BR11,1; – set I/O bit 11 to a HIGH output state at reset.

BR0,1; - COMMAND ERROR! – I/O bit 0 is not an output.

BR15,? – reports the currently defined Output Reset State. Displays the Reset State of I/O bit 15, where 1 is for HIGH and 0 is for LOW.

	QUEUE REQUIREMENTS		
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
BR#,#;	Single Axis	Immediate	
BR#,#;	AA-AM	Immediate	

Related commands: None

BS USER DEFINED OUTPUT BITS SET

Set all of the output bits to a known state at the same time. This command will affect all output bits, setting their states to the specified bit mask nearly simultaneously. The mask must be in ASCII hex format where the least significant bit (bit 0) is on the right. To set an output low, the corresponding bit in the hex mask must be a 0. A one (1) in any bit position will set the corresponding output high.

Note: Output bits should not be used as triggers for applying power to any device unless master power is applied separately and after the MAX is fully configured. The states of the outputs are unpredictable during power-up and reset and can toggle several times before settling at a high level.

Range: 0000 ≤ Hex number ≤ FFFF for the MAXp and MAXv

00 ≤ Hex number ≤ FF for the MAXnet

Example: Assume I/O bit direction BD = FFFF, all outputs. Set output 0 high, 1-3 low, 4-6 high and 7-15 low (0071 = (hex) 0000000 01110001)

Enter: BS0071;

Response: None.

Note: General purpose I/O lines that are defined as inputs are not affected by the BS command.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
BS#;	Single Axis	1	1
BS#;	AA-AM	3	1

Related commands: BD, BH, BL, BX

BW WAIT FOR INPUT TO GO LOW

The BW command is just like the SW command except that it waits for the input line to reach a TTL low rather than a TTL high. Refer to the SW command for more detail.

RANGE: 0 ≤ Bit Number ≤ 15 for MAXp and MAXv

0 ≤ Bit Number ≤ 7 for MAXnet

Example: See the examples for the SW (see page **250**) command.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
BW#;	Single Axis	1	1
BW#;	AA-AM	3	1

Related commands: SW, WA, WT, WQ

BX REPORT USER DEFINED I/O BIT STATES IN HEX

The BX command returns the states of the general purpose I/O bits in a hex_format. The rightmost character represents the least-significant-nibble (4 bits) and, if the nibble is rewritten as bits, the rightmost bit is the least-significant bit. An input set low will be represented as a binary 0 and a high as binary 1, similarly for an output.

Example: Assuming the default I/O bit direction of BD = FF00, bits 0-7 inputs and bits 8-15 outputs.
User input lines 0 and 2 are high and the remaining 6 inputs are low. User outputs 10 and 11 are high and the remaining 6 outputs are low.
Use the BX commands to verify these states.

Enter: BX;

Response: 0C05<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
BX;	Single Axis	Immediate	
BX;	AA-AM	Immediate	

Related commands: BH, BL, BS

CA CLEAR AXIS DONE FLAG

The CA command operates like the IC command, except it clears the done flag of the addressed axis only. In multi-axis modes, the CA command clears the flags of all selected axes. Unlike the IC command, CA will not clear other error flags in the status register such as slip and limit.

Example: After a multi-axis move, clear the Z axis done flag only.

Enter:

```
AA;
MR1000,2000,3000,4000;
GO;
ID;
AZ;
CA;
```

Response: None.

Example: After a multi-axis move, clear the Y and Z axis done flags only.

Enter:

```
AA;
MR1000,2000,3000,4000;
GO;
ID;
CA,1,1;
```

Response: None.

NOTE: In AA or AM mode, a null value in the argument list specifies the done bit of that axis is not to be cleared.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
CA;	Single Axis	Immediate	
CAb,b...;	AA-AM	Immediate	

Related commands: GD, IC, ID, II, IN, IP

CC REPORT UPDATE CYCLE COUNT

Firmware Revision Required: V1.48 and greater

The CC command displays the current count of update cycles since the controller was last reset. The update cycle counter value starts at 0 on power up, and increments by one on each update cycle. The maximum value for the update cycle is 4,294,967,295, then it rolls over to a value of 0 on the next increment. The rate at which the update cycle count increases is set by the #UR command. The CC command together with the WTC command can be used to synchronize axis actions with the host computer system.

Example: Display the current motor update cycle count.

Enter: CC;

Response: =35769<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
CC;	Single Axis	Immediate	
CC;	AA-AM	Immediate	

Related commands: #UR, WTC

#CC DATA CAPTURE TABLE CLEAR

Firmware Revision Required: V1.42 and greater

The #CC command is used to clear the data in the capture table. The capture table clear command takes the form of:

A<axis id>#CC;

The #CC command clears the data capture table data for the current axis.

Example: Clear the capture table for the Y axis.

Enter: AY;
#CC;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#CC;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: #CD, #CE, #CF, #CI, #CM, #CN, #CR, #CS

#CD DATA CAPTURE TABLE DEVICE ASSIGNMENT

Firmware Revision Required: V1.42 and greater

The #CD command is used to assign the input device to be captured in the table along with the motor position and the encoder position. The data capture table device assignment command takes the form of:

A<axis id>#CD<device number>;

This command pairs the specified <device number> as an input device to the specified <axis id> axis. The device specified by the <device number> will be read and the data stored in the capture table entry associated with the <axis id> axis. Valid input <device number> values are as follows:

0 - No device is paired with this axis. Only motor and encoder position data is recorded.

1 - ADC channel 0. Valid for MAXnet, MAXk & MAXv.

2 - ADC channel 1. Valid for MAXnet, MAXk & MAXv.

3 - ADC channel 2. Valid for MAXnet (6+ axis), MAXk & MAXv.

4 - ADC channel 3. Valid for MAXnet (6+ axis), MAXk & MAXv.

5 - ADC channel 4. Valid for MAXv.

6 - ADC channel 5. Valid for MAXv.

7 - GPIO bit status. Valid for MAXnet, MAXk & MAXv.

8 - Auxiliary Encoder 0. Valid for MAXv.

9 - Auxiliary Encoder 1. Valid for MAXv.

10 - Absolute encoder reading. Valid for MAXnet, MAXk & MAXv.

11 - Limit input bit status X-S. Valid for MAXnet, MAXk & MAXv.

12 - Limit input bit status W-K. Valid for MAXnet & MAXk.

13 – BiSS raw reading. Valid for MAXnet, MAXk & MAXv. (requires Firmware 1.49 and greater)

Example: Store ADC0 readings in the capture table with X axis motor and encoder position data.

Enter:

```
AX;
#CD1;
```

Response: None.

Example: Store general purpose input bits with Y axis motor and encoder position data.

Enter:

```
AY;
#CD7;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#CD#;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: #CC, #CE, #CF, #CI, #CM, #CN, #CR, #CS

#CE DATA CAPTURE VALID ENTRIES REPORT

Firmware Revision Required: V1.42 and greater

The #CE command reports the number of entries in the data capture table that contain valid data. The data capture table valid entries report command takes the form of:

A<axis id>#CE;

Example: Report the number of valid capture table entries for the Z axis.

Enter: AZ;
#CE;

Response: =12480

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#CE;	Single Axis	Immediate	
	AA-AM	Immediate	

Related commands: #CC, #CD, #CF, #CI, #CM, #CN, #CR, #CS

#CF DATA CAPTURE FREE ENTRIES REPORT

Firmware Revision Required: V1.42 and greater

The #CF command is used to report the number of available entries in the data capture data array. The capture table free entries report command takes the form of:

A<axis id>#CF;

This command reports the number of entries in the table that do not have valid data entries.

Example: Report the number of available capture table entries for the Z axis.

Enter: AZ;
#CF;

Response: =53056

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#CF;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: #CC, #CD, #CE, #CI, #CM, #CN, #CR, #CS

#CI ENABLE DATA CAPTURE IMMEDIATE

Firmware Revision Required: V1.42 and greater

The #CI command is used to start recording data into the capture table on the next motor update cycle. The capture table immediate command takes the form of:

A<axis id>#CI<optional recording limit>,<optional downsampling count>;

The #CI command enables input of data to the capture table, for all axes, beginning with the next motor update cycle. Entering an <optional recording limit> value will cause the recording of data to stop after the capture table has been filled with the number of entries specified by the <optional recording limit>. If no value is entered for the <optional recording limit>, then a default of 65536 entries will be recorded into the capture table. If recording limit is specified, then an <optional downsampling count> can also be specified. The downsampling count specifies the number of motor update cycles that will be skipped before the next capture table entry will be recorded. If no value is entered for the downsampling count, then a value of zero is used, causing data to be placed in the capture table on every motor update cycle.

Example: Begin recording data immediately on the next motor update cycle until either 65536 table entries are recorded or the #CS command is issued.

Enter: AX;
#CI;

Response: None.

Example: Begin recording data immediately, collecting data for one motor update cycle, then skipping 9 motor update cycles before collecting the next data sample. This data recording process continues until 1000 data samples have been placed into the capture table.

Enter: #CI1000,9;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#CI#,#;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: #CC, #CD, #CE, #CF, #CM, #CN, #CR, #CS

CL CLOSED-LOOP CONTROL

Firmware Revision Required: V1.30 and greater

The CL command takes a single parameter that can be a ‘1’ to enable closed-loop mode or a ‘0’ to disable closed-loop mode.

For servo axes, the CL1; command closes the loop, enabling the PID. For servo axes, this mode is disabled when the CL0; command is entered, when the servo error becomes too large or a limit is encountered.

CL? can be used to query the current setting.

For stepper with encoder feedback axes, the CL1; command enables position correction after a move and activates the HD, HG and HV commands for stepper axes with encoders. For stepper axes with encoders, this mode will be canceled (as though via an CL0; command) if an JG, JF, HM, HR, SO, SP command is entered, if a limit is encountered or the maximum allowable position error is executed.

Example: (Stepper) The following commands could be used to set up the position correction mode on a stepper axis. This sequence sets up a move velocity of 100,000 steps per second and an acceleration of 500,000 steps per second per second. The position correction velocity is set for 50,000 steps per second, a dead band of 10 steps and correction gain of 2,000. The correction is then enabled. A 200,000 step move is performed, then that position is maintained within the 10 step dead band until commanded to a new position.

Enter:

```
AX;
VL100000;
AC500000;
HV50000;
HD10;
HG2000;
CL1;
MR200000;
GO;
```

Response: None

Example: (Servo) Close PID loop for X and T axes

Enter:

```
AA;
CL1,,1;
```

Response: None

NOTE: In AA or AM mode, a null value in the argument list specifies that encoder feedback is not to be enabled for that axis.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
CL[1/0/?];	Single Axis	1*	0
CLb,b...;	AA-AM	1*	0

* Values in table are for a stepper with encoder axis. For a servo axis the command queue requires 2, and the argument queue requires 1.

Related commands: HD, HG, HV, KA, KB, KD, KF, KI, KP, KV

#CM ENABLE DATA CAPTURE AT GO

Firmware Revision Required: V1.42 and greater

The #CM command is used to start recording data into the capture table on the next motor update cycle following the execution of a go command. The enable data capture at go command takes the form of:

A<axis id>#CM<optional recording limit>,<optional downsampling count>;

Entering an <optional recording limit> value will cause the recording of data to stop after the capture table has been filled with the number of entries specified by the <optional recording limit>. If no value is entered for the <optional recording limit>, then a default of 65536 entries will be recorded into the capture table. If a recording limit is specified, then an <optional downsampling count> can also be specified. The downsampling count specifies the number of motor update cycles that will be skipped before the next capture table entry will be recorded. If no value is entered for the downsampling count, then a value of zero is used, causing data to be placed in the capture table on every motor update cycle.

Example: Begin recording data on the motor update cycle following the next go command. Data is recorded every motor update cycle until either 65536 (default) entries are recorded or the #CS command is issued.

Enter: #CM;

Response: None.

Example: Begin recording data on the motor update cycle following the next go command. Record data every 10th motor update cycle until 1000 capture table entries have been filled.

Enter: #CM1000,9;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#CM#,#;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: #CC, #CD, #CE, #CF, #CI, #CN, #CR, #CS

#CN DATA CAPTURE NEXT ENTRIES REPORT

Firmware Revision Required: V1.42 and greater

The #CN command is used to report the next set of entries in the capture table. The capture table next report command takes the form of:

A<axis id>#CN<optional table entry count>;

This command has an optional argument, which specifies the number of table entries to be reported. The default number of table entries to be reported is 32. This command starts reporting data from the table retrieval index, which is set by the #CR command. If not set by the #CR command, the table retrieval index defaults to a value of zero.

The output format of this report is affected by the dual loop commands (DLI,DLP). If dual loop is disabled, then the format of this report is an ASCII string of upper case hexidecimal values in the following format:

IIII MMMMMMMM EEEEEEEE DDDD ... MMMMMMMM EEEEEEEE DDDD

Where:

IIII is the capture table index

MMMMMMMM is the axis motor position

EEEEEEEE is the axis encoder position

DDDD is the input device that is paired with this axis.

The MMMMMMMM EEEEEEEE DDDD string represents one capture table entry. This string is repeated for the specified number of table entries.

If dual loop is enabled, then the format of this report is an ASCII string of upper case hexidecimal values in the following format:

IIII MMMMMMMM EEEEEEEE LLLLLLLL DDDD ... MMMMMMMM EEEEEEEE LLLLLLLL DDDD

Where:

IIII is the capture table index

MMMMMMMM is the axis motor position

EEEEEEEE is the motor axis encoder position

LLLLLLLL is the load axis encoder position

DDDD is the input device that is paired with this axis.

The MMMMMMMM EEEEEEEE LLLLLLLL DDDD string represents one capture table entry.

This string is repeated for the specified number of table entries.

Example: Reports the next 16 table entries.

Enter: AX;
#CN16;

Response: 00C0 00000000 00000000 16EA ... 00000000 00000000 16EA

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#CN##;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: #CC, #CD, #CE, #CF, #CI, #CM, #CR, #CS

#CR DATA CAPTURE TABLE REPORT

Firmware Revision Required: V1.42 and greater

The #CR command is used to report the capture table data stored at the specified table index, for the number of table entries specifed by the optional table entry count. The capture table report command takes the form of:

A<axis id>#CR<table index>,<optional table entry count>;

The default number of table entries reported is 32. This command initializes the table retrieval index for the #CN command to use. The table retrieval index is set to the specified table index value plus the table entry count value. The output of this report is an ASCII string of hexadecimal values in one of the following formats:

IIII MMMMMMMM EEEEEEEE OOOO ... MMMMMMMM EEEEEEEE OOOO

IIII MMMMMMMM EEEEEEEE LLLLLLLL OOOO ... MMMMMMMM EEEEEEEE LLLLLLLL OOOO

Where:

IIII is the capture table index

MMMMMMMM is the axis motor position

EEEEEEEE is the motor axis encoder position

LLLLLLLL is the load axis encoder position

OOOO is the input device that is paired with this axis.

The MMMMMMMM EEEEEEEE OOOO, and MMMMMMMM EEEEEEEE LLLLLLLL OOOO strings represents one capture table entry. This string is repeated for the specified number of table entries. The LLLLLLLL load encoder string is only output when dual loop functionality is enabled (DLI, DLP).

Example: Report the first 32 capture table entries for the X axis, and set the table retrieval index to 32.

Enter: AX;
#CR0;

Response: 0000 00000000 00000000 16EA ... 00000000 00000000 16EA

Example: Report 128 capture table entries, for the X axis, beginning with table index 64. This sets the table retrieval index to 192.

Enter: AX;
#CR64,128;

Response: 0040 00000000 00000000 16EA ... 00000000 00000000 16EA

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#CR#,#;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: #CC, #CD, #CE, #CF, #CI, #CM, #CN, #CS

#CS DATA CAPTURE STOP

Firmware Revision Required: V1.42 and greater

The #CS command is used to stop the recording of data to the capture table. The capture table stop command takes the form of:

A<axis id>#CS;

The #CS command stops recording data into the capture table immediately, for all axes.

Example: Stop recording data into the capture table for the X axis.

Enter:

```
AX;
#CS;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#CS##;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: #CC, #CD, #CE, #CF, #CI, #CM, #CN, #CR

CW CLEAR WHILE

The CW command breaks the WH loop upon execution of the remaining commands in the loop; i.e. the current execution of the loop is finished. The WH loop is always executed at least one time since the test for the flag is at the bottom.

Example: (see WH command page 293)

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
CW;	Single Axis	Immediate	
CW;	AA-AM	Immediate	

Related commands: WG, WH

?DA PRINT A CUSTOM ACCELERATION RAMP

This command will report the entries of a previously defined custom ramp table.

RANGE: 1 ≤ Ramp Table Numbers ≤ 8

Example: Print out custom ramp table #2

Enter: ?DA2;

Response:

```
DAR2<LF>
DAB0.10000,0.20000<LF>
DAB90000,0.80000<LF>
DAB10000,1.00000<LF>
DAE<LF>
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?DA#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: DAB, DAE, DAR, ?DE, ?DS

DAB DEFINE CUSTOM ACCELERATION RAMP BREAKPOINT

The DAB command sets a breakpoint in a custom ramp table. This is the only command that should be used after DAR and before DAE. Each custom ramp may contain up to 25 breakpoints, each defined by a DAB command.

The DAB command takes two parameters; the first specifies the acceleration level that should be used to achieve the second parameter, velocity level. Both levels are expressed in terms of percentage in decimal format; i.e. 1.00 is 100%. At no time should a DAB command be entered in which the velocity parameter is less than the velocity parameter of the prior DAB. The MAX will not flag this as a command error but the results of such a ramp will be unpredictable. Each DAB command sent should be equal to or greater than the DAB command that preceded it. It is the user's responsibility to make sure this command is used properly.

RANGE:
0.0 ≤ Parameter 1 ≤ 1.0
0.0 ≤ Parameter 2 ≤ 1.0

Example: See the DAR command (page **87**) for a complete example of a custom profile.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
DAB#,#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: ?DA, DAR, DAE, ?DS, ?DE, SR

DAE END CUSTOM ACCELERATION RAMP DEFINITION

The DAE command terminates a custom ramp table definition initiated by the DAR command.

Example: See the DAR command (page **87**) for a complete custom ramp table definition.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
DAE;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: ?DA, DAB, DAR, ?DE, ?DS, SR

DAR BEGIN CUSTOM ACCELERATION RAMP DEFINITION

The DAR command starts the definition of a custom ramp table. A parameter supplied with this command, from 1 to 8, specifies which ramp table to create. If a ramp table by that number has already been defined, it will be overwritten.

Once the DAR command has been issued, only the DAB and DAE commands will be valid. A series of ramp table breakpoints may be entered using the DAB command which define the profile breakpoints for this ramp table. Up to 25 breakpoints may be defined but a smaller number may be used. A ramp table containing no breakpoints is invalid and will result unpredictably if used. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

RANGE: 1 ≤ DAR ≤ 8

Example: Create a ramp table definition resembling a jerk-limited linear profile.

Enter:

```
DAR3;          * Store as table #3
DAB.1,.05;     * Ramp at 10% of AC until 5% of VL
DAB.3,.1;      * Ramp at 30% of AC until 10% of VL
DAB.9,.9;      * Ramp at 90% of AC until 90% of VL
DAB.3,.95;     * Ramp at 30% of AC until 95% of VL
DAB.1,1;       * Ramp at 10% of AC until 100% of VL
DAE;           * End table definition
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
DAR#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: ?DA, DAB, DAE, ?DE, ?DS, SR

DBI INVERT STEP DIRECTION BIT

The DBI command inverts the logic of the direction control output of the addressed axis or axes. By default, the direction output of an axis is a TTL low when traveling in the positive direction and high when traveling negative. After using the DBI command, the direction bit will be high when traveling positive and low when traveling negative. This is useful for inverting the logical direction of a motor when the encoder counts opposite the motor direction. This command can be canceled using the DBN command. To make this the default at power up or reset, use the APP command.

The report the current status of the direction bit enter DB?

Example: Set the direction outputs for axes Z and T to output high when traveling positive and low when traveling negative. Leaves X and Y as they are.

Enter:

```
AZ;
DBI;
AT;
DBI;
Or
AA;
DBI,,1,1;
```

Response: None.

Note: In AA or AM mode a null value in the argument list specifies that the sense of that direction bit is not to be changed.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
DBI;	Single Axis	1	0
DBIb,b...;	AA-AM	1	0

Related commands: DBN

DBN NORMALIZE STEP DIRECTION BIT

The DBN command normalizes the logic of the direction control output of the addressed axis or axes, returning their output logic to default; i.e. TTL low when traveling in the positive direction and high when traveling negative. This command negates the effect of the DBI command. To make this the default at power up or reset when DBI has already been made the default, use the APP command.

The report the current status of the direction bit enter DB?

Example: Set the direction outputs for axes Z and T to default output logic; i.e. output high when traveling positive and low when traveling negative. Leave X and Y as they are.

Enter:

```
AZ;
DBN;
AT;
DBN;
Or
AA;
DBN,,1,1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DBN;	Single Axis	1	0
DBNb,b...;	AA-AM	1	0

Related commands: DBI

DC SET DECELERATION RATE

The DC command sets a deceleration rate overriding the AC parameter when the GP and TP commands is used to initiate a move. At power-up, the deceleration value is equal to the acceleration value.

DC? can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE: 1 ≤ DC ≤ 8000000

Example: Send the Y axis on a 100,000 count move that accelerates at 100,000 counts per second per second up to 50,000 counts per second and decelerates at 20,000 counts per second.

Enter:

```
AY;
AC100000;
DC20000;
VL50000;
GP100000;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
DC#;	Single Axis	1	1
DC#,#...,#;	AA-AM	1	1

Related commands: AC, GP, RC, TP, VA, VB, VL

?DE REPORT A CUSTOM RAMP TABLE ENTRY

The ?DE command will return a specific entry from a specific custom ramp table. The first parameter specifies the table to examine and the second parameter specifies the entry to return from the table.

RANGE:
1 ≤ Parameter1 ≤ 8
1 ≤ Parameter2 ≤ 25

Example: We can't remember what the 23rd breakpoint in table 4 was set to. Use the ?DE command to find out.

Enter: ?DE4,23;

Response: <LF> (there is no 23rd entry in table 4)

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?DE#,#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: ?DA, DAB, DAE, DAR, ?DS

DL DUAL LOOP ENABLE/DISABLE

Firmware Revision Required: V1.42 and greater

The DL command is used to enable or disable the dual loop PID. The dual loop enable/disable command takes the form of:

DL<value>;

Where:

<value> is either a '1' to enable the dual loop PID, or a '0' to disable the dual loop PID.

The default is a '0', which disables the dual loop PID. In order to enable the dual loop PID, the current axis must be configured for a servo motor with an encoder, and either the DLI or DLP command, or both, must be set to an axis other than the current axis, and the axis assigned by the DLI or DLP commands must be configured for an encoder. A '?' character can be entered in place ofthe enable/disable <value> parameter, in order to report whether the dual loop PID is currently enabled or disabled.

Example: Enable the dual loop PID for the X axis.

Enter:
```
AX;
DL1;
```

Example: Query the DL command to determine if the dual loop PID is enabled for the X axis.

Enter:
```
AX;
DL?;
```

Response: =1;
The dual loop PID is enabled for the X axis.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DL#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: DLI, DLP, DLR

DLI DUAL LOOP INTEGRAL ENCODER

Firmware Revision Required: V1.42 and greater

The DLI command is used to specify the axis encoder to be used as the integral term of the PID. The dual loop integral encoder command takes the form of:

DLI<axis ID>;

Where:

<axis ID> specifies the encoder axis to be used as the integral term of a dual loop PID.

If the <axis ID> specifies the current axis, then the motor encoder is used for the integral term of the dual loop PID for the current axis. If the <axis ID> specifies a valid axis, other than the current axis, then the <axis ID> specifies the load encoder to be used for the Integral term of the dual loop PID. A '?' character can be entered as the <axis ID> in order to report the dual loop PID integral term assignment.

Example: Assign the Z axis encoder to the X axis as the load encoder for the integral term of the dual loop PID.

Enter:
```
AX;
DLIZ;
```

Example: Query the DLI command to determine the dual loop PID integral term assignment for the X axis.

Enter:
```
AX;
DLI?;
```

Response: =Z;
The Z axis encoder is used as the dual loop integral term for the X axis servo PID.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DLI*;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: DL, DLP, DLR

DLP DUAL LOOP PROPORTIONAL ENCODER

Firmware Revision Required: V1.42 and greater

The DLP command is used to specify the axis encoder to be used as the proportional term of the PID. The dual loop proportional encoder command takes the form of:

DLP<axis ID>;

Where:

<axis ID> specifies the encoder axis to be used as the proportional term of a dual loop PID.

If the <axis ID> specifies the current axis, then the motor encoder is used for the proportional term of the dual loop PID for the current axis. If the <axis ID> specifies a valid axis, other than the current axis, then the <axis ID> specifies the load encoder to be used for the proportional term of the dual loop PID. A '?' character can be entered as the <axis ID> in order to report the dual loop PID proportional term assignment.

Example: Assign the X axis encoder to the X axis as the encoder for the proportional term of the dual loop PID.

Enter: AX;
DLPX;

Example: Query the DLP command to determine the dual loop PID proportional term assignment for the X axis.

Enter: AX;
DLP?;

Response: =X;
The X axis encoder is used as the proportional term for the X axis servo PID.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DLP*;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: DL, DLI, DLR

DLR DUAL LOOP ENCODER RATIO

Firmware Revision Required: V1.42 and greater

The DLR command is used to specify the ratio of motor encoder counts to load encoder counts in a dual loop PID configuration. This command is used when there is a gearing difference between the motor encoder and the load encoder. The dual loop encoder ratio command takes the form of:

DLR<motor count>,<load count>;

Where:

<motor count> specifies the number of motor encoder counts, and <load count> specifies the number of load encoder counts.

The default is one motor encoder count for every load encoder count. A '?' character can be entered in place of <motor count> and <load count> in order to report the current gearing ratio between the motor and load encoders.

Example: Set a gearing ratio of 3 motor encoder counts for every 2 load encoder counts.

Enter: AX;
DLR3,2;

Example: Query the DLR command to determine the gearing ratio between the motor encoder and the load encoder.

Enter: AX;
DLR?;

Response: =3,2;
The dual loop encoder ratio is 3 motor encoder counts for every 2 load encoder counts.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
DLR#,#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: DL, DLI, DLP

DOV OUTPUT DAC VOLTAGE

This command is used to set a DAC output to the specified voltage. This command takes the form of:

DOV<DAC channel>,<voltage>;

Where:

<DAC channel> identifies which DAC the voltage is to be applied to. There is one DAC per axis, plus the number of auxiliary DACs on the MAX controller. The auxiliary DAC channels are specified as a value of 0 for the first auxiliary DAC, then increasing sequentially for each the subsequent auxiliary DAC. Only axes that are not specified as a servo axis may be specified as the DAC channel for the DOV command. If the DAC channel specifies an axis that is configured as a servo axis, then this command will respond with a command error.

RANGE:

<DAC channel> = 0-1 or X-S depending on number of axes for MAXv

<DAC channel> = X-S depending on number of axes for MAXp

<DAC channel> = 0-1 or X-K depending on number of axes for MAXnet and MAXk

<voltage> is a fixed point numeric value specifying the DAC output voltage.

RANGE:

-10.0 <= <voltage> <= +10.0

Starting with V1.30 firmware (see WY command), a question mark can follow the DAC channel identifier to display the current DOV parameter value for that DAC. The actual value output to the DAC is a summation of the DAC zero offset value, set by the DOZ command for the DAC, and the voltage value specified by this command.

Example 1: Set the second auxiliary DAC output to 5.5 volts.

Enter: DOV1,5.5;

Response: None

Example 2: Report the second auxiliary DAC output voltage.

Enter: DOV1,?;

Response: =5.50<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
DOV#,*Volts*;	Single Axis	Immediate	
DOV#,*Volts*;	AA-AM	Immediate	

Related commands: DOZ

DOZ SET DAC ZERO OFFSET VOLTAGE

This command is used to set the output of a DAC to zero volts. The zero offset value is determined by measuring the output of the DAC, and adjusting this commands voltage parameter until the DAC outputs zero volts. This command takes the form of:

```
DOZ<DAC channel>,<voltage>;
```

Where:

<DAC channel> identifies which DAC the voltage is to be applied to. There is one DAC per axis, plus the number of auxiliary DACs on the MAX controller. The auxiliary DAC channels are specified as a value of 0 for the first auxiliary DAC, then increasing sequentially for each subsequent auxiliary DAC. Only axes that are not specified as a servo axis may be specified as the DAC channel for the DOZ command. If the DAC channel specifies an axis that is configured as a servo axis, then this command will respond with a command error.

RANGE:

<DAC channel> = 0-1 or X-S depending on number of axes for MAXv

<DAC channel> = X-S depending on number of axes for MAXp

<DAC channel> = 0-1 or X-K depending on number of axes for MAXnet and MAXk

<voltage> is a fixed point numeric value specifying the DAC output voltage.

RANGE:

-10.0 <= <voltage> <= +10.0

Starting with V1.30 firmware (see WY command), a question mark can follow the DAC channel identifier to display the current DOZ parameter value for that DAC. The actual value output to the DAC is a summation of the DAC zero offset value, set by this command, and the voltage value specified by the DOV command. In order to set a proper zero offset for a DAC, its DOV parameter value must be zero, and if the DAC is associated with an axis, then it cannot be configured as a servo axis.

See APP command to preserve the DOZ settings as power-up/reset values.

Example 1: Set the zero offset of the T axis DAC to -0.15 volts.

Enter: DOZT,-0.15;

Response: None

Example 2: Report the zero offset of the T axis DAC.

Enter: DOZT,?;

Response: =-0.15<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DOZ#, *Volts*;	Single Axis	Immediate	
DOZ#, *Volts*;	AA-AM	Immediate	

Related commands: APP, DOV

?DS REPORT THE SIZE OF A CUSTOM RAMP TABLE

The ?DS command returns the number of segments in the specified custom ramp table.

RANGE: 1 ≤ ?DS ≤ 8

Example: The 3rd custom ramp should be 17 breakpoints long. Make sure this is true.

Enter: ?DS3;

Response: =17<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?DS#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: ?DA, DAB, DAE, DAR, ?DE

EA ENCODER STATUS

The EA command returns encoder status of the currently addressed axis or axes in the following format:

EA COMMAND RESPONSE DESCRIPTION		
CHAR	**SENT**	**DESCRIPTION**
1	E	Slip detection enabled
	D	Slip detection disabled
2	E	Position maintenance enabled
	D	Position maintenance disabled
3	S	Slip or stall detected (reset by execution of EA command)
	N	No slip or stall detected
4	P	Position Maintenance within deadband
	N	Position not within deadband
5	H	Encoder is at home condition
	N	Encoder is not at home
6	LF	Line feed

Example: Examine the status of the Y axis encoder.

Enter: AY;
EA;

Response: DDNNN<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
EA;	Single Axis	Immediate	
EA;	AA-AM	Immediate	

Related commands: QA, QI, RA, RI

EC? REPORT ENCODER CONFIGURATION

EC? reports the encoder configuration of an axis .

Examples: AX;
EC? *A9, 500000<lf>*
Reports that the X axis encoder is configured as a 9 bit absolute encoder with a 500000 Hz data clock.

Example: AY;
EC?*I<lf>*
Reports that the Y axis encoder is configured as a incremental encoder.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
EC?	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: #AE, ECA, ECB, ECI, RR

ECA ABSOLUTE ENCODER CONFIGURATION

The ECA<encoder bits>,<clock rate>; command configures an axis encoder as an absolute encoder with <encoder bits> bits, in its count register. If the <clock rate> field is present the encoder's data clock rate is set. If the clock rate is not specified, the default rate of 500000 Hz is used, where 8 ≤ encoder bits ≤ maximum authorized bit count. **Note**: A controller's maximum absolute encoder bit counts are configured, on an axis by axis basis, at the time of its manufacture. Any controller that has been configured to utilize absolute encoders will be authorized for 12 bit encoders, on its X & Y axes. An axis may be authorized for a maximum of 32 bits. Valid data clock rates include: {4000000 | 2000000 | 1000000 | 500000 | 250000 | 125000 | 62500 | 31250} Hz. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

On MAXnet (axes V – K) and MAXk the data clock is driven on the encoder index I/O.

For MAXnet (axes X – U) and MAXv, each axis configured to utilize an absolute encoder requires the exclusive use of two general purpose I/O bits for its data clock lines. The ECA command allocates the I/O bit assignments as follows:

Axis	I/O Bits
X	0 & 1
Y	2 & 3
Z	4 & 5
T	6 & 7
U	
V	
R	
S	

The axes without data clock on encoder index will enable clock lines off the corresponding I/O bits as listed above when configured for absolute encoding. Axes that do not have corresponding I/O bits assigned for clocking need to share the clock with another absolute encoder using I/O bits 0 – 7. This means that the sharing absolute encoders must use the same clock frequency and same bits resolution for proper operation.

Example: The following command configures the X axis encoder as a 9 bit absolute encoder with a data clock rate of 250,000Hz.

Enter:

```
AX;
PSE;
ECA9,250000;
```

Example: The following command configures the following: X axis at 12 bits at 125,000Hz, Y axis at 16 bits at 125,000Hz, Z axis at 16 bits at 250,000 Hz, T axis at 24 bits at 500,000Hz, and the U axis at 16 bits at 250,000Hz, Note: Z and U axes can share the same I/O bits since the bits resolution and clock frequency are the same.

Enter:

```
AX;
PSE;
ECA12,125000;
AY;
PSE;
ECA16,125000;
AZ;
PSE;
ECA16,250000;
AT;
PSE;
ECA24,500000;
AU;
PSE;
ECA16,250000;
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
ECA#,#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: #AE, EC?, ECB, ECI, RR

ECB BiSS ABSOLUTE ENCODER CONFIGURATION

Firmware Revision Required: V1.49 and greater

The ECB<encoder bits>,<clock rate>,<status bits>,<CRC poly>,<CRC init>; command configures an axis encoder as a BiSS c-mode absolute encoder with <encoder bits> bits, in its count register. The <clock rate> specifies the data clock frequency BiSS-MA. The <status bits> are the number of tailing data bits (eg Warning, Error) of the encoder that are neglected for position computation. <CRC poly> specifies the up to 8-bit CRC algorithm in hex. <CRC init> specifies the CRC register initial value in hex. If the command is prematurely terminated, the default values are used for the omitted parameters.

RANGE: 8 ≤ <encoder bits> ≤ maximum authorized bit count ≤ 60

<clock rate>: {8,000,000 | 4,000,000 | 2,000,000 | 1,000,000 |

500,000 | 250,000 | 125,000 | 62,500 | 31,215} default: 500,000

0 ≤ <status bits> <= maximum authorized bit count ≤ 60 default: 0

0 ≤ <CRC poly> ≤ 1ff (hex) default: 0

0 ≤ <CRC init> ≤ ff (hex) default: 0

If the BiSS CRC checksum is incorrect, the last correctly read value is used for motion planning.

Note: A controller's maximum absolute encoder bit counts are configured, on an axis by axis basis, at the time of its manufacture. Any controller that has been configured to utilize absolute encoders will be authorized for 12 bit encoders, on its X & Y axes. An axis may be authorized for a maximum of 60 bits. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

On MAXnet (axes V – K) and MAXk the data clock is driven on the encoder index I/O.

For MAXnet (axes X – U) and MAXv, each axis configured to utilize an absolute encoder requires the exclusive use of two general purpose I/O bits for its data clock lines. The ECB command allocates the I/O bit assignments as follows:

Axis	I/O Bits
X	0 & 1
Y	2 & 3
Z	4 & 5
T	6 & 7

The axes without data clock on encoder index will enable clock lines off the corresponding I/O bits as listed above when configured for absolute encoding. Axes that do not have corresponding I/O bits assigned for clocking will not be able to use BiSS encoders.

Example: The following command configures the X axis encoder as a 36 bit BiSS absolute encoder with a data clock rate of 4 MHz. Two additional warning and error bits are appended to the position data. CRC polynomial is x6 + x1 + X0, CRC init value 0.

Enter: AX;

```
PSE;
ECB36,4000000,2,43,0;
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ECB#,#,#,#,#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: #AE, EC?, ECA, ECI, RR, #RR

Note: The design is based on the BiSS C-Mode specification Rev C5.

ECI INCREMENTAL ENCODER CONFIGURATION

The ECI command configures an encoder as an incremental encoder. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

Example: Restore the X axis encoder to its default incremental encoder configuration.

Enter:

```
AX;
ECI:
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ECI	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: EC?, ECA, RR

EDI INVERT ENCODER DIRECTION

The EDI command inverts the sign of the encoder counts for the current axis. After receiving this command, the MAX will produce a negative encoder value for positive encoder motion and a positive encoder value for negative encoder motion. To cancel this command, issue an EDN commands. To make inverted encoder direction at power up or reset, use the APP command.

ED? can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

Example: The Y axis encoder is counting opposite the expected direction. Setup the Y axis to produce a negative encoder value when the encoder counts positive to correct the problem.

Enter:

```
AY;
EDI;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
EDI;	Single Axis	Immediate	
EDIb,b...;	AA-AM	Immediate	

Related commands: DBI, DBN, EDN

EDN NORMALIZE ENCODER DIRECTION

The EDN command normalizes the sign of the encoder counts for the current axis, negating the effects of the EDI command. After receiving this command, the MAX will produce positive encoder counts for positive motion and a negative encoder count for negative motion, the default behavior. To make this the default behavior (if it has been changed via EDI/APP), use the APP command. (EDN is the factory default setting.)

ED? can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

Example: The Y axis encoder was rewired and now counts in the correct direction. Return the Y axis encoder sign to normal; i.e. return positive encoder counts for positive motion.

Enter: AY;
EDN;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
EDN;	Single Axis	Immediate	
EDNb,b...;	AA-AM	Immediate	

Related commands: DBI, DBN, EDI

EG ENGAGE ELECTRONIC GEARING

Firmware Revision Required: V1.28 and greater

The electronic gearing engagement commands take the form of:

EG<Master Axis><Gear ratio>;

Where:

<Master Axis> specifies the axis that the current axis is to follow.

Note: The current axis cannot be geared to itself or to an axis that has been geared to another master axis.

<Gear ratio> specifies a gearing ratio as the ratio of geared axis motion to master axis motion. The ratio can be entered in the following forms:

EG<*Master Axis*>; sets up gearing with a default gear ratio of 1 to 1.

For example: EGX; gears the current axis to the X axis using a ratio of 1 to1.

EG<*Master Axis*><Geared axis count>; sets up gearing with a ratio of <Geared axis count>:1.

For example: EGX2; gears the current axis to the X axis with a ratio of 2 to 1.

EG<*Master Axis*><Geared axis count>, <Master axis count>; sets up gearing with a ratio of Geared axis count>: <Master axis count>.

For example: EGX1,2; gears the current axis to the X axis with a gear ratio of 1 to 2.

Negative ratio values indicate that the current axis is to track the master axis in the opposite direction. The gear ratio must be such that (Gear Ratio X the Master Axis' VL value) is less than or equal to the maximum controller step rate of 4,194,303 steps per second.

The commands specify that the current axis (geared axis) is to track the master axis velocity such that the current axis velocity = Ratio X Master Axis velocity.

Once a gearing engagement command is issued the geared axis will only respond to immediate commands until the gearing is cancelled. The EGF command addressed to the geared axis or over-travel on the geared axis will cancel electronic gearing.

Note: The following commands FL, KL, KS, SA, SD or SI flush the axis queue and will also cancel electronic gearing.

EG? Can be used to report the current electronic gearing settings.

Example: Gear the Z axis to the X axis such that the Z axis velocity is always one half that of X.

Enter:

```
AZ;
EGX1,2;
```

Response: None.

Example: Gear the T axis to the Y axis such that the T axis velocity is always twice that of Y but in a direction that is opposite to Y.

Enter:

```
AT;
EGY-2,1;
```

Response: None.

QUEUE REQUIREMENTS		
FORMAT	**MODE**	**REQUIREMENTS**
EG#,#,#;	Single axis	2
-	AM-AM	Not Valid

Related commands: EGF, EGM

EGF TURN OFF ELECTRONIC GEARING

Firmware Revision Required: V1.28 and greater

The EGF command turns off electronic gearing for the current axis.

Example: Turn off electronic gearing on the X axis

Enter: AX;
EGF;

Response: None

QUEUE REQUIREMENTS		
FORMAT	**MODE**	**REQUIREMENTS**
EGF;	Single axis	Immediate
-	AA-AM	Not Valid

Related commands: EG, EGM

EGM REPORT ELECTRONIC GEARING MAP

Firmware Revision Required: V1.28 and greater

The EGM command reports the electronic gearing map of all axes. It prints a comma separated list of characters, one character for each axis. If an axis is not geared to another axis or is a master, then the only character printed is the uppercase letter designator for that axis (X, Y, Z, etc.). If an axis is geared to another axis, then upper case designator is followed by a lower case letter designator for the master axis.

Example: Gear the X axis to the Y axis and then query for the electronic gearing map.

Enter:

```
AX;
EGY1,1;
EGM;
```

Response:

```
Xy,Y,X,Z,T,U,V,R,S,W,K
```

QUEUE REQUIREMENTS		
FORMAT	**MODE**	**REQUIREMENTS**
EGM;	Single axis	Immediate
-	AA-AM	Not Valid

Related commands: EG, EGF

EH DEFINING ENCODER HOME

The EH command either defines the true states of the home input, encoder index, encoder phase B, & encoder phase A for the Encoder Home condition or it reports the home conditions defined for this axis. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

EH? can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

Format: AX-AS: EHhiba; or AX-AS :EH?

The **EHhiba command provides** variants that will completely determine the logic states of the home input, encoder signals A, B, and Index that make for a true encoder home event. This allows the user more flexibility to adapt to the schemes of most encoder manufacturers.

- h represents the desired state of the home input. It can have values of 0 or 1.

- i represents the desired state of the encoder index input. It can have values of 0 or 1.

- b represents the desired state of the encoder phase B input. It can have values of 0 or 1.

- a represents the desired state of the encoder phase A input. It can have values of 0 or 1.

Note: The EH command is only valid for closed-loop Steppers and Servo axes. If the command is entered for an open-loop stepper axis, a command error will result. The EH command is an immediate command, and therefore requires no queue space

Example: The AX;EH1010; command defines the encoder home conditions of the X axis to be Home input high, Index low/PhaseB high/PhaseA low.

The AX;EH? reports the encoder home conditions for the X axis.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
EHhiba;	Single axis	Immediate	
EHhiba,hiba,…;	AA-AM	Immediate	

Related commands: HI, HM, HR

EHD ENCODER HOME DELAY

Firmware Revision Required: V1.30 and greater

To compensate for any spring action that might be present in a coupling between a motor and its encoder the motor position is synchronized with the encoder position at the end of an encoder home operation. The EHD command sets the delay time between when the axis stops and the time when the motor position is set equal to the encoder position. This time delay allows any mechanical oscillations to settle out before the motor is synchronized with its encoder. The default delay is zero. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

EHD? can be used to report the current setting.

V1.31 and less - RANGE: 0 ≤ EHD ≤ 32767 Milliseconds

V1.32 and greater - RANGE: -1 ≤ EHD ≤ 32767 Milliseconds

V1.32 firmware and greater: if the parameter is -1, then the -1 parameter disables the delay and the motor/encoder synchronization. (This is for backwards compatibility with PCIx controllers and MAX controllers with firmware 1.29 and less.)

Example: The following would set a 1000 millisecond encoder home delay for the X axis.

Enter:
```
AX;
EHD1000;
```

Response: None

Example: The following would set a encoder home delay of 500 milliseconds for the Y axis and 750 milliseconds for the T axis.

Enter:
```
AA;
EHD,750,,500;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
EHD#;	Single Axis	1	1
EHD#,#...;	AA-AM	1	1

Related commands: HI, HM, HR

ER ENCODER RATIO

The ER command allows specification of encoder:motor ratio for position maintenance mode. This command is not designed for use with servo motors. ER takes two arguments: encoder counts and motor counts. Both parameters must be integers. The ratio need not be per full revolution; reduce the integers that define the fraction as far as possible and use those values. **Once an encoder ratio has been defined, all motion is programmed in encoder units as opposed to motor steps.** If it is desired to use User Units, UU, with an encoder ratio, then the User Units should be defined in encoder units.

When an encoder ratio is active, all user-supplied axis coordinated that have the encoder ratio applied must result in a coordinate value that falls within the axis coordinate range of (Min. Position Value, Max. Position Value).

ER? can be used to display the floating point value of motor/encoder.

ERF? can be used to display the floating point value of encoder/motor.

ERI? can be used to display the original integers entered. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

The factory default ratio is 1:1. See the APP command to preserve the ER settings as the Power up/Reset values.

NOTES:

1) If an encoder ratio has been defined by the ER command, then the slip tolerance defined by the ES command is defined in encoder units and not in motor units.
2) When closing the loop on an stepper axis with an encoder ratio defined, the practical hold dead-band (HD) can be limited by the encoder ratio. If an encoder ratio has been defined such that the motor count is less than the encoder count, then the practical minimum hold dead-band is the encoder count divided by the motor count rounded up to the next integer.

Parameter 1 = Encoder Counts

Parameter 2 = Motor Counts

Example: You have an encoder connected to a stepper motor through a series of gears. When the motor steps 25,000 times, the encoder produces 10,000 counts. Set up an encoder ratio so hold mode will work correctly.

Enter: ER10000,25000;
or
ER2,5;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
ER#,#;	Single Axis	1	0

-	AA-AM	Not Valid

Related commands: None

#ER REPORT ILLEGAL COMMAND

Firmware Revision Required: V1.30 and greater

The #ER command reports the characters of the first command that caused a command error since the last #ER command. If no command error has occurred when the #ER command is issued then a null string is reported. Once a command error has occurred, the #ER command can be used to report the characters of the command up to the point that caused the error. Once characters that caused a command error have been reported, future #ER commands will report a null string until the next command error occurs. If multiple command errors occur prior to issuing the #ER command, only the command that caused the first error will be reported. Once the #ER command has been used to report the command characters that caused an error, another command error will need to occur before #ER will report anything other than a null string.

Example: Query the controller for the first two characters of the command that caused an error.

```
Enter:     AX;
           ##        first illegal command
           #?        second illegal command
           RP;       legal command
           #ER;
           I99999999999999;        third illegal command
           #ER;

Response:  ##   (first command error reported by first #ER)
           I99999999999999;   (third command error reported by second #ER)
```

NOTE: The second illegal command was never reported.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#ER;	Single Axis	Immediate	
#ER;	AA-AM	Immediate	

Related commands: None

ES ENCODER SLIP TOLERANCE

The ES command is used to define and display slip tolerance values. Slip happens when there is too much difference between the commanded motor position and the actual position. The ES command operates in two modes, one for axis with encoders and one without.

For axis without encoders, this command is used to specify the slip tolerance range around the home switch. This value is added to and subtracted from the home position to create a slip range around the home switch. If the motor position is outside this range when the home switch is active, then a slip condition is flagged.

For axis with encoders, this command takes two parameters: the first specifies the slip tolerance, and the second specifies the secondary slip tolerance. The first parameter specifies the difference between the commanded motor position and the encoder position before a slip or stall is flagged in the status register and in the RL command response. The encoder may get off target by as much as this value before the controller will consider the axis slipped. The flagging of a slip condition for the first parameter must be turned on by the IS command. If a slip error occurs, the IS command must be re-issued to re-enable slip detection. In addition to the IS command enabling the slip detection, the SK command can be used to activate a kill command when the slip condition is detected. The factory default value is 1 for this parameter.

The second optional parameter specifies a slip detection tolerance that is independent of the IS and SK commands, such that when this tolerance is exceeded a slip interrupt will be generated and a kill command will be issued. This second parameter provides a fail-safe slip tolerance that will stop the axis and interrupt the host when the tolerance is exceeded, even if IS and SK are currently disabled. The default setting for this parameter is 65535. Entering a value of zero for this parameter will disable this secondary slip tolerance mode. See the APP command to preserve the ES settings as the Power up/Reset values.

ES? can be used to query the current parameter settings. For axis without encoders, the ES? command will only report the home switch slip range. For axis with encoders, the ES? command will report both slip tolerance parameter values.

NOTES:

1) If an encoder ratio is defined with the ER command, then the slip tolerance defined by the ES command is in encoder units and not in motor units.
2) The GS command allows an open-loop stepper axis to perform a limited form of slip detection by using the home switch as a reference when executing a move command.

RANGE: 0 = ES = 2,147,483,647 (both parameters)

Example: Your application can tolerate being up to 5 steps from the desired position before the controlling program should be notified of a slip condition.

Enter:
```
ES5;
IS;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
ES#**[**,#**]**;	Single Axis	1	2
-	AA-AM	Not Valid	

Related commands: EA, IS, RL, SK

ET ENCODER TRACKING

The ET command turns on the encoder tracking mode. The axis will track its encoder input, thus allowing one axis to follow the activity of another or a thumbwheel for manual positioning or the movement of another device that produces a signal compatible to the encoder inputs. No acceleration or deceleration ramps are generated. The axis will duplicate the encoder input taking into account any encoder ratio (ER) defined for the axis. The ER command allows the user to scale the motor's movements relative to the encoder. This command is intended to be used with stepper motors with encoders and not with servo motors. A kill command (KL) or a limit condition will cancel encoder tracking mode.

Example: Set up the Y axis so it will follow its encoder input.

Enter:

```
AY;
ET;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
ET;	Single Axis	1	1
ETb,b...;	AA-AM	1	1

Related commands: CL, ER

FL FLUSH QUEUE

The FL command will flush an individual axis' queue. This command is similar in operation to the KL and ST commands except that current motion will remain unaffected by the FL command. All unexecuted commands remaining in the current axis queue will be flushed upon receipt of this command.

Example: Several motion commands have been sent to the X axis but a situation arose and now those commands must be cleared out. The currently executing motion must be allowed to complete to avoid damage to the product.

Enter:

```
AX;
FL;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
FL;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: KL, KS, SA, ST

FT PID FILTER TYPE

Firmware Revision Required: V1.48 and greater

The set PID filter type command takes the form of:

```
FT<type>;
```

The FT command assigns the specified PID filter type to the current axis. The <type> parameter is used to specify the PID filter type. A value of 1 for the <type> parameter assigns the MAX family PID filter. A value of 2 for the <type> parameter assigns a traditional PID filter. If a '?' charater is entered for the <type> parameter, then the currently assigned filter type will be displayed. The default value for the FT command <type> parameter is 1, which assigns the MAX family PID filter.

Example: Assign the MAX family PID filter to the Z axis.

Enter:

```
AX;
FT1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
FT[1/2];	Single Axis	1	1
FT[1/2,1/2,…];	AA-AM	1	1

Related commands: None.

FV REPORT INSTANTANEOUS FRACTIONAL VELOCITY

The FV command will report the current instantaneous velocity at which the axis is moving. The velocity is reported to six decimal places. The reported velocity may differ from the programmed maximum velocity if the axis is accelerating or decelerating or stopped.

Example: Jog the Y axis at 1.2345 steps per second. Display the current velocity.

Enter:

```
AY;
JF1.2345;
FV;
```

Response: 1.234500<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
FV;	Single Axis	Immediate	
FV;	AA-AM	Immediate	

Related commands: JF, RC, RV, VL, VRV

GD GO AND RESET DONE

The GD command may be substituted for a GO command. It will reset the done flags and then initiate the move which has been previously programmed with such commands as MA, MR, MT, and ML just as the GO command does. In single axis mode, only the done flag for the selected axis will be reset.

In AA mode, all the done flags will be reset. In the AM mode, only the axes involved in the move will be reset. This allows the host to reset the interrupts on the axis involved in the next move without affecting other axes which may be still active. Note that this command is probably only useful in applications where commands are queued in advance since the interrupt may be reset before the host has the opportunity to service it if the GD command is waiting in the queue.

If this command is issued without having defined a move, the results are undefined. Issuing a GD command to execute an already-executed move also has undefined results. Only one GD command should be issued per defined move.

Example: In the single axis mode, move the Y axis 12345 counts in the negative direction and set the done flag when the move is completed. Then clear the done flag, move the motor 12345 counts in the positive direction, and set the done flag again when the move is completed.

Enter:

```
AY;
MR-12345;
GO;
ID;
MR12345;
GD;
ID;
```

Response: None.

Example: In AA mode, perform a linear absolute move with the X and Y axes to the position 10000,20000 and set the done flag when the move is completed. Then clear the done flag, perform a linear relative move on both axes moving the X axis 10000 steps in the negative direction and the Y axis 20000 steps in the negative direction, and set the flag once again.

Enter:

```
AA;
MT10000,20000;
GO;
ID;
ML-10000,-20000;
GD;
ID;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
GD;	Single Axis	6*	0*
GD;	AA-AM	8*	0*

- * If the axis is stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.
- * If PA (power automatic) mode is active add 2 to the command queue
- * If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

Related commands: MA, ML, MR, MT, GN, GO, GS

GN GO AND NOTIFY WHEN DONE

The GN command will initiate a move which has been previously programmed with such commands as MA, MR, MT, and ML and set the axis done flags when the move is complete. No operand is required with the GN command. If this command is issued without having defined a move, the results are undefined. Issuing a GN command to execute an already-executed move also has undefined results. Only one GN command should be issued per defined move.

Example: In the single axis mode, move the X axis to absolute position 12345.

Enter:

```
AX;
MA12345;
GN;
```

Response: None

Example: In the AA mode, move the X axis 2468 steps in the positive direction and the Y axis 2468 steps in the negative direction.

Enter:

```
AA;
MR2468,-2468;
GN;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
GN;	Single Axis	6*	0*
GN;	AA-AM	7*	0*

* If the axis is a stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.
* If PA (power automatic) mode is active add 2 to the command queue
* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: GD, GS, MA, ML, MR, MT

GO GO

The GO command will initiate a move which has been previously programmed with such commands as MA, MR, MT, and ML. No operand is required with the GO command. If this command is issued without having defined a move, the results are undefined. Issuing a GO command to execute an already-executed move also has undefined results. Only one GO command should be issued per defined move.

Example: In the single axis mode, move the X axis to absolute position 12345.

Enter:

```
AX;
MA12345;
GO;
```

Response: None

Example: In the AA mode, move the X axis 2468 steps in the positive direction and the Y axis 2468 steps in the negative direction.

Enter:

```
AA;
MR2468,-2468;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
GO;	Single Axis	5*	0*
GO;	AA-AM	7*	0*

* If the axis is a stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.

* If PA (power automatic) mode is active add 2 to the command queue

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: GD, GN, GS, MA, ML, MR, MT

GP GO TO POSITION

Firmware Revision Required: V1.30 and greater

The GP command moves to the specified position(s). A GO command is ***not*** required for motion to start. The motion begins without delay in the next motor update cycle. Velocity override is active during the move. Only linear acceleration and deceleration are used during GP moves. The standard axis commands are used to set the velocity (VL), acceleration (AC), and deceleration (DC). The target position parameters are in absolute coordinates by default, but can be entered in relative coordinates by placing a 'R' character immediately after the GP command and preceding the first digit of the target position parameter.

Example: Set the velocity, acceleration, and deceleration parameters for the X axis and go to target position of 175000. When the move is complete, return to position of 0.

Enter:

```
AX;
VL10000;
AC100000;
DC50000;
GP175000;
GP0;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
GP#;	Single axis	1	3
GP#,#,...;	AA-AM	1*	3*

* Queue requirements per axis.

Related commands: AC, DC, PA, VL, VO

GS GO AND MONITOR SLIP TRIGGER

The GS command works exactly like the GO command except that the home switch will be monitored during the motion. If the home switch becomes active the slip flag will be set for the axis. The host application can read the slip flag and see that the home switch was encountered during the move. This is useful in applications that register slip conditions by means other than encoder position verification; in fact, this command is not valid in controls with encoder feedback which includes servo motors.

If this command is issued without having defined a move, the results are undefined. Issuing a GD command to execute an already-executed move also has undefined results. Only one GD command should be issued per defined move.

Example: Move the X axis 50,000 counts in the positive direction. If the motor slips it will close a switch wired to the home input of the X axis. Monitor this switch during the move and set the slip flag for axis X if the switch becomes active.

Enter:

```
AX;
MR50000;
GS;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
GS;	Single Axis	5*	0*
-	AA-AM	Not Valid	

* If the axis is a stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.

* If PA (power automatic) mode is active add 2 to the command queue

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: GD, GN, GO, MA, MR

HD SET STEPPER HOLD DEADBAND

The HD command specifies dead band counts and optional stability and error timers that are used in position detection.

For closed-loop stepper motors the dead-band timer is used for position maintenance mode. If the encoder count is within this distance of target, it is considered in position and no further correction will be made. This parameter interacts with the HG and HV commands; i.e. a larger dead band will allow a larger gain parameter in many applications. For both closed-loop stepper and servo axes, the dead-band parameter sets the tolerance for the in-position interrupt issued by the IP command.

In single axis mode, two additional optional parameters may be entered. The first optional parameter specifies a stability time in milliseconds. The stability time is used when detecting in position for the IP command such that the axis must be within the dead-band tolerance for the number of milliseconds of stability time before the axis is considered in position. The second parameter can specify an error time out for the in position check. If the in position check does not sense in position within the error timeout time, then a slip interrupt will be issued if interrupt on slip has been enabled with the IS command and a kill will be issued if slip kill has been enabled with the SK command. Note that the in position check will continue to wait for the in position condition indefinitely even though the error timeout has expired. The error timeout is used only to trigger an slip interrupt if enabled by the IS command and a slip kill if enabled by the SK command. Any parameters missing from the parameter list will be left unchanged.

In multi-axis modes only the deadband parameter can be set.

See the APP command to preserve the HD setting as the Power up/Reset values. Note, only the dead-band setting is preserved in the archive memory.

HD? can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

The two time parameters are not preserved in the archive memory with the APP command.

RANGE parameter 1: 0 ≤ HD1 ≤ 64,000default = 0

RANGE parameter 2: 0 < HD2 default = 0

RANGE parameter 3: 0 ≤ HD3 default = 0 (disabled)

NOTE: When closing the loop on an stepper axis with an encoder ratio defined, the practical hold dead-band (HD) can be limited by the encoder ratio. If an encoder ratio has been defined such that the motor count is less than the encoder count, then the practical minimum hold dead-band is the encoder count divided by the motor count rounded up to the next integer.

Example: Set the hold dead-band tolerance to 10, the stability timer to 25, and the error timer to 1500 for the X axis.

Enter:
```
AX:
HD10,25,1500;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
HD#**[**,#,#**]**;	Single Axis	1	3
HD#,#...:	AA-AM	1	1

Related commands: HG, HV, IP

HG SET STEPPER HOLD GAIN

The HG command allows the user to specify the position hold gain parameter. This gain parameter is multiplied by the position error in determining the velocity during correction. The velocity used will not exceed the value set with the hold velocity (HV) command. This command is designed to work with stepper motor applications using encoders and is not designed for use with servo motors. The parameter should be set experimentally by increasing it until the system is unstable then reducing it slightly below the threshold of stability. The factory default value is 1. See the APP command to preserve the HG settings as the Power up/Reset values.

HG? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE: 1 ≤ HG ≤ 32,000

Example: (see CL command (page 75))

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
HG#;	Single Axis	1	1
HG#,#...;	AA-AM	1	1

Related commands: HD, HV

HI HOME INDEX

Firmware Revision Required: V1.30 and greater

The HI command is used to select whether the home position is determined only by the home switch or by the encoder in conjunction with the home switch. This affects how the HM and HR commands determine the home position. The encoder home position is determined by the index, phase A, and phase B signals. The TRUE state of the index, phase A, and phase B signals required to determine the home position are defined by the EH command. The TRUE state of the home switch input is defined by the HT command. The default for the HI command is to only use the home switch to determine the home position, and is set by executing the command HI0. The home logic for HI0 is expressed in Boolean terms as:

home = HomeSwitch XNOR TrueState_HomeSwitch

The command HI1 is executed to use both the home switch and the encoder to determine a more precise home position. The home logic for HI1 is the logical AND of the encoder phase A, the encoder phase B, the encoder index, and the home switch. The HI1 mode is expressed in Boolean terms as:

home = (phase_A XNOR TrueState_PhA) AND (phase_B XNOR TrueState_PhB) AND(index XNOR TrueState_Indx) AND (home_switch XNOR TrueState_HomeSwitch)

Use the HT command to define the true home switch state. Use the EH command to define the encoder signal (Home, Index, Phase A and B) states required for the true home condition.

HI? can be used to query the current setting.

Example: Set up the Y axis so it will use the encoder signals to recognize the home position.

Enter:
AY;
HI1;

or
AA;
HI,1;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HI[0/1/?];	Single Axis	Immediate	
HIb,b...;	AA-AM	Immediate	

Related commands: EH, HM, HR, HT, KM, KR

HM HOME IN POSITIVE DIRECTION

The HM command will cause the current axis or specified axes to move in the positive direction at the predefined velocity until the home is detected for each axis. The position counters will be initialized to the positions supplied as parameters.

How home is detected depends on the HI mode. The default mode detects home based on an external home input only. The HI mode detects home based on the external home input and the encoder signals (Index, A and B), as defined by the EH command.

When using the HI mode, the velocity should be less than the update rate to maintain accuracy of the home position loaded. A velocity set to less than or equal to the update rate will provide a homing accuracy of +/-0 counts. Every multiple of the update rate adds +/-1 count to the error range. When using the HI mode, the accuracy of the home position is independent of the velocity.

Each axis will, when home is detected, reset its position counter to the parameter specified in the HM command. Once the counter is reset, the axis will ramp to a stop at the rate specified previously via the AC command. This will result in the axis stopping beyond the home switch so care should be taken to ensure adequate stopping distance is available beyond the switches. The axis can be easily returned to the precise home position by using an MA command after the HM command.

If no parameter is specified in single axis modes, the HM command will use zero as a default value. Parameters must be specified in multi-axis modes to inform the MAX which axes are to be homed.

Example: Find the physical home position of the X axis of the stage. The motor runs until the home switch input is activated and then initializes the position counter to the parameter supplied. Since the motor decelerates to a stop after reaching home, it is necessary to do an MA to the same position as specified in the home command if it is desired to physically position the device at home. The following commands will find home, initialize it to 1000 counts, and then return to home. In many cases it will not be necessary to return home, only find the position and synchronize the controller to it.

Enter:

```
AX;
VL1000;
HM1000;
MA1000;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
HM#;	Single Axis	3*	1*
HM#,#...;	AA-AM	3*	11*

- * If the axis is a stepper and encoder or servo axis add 2 to the argument queue.
- * If PA (power automatic) mode is active add 2 to the command queue.
- * If an auxiliary output bit settle time has been specified add 2 to the command queue and 1 to the argument queue.
- * If the last profile move, just prior to this home command was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the home. This will add to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue
Linear (RTL)	4	4
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: EH, HI, HT, KM, KR, LO, LP, RP

HR HOME REVERSE

The HR command will cause the current axis or specified axis to move in the negative direction at the predefined velocity, until the home is detected for each axis. When the home input is detected, the position and encoder counters are loaded with the parameter(s) supplied in the HR command. Then the axis is ramped to a stop. It behaves exactly like the HM command, except it travels in the reverse direction.

How home is detected depends on the HI mode. The default mode detects home based on an external home input only. The HI mode detects home based on the external home input and the encoder signals (Index, A and B), as defined by the EH command.

Example: In a long stage it may be awkward to travel the full distance to home at less than the update rate. The following will get close to home at higher speed, and then refine the position at lower speed in the reverse direction.

Enter:

```
AX;
VL100000;
HT1;
HM;
VL1000;
HT0;
HR;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HR#;	Single Axis	3*	1*
HR#,#...;	AA-AM	3*	1*

* If the axis is a stepper and encoder or servo axis add 2 to the argument queue.
* If PA (power automatic) mode is active add 2 to the command queue.
* If an auxiliary output bit settle time has been specified add 2 to the command queue and 1 to the argument queue.
* If the last profile move, just prior to this home command, was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the home. This will add to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue
Linear (RTL)	4	4
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: EH, HI, HT, HM, HS, KM, KR, LO, LP

HT HOME TRUE STATE

Firmware Revision Required: V1.30 and greater

The HT command sets the sense of the home switch on the current axis. This command allows either TTL logic high or low to be treated as the "true" state for homing operations. Once this command has been sent to the MAX, the selected TTL level can be made the power-up default by using the APP command. This command sets the home switch "true" state for all home operations.

HT? can be used to query the current setting.

Example: Set home state active high for the Z axis.

Enter: AZ;
HTH;

Or

AA;
HT,,H;

Response: None

Note: In AA or AM modes, a null argument in the parameter list specifies that axis home switch to be unchanged.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
HT[L/H/?];	Single Axis	1	0
HT[L/H,L/H,…];	AA-AM	1	0

Related commands: EH, HI, HM, HR, KM, KR

HV SET STEPPER HOLD VELOCITY

The HV command specifies the maximum velocity to be used when correcting position error. The factory default setting is zero; some value must be set for position correction to occur at all. See the APP command to preserve the HV settings as the Power up/Reset values. This command is not designed for use with servo motors.

Hold gain (HG) will be used to scale the HV value based on the total error that must be corrected. In most cases the HV value will never be reached unless the position error is very wide or the HG value is set very high.

HV? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE: 1 ≤ HV ≤ 4,194,303

Example: (see CL command, see page 75)

NOTE: In AA or AM mode, a null value in the argument list specifies that the hold velocity parameter is not to be changed for that axis.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
HV#;	Single Axis	1	1
HV#,#...;	AA-AM	1	1

Related commands: HD, HG

IC INTERRUPT CLEAR

The IC command or the ASCII character Control-Y (hex 19) is used to clear the done and error flags in the status register and the done flag register on MAX, otherwise the axis would always appear to be “done”. This command will be executed immediately and will usually be placed in the done and error handler interrupt service routine to clear the interrupt and the associated flags. The flags may be polled by an RA or RI command which will also reset the flags.

Note: This command is not recommended in Windows environments using OMS-supplied device drivers and DLLs. The preferred method is to use the device driver and DLL to handle the reporting and clearing status flags.

Example: Clear the flags after an X axis move relative of 5000 steps was flagged as done when an ID executes.

Enter:

AX;
MR5000;
GO;
ID; (done flag set, wait until done interrupt received)
IC; (done flag cleared on MAX)

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IC;	Single Axis	Immediate	
IC;	AA-AM	Immediate	

Related commands: CA, GD, ID, II, IN, IP

ID INTERRUPT WHEN DONE

The ID command will set the done flag for the axis or axes to which the command was issued. An interrupt to the host will be generated if interrupts have been enabled in the MAX. In Windows environments, this interrupt is captured by the device driver which will store the flags to pass on to the host application when requested.

This command allows the MAX to signal the host when a string of commands has been completed.

Though move commands are most commonly used with the ID command, others may be used as well. The ID command may be sent to the MAX to tell it to set done flags whenever the host application could benefit from knowing an event or series of commands has completed.

In AA mode, the done bit is set for each of the axes. In AM mode, the done bit is set for each of the most recently used axes.

Example: Interrupt the host CPU after the execution of Move Absolute is finished. When the move is finished the ID command will be encountered in the command queue and will set the done flags.

Enter:
```
AX;
MA100000;
GO;
ID;
```

Response: None

Example: Wait until an input line becomes false then notify the host that this occurred.

Enter:
```
SW2;
ID;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ID;	Single Axis	1	0
ID;	AA-AM	1	0

Related commands: II, IN, IP, VID, VIP

II INTERRUPT INDEPENDENT

Like the ID command, the II command tells the MAX to interrupt the host when each axis finishes a move. Unlike the ID command, only those axes which have been supplied parameters in the most recent move command will get their done flags set and cause interrupts.

Example: The following command sequence would cause interrupts when the Y and T axes finish. If they do not complete at the same time, two separate interrupts would be generated.

Enter:

```
AM;
MR,1000,,10000;
GO;
II;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	Single Axis	Not Valid	
II;	AA-AM	1	0

Related commands: ID, IN, IP

IN INTERRUPT NEARLY DONE

The IN command allows the MAX to interrupt the host when the axis or combination of axes is nearly complete. When used in an application involving probing a part after a move, the probes could start accelerating down while the stage is finishing its move, improving the overall system throughput. This command is valid in all modes. The IN command must be entered before the GO or GD command since it is executed before the move is complete. The test is only performed during deceleration. If the IN parameter is greater than the ramp down distance, the interrupt will be generated when the control starts decelerating.

When a multi-axes move is used in AA/AM modes, IN specifies a separate near distance for each axis, but only a single interrupt occurs when all the axes in the move are within their respective distances specified by the IN command. Once an interrupt has occurred, the IN command must be re-issued in order to generate another interrupt.

IN? can be used to query the current setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE:
Min Position Range ≤ Parameter (Distance Before End of Move) ≤ MAX Position Range

Example: The following sequence would interrupt the host once when the X axis is complete and the Z axis is within 10,000 counts of being complete. The Y axis completion would be ignored in this example.

Enter:

```
AA;
IN0,,10000;
MR100000,100000;
GO;
MR,,50000;
GO;          (X and Z interrupt occurs when Z is within 10000 steps
             of the Z destination.)
```

Response: None

NOTE: In AA or AM mode, a null value in the argument list specifies the position tolerance for nearly done is not to be set for that axis.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IN#;	Single Axis	1	2
IN#,#...;	AA-AM	1	2

Related commands: ID, II, IP

IO SET I/O BIT DIRECTION

This IO# command selects a general purpose I/O to be an input or output as determined by the designator where 1 is an output and 0 is an input. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

Range:

0 ≤ parameter 1 ≤ 15 for MAXp

0 ≤ parameter 1 ≤ 15 for MAXv

0 ≤ parameter 1 ≤ 7 for MAXnet

0 ≤ parameter 2 ≤ 1 for all

Example: Select I/O bit 4 as an output and I/O bit 5 as an input.

Enter:

```
AA;
IO4,1;
IO5,0;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
IO#,#;	Single Axis	Immediate	
IO#,#;	AA-AM	Immediate	

Related commands: BD, BH, BL, BX

IOE DEFINE ERROR OUTPUTS

Firmware Revision Required: V1.30 and greater

The IOE command defines the state of selected general purpose outputs when a critical error occurs. Critical errors are slip, over-travel limits, and Kill operations caused by general purpose inputs (see IOK command). The IOE command takes two decimal parameters. The first parameter specifies the general purpose output bit number that is to be affected when the critical error occurs. Note that only general purpose I/O that are defined as outputs can be affected by the IOE command. If a bit is specified that is not defined as an output, a command error will occur. The second parameter specifies the state of the output when a critical error occurs. A 1 in the parameter specifies a logic high state on the output when the error occurs and a 0 specifies a logic low state. Multiple commands may be issued to specify multiple outputs to occur simultaneously when the error occurs. Each IOE command that is entered adds an output to be affected when critical errors occur. An IOE,C command clears the IOE status to the power-up default of no outputs on critical error.

Once a critical error has occurred, it is the user's responsibility to reset the outputs to the non-error state before another error occurs so that there will be a transition of the defined output states when the next error occurs (see IOE,R command). When IOE is executed in single axis mode, only critical errors on the currently defined axis will cause the defined error outputs. When IOE is executed in AA or AM modes the single axis settings are overridden and critical errors on any axis will cause the defined error outputs. The power-up default is that no output will occur when a critical error occurs.

IOE? can be used to report the current settings.

RANGE:

0 ≤ parameter #1 ≤ 15 for MAXp

0 ≤ parameter #1 ≤ 15 for MAXv

0 ≤ parameter #1 ≤ 7 for MAXnet

0 ≤ parameter #2 ≤ 1 for all

Example: Specify that general purpose output bit 15 will be in the high state when a critical error occurs on any axis.

Enter:
BDFF00;
AA;
IOE15,1;
… processing…
… an error has occurred on the Y axis host senses error output and takes action
IOE,R; processing is ready to resume so set IO15 low so future errors can set it high

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IOE#,#;	Single Axis	Immediate	
IOE#,#;	AA/AM	Immediate	

Related commands: BD, IOE,C, IOE,R, IOE,X

IOE,C CLEAR DEFINED ERROR OUTPUTS

The IOE,C command clears all previously defined IOE or IOE,X states to the power-up default, which is that no outputs are activated when a critical error occurs. In single axis mode, just the error outputs for the currently defined axis are cleared. In AA/AM modes, the error outputs of each axis are cleared.

Example: Clear any previously defined error outputs so that no outputs are activated when a critical error occurs.

Enter: IOE,C;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
IOE,C;	Single Axis	Immediate	
IOE,C;	AA/AM	Immediate	

Related commands: BD, IOE, IOE,R, IOE,X

IOE,R RESET ERROR OUTPUTS

The IOE,R command sets the state of all outputs specified by the IOE or IOE,X commands to the non-error state. In single axis mode, the outputs defined for the current axis will be reset to the non-error state. In AA/AM modes, the error outputs defined for each axis will be reset to the non-error state regardless of whether all axes have identical error outputs defined.

Example: Set all defined error outputs to the non-error state.

Enter: IOE,R;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IOE,R;	Single Axis	Immediate	
IOE,R;	AA/AM	Immediate	

Related commands: BD, IOE, IOE,C, IOE,X

IOE,X DEFINE ERROR OUTPUTS (HEX)

The IOE,X command is a hexadecimal version of the IOE command. The IOE,X command takes two hexadecimal parameters. The first parameter is a mask that specifies the output bits that will be affected when the critical error occurs. The second parameter specifies the state of the output(s) when a critical error occurs. Bits that are a 1 in the second hexadecimal parameter specify a logic high state on the output when the error occurs and bits that are 0 specify a logic low state. The least significant bit in the hexadecimal value represents I/O bit 0 and the most significant bit represents the highest number I/O bit. Note that only general purpose I/O that are defined as outputs can be affected by the IOE,X command. If any bit specified is not defined as an output, a command error will occur. This command allows the specification of a the complete state of the outputs on a critical error with a single command.

IOE,X can be followed by a '?' to report the error output setting in hexadecimal.

RANGE: 0 ≤ parameter #1 ≤ ffff for MAXp and MAXv

0 ≤ parameter #1 ≤ ff for MAXnet

must be a bit number defined as an output

0 ≤ parameter #2 ≤ ffff for MAXp and MAXv

0 ≤ parameter #2 ≤ ff for MAXnet

Example: Specify that general purpose output bit 8 will be in the low state when a critical error occurs on the X axis.

Enter:

```
BDFF00;
AX;
IOE,X0000,0100;
…                     processing…
…                     an error has occurred on the Y axis
…                     host senses error output and takes action
IOE,R;                processing is ready to resume so set IO8 high
                      so future errors can set it low
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IOE,X#,#;	Single Axis	Immediate	
IOE,X#,#;	AA/AM	Immediate	

Related commands: BD, IOE, IOE,C, IOE,R

IOK DEFINE KILL INPUTS

The IOK command defines the state of selected general purpose inputs that will cause an automatic Kill operation. The command takes two decimal parameters. The first parameter specifies the general purpose input bit number that can cause a kill operation. Note that only general purpose I/O that are defined as inputs can be affected by the IOK command. If a bit is specified that is not defined as an input, a command error will occur. The second parameter specifies the state of the input that will cause a kill operation. A 1 in the parameter specifies a logic high state on the input causes a kill operation and a 0 specifies a logic low state on the input causes a kill operation. Multiple commands may be issued to specify multiple inputs, any of which will cause a kill operation. Each command that is entered adds inputs that will cause a kill operation until an IOK,C command clears the kill inputs to the power-up default of no inputs that cause a kill operation.

If any of the specified input bits are in the defined state, a Kill operation will occur and no queued commands will execute until all the inputs specified are in the opposite state defined for the Kill operation. The Kill operation affects all axes regardless of whether the command was entered in single or multi-axis mode. The power-up default is that no inputs cause a kill operation.

IOK? can be used to report the current settings.

RANGE:

0 ≤ parameter 1 ≤ 15 for MAXp and MAXv

0 ≤ parameter 1 ≤ 7 for MAXp and MAXv

0 ≤ parameter 2 ≤ 1 for all

Example: Specify that whenever general purpose input bit 6 or 7 is in a logic low state that a Kill All operation will be performed.

Enter:
BDFF00;
AA;
IOK6,0;
IOK7,0;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
IOK#,#;	Single Axis	Immediate	
IOK#,#;	AA/AM	Immediate	

Related commands: BD, IOK,C, IOK,X

IOK,C CLEAR DEFINED KILL INPUTS

The IOK,C command clears all previously defined kill inputs states (see IOK or IOK,X) to the power-up default, which is that no inputs cause a kill operation.

Example: Clear any previously defined kill inputs so that no inputs cause a kill operation.

Enter: IOK,C;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
IOK,C;	Single Axis	Immediate	
IOK,C;	AA/AM	Immediate	

Related commands: BD, IOK, IOK,X

IOK,X DEFINE KILL INPUTS (HEX)

The IOK,X command is a hexadecimal version of the IOK command. The IOK,X command defines the state of selected general purpose inputs that will cause an automatic Kill operation. The command takes two hexadecimal parameters. The first parameter is a mask that specifies all input bits that will cause a Kill operation. The second parameter specifies the state of the input(s) that will cause a Kill operation. 1's in the second hexadecimal parameter specify a logic high state on the input and 0's specify a logic low state. The least significant bit in the hexadecimal value represents I/O bit 0 and the most significant bit represents the highest numbered I/O bit. Note that only general purpose I/O that are defined as inputs can be specified by the command. If any specified bit is not defined as an input, then a command error will occur. If any of the specified input bits are in the defined state, a Kill operation will occur and no queued commands will execute until all the inputs specified are in the opposite state defined for the Kill operation. The Kill operation affects all axes regardless of whether the command was entered in single or multi-axis mode. The power-up default parameters are 0000,0000 which specify that no inputs will cause a kill operation.

IOK,X? can be used to report the current settings in hexadecimal.

Example: Specify that whenever general purpose input bit 6 or 7 is in a logic low state that a Kill All operation will be performed.

Enter:

```
BDFF00;
AA;
IOK,X00C0,0000;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
IOK,X#,#;	Single Axis	Immediate	
IOK,X#,#;	AA/AM	Immediate	

Related commands: BD, IOK, IOK,C

IP INTERRUPT WHEN IN POSITION

The IP command operates like the ID command except the interrupt is deferred until the stage is within the specified encoder hold dead-band set by the HD command. If the position hold (CL) is not enabled for an axis, the command will behave like an ID command for that axis.

The IP command uses parameters set by the HD command to determine when to generate the in position interrupt.

HD optional parameter 1 specifies the in position dead-band tolerance.

HD optional parameter 2 specifies the number of milliseconds that the position must be within the hold dead-band before the interrupt will be generated. If no parameter is specified the interrupt will be generated the first update cycle that the position is within the hold dead-band.

HD parameter 3 can be used to specify the number of milliseconds before a slip error is generated. Note that a slip error will only be generated if interrupt on slip is enabled with the IS command. If interrupt on slip is enabled then a slip interrupt will be generated after the specified number of milliseconds. The IP command will continue to wait indefinitely, however, unless slip kill is enabled with a SK command. When slip kill is enabled, in addition to the slip interrupt being generated when the timeout expires, kill will also be generated, thus flushing the IP command from the queue as well as all other commands.

Example: Set the done flag when axis is within it's dead-band of 10 steps for a minimum of 15 milliseconds and generate a slip interrupt if it takes longer than 1000 milliseconds to get in position.

Enter:

```
AX;
HV1000;
HG100;
HD10,15,1000;
CL1;
MR1000;
GO;
IS;
IP;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
IP;	Single Axis	1	2
IPb,b...;	AA-AM	1	2

Related commands: CL, ID, II, IN, IS, HD, SK

IS INTERRUPT ON SLIP

The IS command enables the MAX to interrupt the host on slip or stall detection. Slip detection are disabled if a ER, ET, CL, HM, or HRcommand is entered or if a limit is encountered. If a slip occurs, slip detection must be re-enabled. The factory default slip tolerance (ES) value is 1.

In AA or AM modes, parameters follow the IS command to specify multiple axes to interrupt when the slip occurs on the axis. Each parameter corresponds to an axis in the usual multi-axis order. If the parameter is entered, then axis is selected for the interrupt on slip operation. If no parameter is entered, then axis is not selected for the interrupt on slip operation.

Example: (see ES command on page 118)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IS;	Single Axis	1	0
ISb,b...;	AA-AM	1*	0

* = per axis

Related commands: ES, IOE, RL, SK

JF JOG FRACTIONAL VELOCITIES

The JF command will jog one or more axes at the velocities specified, like the JG command. The parameter may include a fractional part allowing better resolution at low speeds. The velocity set by this command will remain the default velocity until altered by a VL, JG or another JF command.

RANGE: -4,194,303.000 ≤ JF ≤ 4,194,303.000

Example: Jog the Y axis at 2 $^{2/3}$ steps per second.

Enter: AY;
JF2.667;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
JF#;	Single Axis	3	1
JF#,#...;	AA-AM	3	1

* If PA (power automatic) mode is active add 1 to the command queue

* If the axis is a stepper and encoder or servo axis add 1 to the command queue

* If the profile move, just prior to this home command was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the home. This adds to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue
Linear (RTL)	4	4
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: FV, JG, SA, ST, TM

JG JOG

The JG command is a velocity mode and will move one or more axes at the velocities supplied as parameters. The JG command will accelerate to the programmed velocity at the current AC rate and run until altered by an ST, SA, KL, CL (servo models), another JG command, or a limit switch is encountered while limits are enabled.

The jog velocity may be changed by following the command with another JG command of a different velocity. A change in direction between two JG commands will cause an axis to ramp to a stop then back up in the opposite direction.

This command modifies the move velocity parameter (VL) for the affected axis or axes. The JG command does not require any other command to start the motion.

RANGE: -4,194,303 ≤ JG ≤ 4,194,303

Example: Jog the motor at 100,000 counts per second then change to 35,000 counts per second when the second JG is entered, stay at that velocity for 5 seconds, then stop by decelerating to a stop. Next, jog the motor at 5,000 counts per second in the negative direction.

Enter:

```
JG100000;
JG35000;
WT5000;
ST;
JG-5000;
```

Response: None

Note: Output events waiting for completion of JG will begin when JG is up to its requested velocity. In this case, the motor will ramp from zero to 100,000, ramp back down to 35,000, flatten out at 35,000 for 5 seconds, then ramp to a stop, before moving in the negative direction at a velocity of 5,000.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
JG#;	Single Axis	3	1
JG#,#...;	AA-AM	3	1

* If PA (power automatic) mode is active add 1 to the command queue
* If the axis is a stepper encoder or servo axis add 1 to the command queue
* If the profile move, just prior to this home command was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the home. This add to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue
Linear (RTL)	4	4
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: JF, SA, ST, TM

KA SET PID ACCELERATION FEED-FORWARD GAIN COEFFICIENT

KA is the acceleration feed-forward gain used in the PID filter calculations. The acceleration feed-forward gain coefficient is a percentage of the commanded acceleration. Acceleration feed-forward is used with velocity feed-forward, and is used to reduce the overshoot created by the velocity feed-forward gain, which is set with the KV command. The factory default value for acceleration feed-forward gain is zero. See the APP command to preserve the KA settings as the power up/reset values. **KA?** can be used to query the current parameter setting.

RANGE: 0.00 ≤ KA < 32767.00

Example: Define KA to be 13.5 on the Y axis.

Enter:

```
AY;
KA13.5;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KA#;	Single Axis	1	1
-	AA-AM	Not Valid	

Related commands: CL, KB, KD, KI, KF, KO. KOD, KP, KU, KV

KB SET PID MAXIMUM OUTPUT

KB sets the maximum DAC output voltage of the servo axis PID. The KB command takes values in the range of zero to ten, where a value of zero limits the DAC output voltage to zero. The factory default value is 10, which is the maximum DAC output voltage; therefore, no limit is placed on the PID DAC output voltage.

For bi-polar mode, the DAC voltage is limited to +/- the KB value.

BI-POLAR LIMIT: - KB = DAC = + KB

For unipolar mode, the DAC voltage is limited to 5 volts +/- the KB value divided by 2.

UNIPOLAR LIMIT: 5 - (KB/2) = DAC = 5 + (KB/2)

See the APP command to preserve the KB settings as the power up/reset values. **KB?** can be used to query the current parameter setting.

RANGE: 0.0 ≤ KB ≤10.0

Example: Limit the servo DAC output to 9 max on Y axis.

Enter: AY;
KB9;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KB#;	Single Axis	1	1
-	AA-AM	Not Valid	

Related commands: CL, KA, KD, KI, KF, KO. KOD, KP, KU, KV

KD SET PID DERIVATIVE GAIN COEFFICIENT

KD sets the gain coefficient used in the PID filter calculations for the derivative term. The derivative term slows the rate of change of the PID output; is used to reduce the overshoot produced by the integral term (see KI), and improve stability of the system. The factory default value is 160. See the APP command to preserve the KD settings as the power up/reset values. **KD?** can be used to query the current parameter setting.

RANGE: 0.00 ≤ KD < 32767.00

Example: Set the KD gain to 56 on the Z axis.

Enter: AZ;
KD56;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KD#;	Single Axis	1	1
-	AA-AM	Not Valid	

Related commands: CL, KA, KB, KI, KF, KO. KOD, KP, KU, KV

KF SET SERVO AXIS PID FRICTION COEFFICIENT

The KF command sets the friction offset coefficient of the PID for a servo axis PID. A friction offset value assists in the smooth start up motion of the axis, and compensates for friction. The factory default value is 0. See the APP command to preserve the KF settings as the power up/reset values. **KF?** can be used to query the current parameter setting.

Range: 0.0 ≤ KF ≤ 32767.00

Example: Set the Y axis friction coefficient to 100.

Enter:
```
AY;
KF100;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KF#;	Single Axis	1	1
-	AA-AM	Not Valid	

Related commands: CL, KA, KB, KD, KI, KO. KOD, KP, KU, KV

KI SET PID INTEGRAL GAIN COEFFICIENT

KI is the integral gain coefficient used in the PID filter calculations. The integral term of the PID accumulates errors from the past. The KU command is used to set the maximum value that the integral term is allowed to accumulate and contribute to the PID. If the integral term accumulates too much error, or the integral gain is too high, then overshoot can occur. The factory default value is 1. **KI?** can be used to query the current parameter setting. See the APP command to preserve the KI settings as the power up/reset values.

RANGE: 0.00 ≤ KI < 32767.00

Example: Define KI to be 12.3 on the Y axis.

Enter:

```
AY;
KI12.3;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KI#;	Single Axis	1	1
-	AA-AM	Not Valid	

Related commands: CL, KA, KB, KD, KF, KO. KOD, KP, KU, KV

KL KILL

The KL command will flush the command queues and terminate pulse generation of all axes immediately. It is intended for emergency termination of any program and to reset the input queues to a known state.

Step motors may not stop immediately even though no more step pulses are delivered due to inertia of the motor and system load. This may result in slippage of the motor. Therefore, the position counter may not accurately reflect the true position of the motor following this command. All axes should be re-homed to return the position counters to a known state.

Due to the encoders used in servo systems, position will not be lost, so re-homing servo axes is unnecessary.

Example: Stop all previously defined movement and flush the queue of a partially entered incorrect move command (you wanted a negative move not a positive one), before GO is entered.

Enter:

```
AX;
MR5000; (oops!)
KL;
MR-5000;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KL;	Single Axis	1	0
KL;	AA-AM	1	0

* If PA (power automatic) mode is active add 1 to the command queue

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: FL, KS, SA, SD, SI, SO, ST

KM HOME AND KILL

The KM command will move the current axis in the positive direction until home is detected and then kill motion immediately; i.e. without using a deceleration ramp. The position counter will not be reset or cleared. Due to motor and/or payload inertia, the motor may not stop immediately but slip some distance instead. This will result in inaccurate position counters. This command's primary purpose is to move an axis out of the way quickly or to get the axis near home rapidly to speed up the homing process.

Example: Move the Y axis in the positive direction to the home sensor and stop movement as quickly as possible.

Enter: AY;
KM;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KM;	Single Axis	1	1
KMb,b,b,b,b,b,b,b;	AA-AM	1	1

* If PA (power automatic) mode is active add 2 to the command queue.

* If an auxiliary output bit settle time has been specified add 2 to the command queue and 1 to the argument queue.

* If the last profile move, just prior to this home command, was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the Home. This will add to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue
Linear (RTL)	4	4
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: HI, HM, HT, HR, KR, LO, LP

KO SET PID OFFSET COEFFICIENT

The KO command defines the offset coefficient to cause the motor to remain stationary and compensate for additional torque on the motor from loading. The KO command has a range of +/-32,767, which corresponds directly to the 16-bit range of the DAC. Each increment/decrement of the KO value will result in an approximate change in the DAC output voltage of 0.0003 volts. The factory default value is 0. **KO?** can be used to query the current parameter setting. See the APP command to preserve the KO settings as the power up/reset values.

RANGE: -32767.00 ≤ KO ≤ 32767.00

Example: Define the offset coefficient to be –2000 (~ -610mV) on the Y axis.

Enter: AY;
KO-2000;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KO#;	Single Axis	1	1
-	AA-AM	Not Valid	

Related commands: CL, KA, KB, KD, KF,KI, KOD, KP, KU, KV

KOD SET PID OUTPUT DEAD BAND COMPENSATION OFFSET

The KOD command defines the dead band compensation offset values that lets the PID overcome servo amplifiers input dead band. The command takes 1-3 arguments, positiveOffset, negativeOffset and offsetThreshold. The positiveOffset is added to the positive PID output value. The negativeOffset is optional and is subtracted from the negative PID output value. If no negativeOffset is given, the positiveOffset value is used instead. The offsetThreshold is optional and specifies the minimal deviation of the PID from 0 needed to trigger addition/subtraction of the offset values. If the PID output is less than the offsetThreshold range, then the PID output is forced to 0 and only the KO offset is applied.

The KOD parameters values have a range of 0-32,767 for the Offsets and 1-32,767 for the Threshold. These values corresponds directly to the 16-bit range of the DAC. Each increment/decrement of these value will result in an approximate change in the DAC output voltage of 0.0003 volts. To calculate the KO offset or threshold parameter P for a given voltage V, use the formula P = 32767 * voltage / 10 and round to the nearest integer.

The factory default values are 0 for the positiveOffset, 0 for the negativeOffset, and 1 for the offsetThreshhold, which is the command equivalent of KOD0,0,1. KOD? can be used to query the current parameter setting. See the APP command to preserve the KOD settings as the power up/reset values. **KOD?** can be used to query the current parameter setting.

RANGE: 0 ≤ positive*Offset* ≤ 32767

RANGE: 0 ≤ negative*Offset* ≤ 32767

RANGE: 1 ≤ offset*Threshold* ≤ 32767

Example: Define a dead band compensation between -500mV and +550mV on the Y axis.

Enter: AY;
KOD1802,1638;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KOD#[,#[,#]];	Single Axis	1	3
-	AA-AM	Not Valid	

Related commands: CL, KA, KB, KD, KF, KI, KO, KP, KU, KV

KP SET PID PROPORTIONAL GAIN COEFFICIENT

KP is the proportional gain coefficient used in the PID filter calculations. The proportional term of the PID produces an output value that is proportional to the current position error. A high proportional gain value gives a large response to position error, and can result in an unstable system. A low proportional gain value gives a smaller response to position error, and can result in a less responsive system to position error. The factory default value is 10. **KP?** can be used to query the current parameter setting. See the APP command to preserve the KP settings as the power up/reset values.

RANGE: 0.00 ≤ KP < 32767.00

Example: Define KP to be 12.3 on the Y axis.

Enter: AY;
KP12.3;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KP#;	Single Axis	1	1
-	AA-AM	Not Valid	

Related commands: CL, KA, KB, KD, KF, KI, KO, KOD, KU, KV

KR HOME REVERSE AND KILL

The KR command will find home by issuing a JG in the negative direction. When home is found, it will stop generating pulses immediately; i.e. no deceleration ramp will be generated. This command is identical to the KM command except that the direction of motion is reversed.

Example: Move the Y axis in a negative direction to the home sensor and stop movement as quickly as possible.

Enter: AY;
KR;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KR;	Single Axis	1	1
KRb,b... ;	AA-AM	1	1

* If PA (power automatic) mode is active add 2 to the command queue.

* If an auxiliary output bit settle time has been specified add 2 to the command queue and 1 to the argument queue.

* If the last profile move, just prior to this home command, was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the Home. This will add to the queue requirements under the following conditions:

QUEUE REQUIREMENTS		
AXIS RAMP TYPE	**COMMAND**	**ARGUMENT**
Linear (RTL)	4	4
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: HI, HM, HR, HT, KM, LO, LP

KS KILL SELECTED AXES

This command performs the same operation as the KL (kill) command except that individual axes can be killed without affecting others. KS will flush only the selected axes' command queues rather than the entire board. Refer to the KL command for more details.

Example: The Y axis has hit a limit switch and is now executing commands that were waiting in the queue. This axis must be reset but the other axes must be allowed to continue operation.

Enter:

```
AY;
KS;
or
AA;
KS,1;
```

Response: None

NOTE: In AA or AM modes, null values in the argument list specify that motion on that axis is not to be killed.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KS;	Single Axis	1*	0*
KSb,b...;	AA-AM	1*	0*

* If PA (power automatic) mode is active add 1 to the command queue

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: FL, KL, SA, SD, SI, SO, ST

KU SET PID INTEGRATION SUM UPPER LIMIT

The KU command is used to limit the integral terms contribution to the PID filter. The integral term of the PID accumulates errors from the past. The KU command sets the maximum value that the integral term is allowed to accumulate and contribute to the PID. If the integral term accumulates too much error, or the integral gain is too high, then overshoot can occur. The integral gain is set with the KI command. The factory default value for the KU command is 200. **KU?** can be used to query the current parameter setting. See the APP command to preserve the KU settings as the power up/reset values.

RANGE: 0.00 ≤ KU ≤ 32767.00

Example: Set the integration sum upper limit of X axis to 300

Enter:
```
AX;
KU300;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KU#;	Single Axis	1	1
-	AA-AM	Not Valid	

Related commands: CL, KA, KB, KD, KF, KI, KO, KOD, KP, KV

KV SET PID VELOCITY FEED-FORWARD

KV is the velocity feedforward gain used in the PID filter calculations. The velocity feed-forward gain parameter is used as a percentage of the commanded velocity. Velocity feed-forward can reduce the response time of the system, but it can also introduce overshoot. Overshoot can be reduced by using KA command to add acceleration feed-forward, or by reducing the KI and/or KP gains. However, reducing the KI and/or KP gains can make the system less responsive to disturbances. The factory default value is zero. See the APP command to preserve the KV settings as the power up/reset values. **KV?** can be used to query the current parameter setting

RANGE: 0.00 ≤ KV < 32767.00

Example: Set KV to 35.3 on the Y axis.

Enter:

```
AY;
KV35.3;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KV#;	Single Axis	1	1
-	AA-AM	Not Valid	

Related commands: CL, KA, KB, KD, KF, KI, KO, KOD, KP, KU

LE LOOP END

The LE command terminates the most recent LS command. The axis will loop back and repeat the commands within the loop the number of times specified in the LS command. The loop will start repeating as soon as this command is issued.

NOTE: The LE command in AM mode operates differently than in the AA mode. In AM mode the loops are ending for only those axes that had parameters in the most recent multi-axis command preceding the LE command.

Example: Perform a relative move on axis Z 5 times. After each Z move, wiggle the T axis 20 times.

Enter:

```
AZ;
LS5;
MR5000;
GO;
AT;
LS20;
MR50;
GO;
MR-50;
GO;
LE; (terminates the LS20; command)
AZ;
LE; (terminates the LS5; command)
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
LE;	Single Axis	1	1
LE;	AA-AM	1	1

Related commands: LS

LM LIMIT MODE

Firmware Revision Required: V1.30 and greater

The LM command sets the mode of operation of the limit conditions for the addressed axis or axes. The command takes a single parameter that can either be H, S, or F. A parameter of ‘H’ sets the Hard Limit mode, which causes the affected axis or axes to abruptly halt when a limit condition is encountered. A parameter of ‘S’ sets the Soft Limit mode, which causes the affected axis or axes to ramp to a stop when a limit condition is encountered. In the case of either soft or hard limit modes, the axis queue is flushed when the limit condition is detected and a limit interrupt is generated when the axis stops. Any commands issued after the queue is flushed that move in the opposite direction of the limit will cause motion. A parameter of ‘F’ sets the Limit Off mode, which allows motion to continue when limit conditions occur, however limit conditions are still recognized and reported. See the APP and APB commands for archiving the controller’s parameters in non-volatile FLASH.

LM? may be used to query the current setting.

NOTE: In systems not designed to handle motion beyond the limit switch points, Limit Off mode (LMF) can potentially cause damage to the system and/or persons operating the system. This command should be used with extreme caution.

Limit conditions are treated as critical errors and should not be used as simple positioning inputs to the MAX.

Example: Set up the Y and T axes to stop immediately when a limit switch is encountered.

Enter:
AA;
LM,H,,H;
or
AY;
LMH;
AT;
LMH;

Response: None

Note: In AA or AM modes, a null argument in the parameter list specifies that axis is not to be affected.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
LM[H/S/F/?];	Single Axis	2*	0
LM[H/S/F, H/S/F…];	AA-AM	2*	0

* = per axis

Related commands: IOE, IOK, LT

LO LOAD MOTOR POSITION

The LO command sets the motor position independently of the encoder position unlike LP which sets both to the same supplied value. The LP command will override the LO command and reset the motor position. If the LP command is used and a different motor position value than the encoder position is desired, the LO command must be reentered. Any valid position within the allowable position range may be used.

Example: Set the motor position to 50,000 and the encoder position to 100,000 on the T axis

Enter:

```
AT;
LP100000;
LO50000;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
LO#;	Single Axis	1	1
LO#,#...;	AA-AM	1	1

NOTE: For AA and AM modes that any values in the argument list specify that axis is not to be affected.

Related commands: LP, LPE, RE, RM, RP, RU

LP LOAD MOTOR/ENCODER POSITION

The LP command will load the number supplied as a parameter in the absolute position registers of the axis. In models with the encoder option, the parameter will be loaded into the encoder position register and the parameter times the encoder ratio will be loaded into the position counter. In single axis mode, if no parameter is supplied, the value of zero is used. In AA/AM modes, if no parameter is supplied for an axis, the position is left unchanged.

The LO command can be used after this command to set the motor position independently of the encoder position.

RANGE: Min. Position Value ≤ LP ≤ Max. Position Value

Example: The following would load the X axis position register with 1000, and the Z axis position register with 2000.

Enter:

```
AA;
LP1000,,2000;
```

Response: None

Example: The following would load the Y axis position register with 20,000 and the encoder position register with 30,000 counts, in encoder models.

Enter:

```
AY;
ER3,2;
LP30000;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
LP#;	Single Axis	1*	1*
LP#,#...;	AA - AM	1*	1*

* If the axis is a stepper and encoder or servo axis add 1 to the command queue and 1 to the argument queue.

Related commands: LO, LPE, RE, RM, RP, RU

LPE LOAD ENCODER POSITION

The LPE command sets the encoder position independently of the motor position, unlike LP which sets both to the same supplied value. The LP command will override the LPE command and reset the encoder position. If the LP command is used and a different encoder position value than the motor position is desired, the LPE command must be reentered. Any valid position with the allowable position range may be used.

Example: Set the motor position to 50,000 and the encoder position to 100,000 on the T axis.

Enter:

```
AT;
LO50000;
LPE100000;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
LPE#;	Single Axis	1	1
LPE#,#...;	AA-AM	1	1

Related commands: LO, LP, PE, RE, RM, RP, RU

LS LOOP START

The LS command sets the loop counter for the axis being programmed in the single axis mode and all axes in the AA mode. The command expects a loop counter operand following the command. The commands up to the LE loop terminator will be executed the number of times specified by the operand. Loops may be nested up to four levels deep on each axis. The parameter must be less than 2,147,483,647.

The first loop of commands will occur immediately as they are entered. The remaining loops will be executed after the loop terminator (LE) has been entered.

The axis mode (e.g. A* and AA) must be the same when entering and exiting the loop, otherwise the matching loop termination command will not be found by the board's command processor. The axis mode may be switched within the loop provided the board is in the same mode when the LE command is sent as when the LS command was sent.

If you want one axis to wait for another in the loop, you must be in the AA mode throughout the loop. If you are in the single axis mode in the loop, each axis' commands will go into their separate queues and execute independently of each other.

If, when entering a looping sequence of commands, the command queue is filled before the LE loop terminator is entered, the board will hang. This is because there is no space for the LE command.

RANGE: 1 ≤ LS ≤ 2,147,483,647

Example: Execute a 100,000 count relative move on the Z axis 5 times.

Enter:

```
AZ;
LS5;
MR100000;
GO;
LE;
```

Response: None

NOTE: The first move will occur immediately after entering the GO command. The remaining 4 moves will be executed after the loop terminator LE has been entered.

Example: Execute a 100,000 count move relative on the X axis together with a 100 count move on the T axis, followed by a move absolute to 100 counts on the X axis and 200 counts on the T axis, four times.

NOTE: In AM mode, the LS command operates differently than in the AA mode. In AA mode loops are started for all axes but in the AM mode loops are started only for those axes that had parameters in the most recent multi-axes command preceding the LS command.

Enter:

```
AA;
LS4;
MR100000,,,100;
GO;
MA100,,,200;
GO;
LE;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
LS#;	Single Axis	1	1
LS#;	AA-AM	1	1

Related commands: LE, WH, WS

LT SET LIMIT TRUE STATE

Firmware Revision Required: V1.30 and greater

The LT command sets the active true TTL state of the limit switches on the current axes. The default “true” state of the limits are TTL logic low. This command allows either TTL logic state to be treated as the “true” state. Through the execution of the APP command, limits can be made to default as active high or low on power-up or reset.

LT? can be used to report the current setting.

Example: Set the limit switch high true condition for the X axis.

Enter: AX;
LTH;

Response: None

Example: Set a limit switch true condition for the Z axis as high and T axis as low.

Enter: AA;
LT,,H,L;

Response: None

Note: In AA or AM modes, a null argument in the argument list specifies that axis is not to be affected.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
LT**[**L/H/?**]**;	Single Axis	1	0
LT**[**L/H,L/H,…..**]**;	AA-AM	1	0

Related commands: LM, TL

MA MOVE ABSOLUTE

The MA command will set up one or more axes to move to the absolute positions supplied as parameters. In AA mode, an axis may remain stationary by entering a comma but omitting the parameter. The move is actually initiated by a GO or GD command.

In AA mode all axes begin their move at the same time. Each axis will use its predefined acceleration and velocity values to move to the new absolute position. Each axis may or may not get to the destination at the same time because each axis utilizes individual velocities and accelerations. The MT and VP command will ensure all axes reach their target positions simultaneously.

Example: In the single axis mode, move the X axis to absolute position 100,000 counts with the previously entered acceleration and velocity parameters.

Enter:

```
AX;
MA100000;
GO;
```

Response: None

Example: In the AA mode, move the Y axis to absolute position 10,000 counts and the T axis to absolute position 1,000 counts. The other axes will remain in their current positions.

Enter:

```
AA;
MA,10000,,1000;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MA#;	Single Axis	1*	1*
MA#,#...;	AA-AM	1*	1*

* If the axis is a stepper encoder or servo axis add 1 to both the command and argument queues.

* If the axis is using the cosine acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 1 to the command queue and add 10 to the argument queue.

* If the axis is using the S-curve acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 3 to the Command queue and add 41 to the argument queue.

* If the acceleration and velocity values need to be reset to their AC and VL values because the last move just prior to this move was either a MT or ML move, add the following queue requirements:

QUEUE REQUIREMENTS		
Axis Ramp Type	**COMMAND QUEUE**	**ARGUMENT QUEUE**
Linear (RTL)	4	4
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

Related commands: GD, GN, GO, GS, ML, MR, MT, VP

MD TEMPORARY MACRO DEFINE

MD is used to begin defining a temporary macro. A macro can contain up to 250 characters. Macros 0 through 4 are temporary and they will be erased when the controller is reset or power is turned off. Macros 5 through 24 are stored in non-volatile memory and will be preserved when the controller is reset or powered off. This command cannot be used to define a macro directly into numbers 5 through 24. They must be defined with this command and then moved into the non-volatile space with the PT command.

Enter the macro number immediately after the MD command. The macro number must be between 0 and 4. Next enter the command string, which is made up of up to 250 ASCII characters. After entering the command string for the macro, enter a control Z to end the macro definition. The control Z may be ASCII value 26 or the string “^Z” (carat, shift 6 on a US keyboard, followed by a Z.)

Be careful not to exceed 250 ASCII characters or the size of the axis queue when working with macros.

RANGE: 0 ≤ MD ≤ 4

Example: Define macro 2 to set velocities to 20000 on all axes of a two axes board.

Enter:

```
MD2;
AA;
VL20000,20000;
^Z
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
MD#;	Single Axis	Immediate	
MD#;	AA-AM	Immediate	

Related commands: MX, PM

ML MOVE LINEAR

The ML command uses linear interpolation to perform a straight line relative move. Input parameters are relative distances for each axis involved in the move. The ML command should be followed by a GO or GD to start the axes together. The velocity and acceleration parameters are scaled to allow the axes to move and finish together. All axes are scaled to the axis with the longest move time (the master axis). At the end of the move, all involved velocities and accelerations will be restored to their pre-move values. For programming linear interpolation moves with vector velocity and vector acceleration, see the VP command.

Example: In the AA mode, move the Y, Z and T axes 10000, 100 and 1000 counts respectively with all axes starting and finishing together. The other axes remain in their previous positions.

Enter:

```
AA;
ML,10000,100,1000;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
ML#,#...;	AA-AM	2*	2*

* If this is the master axis for this move and its acceleration and velocity values need to be reset to their original AC and VL values, because a previous move altered them, add the following queue requirements:

If this is not the master axis, then the acceleration and velocity values will be modified, and the following queue requirements will be added:

QUEUE REQUIREMENTS		
AXIS RAMP TYPE	**COMMAND QUEUE**	**ARGUMENT QUEUE**
Linear (RTL)	4	4
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

* If the axis is using the cosine acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 1 to the command queue and 10 to the argument queue:

* If the axis is using the S-curve acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 3 to the command queue and add 41 to the argument queue.

Related commands: GD, GN, GO, MA, MR, MT, VP

MM SET NEGATIVE DIRECTION

The MM command sets the direction logic to move in the negative direction. The direction output for the current axis will change (if necessary) to reflect the new direction. All non-direction-specific move commands will now move in the negative direction.

Example: Set the direction line to move in the negative direction on the Y axis.

Enter:
```
AY;
MM;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
MM;	Single Axis	1	0
-	AA-AM	Not Valid	

Related commands: MO, MP

MO MOVE ONE STEP

The MO command will output one count in the current direction (do not use the GO command). The direction may be reversed directly by use of the MM or MP command or indirectly via a move such as JG or MR. This command generates the output signal in one sample interval and thus eliminates the latency of generating a ramp with an MR1; GO command sequence.

Example: Move the Z axis one pulse in the negative direction.

Enter:

```
AZ;
MM;
MO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
MO;	Single Axis	1	0
-	AA-AM	Not Valid	

Related commands: MM, MP

MP SET POSITIVE DIRECTION

The MP command sets the direction logic to move in the positive direction. The direction output for the current axis will change (if necessary) to reflect the new direction. All subsequent non-direction-specific motion commands will now move in the positive direction.

Example: Set the AX direction bit to positive.

Enter:
```
AX;
MP;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MP;	Single Axis	1	0
-	AA-AM	Not Valid	

Related commands: MM, MO

MR MOVE RELATIVE

The MR command will set up one or more axes to move relative from their current positions at the time the move is executed. In the AA mode, an axis may remain stationary by entering a comma but omitting the parameter. The move is actually initiated by a GO or GD command.

In AA mode all axes will start at the same time. Each axis will use its predefined acceleration and velocity values to move to the new position. Each axis may, or may not, get to the destination at the same time, because each axis utilizes individual velocities and accelerations. To ensure all axes reach their destinations simultaneously use the ML command. For programming linear interpolation moves with vector velocity and vector acceleration, see the VP command.

Example: In the single axis mode, move the X axis 2468 steps in the negative direction.

Enter:

```
AX;
MR-2468;
GO;
```

Response: None

Example: In the AA mode, move the X axis 12345 steps in the positive direction and the Y axis 6789 steps in the positive direction. Both axes will start at the same time.

Enter:

```
AA;
MR12345,6789;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MR#;	Single Axis	1*	1*
MR#,#...;	AA-AM	1*	1*

* If the axis is a stepper encoder or servo axis add 1 to both the command and argument queues.

* If the axis is using the cosine acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 1 to the command queue and add 10 to the argument queue.

* If the axis is using the S-curve acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 3 to the command queue and add 41 to the argument queue.

* If the acceleration and velocity values need to be reset to their AC and VL values because the last move just prior to this move was either a MT or ML move, add the following queue requirements:

QUEUE REQUIREMENTS		
AXIS RAMP TYPE	**COMMAND QUEUE**	**ARGUMENT QUEUE**
Linear (RTL)	4	4
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

Related commands: GD, GN, GO, GS, MA, ML, MT, VP

MT MOVE TO

The MT command uses linear interpolation to move two or more axes to the specified absolute positions. The syntax is similar to the ML command. This command is invalid if loops are being used due to the overhead involved. The command will become valid again after executing an ST or KL command. When used in the contour definition mode, only the axes being used in the contour must be provided for in the MT syntax. A GO or GD command initiates the move.

The velocity and acceleration parameters are scaled to allow the axes to move and finish together. All axes are scaled to the axis with the longest move time (the master axis). At the end of the move, all involved velocities and accelerations will be restored to their pre-move values. For programming linear interpolation moves with vector velocity and vector acceleration, see the VP command.

Example: In the AA mode, move the X, Y and T axes to absolute positions 1000, 10000 and 100 counts, respectively, with each starting and finishing together. The unused axis remains in its previous position.

Enter:

```
AA;
MT1000,10000,,100;
GO;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
MT#,#...;	AA-AM	2*	2*

* If the axis is using the cosine acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 1 to the command queue and 10 to the argument queue:

* If the axis is using the S-curve acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 3 to the command queue and add 41 to the argument queue.

* If this is not the master axis, then the acceleration and velocity values will be modified, and the following queue requirements will be added:

QUEUE REQUIREMENTS		
Axis Ramp Type	**COMMAND QUEUE**	**ARGUMENT QUEUE**
Linear (RTL)	4	4
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

Related commands: GD, GN, GO, MA, ML, MR, VP

MX MACRO EXECUTE

The MX command will execute the command string stored in the specified macro. The macro number that is entered as the argument of the command must be between 0 and 24.

RANGE: 0 ≤ MX ≤ 24

Example: Execute macro number 6.

Enter: MX6;

Response: None

NOTE: MX itself is an immediate command. However, the commands contained within the macro may have queue requirements.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
MX#;	Single Axis	Immediate	
MX#;	AA-AM	Immediate	

Related commands: MD, PM, PT

MAXnet only

#NI NETWORK IP ADDRESS

#NI changes or reports the IP address used by the MAXnet controller for the TCP/IP communications mode. IP addresses are entered using commands in the form of #NIxx.xx.xx.xx; where 0≤ xx ≤ 255. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH. The factory default is 10.40.30.60.

#NI? Can be used to query the current IP setting.

NOTE: A new IP address does not become effective until the IP address is saved in flash, using the APP command, and the controller has been reset.

Example: Change the IP address to 192.168.0.10.

Enter:

```
AX;
#NI192.168.0.10;
APP;
```

Reset power to controller to reestablish the TCP/IP connection using the new IP address

Response: None.

Example: Report the IP address.

Enter:

```
AX;
#NI?
```

Response: =192.168.0.10

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#NI[xx.xx.xx.xx;/?];	Single axis	Immediate	Immediate
#NI[xx.xx.xx.xx;/?];	AA-AM	Immediate	Immediate

Related commands: #NM?, #NP

MAXnet only

#NL ETHERNET COMMUNICATION MODE

The #NL command sets only the Ethernet communication mode. It has two arguments: <flag polling mode> and <NoAck>. If <flag polling mode> is set to 1 the flag message is withheld and appended to the next requested query response. If set to 0 flag messages are sent as soon as they occur. This mode is only needed when the controller is being polled at a heavy rate to prevent package collisions. Ethernet flag polling mode is available starting with V1.48 firmware

If <NoAck> is set to 1, no datastreem character ACK (0x06) or NAK (0x15) is returned when a Ethernet package is received. This reduces Ethernet bandwidth usage if the host application does not use the byte for communication control. Ethernet NoAck mode is available starting with firmware version V1.49.

#NL? Can be used to query the current parameter settings.

NOTE: This command is not archived, must be set at each startup.

Example: Turn on Ethernet communication to flag polling mode.

Enter: #NL1;

Example: Turn off flag polling mode and don't send ACK/NAK messages.

Enter: #NL0,1;

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#NLb[,b];	Single Axis	Immediate	
#NLb[,b];	AA-AM	Immediate	

Related commands: none

MAXnet only

#NM? NETWORK MAC ADDRESS

Reports the network MAC address assigned to this controller.

Example: Report the controller’s factory assigned MAC address.

Enter:
```
AX;
#NM?
```

Response: =00-16-82-00-00-04

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#NM?	Single axis	Immediate	Immediate
#NM?	AA-AM	Immediate	Immediate

Related commands: #NI, #NP

#NP NETWORK PORT NUMBER

#NP changes or reports the Port number used by the MAXnet controller for the TCP/IP communications mode. Port numbers are entered using a command of the form of #NP*n*; where 1 ≤ *n* ≤ 65535. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH. The factory default setting is 23 (Telnet).

#NP? can be used to query the current setting.

NOTE: The new port number does not become effective until the port number is saved in flash, using the APP command, and the controller is reset.

Example: Change the Port number to 49155.

Enter:

```
AX;
#NP49155;
APP;
```

Reset power to controller to reestablish the TCP/IP connection using the new Port number

Response: None.

Example: Report the Port number.

Enter:

```
AX;
#NP?
```

Response: =49155

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#NP*n*;	Single axis	Immediate	Immediate
#NP*n*;	AA-AM	Immediate	Immediate

Related commands: #NI, #NM?

#NU NETWORK UPDATE INTERVAL

#NU changes or reports the MAXnet TCP/IP update interval. This parameter determines the frequency with which the TCP/IP communications hardware is updated. Update intervals, in milliseconds, are entered using a command of the form #NU#; where 1 ≤ # ≤ 100. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH. The factory default is 5.

#NU? can be used to query the current setting.

Example: Change the update interval to 5 milliseconds then archive it.

Enter:
```
AX;
#NU5;
APP;
```

Response: None.

Example: Query the update interval.

Enter:
```
AX;
#NU?
```

Response: =5

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#NU#;	Single axis	Immediate	-
-	AA-AM	Immediate	

Related commands: #NI, #NM?

PA POWER AUTOMATIC

The PA command will turn on or off the auxiliary outputs at the beginning of any Go, Jog, or Home command execution and complement the outputs after the move is executed. The auxiliary will be turned on; i.e. pulled high, upon the execution of the move command and off at the end of that move if the parameter is zero (or not specified in the single axis mode). If the parameter is non-zero, the sense is reversed; i.e. the auxiliary output is turned off (driven low) upon the execution of the move command and on at the end of the move. The move execution commands that affect the state of the auxiliary outputs are:

GD, GN, GO, GS, GU, JF, JG, HM, HR, KM, KR, MV and TM.

The SE command can be used to apply a settling time at the end of each move before complementing the auxiliary bit. This is useful when systems using this mode to control the reduced current mode of stepper motor drives need to retain torque for some specific amount of time before allowing the motor drive to reduce current output.

This mode need only be set once and can be turned off by manually controlling the auxiliary outputs with the AB command or issuing a PAA command. Axes can be selectively affected in the AA mode. The values of the included parameters set the state of the auxiliary line during the move. The following queue requirements apply to each move command in the command stream in the AA and single axis modes. See the PAA command to avoid these queue delays. This mode is off by factory default. See the APP command to preserve the PA settings as the Power up/Reset values.

A kill function (KL, KS) will set the auxiliary output to the non-motion state.

For firmware versions 1.32 or greater: limit, slip, and IOK kill functions will also set the auxiliary output to the non-motion state.

PA? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

Example: Turn on the Y axis auxiliary output at the beginning of a move and turn the T axis output off at the beginning of a move, while in the AA command mode. (note the reversed logic; i.e. 0 = on, 1 = off. "on" pulls the signal line to ground. "off" lets it rise to 5 volts or its pull-up reference voltage.)

Enter:
```
AA;
PA,0,,1;
```

Response: None

NOTE: PA selects the mode immediately but places entries in the axis command queue to set the state of the aux bit to the idle state.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PAb;	Single Axis	1*	0
PAb,b...;	AA-AM	1*	0

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

Related commands: AB, AD, PAA, SE

PAA AUXILIARY AUTOMATIC

Firmware Revision Required: V1.30 and greater

The PAA command is similar to the PA command, except that it avoids all queue delays at the start of the move and at the end of the move when the settling time (SE) is greater than zero. A PAA command will automatically set the PA settings to off.

The PAA command will turn on or off the auxiliary outputs whenever the velocity of the axis is non-zero and complement the outputs when the velocity of the axis is zero. If the parameter is zero (or not specified in single axis mode), the auxiliary will be turned on; i.e. pulled high, when the velocity is non-zero. If the parameter is non-zero, the sense is reversed; i.e. the auxiliary output is turned off (driven low) when the velocity is non-zero.

PAA? can be used to query the current setting.

The SE command can be used to apply a settling time after the velocity reaches zero before complementing the auxiliary bit. This is useful when systems using this mode to control the reduced current mode of stepper motor drives need to retain torque for some specific amount of time before allowing the motor drive to reduce current output. However, if the axis is moved by a subsequent command before the settling time has expired, the move will not be delayed until the settling time has expired as it would with the PA command, and as a result the auxiliary bit will not be complemented.

This mode need only be set once and can be turned off by manually controlling the auxiliary outputs with the AB command or issuing a PA command. Axes can be selectively affected in the AA mode. The values of the included parameters set the state of the auxiliary line during the move. This mode is off by factory default. See the APP command to preserve the PAA settings as the Power up/Reset values.

Example: Turn on the Y axis auxiliary output at the beginning of a move and turn the T axis output off at the beginning of a move, while in the AA command mode. (note the reversed logic; i.e. 0 = on, 1 = off. "on" pulls the signal line to ground. "off" lets it rise to 5 volts or its pull-up reference voltage.)

Enter: AA;
PAA,0,,1;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PAAb;	Single Axis	1	1
PAAb,b...;	AA-AM	1*	1*

* per axis

Related commands: AB, AD, PA, SE

PAV POWER AUTOMATIC AT VELOCITY

The PAV command will turn on or off the auxiliary output during the constant velocity portion of a profile move started with GO, GN or GD command. The auxiliary output is activated when the profile reaches full velocity and the output is complemented when the profile begins deceleration. The auxiliary output will be turned on; i.e. pulled high, if the parameter is zero or not specified. If the parameter is non-zero, the sense is reversed; i.e. the auxiliary output is turned off (driven low) during the constant velocity portion of the move. Execution of the PAV command sets the auxiliary output to the deactivated state. This mode needs only be set once and can be turned off by using the AN, AF, PH, or PL commands. Axes can be selectively affected in the AA mode. The values of the included parameters set the state of the auxiliary line during the move in a similar manner as single axis mode. Axes that do not have a parameter set are unaffected.

PAV? can be used to query the current PAV settings.

This mode is off by factory default. The SE command does not apply to the PAV command. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

Example: Set the Y axis auxiliary output high during the constant velocity portion of Y profile moves and set the T axis auxiliary output low during the constant velocity portion of T axis profile moves (note the reversed logic; i.e. 0 = high, 1 = low. "low" pulls the signal line to ground. "high" lets it rise to 5 volts or its pull-up reference voltage.)

Enter:

```
AA;
PAV,0,,1;
```

Response: None

NOTE: PAV selects the mode immediately but places entries in the axis command queue to set the state of the aux bit to the idle state.

QUEUE REQUIREMENTS		
FORMAT	**MODE**	**REQUIREMENTS**
PAVb;	AX – AT	1
PAVb,b,b,b;	AA-AM	1
-	AA/CD	Not Valid

Related commands ?AB, AF, AN

PC? POSITION CAPTURE ENABLE REPORT

Firmware Revision Required: V1.42 and greater

The PC? command is used to determine if the real-time position capture feature is enabled or disabled. The position capture enable report command takes the form of:

PC?;

The PC? command displays the enable/disable status of the real-time position capture feature for the current axis. This command returns the text string "ON" if the position capture is enabled, and "OFF" if position capture is disabled for the current axis.

Example: Display the position capture enable/disable state for the X axis.

Enter: AX;
PC?;

Response: =ON

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PC?;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: PCA, PCB, PCC, PCE, PCF, PCO, PCQ, PCR, PCS

PCA POSITION CAPTURE ASSIGNMENT

Firmware Revision Required: V1.44 and greater

The PCA command is used to assign a general purpose input bit number and its edge as a position capture event trigger, to the current axis. The position capture assignment command takes the form of:

PCA<bit number>,<edge>;

The bit number argument specifies a general purpose I/O bit, and valid values are in the range of 0 to 17. Bit numbers 0 through 15 are for general purpose input bits. If the specified bit number is not an input bit, then an error response is returned. The IO command can be used to define a general purpose I/O bit as either an input or an output. A bit number value of 16 specifies the home switch input. See the HT command for defining the sense of the home switch. A bit number value of 17 specifies the encoder home event. See the EH command for details on defining the encoder home event.

The valid values for the edge argument are a 0, 1, 2, or 3. A value of 0 will disable real-time position capture, and values 1 through 3 will enable real- time position capture for the specified bit number. A value of 1 assigns a falling or negative edge to the specified input bit. A value of 2 assigns a rising or positive edge to the specified input bit. A value of 3 assigns both the rising and falling edges to the specified input bit.

The HM and HR commands use the real-time position capture feature, and temporarily disable PCA assignments when the HM and HR commands are active. Once the home position is reached, the PCA assignments are restored. Therefore, real-time position capture data is not collected while the HM and HR commands are active.

If a PCA command entry results in an error response, then make sure the specified general purpose input bit number is defined as an input, and see the IO command for details.

PCA? can be entered to display the current real-time position capture assignments for the current axis. PCA<bit number>? can be entered to display the current real-time position capture assignment for the specified bit number.

Example: Assign general purpose input bit 1 as a trigger event on its negative edge to the Y axis.

Enter: AY;
PCA1,1;

Response: None.

Example: Assign general purpose input bit 0 as a trigger event on both positive and negative edges to the Y axis.

Enter: AY;
PCA0,3;

Response: None.

Example: Display the general purpose input bit trigger event assignments for the Y axis.

Enter:
```
AY;
PCA?;
```

Response:
```
=3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PCA#,#;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: EH, HM, HR, HT, IO, PC?, PCB, PCC, PCE, PCF, PCO, PCQ, PCR, PCS

PCB POSITION CAPTURE BOUNDARY ASSIGNMENT

Firmware Revision Required: V1.42 and greater

The PCB command is used to further qualify a general purpose input bit trigger event with a position range. By adding a boundary assignment, the input bit will not trigger a capture event when the encoder position is outside the boundary range. The position capture boundary assignment command takes the form of:

PCB<bit number>,<low boundary>,<high boundary>;

This assignment applies to the current axis only, and position capture must be disabled in order to make an assignment. Setting both the low and high boundaries to a value of zero disables the boundary assignment for the specified general purpose input bit number. The bit number may be specified as a '*' character in order for the assignment to apply to all of the general purpose input bits. Specifying a '?' character as the bit number will cause the capture range assignments for the general purpose input bits to be displayed for the current axis. This command is restricted to the general purpose input bit numbers, which are 0 through 15.

Example: Assign general purpose input bit 1 to the X axis with a capture range of 3500 to 4500.

Enter:
AX;
PCB1,3500,4500;

Response: None.

Example: Display the general purpose input bit trigger event boundary assignments for the X axis.

Enter:
AX;
PCB?;

Response:
PCB0=0-0
PCB1=3500-4500
PCB2=0-0
PCB3=0-0

Example: Assign the same trigger event range of 5000-6000 to all of the general purpose input bits for the Y axis.

Enter:
AY;
PCB*,5000,6000;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PCB#,#,#;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: IO, PC?, PCA, PCC, PCE, PCF, PCO, PCQ, PCR, PCS

PCC POSITION CAPTURE TABLE CLEAR

Firmware Revision Required: V1.42 and greater

The PCC command is used to clear the content of the real-time position capture table. The position capture table clear command takes the form of:

PCC;

The PCC command clears all entries in the capture table, and resets the input and output queue pointers for the table.

Example: Clear the contents of the position capture table.

Enter: PCC;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PCC;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: PC?, PCA, PCB, PCE, PCF, PCO, PCQ, PCR, PCS

PCE POSITION CAPTURE ENTRY REPORT

Firmware Revision Required: V1.42 and greater

The PCE command is used to display the specified number of entries from the capture table. The position capture table entry report command takes the form of:

PCE<entry count>;

If the entry count is not specified, then a single entry from the capture table is displayed. The position capture tables are implemented as ring buffers, with 800 entries. This command advances the output queue pointer by the number of entries displayed, and reduces the number of filled entries by the number of entries displayed. If the specified entry count is greater than the number of filled entries in the capture table, then only the filled table entries will be displayed. For details on the values displayed, see the PCQ command.

Example: Display two entries from the position capture table.

Enter: PCE2;

Response:
```
=X,259537,1,0,0
=X,269537,1,0,0
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PCE#;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: PC?, PCA, PCB, PCC, PCF, PCO, PCQ, PCR, PCS

PCF POSITION CAPTURE OFF

Firmware Revision Required: V1.42 and greater

The PCF command is used to disable the real-time position capture feature. The position capture off command takes the form of:

PCF;

The PCF command disables the position capture feature for the current axis.

Example: Disable the position capture feature for the X axis.

Enter: AX;
PCF;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PCF;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: PC?, PCA, PCB, PCC, PCE, PCO, PCQ, PCR, PCS

PCO POSITION CAPTURE ON

Firmware Revision Required: V1.42 and greater

The PCO command is used to enable position capture for the current axis. The position capture on command takes the form of:

PCO;

Position capture can only be enabled if the current axis has an incremental encoder, and a general purpose input bit has been configured for the current axis with the PCA command. See the ECI command for configuring incremental encoders.

Example: Enable the position capture feature for the X axis.

Enter: AX;
PCO;
Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PCO;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: ECI, PC?, PCA, PCB, PCC, PCE, PCF, PCQ, PCR, PCS

PCQ POSITION CAPTURE TABLE QUERY

Firmware Revision Required: V1.42 and greater

The PCQ command is used to display the data collected by the real-time position capture feature. The position capture table query report command takes the form of:

PCQ;

The PCQ command displays all of the filled entries of the position capture table. The first value displayed is the axis associated with the capture event. The second value displayed is the encoder position when the capture event occurred, in decimal. The third, fourth, and fifth values displayed are all hexadecimal. The third value displayed is the positive edge home switch (0x01), the encoder home (0x02), and negative edge home switch (0x04). The fourth value displayed is the positive edge general purpose input bits. The fifth value displayed is the negative edge general purpose input bits that triggered the capture event. If a given bits value is a one, then that bit caused the capture event. The position capture table is implemented as a ring buffer, with 800 entries. This command does not modify the queue insert and removal indices.

Example: Display the filled entries in the position capture table.

Enter: PCQ;

Response:

```
=X,279537,1,0,0
=X,289537,1,0,0
=X,299537,1,0,0
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PCQ;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: PC?, PCA, PCB, PCC, PCE, PCF, PCO, PCR, PCS

PCR POSITION CAPTURE REMOVE TRIGGER

Firmware Revision Required: V1.42 and greater

The PCR command is used to remove the assignment of the general purpose input bit as a position capture event trigger for the current axis. The position capture remove trigger command takes the form of:

PCR<bit number>;

This command also disables the boundary range for this general purpose input bit. Position capture must be disabled for the current axis in order to remove an assignment. The bit number may be specified as a '*' character in order for the command to apply to all of the general purpose input bits. This command removes assignments made to the general purpose input bit by the PCA and the PCB commands. The position capture feature must be disabled in order to execute this command.

Example: Remove position capture assignments for general purpose input bit 2 for the X axis.

Enter: AX;
PCR2;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PCR#;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: PC?, PCA, PCB, PCC, PCE, PCF, PCO, PCQ, PCS

PCS POSITION CAPTURE TABLE SIZE REPORT

Firmware Revision Required: V1.42 and greater

The position capture table size report command takes the form of:

PCS;

The PCS command reports the number of entries that have been filled in the capture table.

Example: Display the number of filled entries in the position capture table.

Enter: PCS;

Response: =15

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PCS;	Single Axis	Immediate	
-	AA-AM	Immediate	

Related commands: PC?, PCA, PCB, PCC, PCE, PCF, PCO, PCQ, PCR

PE REPORT ENCODER POSITIONS

The PE command reports the encoder positions of all encoder and/or servo axes. All encoder positions will be reported even in single axis mode. (This is the same as AA;RE;.)

Example: Report the encoder positions of an eight axis servo board.

Enter: PE;

Response: 0,50,156,0,0,1506,0,0<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PE;	Single Axis	Immediate	
PE;	AA-AM	Immediate	

Related commands: PP, RE, RP

PM PRINT MACRO

The PM command will return the command string stored in the specified macro number as a command response. The macro number entered as the argument for this command must be between 0 and 24.

RANGE: 0 ≤ PM ≤ 24

Example: Print the command string contained in macro 19.

Enter: PM19;

Response: If a macro string is defined for macro 19, the macro string will be the response. If no macro is defined there will be no response.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PM#;	Single Axis	Immediate	
PM#;	AA-AM	Immediate	

NOTE: Macros are stored as ASCII character strings. If <LF> character is used as command terminator it will be sent back to the host computer by the PM command. If the application software stops reading a character string first it will appear that the PM command did not return the macro contents. To avoid this issue, save macros without <LF> terminator. Use semi-colons instead to terminate commands

Related commands: MD, MX, PT

PP REPORT MOTOR POSITIONS

The PP command reports the motor positions of all axes in ASCII format. All axes will be reported even in single axis mode. (This is the same as AA; RP)

Example: Report the motor positions of an eight axis controller.

Enter: PP;

Response: 0,0,0,125,0,200,0,565<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PP;	Single Axis	Immediate	
PP;	AA-AM	Immediate	

Related commands: PE, RE, RP

PS CONFIGURE CURRENT AXIS

The PS command specifies that the motor type for this axis is a stepper motor with encoder feedback (PSE), servo (PSM), stepper without encoder (PSO), or piezo servo (PSZ). To preserve this setting after MAX is powered down, use APP to archive this setting in flash memory.

The piezo servo motor selection (PSZ) (Firmware Revision Required: V1.48 and greater) uses positive voltages only for the servo output, and the motor direction is controlled by the direction output as the clockwise input to the piezo drive, and the auxiliary output as the counterclockwise input to the piezo drive. Because the auxiliary output is used for direction control, the auxiliary control commands are not available when PSZ has been selected.

PS? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

Example: Set up the X axis to be a stepper with encoder feedback.

Enter: AX;
PSE;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PS**[**O/E/M/Z/?**]**;	Single Axis	Immediate	
PS**[**O/E/M/Z,O/E/M/Z,….;**]**	AA-AM	Immediate	

Related commands: None

PT PRESERVE A TEMPORARY MACRO

Use PT to save a temporary macro permanently by copying it to non-volatile memory. The temporary macro number, which is entered as an argument for this command, must be between 0 and 4. The non-volatile macro number, which is also entered as an argument for this command, must be between 5 and 24.

RANGE:
0 ≤ Parameter 1 ≤ 4
5 ≤ Parameter 2 ≤ 24

Example: Copy temporary macro 3 to non-volatile macro 19.

Enter: PT3,19;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**Min**	**Max**
PT#,#;	Single Axis	Immediate	
PT#,#;	AA-AM	Immediate	

Related commands: MD, MX, PM

QA QUERY AXIS STATUS

The QA command returns the status of the single addressed axis like the RA command except the limit and done flags will not be cleared. Refer to the RA command for details.

Example: Check the status of the X axis.

Enter:
```
AX;
QA;
```

Response: PNNH<LF>

Refer to the table "Character Meaning" in the RA command.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
QA;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: EA, QI, RA, RI

QI QUERY INTERRUPT STATUS

The QI command returns the same information as the QA command but for all axes at once. The 4 character fields for each axis are separated by commas. The state of the status flags for all axes without clearing the controller's copy of the done flags.

Example: Check the status of an eight axis board.

Enter: QI;

Response: PNNN,MNNN,PDNN,MNLN,PDNN,MNNN,MNNN,MNNN<LF>

Refer to the table "Character Meaning" in the RA command.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
QI;	Single Axis	Immediate	
QI;	AA-AM	Immediate	

Related commands: EA, QA, RA, RI

QL QUERY ALL LIMIT SENSORS

Firmware Revision Required: V1.21 and greater

QL is an immediate command and is valid in single axes, AA, and AM modes of operation.

The response of the QL command is a hexadecimal value representing the state of the response of the positive and negative limit sensors of each axis. A limit sensor in the TTL "High" state will have a value of ONE (1) and a limit sensor in a TLL "Low" state will have value of Zero (0). The order of the limit sensors is as follows:

[1]02000000	K Axis Positive Limit	[1]00020000	K Axis Negative Limit
[1]01000000	W Axis Positive Limit	[1]00010000	W Axis Negative Limit
8000	S Axis Positive Limit	0080	S Axis Negative Limit
4000	R Axis Positive Limit	0040	R Axis Negative Limit
2000	V Axis Positive Limit	0020	V Axis Negative Limit
1000	U Axis Positive Limit	0010	U Axis Negative Limit
0800	T Axis Positive Limit	0008	T Axis Negative Limit
0400	Z Axis Positive Limit	0004	Z Axis Negative Limit
0200	Y Axis Positive Limit	0002	Y Axis Negative Limit
0100	X Axis Positive Limit	0001	X Axis Negative Limit

[1] All boards with more than 8 axes show a 8-digit response!

Example: Query all limit sensors on an 8 axes controller.

Enter: QL;

Response: A500<LF> (This means that the S, V, Z, and X positive limit sensors are high and the R, U, T, and Y positive limit sensors are low. All negative limit sensors are low.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
QL;	Single Axis	Immediate	
QL:	AA-AM	Immediate	

Related commands: None

RA REPORT AXIS STATUS

The RA command returns the state of the limit and home switches and the done and direction flags for the currently addressed axis. The done flag register will be reset by this command.

See RI command for single axis mode.

The status is returned in the following format:

CHARACTER MEANING		
CHAR	SENT	DESCRIPTION
1	P	Moving in positive direction
	M	Moving in negative direction
2	D	Done (ID, II or IN command has been executed, set to N by this command or IC command)
	N	No ID executed yet
3	L	Axis in overtravel. Char 1 tells which direction. Set to N when limit switch is not active.
	N	Not in overtravel in this direction
4	H	Home switch active. Set to N when home switch is not active.
	N	Home switch not active
5	LF	Line feed

Example: The Y axis just encountered a limit, verify its status.

Enter: AY;
RA;

Response: PNLN<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RA;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: EA, QA, QI, RI

RC REPORT INSTANTANEOUS ACCELERATION

The RC command will return the current instantaneous acceleration rate of the current axis. This may differ from the programmed acceleration if a ramp is being generated, that is, when the axis is accelerating or decelerating. When the stage is stopped, the parameter returned will be zero (0). When the stage is running at programmed speed; i.e. not accelerating, the parameter returned will be zero (0).

Example: Display current acceleration values for all axes on an eight axis board.

Enter:

```
AA;
RC;
```

Response:

```
2000000, 2000000, 2000000, 2000000, 2000000, 2000000, 2000000,
2000000<LF>
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RC;	Single Axis	Immediate	
RC;	AA-AM	Immediate	

Related commands: AC, RV, VRC

RDB RESTORE DEFAULT PARAMETER IN BACK-UP

The RDB command restores the parameters from the “backup” archive in flash memory.

Example: Restore the currently assigned parameters from the back up archive

Enter: RDB;

Response: None.

Note: This command places entries in all axis command queues to set up the motion profile parameters.

QUEUE REQUIREMENTS				
FORMAT	MODE	AXIS RAMP TYPE	COMMAND	ARGUMENT
RDB;	Single Axis	Linear Acceleration	5	6
RDB;	AA-AM	Linear Acceleration	5	6
RDB;	Single Axis	Short Parabolic Acceleration	5	10
RDB;	AA-AM	Short Parabolic Acceleration	5	10
RDB;	Single Axis	Standard Parabolic Acceleration	5	24
RDB;	AA-AM	Standard Parabolic Acceleration	5	24
RDB;	Single Axis	Cosine Acceleration	5	24
RDB;	AA-AM	Cosine Acceleration	5	24
RDB;	Single Axis	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDB;	AA-AM	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDB;	Single Axis	S-curve Acceleration	10	107
RDB;	AA-AM	S-curve Acceleration	10	107

Related commands: APB, APP, RDF, RDP

RDF RESTORE FACTORY DEFAULT VALUES

RDF restores the current parameter set to the factory default values. This process does not alter the parameters stored in flash via the APP command. To restore the flash memory to factory default, the RDF command must be issued followed by the APP command.

NOTE: The RDF command does not restore communication parameters (IP address, port number, and serial baud rate) to factory default.

Example: Assign the current parameter set to be the factory default values.

Enter: RDF;

Response: None.

QUEUE REQUIREMENTS				
FORMAT	**MODE**	**AXIS RAMP TYPE**	**COMMAND**	**ARGUMENT**
RDF;	Single Axis	Linear Acceleration	5	6
RDF;	AA-AM	Linear Acceleration	5	6
RDF;	Single Axis	Short Parabolic Acceleration	5	10
RDF;	AA-AM	Short Parabolic Acceleration	5	10
RDF;	Single Axis	Standard Parabolic Acceleration	5	24
RDF;	AA-AM	Standard Parabolic Acceleration	5	24
RDF;	Single Axis	Cosine Acceleration	5	24
RDF;	AA-AM	Cosine Acceleration	5	24
RDF;	Single Axis	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDF;	AA-AM	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDF;	Single Axis	S-curve Acceleration	10	107
RDF;	AA-AM	S-curve Acceleration	10	107

FACTORY DEFAULT SETTINGS
Motor type is open-loop stepper (PSO)
Axis direction bit output state is normal(DBN)
Over travel limits are active low(LTL)
Home inputs are active low(HTL)
I/O direction (BDFF00)
Limits are set hard (LMH)
Acceleration is 2000000 (AC2000000)
Use linear acceleration ramps (RTL)
Early deceleration factor is zero
Output Reset State (BR11111111--------)
Velocity is 200000 (VL200000)
Base Velocities zero(VB0)
Analog Input zero offset (AO0,0)
User units are off(UF)
Auxiliary output bits settle times are zero(SE0)
Auxiliary output bits power control is disabled(AB0)
Auxiliary output bit power up states are high(ADH)
Backlash Compensation (BC0)
Software over travel limits are disabled (TL0,0;)
Stand-alone mode disabled (SXM0)
Macro links are cleared (SXC)
Motion update rate 1024 Hz (#UR1024)
User Units disabled (UF)

ENCODER MODELS
Motor/encoder ratio is one to one(ER1,1)
Encoder slip tolerance is one(ES1)
Position maintenance dead band is zero(HD0)
Position maintenance velocity limit is zero(HV0)
Position maintenance hold gain is one(HG1)
Encoder Home pattern (EH101 resp. EH0101)
Encoder type is set for incremental encoder (ECI)
Encoder direction is set to normal (EDN)
Encoder home delay set to zero (EHD0)
Encoder slip kill is disabled (SK0)

SERVO MODELS
PID output is bipolar(SVB1)
PID output polarity is normal(SVP+)
PID proportional gain is 10 (KP10.00)
PID derivative gain coefficient is 160 (KD160)
PID integral gain coefficient is 1.0 (KI1.0)
PID acceleration feed-forward is zero(KA0.00)
PID velocity feed-forward is zero(KV0.00)
PID offset coefficient is zero(KO0)
PID maximum output is 10 (KB10.0)
PID friction coefficient is zero (KF0)
PID integration sum upper limit is 200 (KU200)
PID output dead band comp. offsets (KOD0,0,1)

Related commands: APB, APP, RDB, RDP

RDP RESTORE POWER-UP DEFAULT VALUES

The RDP command restores the motion parameters using power-up defaults.

Example: Restore the power-up default parameters from flash memory.

Enter: RDP;

Response: None.

NOTE: This command places entries in all axis command queues to set up the motion profile parameters.

QUEUE REQUIREMENTS				
FORMAT	**MODE**	**AXIS RAMP TYPE**	**COMMAND**	**ARGUMENT**
RDP;	Single axis	Linear Acceleration	5	6
RDP;	AA-AM	Linear Acceleration	5	6
RDP;	Single Axis	Short Parabolic Acceleration	5	10
RDP;	AA-AM	Short Parabolic Acceleration	5	10
RDP;	Single Axis	Standard Parabolic Acceleration	5	24
RDP;	AA-AM	Standard Parabolic Acceleration	5	24
RDP;	Single Axis	Cosine Acceleration	5	24
RDP;	AA-AM	Cosine Acceleration	5	24
RDP;	Single Axis	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDP;	AA-AM	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDP;	Single Axis	S-curve Acceleration	10	107
RDP;	AA-AM	S-curve Acceleration	10	107

Related commands: APB, APP, RDB, RDF

RE REPORT ENCODER POSITION

The RE command returns the current encoder position of the currently addressed axis or axes in encoder counts.

Example: Examine the current encoder position of the Y axis.

Enter:

```
AY;
RE;
```

Response: 12345<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RE;	Single Axis	Immediate	
RE;	AA-AM	Immediate	

Related commands: PE, PP, RP

RI REPORT AXES STATUS

The RI command is a multi-axis mode command that returns the same status information for all axes as the RA command does in single axis mode. The 4 character fields for each axis are separated by commas. The done flag is reset by this command as it would be via the RA command.

Example: Check the status of an 8 axis board.

Enter: RI;

Response: MDNN,MDNN,PNLN,PNNN,PNLN,PNNN,PNNN,PNNN,<LF>

Refer to the table "Character Meaning" in the RA command.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
RI;	Single Axis	Immediate	
RI;	AA-AM	Immediate	

Related commands: EA, QA, QI, RA

RL REPORT SLIP STATUS

The RL command returns the slip detection status of all axes. S is returned if a slip condition has occurred for that axis, or else an N is returned. The number of characters returned corresponds to the number of axes available on the board. Open-loop stepper always returns “n” and servo axes always returns “N” in its RL response. The RL command resets slip conditions for the axes that have slipped.

Example: On an eight axis board, see if any axis has slipped.

Enter: RL;

Response: NNSNNNNN<LF> (The Z axis has slipped.)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RL;	Single Axis	Immediate	
RL;	AA-AM	Immediate	

Related commands: ES, IS

RM RESET MODULO POSITION

The RM command divides the axis position counter by the parameter supplied and replaces the axis position counter with the resulting remainder. For an axis with encoder feedback, the encoder position is also adjusted. For encoder feedback axes with an encoder ratio defined, the parameter supplied should be in encoder count units, and the encoder position will be adjusted with the supplied parameter directly; the motor position will be adjusted with the supplied parameter multiplied by the encoder ratio factor.

This command can be used in applications where there is a continuously rotating axis that will eventually overflow the position register unless the position is reset to be within the specified range, or it is desired that an axis position stay within an absolute bounded range, such as a rotary axis where the position is desired to stay within the range of 0 to 360 degrees.

RANGE: 1 < modulo divisor ≤ 100,000,000

NOTE: The sign of the parameter is ignored and always treated as a positive number.

NOTE: If User Units are active then the supplied parameter times the User Units factor must be within the specified range. If an encoder ratio is active, then the supplied parameter times the encoder ratio must be with the specified range.

Example: The current position of a rotating stage with a full-revolution count of 6000 is needed. Since this stage has been rotated several times without regard for the position, the position counter has reached 163,279. Send an RM6000; command to set the position relative to the full position count. Then send an RP command to find out what the real position of the axis is.

Enter:
(Current position is 163,279)
RM6000;
(Current position is now 1,279)
RP:

Response: 1279

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
RM#;	Single Axis	1	1
RM#;	AA-AM	1*	1*

* = per axis

Related commands: LO, LP, PE, PP, RE, RMC, RP, RU

RMC RESET MODULO POSITION CONTINUOUS

The RMC command sets the modulo divisor that is used to continuously monitor the axis position to keep it within the range of 0 ≤ position < modulo divisor. It accomplishes this by dividing the axis position counter by the modulo divisor and replacing the axis position counter with the resulting remainder. For an axis with encoder feedback, the encoder position is also adjusted. For encoder feedback axes with an encoder ratio defined, the parameter supplied should be in encoder count units, and the encoder position will be adjusted with the supplied modulo divisor parameter directly; the motor position will be adjusted with the modulo divisor multiplied by the encoder ratio factor. Using a parameter of zero (0) will disable the function.

Whenever this function is active for an axis, the position reporting commands (PE, PP, RE, RP, RU) will always report position values within the range from zero to one less than the modulo divisor. Position data stored in shared memory is also always within the same range. When this function is active and then disabled with the RMC0; command the axis position will still be within the range 0 up to the modulo divisor, however, future adjustments to the keep the positions within this range will no longer be made.

This command can be used in applications where there is a continuously rotating axis that will eventually overflow the position register unless the position is reset to be within the specified range, or it is desired that an axis position stay within an absolute bounded range, such as a rotary axis where the position is desired to stay within the range of 0 to 360 degrees. It may be helpful to use relative, as opposed to absolute, command coordinates when programming target endpoints for an axis with RMC active.

RANGE: 0 ≤ modulo divisor ≤ 100,000,000

NOTE: The sign of the parameter is ignored and always treated as a positive number.

NOTE: If User Units are active then the supplied parameter times the User Units factor must be within the specified range. If an encoder ratio is active, the modulo divisor times the encoder ratio must be within the specified range.

Example: Send an RMC6000; command to set the modulo position divisor. Move Relative 15,500 counts. When the move is complete send an RP command to find out what the real position of the axis is.

Enter:
RMC6000;
MR15500;
GO;
ID;
…wait until done… any RP commands issued prior to done would report a position in the range of 0 to 5,999
RP:

Response: 3500

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RMC#;	Single Axis	1	1
RMC#;	AA-AM	1*	1*

* = per axis

Related commands: LO, LP, PE, PP, RE, RM, RP, RU

RP REPORT POSITION

The RP command returns the current position of the currently addressed axis in single axis mode or all positions separated by commas in AA or AM mode. The position will be returned to the host in ASCII format. This command is not queued; i.e. the current position will be returned immediately even if the axis is in motion.

Example: The current position on the Y axis is 12345. Use the RP command to verify the position.

Enter:
```
AY;
RP;
```

Response:
```
12345<LF>
```

Example: Verify the positions of all axes (on an eight axis board).

Enter:
```
AA;
RP;
```

Response:
```
100,200,300,400,500,600,700,800<LF>
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
RP;	Single Axis	Immediate	
RP;	AA-AM	Immediate	

Related commands: RE, PE, PP

RQC REPORT AXIS COMMAND QUEUE

The RQC command returns the number of available entries in the axis command queue.

Max Value = 2559

Example: Report the command queue space remaining for all axes (on a four axes controller).

Enter: AA;
RQC;

Response: 2559,2559,2559,2559<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
RQC;	Single Axis	Immediate	
RQC;	AA-AM	Immediate	

Related commands: None

RR REPORT ABSOLUTE ENCODER REFERENCE POSITION

The RR command reports the current count register reading of an absolute encoder.

In single axis mode RR; reports the reading of the currently addressed absolute encoder. In AA or AM mode RR; reports the current readings of all absolute encoders.

Example: Report the the current absolute encoder position of axis X.

Enter:

```
AX;
RR
```

Response: 100

Example: Report current absolute encoder positions for X, Z & U (Y & T are not configured as absolute encoders).

Enter:

```
AM;
RR
```

Response: 100,,300,,500

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
RR	Single Axis	Immediate	
RR	AA-AM	Immediate	

Related commands: #AE, ECA, ECB, ECI

#RR REPORT RAW BiSS ABSOLUTE ENCODER

Firmware Revision Required: V1.49 and greater

The #RR command reports the current raw BiSS absolute encoder reading in hex with the CRCok-bit appended. The reported hex value consists of the <encoder data>, <status bits> and a <CRC ok> bit. It does not include any Ack, Busy, Start, CDS or CRC checksum bits.

In single axis mode #RR; reports the reading of the currently addressed BiSS absolute encoder reading in hex with a CRCok bit appended. In AA or AM mode #RR; reports the current readings in hex of all BiSS absolute encoders with CRCok bit appended.
The value is random with CRCok=0, if no start sequence is detected in the BiSS data stream.

Example: Get the last raw BiSS encoder reading of axis X configured with ECB36,4000000,2,43,00;.

Enter: AX
#RR

Response: *5776D2A7*
The response breakdown is as follows:
Position: 0xAEEDA54 = 183425620; Status bits: 0x3; CRCok: 1.

Example: On a 5 axis controller get the last raw BiSS encoder reading of all axes, where X is configured with ECB36,4000000,2,43,00; and axis Z is configured with ECB26,2000000,1,43,00;.

Enter: AM
#RR

Response: 1964D2EF,,27BB8F,,
The response breakdown is as follows: X: pos 0x32C9A5D, status 0x3, CRCok; Z: pos 0x9EEE3, status 0x1, CRCok; Axes Y, T and U are not BiSS encoders

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#RR	Single Axis	Immediate	
#RR	AA-AM	Immediate	

Related commands: #AE, ECB, EC?, RR

RT RAMP TYPE SELECT

Firmware Revision Required: V1.30 and greater

The RT command selects a linear or previously defined ramp profile for use with a currently active axis. The alpha parameter following the RT command select the ramp type. L specifies linear and takes no further parameters. S specifies S-curve and takes an additional numeric parameter to specify one of 8 possible previously defined S-curve profiles. U specifies User-defined and takes an additional numeric parameter to specify one of 8 possible User-defined profiles. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

RT? can be used to query the current setting in single and multi-axis mode.

Example: Select linear ramp for the Y axis and S-curve ramp number 1 for use with Z axis.

Enter:

```
AJ1,.25;
AY;
RTL;
AZ;
RTS1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
RT[L/S#/U#/?];	Single Axis	15	5 + (2 x number of segments in ramp definition)
RT?;	AA-AM	Query only	

Related commands: AJ, DAB, DAE, DAR

RU REPORT POSITION IN USER UNITS

The RU command returns the current position in user units (see UU command). The format of the response is a floating-point number.

Example: One revolution of a motor is 2000 steps. Define user units so moves can be referenced in revolutions. Move the Z axis 3 1/2 revolutions. Use RU to display the position when the move is complete.

Enter:

```
AZ;
UU2000;
LP0;
MR3.5;
GO;       (Wait until move is complete.)
RU;
```

Response: 3.50000<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RU;	Single Axis	Immediate	
RU;	AA-AM	Immediate	

Related commands: RE, RP, UU

RV REPORT INSTANTANEOUS VELOCITY

The RV command will return the current instantaneous velocity at which the axis is moving. This may differ from the programmed maximum velocity if the axis is accelerating or decelerating or stopped. If the JF command is executing, the command only reports the integer part of the velocity.

Example: Jog the Y axis at 12345 steps per second. Display the current velocity.

Enter:
AY;
JG12345;
RV;

Response: 12345<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RV;	Single Axis	Immediate	
RV;	AA-AM	Immediate	

Related commands: FV, RC, VL, VRV

SA STOP ALL

The SA command flushes all queues and causes all axes to decelerate to a stop at the rate previously specified in an AC command. All status and position information is retained. Even when executed in a single axis mode, this command will cause all axes to stop.

Example: Send all axes on a move, then ramp them to a stop before they finish.

Enter:

```
AA;
VL100,100,100,100,100,100,100,100;
MR1000,2000,3000,4000,3000,2000,1000,4000;
GO                                                    (wait awhile)
SA;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SA;	Single Axis	1	0
SA;	AA-AM	1	0

Related commands: KL, KS, SD, SI, SO, ST

MAXnet only

SB SET BAUD RATE

The SB command sets or reports the controller's serial communications baud rate. Valid baud rates include 9600, 19200, 38400, 57600 and 115200. Factory default baud rate is 9600.

SB? can be used to query the current setting.

NOTES:

1) A baud rate does not become effective until it is saved in flash, using the APP command, and the controller is power cycled off then on.
2) The MAXnet controller uses a CTS/RTS hardware handshake for RS232 flow control.

Example: Set the serial communications baud rate to 19200 baud.

Enter:

```
SB19200;
APP;
```

Reset power to controller to reestablish the connection using the new baud rate

Response: None.

Example: Query the serial communications baud rate.

Enter: SB?

Response: =19200

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SB**[#/?]**;	Single axis	Immediate	Immediate
SB**[#/?]**;	AA - AM	Immediate	Immediate

Related commands: None

SD STOP AND RESET DONE

The SD command may be substituted for the SA command. It will reset the done flags for all axes, stop all axes at the rates previously specified via the AC command, and then flush all axis command queues. This allows the host to be interrupted when all axes have stopped by using the ID command after the SD. The SA; ID combination may flag the completion early if one of the axes is already done from a previously executed ID.

Example: Stop all axes and reset all done flags. When all axes have stopped set all done flags.

Enter:

```
AA;
SD;
ID;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SD;	Single Axis	2*	0*
SD;	AA-AM	2*	0*

* If PA (power automatic) mode is active add 1 to the command queue.

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

Related commands: KL, KS, PAA, SA, SI, SO, ST

SE AUX POWER AUTOMATIC SETTLING TIME

The SE command allows specification of a settling time, in milliseconds, to be used before the auxiliary output is complemented when using PA mode. The parameter may be any value up to 1000 milliseconds. Specification of a parameter of zero disables SE mode.

The factory default settling time is zero. See the APP command to preserve the SE settings as the Power up/Reset values.

SE? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE: 0 ≤ SE ≤ 1000

Example: Turn on the Z axis auxiliary output upon execution of a move and have it remain on for 500 milliseconds after the move is complete.

Enter:

```
AZ;
PA;
SE500;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SE#;	Single Axis	Immediate	
SE#,#...;	AA-AM	Immediate	

Related commands: PA, PAA

SEA SYNCHRONIZE ENCODER POSITION WITH ABSOLUTE ENCODER POSITION

The SEA command synchronizes an encoder position with the absolute encoder by setting the encoder position equal to the absolute encoder position. An optional offset parameter value can be entered which will be added to the absolute encoder reading when setting the encoder position. If no value is specified for the offset, it will default to a value of zero. In AA or AM modes, the offset parameter is required and should be set to a value of zero if the offset is not to be used.

The SEA and SMEA commands share the offset parameter.

The absolute encoder synchronization mode (SEA or SMEA) and the offset parameter value can be preserved as power up/reset values with the APP command.

RANGE: -2147483647 <= offset <= 2147483647

Example: The following would set the X axis encoder position to the current X axis absolute encoder position plus an offset of -30.

Enter:
AX;
SEA-30;

Response: None

Example: The following would synchronize the encoder positions of the Y & T axes with their respective absolute encoder positions.

Enter:
AA;
SEA,0,,0;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SEA{#};	Single Axis	1	1
SEA#,#...;	AA-AM	1	1

Related commands: SMEA

SI STOP INDIVIDUAL

This command can be used to stop only certain axes. In a single axis mode, the SI command behaves identically to ST. In a multi-axis mode, however, SI can be used to stop any number of axes and can be used in place of SA. Like SA, SI will ramp those axes to be stopped using the rate previously specified via the AC command. This command is useful for stopping a specific axis when the current axis mode is unknown and for stopping several axes without affecting current motion on other axes. This command is not appropriate during VC and VP moves.

Each parameter represents an axis from X through S. Any non-zero value in a parameter will cause the corresponding axis to be stopped.

Example: Start a motion on all four axes. When input bit 1 becomes true, stop axes Y and T without affecting X and Z.

Enter:

```
AM;
MR15000,30000,20000,40000;
GO;
SW1;
SI,1,,1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SI;	Single Axis	1*	0*
SIb,b...;	AA-AM	1*	0*

* If PA (power automatic) mode is active add 1 to the command queue.

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

Related commands: KL, KS, SA, SD, SO, ST

SK SLIP KILL

The SK command controls slip kill mode. A parameter of 1 enables the mode and a parameter of 0 disables the mode. When this mode is enabled, if the motor slips beyond the dead band set by the ES command, the MAX will kill motion on the selected axes as though a KL command had been issued to the axes. A '?' can follow the SK command to report the current enable/disable status with a 1 for enabled or a 0 for disabled. APP will preserve the slip kill enable/disable setting in non-volatile FLASH.

By default, the only axis that is killed when a slip occurs is the axis that slipped. However, any combination of axes can be killed when any axis slips. Which axes are killed when an axis slips is defined with the SK command followed by an alpha parameter that specified an axis (x,y,z,t,u,v,r or s), followed by numeric multi-axis parameters that specify which axes to kill when a slip occurs on the axis specified by the alpha parameter. Each numeric parameter corresponds to an axis in the usual multi-axis order. If the parameter is 1, the axis is selected for the kill operation. If the parameter is zero, the axis is de-selected for the kill operation. If the parameter is missing for an axis, then the setting for that axis is unaffected. A '?' following the alpha axis specification will report the current axes selected for the kill operation. Selected axes are designated by a '1' character and non-selected axes are designated by a '0' character. Note that these parameters provide the ability to NOT kill the axis that actually slipped – this is done by using a zero parameter for the axis specified.

Example: X axis is sent on a move. Its encoder cable was not connected to the controller (oops!). The slip condition is detected when the difference between the motor position counts and encoder counts exceed 20. At this time the controller issues a KL (Kill) command to the X axis and generates a slip interrupt to the host.

Enter:

```
AX;
ES20;
SK1;
IS:
LP0;
MA30;
GO;
```

Response: None.

Example 2: X axis is sent on a move. Its encoder cable was not connected to the controller (oops!). The slip condition is detected when the difference between the motor position counts and encoder counts exceed 20. At this time the controller issues a KL (Kill) command to the X Y,Z, and T axes and generates a slip interrupt to the host.

Enter:

```
AX;
ES20;
SKX1,1,1,1;
SK1;
IS:
LP0;
MA30;
GO;
```

Response: None.

Enabling/Disabling slip kill mode:

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SK#;	Single Axis	1	2
SK#,#,#,…;	AA-AM	1*	2

* = per axis

Defining kill axes:

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SKx#,#,#,…;	Single Axis	1	2
SKx#,#,#,…;	AA-AM	1*	2

* = per axis

Related commands: ES, IS, RL

SME SYNCHRONIZE MOTOR POSITION WITH ENCODER POSITION

The SME command synchronizes a motor position with an encoder by setting the motor position equal to the encoder position.

Example: The following would set the X axis motor position to the current X axis encoder position.

Enter: AX:
SME;

Response: None

Example: The following would synchronize the motor positions of the Y & T axes with their respective encoder positions.

Enter: AA;
SME,1,,1;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SME;	Single Axis	1	1
SMEb,b...;	AA-AM	1	1

Related commands: LP

SMEA SYNCHRONIZE MOTOR AND ENCODER POSITION WITH ABSOLUTE ENCODER POSITION

The SMEA command synchronizes the motor and encoder positions with the absolute encoder position by setting the motor and encoder positions equal to the absolute encoder position. An optional offset parameter value can be entered which will be added to the absolute encoder reading when setting the motor and encoder positions. If no value is specified for the offset, it will default to a value of zero. In AA or AM modes, the offset parameter is required, and should be set to a value of zero if the offset is not to be used. The SEA and SMEA commands share the offset parameter.

The absolute encoder synchronization mode (SEA or SMEA) and the offset parameter value can be preserved as power up/reset values with the APP command.

RANGE: -2147483647 <= offset <= 2147483647

Example: The following would set the X axis motor and encoder positions to the current X axis absolute encoder position.

Enter:

```
AX;
SMEA;
```

Response: None

Example: The following would synchronize the motor and encoder positions of the Y & T axes with their respective absolute encoder positions plus an offset of 50.

Enter:

```
AA;
SMEA,50,,50;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SMEA{#};	Single Axis	1	1
SMEA#,#..;	AA-AM	1	1

Related commands: SEA

SO STOP BY RAMPING FROM DISTANCE

The SO command instructs the MAX to continue moving until reaching a specified distance (parameter 2) from a specified stop point (parameter 1). The axis will then ramp to a stop within the specified distance. This allows the user to control the point at which deceleration begins, the rate of deceleration, and the stop point, all with a single command.

RANGE:

Min. Position Range ≤ Parameter 1 (Stop Position) ≤ Max. Position Range
Min. Position Range ≤ Parameter 2 (Distance from Stop Position to Start Decelerating) ≤ Max. Position Range

NOTE: The position range is dependent on the MAX's settings.

Example: The X axis is jogging at 10,000 steps per second. We want the axis to stop at position 50,000 but it must not start ramping until reaching position 46,000.

Enter: SO50000,4000;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SO#,#;	Single Axis	3*	2*
-	AA-AM	Not Valid	

* If the axis is a stepper and encoder or servo axis, add 1 to the command queue
* If PA (power automatic) mode is active, add 1 to the command queue
* If an aux bit settling time has been specified, add 2 to the command queue and add 1 to the argument queue

Related commands: KL, KS, SA, SD, SI, ST

SR SELECT CUSTOM RAMP

The SR command selects a previously defined custom ramp profile for use with a currently active axis. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

RANGE: 1 ≤ SR ≤ 8

Example: Select custom ramp number 4 for use with axis Y.

Enter:
```
AY;
SR4;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SR#;	Single Axis	15	5 + (2 x number of segments in ramp definition)
-	AA-AM	Not Valid	

Related commands: DAB, DAE, DAR, RT

ST STOP

The ST command flushes the queue for the current axis or axes only and causes the axis/axes to decelerate to a stop at the rate previously specified via the AC command. This command is used to stop one or more motors in a controlled manner from jog mode or an unfinished GO or GD command. This command is executed immediately upon receipt. All status and position information is retained. When executed in a multi-axis mode, the ST command is equivalent to the SA command.

Example: Move the Y axis for a while at 1200 steps/second and then ramp to a stop.

Enter:

```
AY;
JG1200;
(Wait awhile)
ST;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ST;	Single Axis	1*	0*
ST;	AA-AM	9*	61*

* If PA (power automatic) mode is active add 1 to the command queue.

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

Related commands: KL, KS, SA, SD, SI, SO

SV SERVO VOLTAGE CONTROL

Firmware Revision Required: V1.30 and greater

The SV command allows several servo voltage parameters to be set or queried. The first parameter is an alpha character that determines which servo voltage function is being controlled.

If the first parameter is ‘P’, the servo polarity is set by the parameter following the ‘P’, which can be ‘+’ or ‘-‘, plus setting the MAX PID to produce a positive voltage for positive error and a negative voltage for negative error, the default behavior. A ‘-‘ character following the ‘P’ inverts the PID voltage output.

If the first parameter is a ‘B’, the servo bi-polar mode is set by the parameter following the ‘B’, which can be a ‘1’ or a ‘0’. A ‘1’ enables bi-polar mode and a ‘0’ disables bi-polar mode. When bipolar is enabled, a zero torque reference will result in a 0VDC output (+/- offset voltage). The analog output will range between +10VDC and -10VDC when bipolar is enabled. The default behavior is bi-polar enabled. When bi-polar is disabled, the control is in unipolar mode and the PID analog output will range between 0.0VDC and +10VDC. At maximum positive velocity, the board outputs +10VDC. At maximum negative velocity, the board output approaches 0.0VDC. To maintain position the board outputs 5VDC.

See the APP and APB commands for archiving the controller’s parameters in non-volatile FLASH.

SVV? can be used to query the current PID output in volts.

Example: The Y axis encoder is counting opposite the expected direction. Setup the Y axis to produce a negative voltage when moving positive instead of a positive voltage to correct the problem.

Enter: AY;
SVP-;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SVP**[**+/-/?**]**; SVB**[**0/1/?**]**; SVV?;	Single Axis	Immediate	
SVP**[**+/-/?**]**; SVB**[**0/1/?**]**; SVV?;	AA-AM	Immediate	

Related commands: None

SW SYNC WAIT FOR USER-DEFINED INPUT HIGH

The SW command allows synchronization of multi-axis moves or other tasks on one or more MAX boards by using one of the general purpose input lines. This command causes the MAX to stop processing new commands until the general purpose input line has been released (allowed to go high) before proceeding with the next command.

The SW command can also be used to cause an axis to wait until the others are finished. To do this, wire-OR the auxiliary lines from several axes together and connect them to a general purpose input line. Use power automatic (PA) mode with the SW command on that line. All commands after that will wait until all axes release their auxiliary lines; i.e. come to a complete stop.

RANGE: 0 ≤ Bit Number ≤ 15 for MAXp and MAXv

0 ≤ Bit Number ≤ 7 for MAXnet

NOTE: The parameter used to specify Bit Number must be configured as an input. See BD and IO.

Example: The following command sequence will cause the X axis move to wait until the Y axis has finished its move and turned off its auxiliary output which has been wired to the general purpose input 0 line.

Enter:

```
AY;
AB1;
MR2000;
GO;
AB0;
AX;
SW0;
MR10000;
GO;
```

Response: None.

The SW command provides a way to synchronize moves on two or more controllers. The following example shows one way to do this.

Example: You have 3 four axis controllers, for a total of 12 axes to move together. Call board 1 the "master" and boards 2 and 3 the "slaves". Wire board 1's X axis auxiliary line to the two slave boards' general purpose input 0 line. Send to the master the command "AX;PA0;", setting the master's X axis auxiliary line low until its move starts. This also sets the slaves' general purpose input 0 line low. Enter the "SW0;" command to the two slaves, followed by the move and GO commands. On the master, enter the move command, followed by the GO command. When the master's move starts, the PA command will set the auxiliary line high releasing the wait on the slave boards. All three boards will start their moves. This provides synchronization to within 480μs of each board.

Procedure: Wire board 1's X axis auxiliary line to board 2's and board 3's general purpose input 0 line.

Enter:

```
(Board 1)AX;
          PA0;
(Board 2) AA;
          SW0;
          MR200,200,200,200;
          GO;
(Board 3) AA;
          SW0;
          MR300,300,300,300;
          GO;
(Board 1) AA;
          MR100,100,100,100;
          GO;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SW#;	Single Axis	1	1
SW#;	AA-AM	1	1

Related commands: BW, WA, WQ, WT

SXC CLEAR MACRO LINKS

This command clears all macro links of input bits to macro executions.

Example: Clear all previously defined macro links to input bits.

Enter: SXC;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SXC;	Single Axis	Immediate	
SXC;	AA-AM	Immediate	

Related commands: APP, SXL, SXM

SXK#,#,#[,#]   DEFINE KILL LINK

The SXK command links the execution of the kill function (KL) to the defined change of state of the assigned input bit. The SXK setting is active independent of the stand-alone mode (SXM) setting. The inputs are sampled every update cycle to determine if a kill operation is required for the current SXK setting. The APP command can be used to preserve the settings as the power-up/reset values. The default setting is no input bit is linked to the kill command.

First Parameter

This specifies the standard input bit number. Any general purpose I/O bit configured as an input can be used.

Second Parameter

Valid Bit States are 0 and 1

If the value of the selected bit state is ZERO, the selected macro will be executed if the selected bit *changes* from a TTL high to a TTL low.

If the value of <Bit State> is ONE, then the selected Macro will be executed when the selected bit *changes* from a TTL low to a TTL high.

NOTE: The kill operations defined by the SXK are active independent of stand-alone mode (SXM).

Third Parameter

If the value is ZERO the KILL function is disabled.

If the value is ONE, the KILL function linkage for the specified Bit State is enabled and will replace any previously defined kill linkage

Optional Fourth Parameter

This parameter optionally defines a debounce time in milliseconds. This will lock out additional executions of the kill command until the debounce time has expired. If no debounce time is specified, a default of 15 milliseconds will be used.

SXK? can be used to query the current settings.

Example: I/O bit 2, as it goes from high to low, is to be linked with the KILL (KL) function.

Enter: SXK2,0,1;

Response: None

Example: Query the current SXK settings.

Enter: SXK?

Response: =2,0,1,15

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SXK#,#,#**[**,#**]**;	Single Axis	Immediate	
SXK#,#,#**[**,#**]**;	AA-AM	Immediate	

Related commands: APP, BD, KL

SXL#,#,#; DEFINE MACRO LINK

This command creates the link to execute the previously defined permanent Macro when it senses the change in state of the defined standard input bit. The APP command can be used to preserve the settings as the power-up/reset values. The default setting is no input bits are linked to the macros.

When a macro is triggered by an input state change, all of the commands in the macro must be parsed before another input state change can trigger any subsequent macro execution. Thus, if several different macros are linked to inputs and one of the inputs triggers a macro execution, input state changes will be ignored until all the commands in the triggered macro have been parsed. Note that the execution of the commands placed in the queue by the macro do not necessarily have to complete before input state changes can trigger another macro. Consider the case of a macro that contains loop commands. When this macro is triggered by an input state change, its commands will be parsed and the queued commands within the loop may still be executing for some time after the entire macro is parsed. After the macro is parsed, other inputs may trigger a macro even though the loop is still executing.

First Parameter

This specifies the standard input bit number. Any general purpose I/O that is configured as an input can be used.

Second Parameter

Valid Bit States are 0 and 1

If the value of the selected bit state is ZERO, the selected macro will be executed if the selected bit *changes* from a TTL high to a TTL low.

If the value of <Bit State> is ONE, then the selected Macro will be executed when the selected bit *changes* from a TTL low to a TTL high.

NOTE: Each bit state can be linked with a macro. So, up to two macros can be assigned to an input bit. For example, macro 10 could be executed when I/O 0 goes low and macro 11 could be executed when I/O 0 goes high. It is possible to link multiple input bits to the same macro.

Third Parameter

Specifies the Macro number to be executed when the conditions are met. A macro link can be deleted by specifying the bit number and the Bit State along with a Macro number zero. The macro assigned to a given input bit and Bit State can be changed by issuing the same SXL command using a different macro number.

SXLn,b,?; can be used to query the selected linkage definition.

Optional Fourth Parameter

This parameter optionally defines a debounce time in milliseconds. This will lock out additional executions of the macro until the debounce time has expired. If no debounce time is specified, a default of 15 milliseconds will be used.

Example: When I/O bit 1 goes from low to high, macro 20 will be executed. Debounce time is set to 5 milliseconds.

Enter: SXL1,1,20,5;

Response: None

Example: Upon Power-up, the user presses an “Activate Button-short to ground”, (linked to bit 1). This should cause the controller to position the X-axis at a position 1000 steps from zero and the Y-axis at 2000 steps from zero by executing Macro 10. The Stop button (short to ground) is linked to bit 2.

Enter:

```
MD0;           begin the definition of Macro #0
AX;
MA1000;
GO;            move x-axis to position 1000
AY;
MA2000;
GO;            move y-axis to position 2000
<control Z>    terminate the definition of Macro #0
PT0,10;        store macro #0 to non-volatile macro #10
SXL1,0,10;     define I/O bit #1, active low will execute macro #10
MD1;           begin the definition of Macro #1
ST;            stop everything
<control Z>    terminate the definition of Macro #1
PT1,9;         store macro #1 to non-volatile macro #9
SXL2,0,9;      define I/O bit #2, active low will execute macro #9
SXM1;          enable the stand-alone mode
APP;           set the current parameters to the power-up defaults
```

Response: None

Example: Query current SXL settings.

Enter: SXL?;

Response: =2,0,9,15;

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SXL#,#,#[,#];	Single Axis	Immediate	
SXL#,#,#[,#];	AA-AM	Immediate	

Related commands: APP, BD, MD, MX, PT, SXC, SXK, SXM

SXM# ENABLE/DISABLE STAND-ALONE MODE

This command enables, disables or queries the SXM mode (stand-alone mode). The APP command can be used to preserve the settings as the power-up/reset values. The default setting is stand-alone mode disabled.

SXM mode value of 1 enables the stand-alone mode

SXM mode value of 0 disables the stand-alone mode.

SXM? can be used to query the current setting.

Example: Enable stand-alone mode

Enter: SXM1;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SXM#;	Single Axis	Immediate	
SXM#,	AA-AM	Immediate	

Related commands: APP, BD, MD, MX, PT, SXC, SXL

TL SET SOFTWARE OVERTRAVEL LIMITS

The TL command sets logical limits on the range of travel for an axis. Two parameters must be supplied; one for the upper travel limit and the other for the lower travel limit, both as absolute positions. If the axis reaches either of these logical limits, the MAX will flag a limit condition just as it would be using the physical limit switch inputs. In soft limit mode (LMS), the control will decelerate to a stop at the travel limit defined by the TL command. In hard limit mode (LMH), the control will halt abruptly if the travel limit is reached. In both soft and hard limit modes, the axis queue will be flushed and a limit interrupt will be issued to notify the host. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

TL? can be used to query the current parameter settings. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

Note: Software overtravel limits and physical limit switch inputs can be enabled at the same time.
Note: TL0,0; Turns software limits off.

RANGE: ± Position Range

Example: Set logical position limits for the X axis of +/-1,000,000.

Enter: AX;
TL1000000,-1000000;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
TL#,#;	Single Axis	1	2
-	AA-AM	Not Valid	

Related commands: LM, LT

TM TIMED JOG

The TM command performs a jog at the current velocity limits defined for the axis/axes for the specified number of milliseconds. In multi-axis mode, all axes begin moving at the same time and ramp to a stop when their respective jog times have elapsed. The overall jog time will be the parameter passed to the TM command plus deceleration time and acceleration time.

A negative argument will result in motion in negative direction.

RANGE: -200,000 ≤ TM ≤ 200,000

Example: Jog the X axis for 1000 milliseconds.

Enter: AX;
TM1000;

Response: None.

Example: Jog the X axis for 1000 milliseconds and the Z axis for 2000 milliseconds, starting both at the same time.

Enter: AA;
TM1000,,2000;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
TM#;	Single Axis	5*	1*
TM#,#,…	AA-AM	5*	1*

* If PA (power automatic) mode is active add 1 to the command queue
* If the axis is a stepper and encoder or servo axis add 1 to the command queue

Related commands: JF, JG, SA, ST

TP TRACK POSITION

Firmware Revision Required: V1.30 and greater

The TP command creates a mode which allows the controller to receive new target positions on-the-fly, that is, while in motion towards a previous TP target. The target positions can be specified in either absolute or relative coordinates. The default is absolute, but this can be overridden by placing an ‘r’ prior to numeric coordinate specification, an ‘a’ prior to numeric coordinates specifies absolute coordinates. So TP and TPa commands are equivalent and both specify absolute coordinates. TPr commands specify relative coordinates. The first TP command puts the controller in Track Position mode and starts to move towards the initial target position. No GO command is required. The motion profile will be trapezoidal in this mode, using the current setting of the velocity (VL), acceleration (AC), and deceleration (DC) parameters, which may also be modified on-the-fly and will take effect immediately when in TP mode. New target positions may create the need for a change of direction, in which case the motion will decelerate to a stop and then accelerate to velocity towards the new target position, using the current deceleration (DC), acceleration (AC) and velocity (VL) parameters respectively. New target positions may also be so close to the current position that deceleration with the DC parameter is not possible without overshooting the new target. In this case, the controller decelerates using the DC parameter, overshooting the new target by the distance necessary to achieve zero velocity, and then changes directions and proceeds towards the new target using the current acceleration and velocity parameters.

TP mode is canceled by any stop of kill command, or any command that flushes the axis queue. TPX can also be used to exit TP mode.

TPS can be used just like the TP command, the only difference being that when the move is complete, the controller will not remain in TP mode.

Velocity override can be used in this mode. Power automatic can be used in this mode, but it must be enabled prior to entering TP mode. For closed-loop steppers, hold mode can be used in TP mode, but all the hold parameters (HV, HG, HD, and HN must be entered prior to enabling TP mode).

When the control is not in TP mode and a TP command is entered, all previously queued commands will execute before TP mode is entered. In general, queued commands entered while in TP mode will not execute. Some queued commands, however, operate differently in TP mode. These commands are AC, DC, and VL, which are normally queued commands, but in TP mode are immediate commands.

TP can be active for all axes in AA mode or a subset of axes in AM mode. Each axis performs the position tracking function simultaneously, but independently of the other axes.

TP? can be used to query for the last TP destination entered. If TP mode is not active or the control is not currently moving towards a TP target when TP? is issued, then “---“ is returned in the response.

TPID#; can be used to interrupt with the done flag when the target position is reached.

TPII#,#,…; can be used in AA/AM modes to interrupt with the axis’ done flag when each individual axis reaches its target position.

Example: Set the velocity, acceleration, and deceleration parameters for the X axis and then enter position tracking mode with an initial target position of 175000. After some time, but prior to the X axis reaching the target of 175000, modify the target to return to position 0 and change the velocity at which to return to 0 to 30000. The control will decelerate using DC to a stop and then reverse directions, using AC to accelerate to a velocity of 30000, if possible, and DC to decelerate to zero velocity at position 0.

Enter:

```
AX;
VL10000;
AC100000;
DC50000;
TP175000;
   …
TP0;
VL30000;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
TP#;	Single Axis	1	1
TP#,#,...;	AA-AM	1 *	1*

* One queue and one argument entry for each axis.

Related commands: AC, DC, PA, TPS, TPX, VL, VO

TPS TRACK POSITION STOP

Firmware Revision Required: V1.30 and greater

The TPS command sets a new target position and will exit TP mode when the target position is reached. This is true whether or not the axis or axes was in TP mode prior to entering the TPS command. If other TP mode commands are entered prior to reaching the TPS target position, they will override the previous TPS command and when the target position is reached, TP mode will not be exited. If another TPS command is entered prior to reaching the previous TPS target, then motion will proceed toward the new TPS target and, if and when the target is reached, TP mode will be exited.

Example: Move the X axis to 17500 and then exit TP mode.

Enter:
```
AX;
TPS175000;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
TPS#;	Single Axis	1	1
TPS#,...;	AA-AM	1*	1*

* One queue and one argument entry for each axis.

Related commands: AC, DC, PA, TP, TPX, VL, VO

TPX TRACK POSITION EXIT

Firmware Revision Required: V1.30 and greater

TPX causes TP mode to exit TP mode as soon as the current TP target position is reached. When the TPX command is issues, no more command characters are processed until the TPX mode exit is complete, which can be as long as one update cycle.

Example: With the X axis is in TP mode, exit a TP mode for the X axis.

Enter:
AX;
TP175000;
… other commands
AX;
TPX;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
TPX;	Single Axis	Immediate*	Immediate*
TPX;	AA-AM	Immediate*	

* The command will not take effect until the current target position is reached.

Related commands: AC, DC, PA, TP, VL, VO

MAXv only

#UC USER CUSTOM CONFIGURATION

Firmware Revision Required: V1.31.1 and greater

This command is used for customer configuration settings that alter the behavior of the controller.

The command takes the form of:

#UC#[,n];

Where **<#>** specifies the custom command and **<*n*>** specifies the optional custom command parameters.

Currently valid values for <#> are:

Custom Configuration Commands		
<#> Value	<n> Value	Description
1	0*	sets MAXv 64-bit VME backplane configuration of pin D28 to S Aux
	1	sets MAXv 64-bit VME backplane configuration of pin D28 to I/O 6
…		
…		
…		

* = Power-up default value

Example: Select the IO6 signal to be output on the VME 64-bit backplane, Row D-pin 28.

Enter: AX;
#UC1,1;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#UC#**[**,n…**]**;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: None

UF USER UNITS OFF

The UF command turns off user units returning all numeric commands and responses to their factory default raw representations. This command is equivalent to and preferred over UU1; since it turns off the mode thus minimizing unnecessary overhead. See the APP and APB commands to preserve the UF settings to flash memory.

Example: Turn off user unit conversion on the X, Y, and Z axes.

Enter:
AX;
UF;
AY;
UF;
AZ;
UF;
Or
AA;
UF1,1,1;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
UF;	Single Axis	Immediate	
UFb,b…;	AA-AM	Immediate	

Related commands: UU

#UR SET UPDATE RATE

This command sets the update rate of the controller. Changing the update rate affects current axis positions! This command is intended to be used at init only. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

#UR? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

#UR#;
# specifies the number of updates per second.

RANGE: 1024, 2048, 4096, 8192

Any other value besides those specified in the range will result in a command error.

Example: Set the update rate to 2048 times per second.

Enter: #UR2048;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#UR#;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: None

UU SET USER UNITS

The UU command converts all move velocities, distances, etc. to user specified units by multiplying by the specified parameter. The UF command is used to terminate this mode. Factory default is with this command off. See the APP or APB commands on to preserve the UU settings to flash memory. If it is desired to use User Units with an encoder ratio, then the User Units should be defined in encoder units.

When User Units is active, all user-supplied axis coordinates that have the user units applied must result in a coordinate value that falls within the axis coordinate range of (Min. Position Value, Max. Position Value).

UU? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

NOTE: While the user units' mode provides a certain level of convenience to the user, it does so at a cost, namely accuracy and control. User unit conversions may cause round off error and may possibly truncate key information.

Example: The motor, driver, and gear ratio you are using requires 10,000 steps to move one inch. Set up the X, Y, and Z axes so you can enter move information in inches.

Enter:

```
AX;
UU10000;
AY;
UU10000;
AZ;
UU10000;
Or
AA;
UU10000,10000,10000;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
UU#;	Single Axis	Immediate	
UU#,#...;	AA-AM	Immediate	

Related commands: RU, UF

VA SET VECTOR ACCELERATION

Firmware Revision Required: V1.30 and greater

The VA command sets the vector acceleration and deceleration used by VC/VP commands. The first parameter sets the acceleration and the second, optional parameter, sets the deceleration. The vector acceleration and/or deceleration can be changed between VC/VP commands and the new vector parameters will take effect on the next VC/VP command following the VA command. No action is taken when the command is entered, but instead it affects all subsequent VC/VP commands. The factory default is 10000 steps/second/second for both acceleration and deceleration.

VA? can be used to query the current parameter settings are reported.

The acceleration parameter is used for increasing velocities and the deceleration parameter is used for decreasing velocities. The only exception to this is when the velocity is decreasing solely because of a decreasing velocity override value, in which case the acceleration parameter is used in order for the velocity override to increase and decrease symmetrically. This behavior of the velocity override to use the acceleration parameter to decrease velocity is superseded whenever deceleration is required for any other reason, such as a stop command or to reach a lower ending velocity at the end of a segment.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

RANGE: 1 ≤ VA ≤ 8,000,000

Example: Set the vector acceleration and deceleration to 7500.

Enter: VA7500;

Response: None

Example: Set the vector acceleration to 15000 without changing the deceleration rate.

Enter: VA15000,;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VA#[,#];	AA	Immediate	Immediate
VA[[n]]#[,#];	AM	Immediate	Immediate

Related commands: VC, VO, VP, VV

VB SET VELOCITY BASE

The VB command allows the acceleration ramp to start off at a specified velocity. This allows faster acceleration and the ability to pass through resonance quickly in some applications. The velocity jumps instantly to the specified velocity, and then accelerates as usual. The deceleration is the same in reverse. This mode is active only for linear ramps; it is ignored for cosine and parabolic ramps but not flagged as a command error. The parameter must be greater than zero and less than the programmed velocity, where the factory default is zero steps per second. This command is not valid with the JG command nor will it work in conjunction with the DC command. See the APP command to preserve the VB settings as the power-up/reset values.

If the VL command is used after the VB command and the velocity value set with VL is less than the previously set VB value, the initial velocity used at the start of a move will be the VL value minus one. This will result in a one-step acceleration ramp and must be taken into consideration in applications making use of the VB command.

VB? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE: 0 ≤ VB < VL value

Example: In the single axis mode, set the Y axis velocity base to 200.

Enter: AY;
VB200;

Response: None.

Example: In the AA mode, set the X and Y axes velocity bases to 200.

Enter: AA;
VB200,200;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
VB#;	Single Axis	1	1
VB#,#...;	AA-AM	1	1

Related commands: AC, DC, VL

VC VECTOR CIRCULAR INTERPOLATION

Firmware Revision Required: V1.30 and greater

The VC command defines a move in a two-dimensional circular pattern from the entry position. The first two parameters are the center of the circle in absolute units, the third parameter is the arc distance to move in degrees or radians, depending on whether the value is preceded by the character ‘d’ for degrees or ‘r’ for radians. The default is degrees if no alpha character precedes the arc distance value. Positive arc distance equal counterclockwise movement. Negative arc distance equal clockwise movement. The radius of the circle is the linear distance between the current position (the end-point of the previous move) and the center point of the circle specified by the first two parameters of the VC command. The vector velocity specified by the VV command and vector acceleration specified by the VA command are used as they are for the VP command. The VC command can optionally be followed by a ‘R’ preceding the numeric parameters and this allows the center point to be specified in coordinates relative to the start point of the circle. An ‘A’ can be also be used to specify absolute center point coordinates, which is the default.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

RANGE:

Min Pos. Value ≤ Parameter 1 (First Coordinate for Center of Circle) ≤ Max Pos. Value

Min Pos. Value ≤ Parameter 2 (Second Coordinate for Center of Circle) ≤ Max Pos. Value

Example: Create an oval racetrack shape with velocity override from analog input channel 2.

Enter:

```
AA;
VOA2,5;
VV4000,4000;
VC0,8192,90;
VP8192,32768;
VC0,32768,180;
VP-8192,8192;
VV4000,0;
VC0,8192,90;
VE;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	Single Axis	Not Valid	
VC[A/R]#,#,#;	AA	1	28
VC[[n]][A/R]#,#,#,	AM	1	28

Related commands: VA, VCP, VE, VO, VP, VV

VCP SET VECTOR CIRCULAR PLANE

Firmware Revision Required: V1.30 and greater

The VCP command defines the axes used for the two-dimensional plane of circular motion. The two parameters specify the two axes of the plane to be used for VC commands. A terminator must be used following the last parameter. The default plane for circular interpolation is X,Y.

VCP? can be used to query the current setting.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

Example: Specify the YZ plane for circular interpolation.

Enter: VCPyz;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VCPaa;	AA	Immediate	Immediate
VC**[**[n]**]**Paa;	AM	Immediate	Immediate

Related commands: VC, VE

VE VECTOR END

Firmware Revision Required: V1.30 and greater

The VE command is used to end a sequence of VC/VP commands. This is required if any queued commands need to be used between two VC/VP segments that control stepper motors and the vector velocity has not been decelerated to zero. In multi-axis mode, this command will only affect axes for which a parameter is entered.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

NOTE: A terminator is required for this command

Example: Perform a sequence of vector segments, stop without deceleration and report when done. Without the VE command prior to the ID command, the X and Y axes would have continued to move at the vector velocity indefinitely.

Enter:

```
AM;
VA10000;
VV1000;
VPR1000,100;
VPR250,5000;
VPR10000,10000;
VE;
ID;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VE;	AA	1	0
VE**[**[n]**]**;	AM	1	0

Related commands: VA, VC, VP, VV

VG VECTOR GO (RESUME)

Firmware Revision Required: V1.30 and greater

The VG command resumes motion after it has been suspended by a VZ or VH command.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

NOTE: A terminator is required for this command.

Example: A motion sequence has been started and must be suspended in order to respond to an unexpected event and then motion is resumed.

Enter:

VPR10000,1000;
VPR5000,25000;
VH1; (Note that the vector may not be at 5000,25000 at this point. That depends entirely on when the VH1; command was issued)
VG; (Motion now continues from wherever it left off)
VPR2500, 1000;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VG;	AA	Immediate	Immediate
VG[[n]];	AM	Immediate	Immediate

Related commands: VC, VH, VP, VZ

VH VECTOR HALT

Firmware Revision Required: V1.30 and greater

The VH command stops motion. How the motion is stopped depends on the parameter following the VH command. If the parameter is zero, motion is stopped immediately without deceleration. If the parameter is 1, motion is stopped by decelerating to zero velocity. Note that the queues are not flushed and all queued commands remain in the queue. If there is a current queued VC/VP command, it is simply suspended until motion is resumed with a VG command. If there is no VC/VP command currently queued, the next VC/VP command will not start motion unless or until a VG command is entered subsequent to the VH command. Note that once a VH; command is entered, a VG command must be re-entered in order to resume VC/VP commands; entering a VC/VP command after a VH command will not resume motion, it will only place the motion command in the queue, which is stalled waiting for a VG command. Note also that VH can be used prior to issuing a series of VC/VP commands in order to pre-load the queue, which will start executing when a VG command is entered.

When single segment mode is enabled with a VZ1; command, a VH command must also be issued in order for the controller to stop at the end of the segment and wait for a VG command. VA1; without a VH command will decelerate to stop at the end of each segment and then proceed to the next segment.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

NOTE: Using the KS command during VP commands requires that all active axes in the VP command be specified. If only a partial set of the active axes is specified then the results are undefined.

Example: A motion sequence has been started and must be suspended in order to respond to an unexpected event and then motion is resumed.

Enter:

```
VPR10000,10000;
VPR5000.25000;
VH0;
…
VG;
VPR2500,1000;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	Single Axis	Not Valid	
VH#;	AA	Immediate	Immediate
VH[[n]]#;	AM	Immediate	Immediate

Related commands: VC, VG, VP, VZ

VID INTERRUPT WHEN VECTOR DONE

Firmware Revision Required: V1.30 and greater

The VID command enables and disables the interrupt when vector done mode. A non-zero parameter following the command will enable the mode and a zero parameter will disable the mode. The done flags will be set for each axis involved in a vector move when the vector move is complete. Note that the setting of the done flags is done without the use of the queue so that multiple consecutive vectors can be executed when VID mode is enabled and each vector will report done without any interruption of the smooth transition between vector segments.

VID? can be used to query the current setting.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VID**#**;	AA	Immediate	Immediate
VID**[**[n]**]**#;	AM	Immediate	Immediate

Related commands: VIP

VIO VECTOR INPUT/OUTPUT CONTROL

Firmware Revision Required: V1.30 and greater

The VIO command allows the state of the general purpose inputs and outputs to be defined at the start of vector motion of a VC/VP commands. No action is taken when the command is entered, but rather it affects the action of all subsequent VC/VP commands. The first parameter specifies the desired state of the inputs and outputs at the start of vector motion. The outputs will be set to the defined state at the start of the vector. The inputs must be in the defined state for the vector to start or else a halt will be performed. Whether or not deceleration occurs depends on the fourth parameter, a parameter of zero halts without deceleration while a parameter of 1 halts with deceleration. The default action is to halt without deceleration.

An optional second parameter defines the state of the outputs in the next update cycle following the start of vector motion. This second parameter setting can be used to create a one update cycle wide pulse output by specifying the negated state of the first parameter. If the second parameter is omitted, it defaults to being equal to the first parameter, thus providing a steady state output. An optional third parameter can define a mask such that only the I/O lines with a 1 bit defined in the mask will be affected when setting the outputs specified by the first and second parameters. This allows other processes to have control of some of the defined outputs without interference by another process using the VIO commands. If the third parameter is omitted, it defaults to all output bits enabled and all input bits disabled. The VIO settings will affect all subsequent VC/VP commands until cancelled by a VIO command with no parameters. All parameters are specified in hexadecimal.

VIO? can be used to query the current settings.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

Example: Provide a one update cycle wide pulse output on IO channel 15 at the beginning of all VP vector segments until cancelled by the VIO; command with no parameters. No other I/O channels will be affected due to the mask setting of 8000.

Enter:

```
VIO0,8000,8000;
VPR10000, 10000;
VPR5000.25000;
…
VIO;
VPR2500,1000;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VIO#[,#,#,#];	AA	Immediate	Immediate
VIO[[n]]#[,#,#,#];	AM	Immediate	Immediate

Related commands: VC, VP

VIP INTERRUPT WHEN VECTOR IN POSITION

Firmware Revision Required: V1.30 and greater

The VIP command enables and disables the interrupt when vector in position mode. A non-zero parameter following the command will enable the mode and a zero parameter will disable the mode. When the mode is enabled, The VIP command sets the done flags for each axis involved in a vector move when all axes involved are within their respective specified encoder dead-bands. Because this command must check for each axis being within its defined encoder dead-band, their will be at least one update cycle after the move is complete to perform this check. Consequently, if multiple consecutive vectors are executed in VIP mode, there will be at least one update cycle delay between each vector.

VIP? can be used to query the current setting.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VIP**[**#**]**n;	AA	1	0
VIP**[**[n]**]**#;	AM	1	0

Related commands: HD, VID

VL SET VELOCITY MAXIMUM

The VL command sets the maximum velocity of one or more axes to the operands which follow the command. The factory default is 200,000 steps per second. See the APP command to preserve the VL settings as the power-up/reset values.

VL? can be used to query the current parameter setting. Starting with V1.30 firmware (see WY command), a question mark can follow the command in order to query the last parameter setting.

RANGE: 1 ≤ VL ≤ 4,194,303

Example: In single axis mode, set the X axis velocity to 10,000 counts per second.

Enter:

```
AX;
VL10000;
```

Response: None.

Example: In the AA mode, set the peak velocities of the X and T axes to 5,000 and 50,000 respectively. Leave the other axes with their previous values.

Enter:

```
AA;
VL5000,,,50000;
```

Response: None.

QUEUE REQUIREMENTS				
FORMAT	**MODE**	**Axis Ramp Type**	**COMMAND**	**ARGUMENT**
VL#; or VL#,#...;	Single Axis or AA-AM	Linear (RTL)	1	1
VL#; or VL#,#…;	Single Axis or AA-AM	Custom Ramps (SR)	1	Number of ramp segments
VL#; or VL#,#...;	Single Axis or AA-AM	S-curve (SS)	3	43

Related commands: AC, DC, VB

VO VELOCITY OVERRIDE

Firmware Revision Required: V1.30 and greater

The VO command allows the user to input a percentage that overrides the current vector velocity used by the VC/VP, GP and TP commands. The VO parameter value takes effect as soon as the controller executes the immediate command. This means that VC/VP, GP and TP segments currently in motion will immediately start accelerating or decelerating to the new overridden velocity. The minimum override percentage is zero which will cause motion to suspend until a non-zero override percentage is active. The maximum override percentage is 200, which would double the current vector velocity. Note: However, the velocity can never be overridden to a value greater than the maximum velocity (see VL). The reported value reflects the velocity override setting, not the percentage of the actual velocity relative to the programmed velocity; they may be different if the overridden velocity has reached maximum.

The VO command has multiple formats to set the percentage value:

VO=# sets the override percentage directly to #

VO# or VO+# adds # to the current override percentage

VO-# subtracts # from the current override percentage

VOA#1,#2 sets the source of the override to the analog input channel #1 and uses #2 as the analog input value, in volts, that represents the maximum override percentage (maximum = 200). An appropriate analog input offset may need to be set to guarantee the velocity override can reach a value of zero.

VO? can be used to query the current setting.
VOA? can be used to query the current analog input velocity override setting.

Using any of the commands that set the override percentage from user input (for example, VO=/+/-) will cancel the analog input mode (VOA) if it is active.

NOTES:

1) For stepper motors, when motion stops before the end of a segment because the override value is set to zero percent or a stop command is issued, the motor will not necessarily stop on a whole step, that is, the STEP signal output may be in either a high or a low state. This is necessary in order to maintain position of the multi-axis vector.
2) The override percentage represents a new “desired” velocity. The motor(s) will accelerate or decelerate using the VA acceleration values toward the new velocity.

Example: Set the velocity override value to 25 percent, and then increase it by 10 to 35, then decrease it by 5 to 30.

Enter:

```
VO=25;
VO+10;
VO-5;
```

Response: None

Example: Use the #1 analog input channel as a velocity override dial. Attach the center tap of a potentiometer to the analog input and the poles of the pot to +5VDC and ground.

Enter: VOA1,5;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
VO**[**+/-#,=,?**]**;	Single Axis	Immediate	Immediate
VO**[**+/-#,=,?**]**;	AA	Immediate	Immediate
VO**[**[n]**][**+/-#,=,?**]**;	AM	Immediate	Immediate

Related commands: GP, TP, VA, VC, VP, VV

VP VECTOR POSITION – LINEAR INTERPOLATION

Firmware Revision Required: V1.30 and greater

The VP command uses linear interpolation to perform a straight line move for one or more axes. The parameters are axis coordinate end-points input in the standard axis order (X, Y, Z, T, U, V, R, S) in which any axis may be omitted by entering a comma and omitting the numeric value. Any unspecified axes remain unaffected. By default the axis coordinate parameters are absolute coordinates with respect to the zero position of each axis. The parameters will be interpreted as relative to the current position if the command is followed by a ‘R’ character or they will be interpreted as absolute if the command is followed by an ‘A’ character, i.e. ‘VPR’ or ‘VPA’. The command starts the move immediately.

The vector velocity parameters are set with the VV command. The vector acceleration and deceleration parameters are set with the VA command. Only linear acceleration is supported during this move. The final vector velocity parameter can be set equal to any non-zero value to allow consecutive commands to create continuous multi-segment motion at the vector velocities. A VV command that sets the final velocity parameter equal to the starting velocity will cause constant velocity during the segment and a smooth transition at the vector velocity for all subsequent vector segments until the VV parameter is changed. VV command that sets the final velocity parameter to zero will cause subsequent VP commands to decelerate to a stop at the end of the vector segment, or in a subsequent segment if the deceleration rate is too slow to allow stopping within the current segment. The vector velocity may be overridden during the move with the VO command. Increasing and decreasing velocities with the override feature will generally use the acceleration parameter to change velocities. An exception to this occurs when the final velocity of a segment warrants deceleration and the current position is close enough to the target position to warrant deceleration, in which case the deceleration parameter is used to ramp to the final velocity.

Power automatic mode is supported with this command and the auxiliary output for each axis involved in the move will be in the defined state for motion as long as the axis is in motion. If the override value is at zero, the auxiliary outputs will revert to the defined state for non-motion until the override value is non-zero. In addition, the VIO command can be used to set general purpose outputs to any value at the beginning of any vector segment.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

NOTES

1) Any stepper motor that changes direction in consecutive segments will have a practical velocity limit determined by the timing requirements of the stepper motor driver regarding the delay required between changing the direction signal and issuing the next transition on the step signal.
2) Any axis involved in a sequence of linear interpolated commands that needs to be locked out from being moved by any other multitasking process should have a parameter entered in each VP command even if the axis is being commanded to go nowhere. . For example, if the Z axis will be involved in a series of X & Y interpolated segments at some point and should not be used by any other series of segments, all X & Y VP segments should include the Z axis in the parameter list even if the Z position value does not change.

Example: In the AA mode, set the vector acceleration and deceleration to 10000 and set the vector velocity to 1500. Then move the Y, Z and T axes 10000, 100 and 1000 counts from their current positions respectively, using linear interpolation. counts from their current positions respectively, using linear interpolation. Stop at the final position.

Enter:

```
AA;
VA10000,10000; (note that this is equivalent to VA10000;)
VV1500,0;
VPR,10000,100,1000;
VE1,1,1;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	Single Axis	Not Valid	
VP[A/R]#,#...;	AA	2	55
VP[[n]][A/R]#,#...;	AM	2	55

Related commands: VA, VG, VH, VIO, VO, VV, VZ

VRC REPORT INSTANTANEOUS VECTOR ACCELERATION

Firmware Revision Required: V1.30 and greater

The VRC command will report the current instantaneous vector acceleration of a VP motion segment. This will be zero if the move is currently at constant velocity or stopped; it will be the vector acceleration value if the move is currently accelerating and it will be the vector deceleration value if the move is currently decelerating.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

Note: This command requires a terminator.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VRC;	AA	Immediate	Immediate
VRC[[n]];	AM	Immediate	Immediate

Related commands: VRV

VRV REPORT VECTOR INSTANTANEOUS VELOCITY

Firmware Revision Required: V1.30 and greater

The VRV command will report the current instantaneous vector velocity of a VP motion segment. This may differ from the programmed vector velocity if the move is accelerating or decelerating.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

NOTE: This command requires a terminator.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VRV;	AA	Immediate	Immediate
VRV[[n]];	AA/AM	Immediate	Immediate

Related commands: VRC

VSD SET VECTOR SCAN DISTANCE

Firmware Revision Required: V1.30 and greater

The VSD command is used to specify an axis that will output a step pulse at the specified distance along any VP/VC vector contour. The step distance will remain constant along the vector contour regardless of the vector velocity, including velocity override, and regardless of where the vector segment endpoints fall relative to the step pulse distance. A parameter of zero will disable the vector scan distance stepping and , in addition, will align the scan axis to the nearest whole step. The distance parameter can be less than 1.0 if a scan pulse rate greater than the vector velocity is desired.

VSD? can be used to query the current settings.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

RANGE ≥ 0

Note: The VSD step output is accurate to the number of steps per update cycle along the vector. To increase the accuracy, either lower the vector velocity and/or increase the update rate.

Example: In the AA mode, set the vector scan distance for the T axis to 1000 steps and then execute a series of vector moves. This causes the T axis to generate a compete step signal cycle every 1000 steps along the contour. There is a total of 67,855 steps along the vector contour, therefore a total of 67 step pulses will be generated, one every 1000 steps along the vector contour.

Enter:

```
AA;
VSD,,,1000;
VC0,8192,90;
VP8192,32768;
VC0,32768,180;
VP-8192,8192;
VV4000,0;
VC0,8192,90;
VE;
VSD,,,0;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VSD#,#...;	AA	1	2
VSD[[n]]#,#...;	AM	1	2

Related commands: VC, VO, VP, VV

VV SET VECTOR VELOCITY

Firmware Revision Required: V1.30 and greater

The VV command sets the starting and ending vector velocity used by VC/VP commands. These vector velocities can be changed between VC/VP commands and the new vector velocity will take effect on the next VC/VP command following the VV command. No action is taken when the command is entered, but instead it affects all subsequent VC/VP commands. The factory default is 1000 steps/ second for both starting and ending velocities. Either velocity can be entered without affecting the other parameter, however if only the first parameter is entered and it is not followed by a comma, then the ending velocity (second parameter) is set equal to the starting velocity (first parameter).

At the start of a segment, the current velocity will start ramping toward the starting velocity, up using the acceleration parameter or down using the deceleration parameter as required in order to reach the starting velocity. As the segment approaches the end-point, the velocity will ramp up using the acceleration parameter or down using the deceleration parameter as required to reach the ending velocity. If the segment is too short to reach the ending velocity with the current acceleration or deceleration setting, then it will reach whatever velocity it can in that segment and continue to try to reach the ending velocity in the next segment (assuming the ending velocity parameter has not been changed). The ending velocity can be set to zero to decelerate to a stop.

VV? can be used to query the current settings.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

RANGE: 1 ≤ VV ≤ 4,194,303

Example: Set the starting and ending vector velocity to 2500.

Enter: VV2500;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VV#[,#];	AA	Immediate	Immediate
VV[[n]]#[,#];	AM	Immediate	Immediate

Related commands: VA, VO, VP

VZ SET VECTOR STOP MODE

Firmware Revision Required: V1.30 and greater

The VZ command enables and disables the single segment mode. A parameter equal to 1 following the VZ command enables the single segment mode and a parameter equal to 0 disables the single segment mode. When single segment mode is enabled, motion stops at the end of the current segment by decelerating to zero velocity, regardless of the ending vector velocity parameter setting that was in effect when the current segment was programmed. The controller will continue on to the next segment, decelerating to zero velocity at each end-point. However, if a VH command is entered, or had been previously entered, then the controller will not proceed to the next segment command until a VG command is entered. When motion is resumed with a VG command, the controller will continue to decelerate to zero velocity at the end of each segment until this mode is reset with a VZ0; command. If no parameter is entered, the default is zero, so VZ; command with no parameter is equivalent to VZ0;.

When single segment mode is enabled with a VZ1; command, a VH command must also be issued in order for the controller to stop at the end of the segment and wait for a VG command. VZ1; without a VH command will decelerate to stop at the end of each segment and then proceed to the next segment.

If hold mode is enabled for an axis that is stopped, either mid-segment or at the end of a segment, the axis will hold to the stopped position until motion is resumed.

VZ? can be used to query current setting.

AM multi-tasking mode supports the optional specification of a task index in brackets, allowing most variable velocity contouring commands to apply to a specific task without affecting other tasks that are executing simultaneously.

Example: VP motion is disabled with the VH0; command and the single segment mode is initially enabled with a VZ1; command to stop at the endpoint of each vector segment. A VG command must be entered to start motion on the first and all subsequent VP vector segments until a VZ0; command is entered, at which time the motion will continue automatically from one segment to the next.

Enter:

```
VH0;
VZ1;
VG;
VPR1000,15000;
VPR1500,2500;
…
VZ0;
VPR2000,4000;
VPR600,800;
…
VPR300,9000;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
VZ**#**;	AA	Immediate	Immediate
VZ**[**[n]**]#**;	AM	Immediate	Immediate

Related commands: VA, VG, VH, VO, VP, VV

WA WAIT FOR AXES TO SYNCHRONIZE

The WA command, valid only in AA mode, allows a command to wait until all moves on all axes are finished before executing any further commands. Some commands which can affect a non-moving axis, such as AB and PA, may execute before a previous move on other axes has finished, especially while in a looping (LS-LE, WS-WD) mode. By preceding these commands with a WA, they will not execute until all previously defined moves have finished.

Example: The Z axis auxiliary line controls a laser beam that you only want on while the Z axis moves in a positive direction. The X and Y axes position the laser. You want to repeat the action 10 times.

Enter:

```
AA;
VL1000,1000,1000;
AC10000,10000,10000;
LS10;
MR1000,1000;
GO;
WA;
AB,,1;
MR,,500;
GO;
AB,,0;
MR,,-500;
GO;
LE;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
WA;	AA-AM	2*	0*

* This command places entries in all axes' queues in AM mode as well as AA mode.

Related commands: BW, SW, WQ, WT

#WC WATCHDOG TIMEOUT COUNTER CLEAR

The #WC command clears the watchdog time out counter and allows normal command processing to proceed, following a watchdog time out. Note if the controller's application code does not periodically refresh the watchdog timer a watchdog time out event is declared, the watchdog time out counter is incremented, the application code is restarted and all axes are reset to their power up states. As long as the watchdog timer out counter is not equal to zero all commands, except for #WC, #WS and WY, are disabled.

Example: #WC

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
#WC	Single Axis	Immediate	
#WC	AA-AM	Not Valid	

Related commands: #WS

WD WHILE END OF WS LOOP

The WD command serves as the loop terminator for the WS command.

Example: (see WS command on page 297)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
WD;	Single Axis	1	1
WD;	AA-AM	1*	1*

* In AA or AM mode, entries are made in all axes' queues.

Related commands: WS

WG WHILE END OF WH LOOP

The WG command serves as the terminator for the WH command.

Example: (see WH command page 293)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
WG;	Single Axis	1	1
WG;	AA-AM	2*	1*

* In AA or AM mode, entries are made in all axes' queues.

Related commands: CW, WH

WH WHILE

The WH command will execute all commands between it and the terminating WG command as a loop until terminated by a CW command. This allows repeated execution of a command sequence which can be terminated by the host. These commands may not be nested but may be executed sequentially.

While it is possible to nest WH/WG loops, it is not practical because a single CW commands will cause all currently executing WH/WG loops to terminate.

Example: You have a 3 axis platform that you use to drill holes in the center of a ¼ inch thick sheet of metal. The sheet is 6 inch square. The driver / motor / lead - screw pitch provide 10000 steps per inch. The operator must manually insert and remove the square from the platform. The X and Y axes move a drill into the desired position. The Z axis lifts and lowers the drill. The operator presses a switch which tells the motion controller that the square is in place and ready to be drilled. The operator will continuously remove and replace the squares until ready to take a break. The following is a description of how to set up an OMS board to perform this task.

Procedure: Connect a normally closed momentary switch between user I/O input line 0 and ground. This will be the “Ready to Drill” switch.

Enter:

```
AX;
UU10000;        *set up user units so we can reference move to
                inches
AY;
UU10000;        *10000 steps = 1 inch
AZ;
UU10000;        *10000 steps = 1 inch
AX;
VL.1;
AC10;           *set up X axis homing velocity and acceleration
AY;
VL.1;
AC10;           *set up Y axis homing velocity and acceleration
AZ;
VL.1;
AC10;           *set up Z axis homing velocity and acceleration
AX;
HR;
AY;
HR;
AZ;
HR;             *send each axis to home
AA;
VL3,3,.5;       *set normal move velocity for X, Y and Z axes
WH;             *start of loop to drill squares indefinitely
SW0;            *(operator removes/replaces square into platform)
MA3,3;          *wait until operator presses switch
GO;             *move to center of square
MA,,.5;
```

```
GO;                        *move the drill through the square (1/2 inch
                              move on the Z axis drill through the square)
MA,,0;
GO;                        *lift the drill
MA0,0;
GO;                        *move the platform to home position
WG;                        *loop back to starting WH command
(CW;)                      *operator wants a break so he/she sends CW from
                              keyboard and presses switch once more (since
                              loop will most likely be waiting for the switch at
                              this point)
                           *the loop ends and the following commands execute
MA0,0,0;
GO;                        *move to home position
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
WH;	Single Axis	1	1
WH;	AA	2*	1*
-	AM	Invalid	

*   In AA or AM mode, entries are made in all axes' queues.

Related commands: CW, LS, WG, WS

WQ WAIT FOR QUEUE TO EMPTY

The WQ command is a special command that stops the board from processing any new commands until the command queue for the current axis mode is empty; i.e. all previous moves have finished. This command is not valid in looping (LS-LE, WS-WD) modes.

Example: Move the Y axis 1,000 steps and wait until the move is complete before asking for the position.

Enter:

```
AY;
MR1000;
GO;
WQ
RP;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
WQ;	Single Axis	Immediate	
WQ;	AA-AM	Immediate	

Related commands: BW, SW, WA, WT

#WS REPORT WATCHDOG TIMEOUT COUNTER STATUS

The #WS command reports the contents of the watchdog time out counter. Note if the counter is not equal to zero at least one watchdog time out has occurred.

Example: #WS=1
One watchdog time out has occurred.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#WS	Single Axis	Immediate	
#WS	AA-AM	Immediate	

Related commands: #WC

WS WHILE USER-DEFINED INPUT IS HIGH/LOW

The WS command will execute the commands between the WS and WD commands as a loop while the specified general purpose input line is true; i.e. low (default). When the input is high at the execution of the WD command the MAX will exit the loop and execute the commands which follow the WD command. The test is at the bottom of the loop; i.e. the commands between WS and WD will always be executed at least once.

WS#**[**,*state***]**;

If the optional second parameter, *state*, is entered then it specifies the input condition to continue the loop. If the *state* parameter is zero, the loop will continue while the input is low. If the *state* parameter is non-zero, the loop will continue while the input is high. If the *state* parameter is not entered, the default behavior is to loop while the input is low.

If the input line specified is already in the specified state to exit the loop when the WS/WD loop is issued to the MAX, those commands will be executed only once.

#RANGE : 0 ≤ Parameter1 ≤ 15

Example: Execute a continuous loop, moving the X axis 10,000 counts and then move the Y axis -1000 counts, until an external device terminates the loop by setting general purpose input 1 high.

Enter:

```
AA;
WS1;
MR10000;
GO;
MR,-1000;
GO;
WD;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
WS#[,#];	Single Axis	1	1
WS#[,#];	AA	1*	1*
-	AM	Invalid	

* In AA or AM mode, entries are made in all axes' queues

Related commands: LS, WD, WH

WT WAIT

The WT command will wait for a specified number of milliseconds before proceeding with the next command in the queue. In the AA mode, all axes will wait and entries are made in all axis queues. Immediate commands will not wait since they are not queued.

RANGE: 1 ≤ WT ≤ 200,000

Example: You want to produce pulses on the X axis at 5,000 steps/second for 2 seconds, then 10,000 pulses/second for 3 seconds, and then stop.

Enter:

```
AX;
JG5000;
WT2000;
JG10000;
WT3000;
ST;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
WT#;	Single Axis	2	1
WT#;	AA-AM	2	1

Related commands: BW, SW, WA, WQ

WTC WAIT FOR COUNT

Firmware Revision Required: V1.48 and greater

The WTC command takes the form of:

WTC<count>;

The WTC command will wait until the update cycle counter reaches the specified count value before proceeding with the next command in the queue. In the AA mode, all axes will wait and count value entries are made in all axis queues. Immediate commands will not wait since they are not queued. The rate at which the update cycle counter increments is set by the #UR command, which specifies the update rate for the controller. The #UR? command will display the current update rate. The CC command displays the current update cycle counter value. The #UR? and CC commands can be used to determine the count parameter value for the WTC command.

The waiting period is limited to 7 days. If the internal current cycle counter value is outside that period the command will treat it as count reached and proceed with the next command.

RANGE: 0 ≤ WTC ≤ 4,294,967,295

Example: You want to start the X axis jogging at 10,000 steps/second when the update cycle counter reaches 307,200.

Enter:

```
AX;
WTC307200;
JG10000;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
WTC#;	Single Axis	2	1
WTC[#,#,…];	AA-AM	2	1

Related commands: CC, #UR, WT

WY WHO ARE YOU

The WY command returns the model type, firmware revision number, and number of controlled axes of the board being addressed.

Example: You want to examine the board identification string of a four axis MAX controller with its firmware revision at 1.42.

Enter: WY;

Response: MAX*-4000 ver:1.42, s/n:000217, FPGA:B5:A7 BOOT:1.03 - Oregon Micro Systems

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
WY	Single Axis	Immediate	
WY	AA-AM	Immediate	

Related commands: None

ZD PIEZO DIRECTION DELAY

Firmware Revision Required: V1.48 and greater

The set piezo motor direction change delay command takes the form of:

```
ZD<delay>;
```

The ZD command assigns the specified delay value as the number of milliseconds of delay before changing the motor rotation direction. The <delay> parameter is specified in milliseconds. Every time the piezo motor PID needs to change the motor direction, this delay is executed. If a '?' character is entered for the <delay> parameter, then the currently assigned piezo motor direction delay value will be displayed in milliseconds. The default value for the <delay> parameter is 11 milliseconds.

The resolution on the ZD delay is determined by the update rate of the controller, which is set by the #UR command. For example, a controller update rate of 1024Hz and a ZD delay of 10 milliseconds gives an actual delay of 9.765 milliseconds, where a ZD delay of 11 milliseconds gives an actual delay of 10.742 milliseconds.

The ZD command only take effect when the PSZ command has set the axis motor type to a piezo servo motor.

RANGE: 0 ≤ ZD ≤ 1000

Example: Set the T axis piezo motor direction delay to be 10 milliseconds.

Enter:
```
AT;
ZD10;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
ZD#;	Single Axis	1	1
ZD[#,#,…];	AA-AM	1	1

Related commands: PS, #UR, ZM

ZM PIEZO DAC MINIMUM

Firmware Revision Required: V1.48 and greater

The set piezo motor DAC minimum command takes the form of:

```
ZM<dac>;
```

The ZM command assigns the specified DAC value as the minimum value that will turn the motor. PID DAC values less than this minimum will cause the CW and CCW outputs to go to the off state to stop the motor. The <dac> parameter is specified in DAC output values. The piezo motor PID compares the DAC output with this value to determine if the motor needs to be stopped with the CW and CCW inputs to the piezo motor drive. The CW input is connected to the axis direction output, and the CCW input is connected to the axis auxiliary output. The difference between the ZM command <dac> value, and the KB command value defines the effective range of values the piezo servo PID has for motor speed control. If a '?' character is entered for the <dac> parameter, then the currently assigned piezo motor minimum DAC value will be displayed in milliseconds. The default value for the <dac> parameter is 1.

The ZM command only take effect when the PSZ command has set the axis motor type to a piezo servo motor.

RANGE: 0 ≤ ZM ≤ 10,000

Example: Set the Y axis piezo motor minimum DAC value to be 1022.

Enter: AT;
ZM1022;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
ZM#;	Single Axis	1	1
ZM[#,#,…];	AA-AM	1	1

Related commands: KB, PS, ZD

This page is intentionally left blank.

3. LEGACY COMMAND SUMMARY

The commands in this chapter are still supported to provide an upgrade path for users of our older motion control products. Use of these commands is discouraged in new applications as newer commands are available to perform the same tasks with greater flexibility and/or capability. These legacy commands are documented here for the sake of completeness

The legacy command functionality is provided with the standard command set, generally with increased capabilities.

LEGACY COMMANDS ALPHABETICALLY				

NON-LEGACY COMMAND	COMMAND	PAGE	Q = QUERY
			C=CMD
				COMMAND DESCRIPTION

AB?	?AB	312	Q	Report auxiliary bit state
AC?	?AC	313	Q	Report AC command setting
AD?	?AD	314	Q	Report default auxiliary bit state
AB	AF	315	C	Auxiliary off
AJ# - AJ?	?AJ	316	Q	Report customer S-curve parameters
AB	AN	317	C	Auxiliary on
A?	?AQ	318	Q	Query current axis
BC?	?BC	319	Q	Report backlash compensation
SVB	BI	320	C	Select servo voltage bipolar output mode
VC/VP/VV	CD	321	C	Contour define
VE	CE	323	C	Contour end
VC/VP	CG	324	C	Contour priority
VE	CK	325	C	Contour end and kill
AJ	CN	326	C	Cosine on
VC	CR	327	C	Circular interpolation
VO/VV	CV	328	C	Contour velocity
VC/VP	CX	329	C	Contour execute
DBI/DBN	?DB	330	Q	Report step direction bit logic
DC?	?DC	331	Q	Report deceleration rate
ER?	?ER	332	Q	Report motor:encoder ratio
ES?	?ES	333	Q	Report encoder slip tolerance
TP	FP	334	C	Force position
EG	FX	335	C	Enable axis gantry mode
GP/TP	GU	336	C	Go asymmetrical
HD?	?HD	337	Q	Report position maintenance deadband
HI	HE	338	C	Home encoder
CL	HF	339	C	Hold off
HTH	HH	340	C	Home input active high
HG?	?HG	341	Q	Report position maintenance gain

LEGACY COMMANDS ALPHABETICALLY				

NON-LEGACY COMMAND	COMMAND	PAGE	Q = QUERY
			C=CMD
				COMMAND DESCRIPTION

HTL	HL	342	C	Home input active low
HI?	?HM	343	Q	Report home state selection
CL	HN	344	C	Hold on
HI	HS	345	C	Home switch
HT?	?HS	346	Q	Report home switch true state selection
HV?	?HV	347	Q	Report stepper hold velocity
KA?	?KA	348	Q	Report PID acceleration feedforward
KB?	?KB	349	Q	Report axis PID upper bound limit
KD?	?KD	350	Q	Report PID derivative gain
KF?	?KF	351	Q	Report PID friction offset
KI?	?KI	352	Q	Report PID integral gain
KO?	?KO	353	Q	Report PID offset
KP?	?KP	354	Q	Report PID proportional gain
KU?	?KU	355	Q	Report PID integration sum upper limit
KV?	?KV	356	Q	Report PID velocity feedforward
RT	LA	357	C	Linear ramp per axis
LMF	LF	358	C	Limits off
LTH	LH	359	C	Set limits active high
LTL	LL	360	C	Set limits active low
LM?	?LM	361	Q	Report limit switch selection
LMH	LN	362	C	Limits on
LT?	?LS	363	Q	Report limit active state
VP/VV	MV	364	C	Move velocity
VO/VV	NV	366	C	New contour velocity
PA?	?PA	367	Q	Report Aux power automatic state
AJ/RT	PF	368	C	Parabolic acceleration off
ABH	PH	369	C	Aux bit high
ABL	PL	370	C	Aux bit low
CL?	?PM	371	Q	Report hold state
AJ/RT	PN	372	C	Parabolic acceleration on
AJ/RT	PR	373	C	Parabolic ramp per axis
PS?	?PS	374	Q	Report motor type
BD?	RB	375	Q	Report user-defined I/O bit direction
N/A	RQ	376	Q	Report contour queue size
RT?	?RT	377	Q	Report ramp type
AJ/RT	SC	378	C	Cosine ramp per axis

LEGACY COMMANDS ALPHABETICALLY

NON-LEGACY COMMAND	COMMAND	PAGE	Q = QUERY
			C=CMD
				COMMAND DESCRIPTION

SVV?	?SD	379	Q	Report servo DAC volts
SE?	?SE	380	Q	Report Aux power automatic settling time
LMH	SF	381	C	Soft limit off
SK?	?SK	382	Q	Report axis slip kill mode selection
LMS	SL	383	C	Soft limit on
LM?	?SL	384	Q	Report soft limit status
SVB?	?SO	385	Q	Report servo analog output mode
GP/TP	SP	386	C	Stop at position
RT	SS	387	C	Selects custom S-curve profile
SVP?	?SV	388	Q	Report servo voltage inversion state
SVP-	SVI	389	C	Invert servo voltage
SVP+	SVN	390	C	Normalize servo voltage
SK	TF	391	C	Turn off slip kill mode
TL?	?TL	392	Q	Report software overtravel limit
SK	TN	393	C	Turn on slip kill mode
EG	TX	394	C	Track the X axis
SVB0	UN	395	C	Set servo output unipolar mode
UR?	?UR	396	Q	Report the controller’s motor update rate
UU?	?UU	397	Q	Report axis user units
VB?	?VB	398	Q	Report base velocity
VL?	?VL	399	Q	Report maximum velocity setting
VP/VV	VS	400	C	Velocity streaming

3.1. LEGACY I/O CONTROL COMMANDS

3.1.1.LEGACY AUXILIARY CONTROL COMMANDS

Each axis of the MAX has an associated auxiliary output line. Though this line can be used as a general purpose output, it also has a special purpose: Power-Automatic Mode. In power-automatic mode, the auxiliary line will invert at the beginning of every motion and return to normal at the end. The “normal” state of this line is user-controllable as is the amount of time to delay, allowing the motor to settle, before returning the line to normal at the end of a move. The following commands provide this control as well as feedback regarding the state and function of each auxiliary line.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

PH	369	C	Aux bit high
PL	370	C	Aux bit low

3.2. LEGACY ENCODER COMMANDS

3.2.1.LEGACY ENCODER SLAVE MODES

Encoder tracking modes connect a motor to an axis at a given ratio. For each turn of the encoder, the motor will move proportionately.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

TX	394	C	Track the X axis

3.2.2.LEGACY PROFILE COMMANDS

Often, the default linear acceleration profile is not optimum for a given system. To meet the needs of those systems, the MAX has a number of commands that allow partial or even complete customization of the profile. The commands below allow the use of parabolic, cosine, and even custom ramps.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

CN	326	C	Cosine on
PF	368	C	Parabolic acceleration off
PN	372	C	Parabolic acceleration on
PR	373	C	Parabolic ramp per axis
SC	378	C	Cosine ramp per axis

3.2.3. LEGACY MOVE EXECUTION COMMANDS

The following commands initiate moves defined by commands in section Move Specification Commands. A number of different commands are available, tailored to various application needs.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

GU	336	C	Go asymmetrical

3.2.4. VELOCITY STAIRCASING COMMANDS

The following commands describe the velocity staircase mode. This mode is useful in applications requiring a change in velocity at a prescribed position without stopping. Similar to the jogging commands, velocity stair casing will move an axis at a specified velocity. The difference is that the next stair casing command in the queue will not be processed until a specified position is reached. Stair casing also allows the host to specify a position for the motor to stop, unlike the jogging commands.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

FP	334	C	Force position
MV	364	C	Move velocity
SP	386	C	Stop at position

3.2.5. VELOCITY STREAMING COMMANDS

Velocity streaming is a specialized form of velocity stair casing. Streaming simply produces specified velocities without ramping or other processing. In effect, streaming allows the host to put velocities directly into the MAXs' internal velocity registers for the X and Y axes.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

VS	400	C	Velocity streaming

3.2.6. CONSTANT VELOCITY CONTOURING COMMANDS

The MAX will attempt to generate any profile which it is asked to do. It is the responsibility of the host to be sure the acceleration required when generating a circle or any other change in direction is possible within the mechanical constraints of the system. All corners must be defined by arcs and tangents to those arcs, else the change in direction will be instantaneous and generate very large accelerations. The arc radius must be chosen so that the acceleration constraints of the system are met.

Constant velocity contouring is similar to a series of discrete move commands except that it allows multiple discrete moves to be executed in series, without stopping, maintaining a constant vector velocity among the involved axes. The commands below are those that are available in contouring mode. There is just one contour queue on MAX to handle all axes commands while in contour mode. The size of the contour queue is 32763.

COMMAND	**PAGE**	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

CD	321	C	Contour define
CE	323	C	Contour end
CG	324	C	Contour priority
CK	325	C	Contour end and kill
CR	327	C	Circular interpolation
CV	328	C	Contour velocity
CX	329	C	Contour execute
NV	366	C	New contour velocity
RQ	376	Q	Report contour queue size

3.3. LEGACY I/O CONTROL COMMANDS

3.3.1. LEGACY AUXILIARY CONTROL COMMANDS

Each axis of the MAX has an associated auxiliary output line. Though this line can be used as a general purpose output, it also has a special purpose: Power-Automatic Mode. In power-automatic mode, the auxiliary line will invert at the beginning of every motion and return to normal at the end. The “normal” state of this line is user-controllable as is the amount of time to delay, allowing the motor to settle, before returning the line to normal at the end of a move. The following commands provide this control as well as feedback regarding the state and function of each auxiliary line.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

AF	315	C	Auxiliary off
AN	317	C	Auxiliary on
?SE	380	Q	Report Aux power automatic settling time

3.3.2. LEGACY HOMING COMMANDS

Homing commands detail the commands available for customizing and initiating homing operations.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

HE	338	C	Home encoder
HH	340	C	Home input active high
HL	342	C	Home input active low
?HM	343	Q	Report home state selection
HS	345	C	Home switch
?HS	346	Q	Report home switch true state selection

3.3.3. LEGACY LIMIT CONTROL COMMANDS

Limit conditions are treated as critical errors in the MAX. When a limit is encountered, the axis involved will cease motion and flush any pending motion commands for that axis. However, since needs vary from application to application, the following commands will allow limit behavior customization to fit almost any system.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

LF	358	C	Limits off
LH	359	C	Set limits active high
LL	360	C	Set limits active low
?LM	361	Q	Report limit switch selection
LN	362	C	Limits on
?LS	363	Q	Report limit active state
SF	381	C	Soft limit off
SL	383	C	Soft limit on
?SL	384	Q	Report soft limit status

3.4. LEGACY SERVO CONTROL COMMANDS

The following commands are valid only for servo axes and should never be executed while the specific axis is in motion.

3.4.1. LEGACY SERVO VOLTAGE CONTROL COMMANDS

Different servo amplifiers have different requirements for their control inputs. Some simply behave differently despite similar input requirements. To enable the use of a wide range of amplifiers, the MAX will accept the following commands for use in configuring servo outputs.

COMMAND	PAGE	Q = QUERY
		C=CMD
			COMMAND DESCRIPTION

BI	320	C	Select servo voltage bipolar output mode
?SO	385	Q	Report servo analog output mode
?SV	388	Q	Report servo voltage inversion state
SVI	389	C	Invert servo voltage
SVN	390	C	Normalize servo voltage
UN	395	C	Set servo voltage unipolar output mode

4. LEGACY COMMAND DESCRIPTIONS

?AB REPORT AUXILIARY BIT STATE

This command returns the logic state of the auxiliary output of the current axis, (AN or AF).

Example: Determine if the X axis auxiliary output is set on.

Enter: AX;
?AB

Response: =on<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?AB	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: AF, AN, PA, ?SE, SE

?AC REPORT AC COMMAND SETTING

This command will reply with the current setting for the maximum acceleration value for the selected axis.

Example: Report the current AC value for this axis.

Enter: AC125000;
?AC

Response: =125000<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?AC	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: AC, RC

?AD REPORT DEFAULT AUXILIARY BIT STATE

This command reports the power up default selection for the auxiliary output of the current axis.

Example: Report the power up state of the Y axis auxiliary output

Enter:
```
AY;
?AD
```

Response:
```
=on<LF>
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?AD	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: AD

AF AUXILIARY OFF

The AF command turns off the selected auxiliary output. That is, it causes the signal to be driven low. The AF command may be used to change power level on driver modules so equipped or as a user specified output.

A parameter must be supplied for the desired axes when used in the AA mode so that the other axes are not affected. No parameter is required in a single axis mode. Note this command will turn power automatic (PA) mode off.

Example: Turn off the Y axis auxiliary output in the single axis mode.

Enter: AY;
AF;

Response: None

Example: Turn off the X and Z axes auxiliary outputs when in the AA command mode. The Y axis is unchanged in this example.

Enter: AA;
AF1,,1;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AF;	Single Axis	1	0
AFb,b,b,b,b,b,b,b;	AA-AM	1	0

*When AF is used in a contour definition the Aux bits of all axes included in that definition will be turned off when the contour is executed.

Related commands: AN, BH, BL, BS, PA

?AJ REPORT CUSTOM S-CURVE PARAMETERS

This command reports the parameters for a given custom S-curve profile.

Range: 1 ≤ Profile Number ≤ 8

Example: Report the parameters for custom S-curve profile number 5.

Enter:

```
AX;
?AJ5;
```

Response:

```
=5.0000000,0.50000000,0.50000000,1.00000000,0.10000000,
0.10000000, 1.00000000<LF>
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?AJ#;	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: AJ, ?RT, SS

AN AUXILIARY ON

The AN command turns on the selected auxiliary output. That is, it allows the signal to be pulled high. This is the default mode for the auxiliary line at power up or reset. The AN command may be used to change power level on driver modules so equipped, trigger another board's input or as a user specified output.

A parameter must be supplied for the desired axes when used in the AA mode so that the other axes are not affected. No parameter is required in a single axis mode. Note this command will turn power automatic (PA) mode off.

Example: Turn on the Y axis auxiliary output in the single axis mode.

Enter:
AY;
AN;

Response: None.

Example: Turn on the X and Z axes auxiliary outputs when in the AA command mode. The Y axis is unchanged in this example.

Enter:
AA;
AN1,,1;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AN;	Single Axis	1	0
ANb,b,b,b,b,b,b,b;	AA-AM	1	0

*When AN is used in a contour definition the aux bits of all axes included in that definition will be turned on when the contour is executed.

Related commands: AF, BH, BL, BS, PA

?AQ QUERY CURRENT AXIS

The ?AQ command reports the mode that the current axis is in, i.e. AA mode, AM mode, etc.

Example: Determine what mode the X axis is in.

Enter:
```
AX;
?AQ
```

Response: =ax<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?AQ	Single Axis	Immediate	
?AQ	AA-AM	Immediate	

Related commands: None

?BC REPORT BACKLASH COMPENSATION

The ?BC command reports the backlash compensation factor for the currently active axis. This is a numeric value of the number of steps added.

Example: Determine backlash compensation factor of axis X

Enter:
```
AX;
?BC
```

Response: =23<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?BC	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: AN, AF, BC, BL, BS, BX

BI SELECT SERVO VOLTAGE BIPOLAR OUTPUT MODE

The BI command sets the analog servo output of the current axis to bipolar. When bipolar is selected, a zero torque reference will result in a 0VDC output (+/- offset voltage). The analog output will range between +10VDC and -10VDC when bipolar is enabled. The BI command is valid only in the single axis mode and is the default mode at power up or reset. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

Example: Set up servo axis X for bipolar operation.

Enter: AX;
BI;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
BI;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: DBI, DBN, ?SO, SVI, SVN, UN

CD CONTOUR DEFINE

The CD command enters contour definition mode and defines a constant velocity contour. The only way to exit this mode is to issue a CE or CK command. Commands following the CD command must be in multi-axis format and be commands valid in CD mode. All multi-axis commands entered in CD mode can address only those axes assigned in the CD command. No commands entered will be executed until a CX command is received which must be issued outside of CD mode.

The CD command takes up to eight parameters. Each parameter specifies a starting point for each axis to be involved in the contour in absolute coordinates. If an axis is not to be involved in the contour, its parameter position should be skipped just as it would be in any other multi-axis command. Commands issued within CD mode must be formatted to include only those axes indicated in the CD command. Those that are not in the CD command simply do not exist in the formatting of commands entered in CD mode. For example, if a CD command is issued that uses the X, Y, and T axes (such as CD100,300,,400;), commands entered within CD mode will only consider the X, Y, and T axes. A MT command that moves X to 200, Y to 600, and T to 200 would take the form: MT200,600,200;. Note the lack of a placeholder comma for the Z axis.

Contours that will include circular interpolation (CR) must be defined for only 2 axes in the CD command. Contours involving more than 2 axes may not use the CR command. The size of the contour definition buffer for the MAX is 32,763 positions.

When the contour is executed, the MAX will use the distance between the current position and the contour starting point to linearly ramp up each axis such that all involved axes reach a combined, vectored velocity equal to the value set with the CV command. If this distance is zero, no ramp will be generated resulting in an instantaneous jump to contour velocity. Most stepper systems cannot achieve this and servos will tend to oscillate wildly before settling down if at all. Care should be taken to allow sufficient ramping distance between the contour starting position and the current position when the CX command is issued.

Once the contour is completely executed, the MAX will ramp the axes to a stop using the rate defined with the AC command. This ramp down will take each axis beyond the final point of the contour. Without manually calculating the ramp down distance for each axis, there is no way to force the contour to come to a complete stop at a predetermined point.

Example: The following demonstrates cutting a hole with a 10,000 count radius using constant velocity contouring and circular interpolation. The contouring velocity is set to 1000 pulses per second. A contour is then defined beginning at coordinates 0,0 on the X and Y axes. The auxiliary output of the Z axis is turned on, which could turn on the cutting torch or laser starting the cut at the center of the circle. A half circle is cut from the center to the outside of the hole, positioning the cutting tool at the start of the desired hole. The hole is then cut, the torch turned off, the stage stopped and the definition is complete. The stage is then positioned and the hole cut with the CX command. The AN and AF commands must have commas for all axes since they can all be addressed from within the contour definition.

Enter:

```
AA;
CV1000;
CD0,0;
```

```
AN,,1;
CR5000,0,-3.1415926;
CR0,0,-6.2831853;
AF,,0;
MT10000,-1000;
CE;
MT10000,0;
GO;
CX;
```

Response: None.

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
-	Single Axis	Not Valid		
CD#,#,#,#,#,#,#,#;	AA-AM	5 + number of axes in the contour		
-	AA/CD	N/A		

Related commands: AF, AN, BH, BL, CE, CK, CR, CV, CX, NV, MT

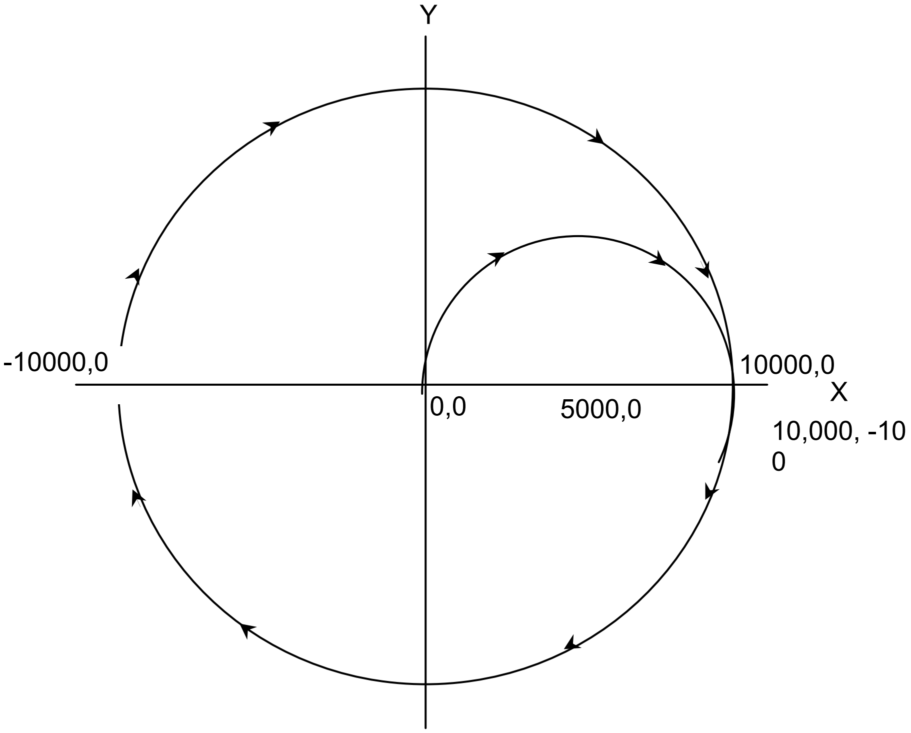

FIGURE 4-1

CE CONTOUR END

The CE command marks the end of the contour sequence. It will terminate the CD mode and, when executed, ramp to a stop and exit to the AA command mode. The end of the contour should contain at least a short linear segment just prior to the CE command to initialize the parameters for the deceleration of the stage.

Example: (see CD command on page 321)

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
-	Single axis	Not Valid		
-	AA-AM	Not Valid		
CE;	AA/CD	N/A	N/A	2

Related commands: CD, CK

CG CONTOUR PRIORITY

The CG command is a form of the CX command, use if CE is followed by a MT command. When CG is used to execute the contour, the following MT commands will be on hold until the contour execution is complete. After the contour has been executed, the MT commands that follow can be parsed.

CG is preferred over the CX when the MT commands are issued after a contour is executed. The CG command ensures that the MT command that follows starts from a known position. This makes for a more accurate calculation of the MT move.

Example: See page 321 (CD), Make sure that the hole is completely cut before executing the move to the new command position.

Enter:

```
AA;
CV1000;
CD0,0;
AN,,1;
CR5000,0,-3.1415926;
CR0,0,6.2831853;
AF,,0;
MT10000,-1000;
CE;
CG;
MT-10000,0;
GO;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
CG;	AA-AM	7*	2*
CG;	AA/CD	Not Valid	

* If PA (power automatic) mode is selected add 2 to the command queue.
* If an aux bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.
* If the axis is stepper and encoder or servo add 1 to the command queue.
* Add the following queue requirements for the ramp types listed.

QUEUE REQUIREMENTS		
Axis Ramp Type	**COMMAND QUEUE**	**ARGUMENT QUEUE**
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	8
All other Parabolic Forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

Related commands: AF, AN, BH, BL, CD, CK, CR, CV, CX, NV, MT

CK CONTOUR END AND KILL

The CK command will end the contour sequence, like the CE command, except there is no ramp down; i.e. motion will stop abruptly. It is used in place of the CE command.

NOTE: This command should be used with caution to prevent the stage from slipping or losing its correct position.

Example: Same scenario as CD command, but we want to end the contour with the minimum ramp down.

Enter:

```
AA;
CV1000;
CD0,0;
AN,,;
CR5000,0,-3.1415926;
CR0,0,-6.2831853;
AF,,0;
MT10000,-1000;
CK;
MT-1000,0;
GO;
CX
```

Response: None.

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
-	Single Axis	Not Valid		
-	AA-AM	Not Valid		
CK;	AA/CD	N/A	N/A	2

Related commands: CD, CE

CN COSINE ON

The CN command enables cosine velocity ramps; i.e. half sinusoid acceleration profiles, for all axes. The cosine profile is not truncated in moves that cannot reach full velocity, but instead the velocity is reduced sufficiently to preserve the cosine profile. This command should not be given while an axis is in motion or the results may be unpredictable. This command affects all axes even if issued in the single axis mode. The PF command is used to return to linear motion profiles. See the APP command to preserve the CN setting as the Power up/Reset ramp.

Example: Set the board to be in cosine mode.

Enter: CN;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
CN;	Single Axis	5	24
CN;	AA-AM	5	24
-	AA/CD	Not Valid	

Related commands: LA, PF, PN, PR, ?RT, SC, SR

CR CIRCULAR INTERPOLATION

The CR command defines a move in a circular pattern from the entry position. The first two parameters are the center of the circle in absolute units and the third parameter is the distance to move in radians. Positive radians equal counterclockwise movement. Negative radians equal clockwise movement. The radius of the circle is the linear distance between the current position and the first two parameters of the CR command.

The CR command generates a circle by breaking the circle into a series of linear segments. The number of segments will be equal to the total distance traveled divided by the contour velocity divided by the update rate. If finer resolution is required, the MT command may be used to produce smaller segments but the segments must be calculated by the user

RANGE:
Min Pos. Value ≤ Parameter 1 (First Coordinate for Center of Circle) ≤ Max Pos. Value
Min Pos. Value ≤ Parameter 2 (Second Coordinate for Center of Circle) ≤ Max Pos. Value

Example: (see CD command on page 321)

QUEUE REQUIREMENTS				
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**	**CONTOUR**
-	Single Axis	Not Valid		
-	AA-AM	Not Valid		
CR#,#,#;	AA/CD	N/A	N/A	8

Related commands: CD, MT

CV CONTOUR VELOCITY

The CV command allows specification of a contouring velocity. It is executed from the AA mode before a contour definition. A contour defined by a CD command cannot be executed if followed by a CV command. Changing this parameter will make any previously defined contours invalid. The contour velocity defaults to 1000 at power up or reset. Use WQ between contour definitions to avoid having a CV associated with a second contour definition affect a prior contour still in motion. A CV cannot be issued between a CD and CE command. To change the contour velocity within a contour definition use the NV command.

RANGE: $1 \leq CV \leq 4{,}194{,}303$

Example: (see CD command on page 321)

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
CV#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: CD, NV

CX CONTOUR EXECUTE

The CX command will execute the previously defined contour sequence. The stage must be positioned such that it can accelerate to speed by the absolute position specified by the CD command it is executing and must be traveling in the proper direction. Once a contour is defined it may be executed at any time by executing a CX command until it is replaced by another contour definition. The CX command cannot be placed within a loop or while construct.

Example: (see CD command on page 321)

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
CX;	AA-AM	6*	2*
-	AA/CD	Not Valid	

- * If the axis is a stepper and encoder or servo add 1 to the command queue.
- * If PA (power automatic) mode is active add 2 to the command queue.
- * If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

QUEUE REQUIREMENTS		
Axis Ramp Type	**COMMAND QUEUE**	**ARGUMENT QUEUE**
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	8
All other Parabolic Forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

Related commands: CD, CE, CK, ID

?DB REPORT STEP DIRECTION BIT LOGIC

The ?DB command reports the direction bit logic: inverted or normal. If the direction bit is low when moving positive, this command will return ‘n’ for ‘normal’. If the direction bit is high when moving positive, this command will return ‘i’ for inverted..

Example: Report whether the direction bit for the T axis is low or high when making positive moves

Enter: AT;
?DB

Response: =i<LF> (The inverted result indicates the T axis direction bit is high for positive moves)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?DB	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: DBI, DBN, ?SV

?DC REPORT DECELERATION RATE

The ?DC command reports the deceleration rate that has been defined by the DC command or the AC command.

RANGE= 1≤ DC ≤ 8000000

Example: Determine deceleration rate of none symmetric motion profiles of X axis.

Enter:
```
AX;
?DC
```

Response:
```
=20000<LF >
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?DC	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: AC, DC, GP, TP

?ER REPORT MOTOR:ENCODER RATIO

The ?ER command reports the motor-to-encoder ratio as set with the ER command.

Example: Find out what the last ER command sent was.

Enter: ?ER

Response: =2.000000<LF>
(The encoder produces 1 count for every 2 steps of the motor.)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?ER	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: ER

?ES REPORT ENCODER SLIP TOLERANCE

The ?ES command reports the current value of the slip detect tolerance assigned to an axis.

Example: Report the current value for encoder slip detection tolerance

Enter: ?ES

Response: =15,15<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?ES	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: ES

FP FORCE POSITION

The FP command will flush the command queue and attempt to stop at the specified position. The axis will overshoot if there is insufficient distance left to stop at the programmed acceleration. This command should not be given to a servo axis while it is in motion; the results may be unpredictable. Force the axis to stop at a specified position.

Example: Force axis to stop at 25,000.

Enter: FP25000;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
FP#;	Single Axis	2*	1*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

* If PA (power automatic) mode is active add 1 to the command queue

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: MM, MP, MV, SP

FX ENABLE AXIS GANTRY MODE

The FX command pairs the current axis with the X axis to form a gantry crane. (i.e. the axis follows the X axis.)

Example: Pair the X and Y axis to form a gantry crane.

Enter:
```
AY;
FX;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
FX;	Single axis	1	0
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: None.

GU GO ASYMMETRICAL

The GU command initiates a previously defined move using the AC value for acceleration and the DC value for deceleration. This command may be used with only one axis at a time; i.e. it is not valid with the ML and MT commands. For multi-axis move with asymmetrical acceleration and deceleration, see the VA and VP commands.

If this command is issued without having defined a move, the results are undefined. Issuing a GU command to execute an already-executed move also has undefined results. Only one GU command should be issued per defined move. GU is used only with linear acceleration ramps. Use S-curve command (see AJ, SS) for defining more complex asymmetrical motion profiles.

Example: Move the Y axis to position 1,500 using the current acceleration and velocity and a deceleration of 5,000 counts per second per second.

Enter:
AY;
DC5000;
MA1500;
GU;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
GU;	Single Axis	3*	0*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

* If the axis is a stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.
* If PA (power automatic) mode is active add 2 to the command queue
* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: AC, DC, GD, GN, GO, GS, MA, MR, VA, VP

?HD REPORT POSITION MAINTENANCE DEADBAND

The ?HD command reports the current settings of the HD command.

Example: Find out what HD was last set to.

Enter: ?HD

Response: =5,20,1500<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?HD	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: HD, ?HG, ?HV

HE HOME ENCODER

The HE command enables encoder index, and phase A and B signals to be considered in addition to the home switch input when an HM or HR command is executed. This command does not execute the home motion which must be initiated with an HM, HR, KM, or KR command. Default home behavior (considering only the home switch for home commands) may be reestablished via the HS command. In the HE mode, home is defined as the logical AND of the encoder index, the external home enable and the encoder quadrant where channel A is positive and channel B is negative. The default home logic expressed in Boolean terms is:

home = phase_A ∗ / phase_B ∗ index ∗ / home_switch

The default logic can be changed to any required combination with the use of HL and HH commands to set the true home switch state and the EH commands to set the encoder signal (Index, A and B) states required for a true home condition.

Example: Set up the Y axis so it will use the encoder signals to recognize the home position.

Enter:
AY;
HE;
or
AA;
HE,1;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HE	Single Axis	Immediate	
HEb,b,b,b,b,b,b,b;	AA-AM	Immediate	

Related commands: EH, HH, HL, HM, HR, HS, KM, KR

HF HOLD OFF

The HF command disables position hold, stall detection and tracking modes, for stepper with an encoder axis. If the current axis is a servo, this command will open the loop and turn off the PID. If the current mode is multi-axis, then the selected axes will go into open-loop mode. Hold off is the default mode at power up or reset.

Example: Turn off encoder hold mode on the X axis.

Enter:
```
AX;
HF;
```

Response: None

NOTE: In AA or AM mode, a null value in the argument list specifies feedback for that axis is not to be disabled.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
HF;	Single Axis	3	0
HFb,b,b,b,b,b,b,b;	AA-AM	3	0

Related commands: HN, ?PM

HH HOME INPUT ACTIVE HIGH

The HH command sets the sense of the home switch on the current axis to active TTL high. This command allows TTL logic high to be treated as the “true” state for applications where this is more convenient. Once this command has been sent to the MAX, active high homes can be made the power-up default by using the APP command; refer to the APP command for details. This command sets the home switch “true” state to TTL logic high for both the HS and HE modes of the home operation.

Example: (see HL command on page 342)

Note: In AA or AM modes, a null argument in the parameter list specifies that axis home switch to be unchanged.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
HH;	Single Axis	1	0
HHb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: EH, HE, HL, HM, HR, HS, KM, KR

?HG REPORT POSITION MAINTENANCE GAIN

The ?HG command reports the current setting of the position maintenance hold gain constant for the current axis. This command works only with stepper + encoder axes.

Example: Position corrections seem slow. Check the setting of HG to be sure it is correct.

Enter: ?HG

Response: =100<LF >

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?HG	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA	Not Valid	

Related commands: ?HD, HG, ?HV

HL HOME INPUT ACTIVE LOW

The HL command sets the sense of the home switch on the current axis to active TTL low. This is the power-up and reset default “true” state unless HH has been saved as a user power-up default (See the APP command.). This command sets the “true” state of a home input to a TTL logic low.

This command sets the home switch “true” state to TTL logic low for both HS and HE modes of home operation.

Example: The stage is moved through home with the home switch set for active low at low speed to meet the less than 2048 steps per second requirement of the home command.

Enter:

```
AX;
VL2000;
HL;
HM0;
```

Response: None

Note: In AA or AM modes, a null argument in the parameter list specifies that axis home switch to be unchanged.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
HL;	Single Axis	1	0
HLb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: EH, HE, HH, HM, HR, HS, KM, KR

?HM REPORT HOME STATE SELECTION

Reports homing mode selected for an axis. The possible responses are ‘switch’ for home switch (HS) mode and ‘encoder’ for home encoder (HE) mode.

Example: Report the homing mode assigned to the X axis.

Enter: AX;
?HM

Response: =s<LF> or =e<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?HM	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: HE, HH, HL, HM, HR, HS, KM, KR

HN HOLD ON

For servo axes, the HN command closes the loop, enabling the PID. For servo axes, this mode is disabled when the HF command is entered, when the servo error becomes too large or a limit is encountered.

For stepper with encoder feedback axes, the HN command enables position correction after a move and activates the HD, HG and HV commands for stepper axes with encoders. For stepper axes with encoders, this mode will be canceled (as though via an HF command) if an JG, JF, HM, HR, SO, SP command is entered, if a limit is encountered or the maximum allowable position error is executed.

Example: (Stepper) The following commands could be used to set up the position correction mode on a stepper axis. This sequence sets up a move velocity of 100,000 steps per second and an acceleration of 500,000 steps per second per second. The position correction velocity is set for 50,000 steps per second, a dead band of 10 steps and correction gain of 2,000. The correction is then enabled. A 200,000 step move is performed, then that position is maintained within the 10 step dead band until commanded to a new position.

Enter:

```
AX;
VL100000;
AC500000;
HV50000;
HD10;
HG2000;
HN;
MR200000;
GO;
```

Response: None

Example: (Servo) Close PID loop or X axis after having modified one of the PID parameters.

Enter:

```
AX;
HN;
```

Response: None

NOTE: In AA or AM mode, a null value in the argument list specifies that encoder feedback is not to be enabled for that axis.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HN;	Single Axis	1*	0
HNb,b,b,b,b,b,b,b;	AA-AM	1*	0

* Values in table are for a stepper with encoder axis. For a servo axis the command queue requires 2, and the argument queue requires 1.

Related commands: HD, HF, HG, HV, ?PM

HS HOME SWITCH

The HS command disables HE mode and returns the MAX to the default home behavior. Default behavior defines a home state to be active when the home switch input is either low in default or HL mode or high when in HH mode. This mode can also be used with encoders which contain internal home logic by connecting their output to the MAX home input for the appropriate axis. The active level of this input may be controlled by the HH and HL commands.

Example: Set up the Y axis so it will ignore the encoder signals and only use the home input to recognize the home position.

Enter:
```
AY;
HS;
```
or
```
AM;
HS,1;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
HS	Single Axis	Immediate	
HSb,b,b,b,b,b,b,b;	AA-AM	Immediate	

Related commands: HE, HH, HL, HM, HR, KM, KR

?HS REPORT HOME SWITCH TRUE STATE SELECTION

The ?HS command reports if the true state of the home switch is set to be active high or low.

Example: Determine if true home switch selection of Z axis is high.

Enter: AZ;
?HS

Response =h<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?HS	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: HE, HH, HL, HM, HS, KM, KR, LO, LP

?HV REPORT STEPPER HOLD VELOCITY

The ?HV command reports the current setting of the position maintenance hold velocity limits for the current axis. This command works only with stepper + encoder axes.

Example: Check the peak correction velocity for the T axis

Enter: AT;
?HV

Response: =20000<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?HV	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: ?HD, ?HG, HV

?KA REPORT PID ACCELERATION FEEDFORWARD

The ?KA command reports the current setting of the acceleration feedforward constant (KA) for the current servo axis. Default value is 0.00

Example: Find out what the current KA value is for servo axis Y

Enter:

```
AY;
?KA;
```

Response:

```
=10.50<LF>
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?KA	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: KA, ?KV

?KB REPORT AXIS PID UPPER BOUND LIMIT

The ?KB command requests the value of the KB parameters for the upper bound limit to the DAC.

Example: Determine PID upper bound limit for X axis

Enter: AX;
?KB

Response: =5.00<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KB	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: DBI, DBN, ?SV

?KD REPORT PID DERIVATIVE GAIN

The ?KD command reports the current setting of the derivative gain coefficient (KD) in the PID of the current servo axis.

Example: Forgot to write down the Y axis KD setting which is working well. Report the setting so it can be recorded.

Enter:

```
AY;
?KD
```

Response:

```
=5.12<LF >
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?KD	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: KD, ?KI, ?KP

?KF REPORT PID FRICTION OFFSET

The ?KF command reports the value of the frictional offset coefficient.

Example: Verify Z axis friction offset coefficient is 12.

Enter:
```
AZ;
?KF
```

Response:
```
=12<LF>
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KF	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: KF, ?KI, ?KP

?KI REPORT PID INTEGRAL GAIN

The ?KI command reports the current setting of the integral gain constant (KI) in the PID of the current servo axis.

Example:	Report the setting of the KI coefficient on the Z axis
Enter:	AZ; ?KI
Response:	=1.00<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KI	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: ?KD, KI, ?KP

?KO REPORT PID OFFSET

The ?KO command reports the voltage offset (KO) setting for the current servo axis.

Example: The open-loop offset is 218. Make sure the closed-loop offset is the same.

Enter: ?KO

Response: =218<LF >

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KO	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: KO

?KP REPORT PID PROPORTIONAL GAIN

The ?KP command reports the current setting of the proportional gain coefficient (KP) in the PID of the current servo axis.

Example: Find out what the X axis proportional gain is set to.

Enter:
```
AX;
?KP
```

Response:
```
=10.00<LF >
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?KP	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: ?KD, ?KI, KP

?KU REPORT PID INTEGRATION SUM UPPER LIMIT

Report servo axis PID integration sum upper limit.

Example: Check to make sure the integration sum upper limit of X axis is less than 200.

Enter: AX;
?KU

Response: =150<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?KU	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: HF, HN, KD, KP, KU

?KV REPORT PID VELOCITY FEEDFORWARD

The ?KV command reports the current velocity feedforward coefficient (KV) of the current servo axis. Default value is 0.00

Example: Make sure the velocity feedforward setting of axis T is zero

Enter:
```
AT;
?KV
```

Response: =0.00<LF>

<table>
<tr><th colspan="4">QUEUE REQUIREMENTS</th></tr>
<tr><th>FORMAT</th><th>MODE</th><th>COMMAND</th><th>ARGUMENT</th></tr>
<tr><td>?KV</td><td>Single Axis</td><td colspan="2">Immediate</td></tr>
<tr><td>-</td><td>AA-AM</td><td colspan="2">Not Valid</td></tr>
</table>

Related commands: ?KA, KV

LA LINEAR RAMP PER AXIS

The LA command specifies that the linear acceleration ramp is to be used by the selected axes. This is the factory default for all axes. See the APP command to preserve the LA settings as the power up/reset values.

Example: Select a linear ramp for the X axis.

Enter:

```
AX;
LA;
```

Response: None

Example: Select the linear ramp for the Y and T axes.

Enter:

```
AA ;
LA,1,,1;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LA;	Single Axis	5	6
LAb,b,b,b,b,b,b,b;	AA-AM	5	6

Related commands: ?RT, SR

LF LIMITS OFF

The LF command disables the limit switches for the addressed axis or axes. This allows the stage to move beyond the limit switches and should be used with caution. The MAX will still recognize that a limit switch has been closed if the stage runs into one and will report this information via the query commands (See the QA command). However, the limit switch closure will have no effect on motion; i.e. the axis will not be forced to stop as a result.

NOTE: In systems not designed to handle motion beyond the limit switch points, this can potentially cause damage to the system and/or persons operating the system. This command should be used with extreme caution.

Example: Set up a board to ignore the Y axis limit switches.

Enter:
AY;
LF;
or
AA;
LF,1;

Response: None

Note: In AA or AM modes, a null argument in the parameter list specifies that axis home switch to be affected.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LF;	Single Axis	1	0
LFb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: LH, LL, LN, SF, SL

LH SET LIMITS ACTIVE HIGH

The LH command sets the senses of the limit switches on the current axis to active high. The default “true” states of the limits are TTL logic low. This command allows TTL logic high to be treated as the “true” state for applications where this is more convenient. Through the execution of the APP command, limits can be made to default as active high on power-up or reset; see the APP command for details.

Example: Select the limit switch high true condition for the X axis.

Enter:
```
AX;
LH;
```

Response: None

Example: Select a high true limit condition for the Z and T axes.

Enter:
```
AA;
LH,,1,1;
```

Response: None

Note: In AA or AM modes, a null argument in the argument list specifies that axis is not to be affected.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LH;	Single Axis	1	0
LHb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: LF, LL, LN, ?LS, SF, SL, TL

LL SET LIMITS ACTIVE LOW

The LL command specifies that over travel occurs when the limit input signal is low (active low). This is the factory default mode. See the APP command for information on how to preserve the LL settings as the power up/reset values.

Example: Select the limit switch low true condition for the X axis.

Enter: AX;
LL;

Response: None

Example: Select a low true limit condition for the Y and T axes.

Enter: AA;
LL,1,,1;

Response: None

Note: In AA or AM modes, a null argument in the argument list specifies that axis is not to be affected.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LL;	Single Axis	1	0
LLb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: LF, LH, LN, ?LS, SF, SL, TL

?LM REPORT LIMIT SWITCH SELECTION

The ?LM command reports the limit detection enable mode of an axis.

Example: Query MAX to learn if limit detection is enabled for the Z axis.

Enter:

```
AZ;
?LM
```

Response: =On<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?LM	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: LF, LN

LN LIMITS ON

The LN command restores the operation of the limit switches for the addressed axis or axes. This is the default mode at power up or reset. With limits enabled, if the axis encounters a limit condition in the course of executing any motion in the direction of the limit, the axis will be instructed to stop, the axis queue will be flushed, and the host will be notified of the event. Limit conditions are treated as critical errors and should not be used as simple positioning inputs to the MAX.

Example: Set up the Y and T axes to stop immediately when a limit switch is encountered.

Enter:

```
AA;
LN,1,,1;
or
AY;
LN;
AT;
LN;
```

Response: None

Note: In AA or AM modes, a null argument in the parameter list specifies that axis is not to be affected.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LN;	Single Axis	1	0
LNb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: IOE, LF, LH, LL, SF, SL

?LS REPORT LIMIT ACTIVE STATE

The ?LS command reports the active state of the limits for the current axis. The LL and LH commands are used to set this value.

Example: Find out whether the Y axis limits are active high or active low.

Enter:
```
AY;
?LS
```

Response: =l<LF> or =h<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?LS	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: LH, LL

MV MOVE VELOCITY

The MV command causes the current axis to move to a new absolute position (parameter 1) at a new velocity (parameter 2). When the destination is reached control will be passed to the next command which should be another MV command or a SP command. If the command is not received in time the controller will continue to move at the specified velocity. Note that this is a slave mode and it is the responsibility of the user to provide the commands in time. They may be queued ahead of time. If a new MV command is sent after the controller has already passed the destination specified in the command, the controller will continue to move at the old velocity.

The MAX will ramp up or down as needed at the rate previously set with the AC command and travel at the new velocity until the new position is reached. The controller will not reverse direction if the position has already passed, but will behave as explained above. Thus the direction of the move must be specified before starting the move with the MP or MM commands. All destinations must be in absolute position; no position relative moves are allowed due to the nature of these commands. Cosine and parabolic acceleration will not apply.

RANGE:

Min Position Range ≤ Parameter 1 (Absolute Position) ≤ Max Position Range

1 ≤ Parameter 2 (Velocity) ≤ 4,194,303

NOTE: The position range depends on the settings used on MAX, see specifications for more details.

Example: Generate a velocity staircase with the breakpoints given in absolute position. Default acceleration (AC) of 200,000

Enter:

```
MP;
MV10000,30000;
MV20000,50000;
MV30000,10000;
SP35000;
```

Response: None

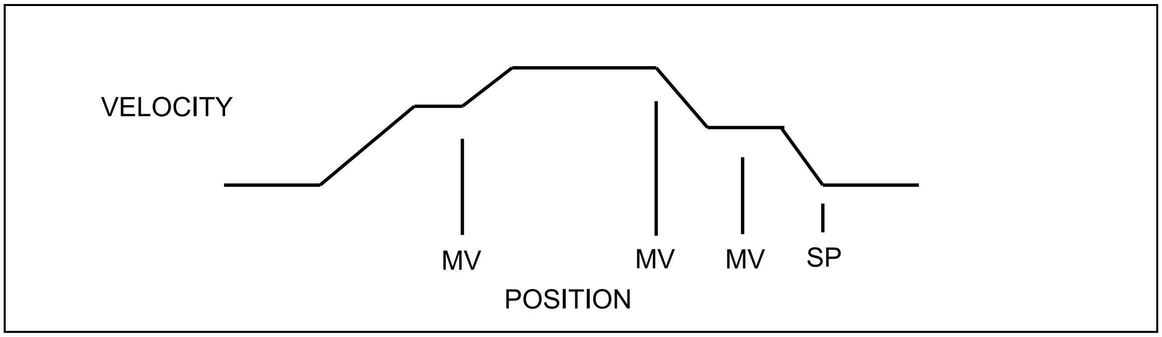

FIGURE 4-2 VELOCITY STAIRCASE PROFILE

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
MV#,#;	Single Axis	2*	2*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

*   If PA (power automatic) mode is active add 1 to the command queue

*   If this axis is a stepper encoder or servo axis add 2 to the command queue

Related commands: FP, MM, MP, SP

NV NEW CONTOUR VELOCITY

The NV command will set a new velocity for a constant velocity contour currently in execution. When the velocity changes, the MAX will switch to the new velocity at the start of the next command in the contour queue without ramping. An NV command issued without a contour currently executing will have no effect. For continuously variable velocity contouring, see the VC, VO, and VP commands.

RANGE: 1 ≤ NV ≤ 4,194,303

Example: The contour is executing at 25,000 counts per second. Change the velocity to 30,000 counts per second upon execution of the next command in the contour queue.

Enter: NV30000;

Response: None

QUEUE REQUIREMENTS				
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**	**CONTOUR**
-	Single Axis	Not Valid		
-	AA-AM	Not Valid		
NV#;	AA/CD	Immediate		

Related commands: CD, CV

?PA REPORT AUX POWER AUTOMATIC STATE

The ?PA command reports whether the current axis has power automatic mode enabled. The PA command is used to set this value. If power automatic mode is enabled, it will report whether the Aux bit goes high or low during a move. The axis is stationary if the aux bit is low.

Example: Determine if X axis power automatic is high or low.

Enter:
```
AX;
?PA
```

Response:
```
=0<LF>
```

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?PA	Single Axis	Immediate	
-	AA-AM	Not Valid	

*Power automatic mode is enabled for X axis such that the aux bit is high during a move.

Related commands: PA, SE

PF PARABOLIC ACCELERATION OFF

The PF command restores all axes to linear acceleration and deceleration ramps. This command should not be given while an axis is in motion or the results may not be predictable. This command affects all axes even if issued in the single axis mode. PF is the factory default setting. See the APP command to restore the PF setting as the power up/reset mode.

Example: Turn off cosine or parabolic ramps, returning to linear.

Enter: PF;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PF;	Single Axis	5	6
PF;	AA-AM	5	6
-	AA/CD	Not Valid	

Related commands: CN, LA, PN, PR, ?RT, SC, SR

PH AUX BIT HIGH

The PH command will turn the currently active axis auxiliary port bit ‘on’ (high). This command disables power automatic (PA) mode.

Example: Set aux bit of axis T to high.

Enter: AT;
PH;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PHb;	Single Axis	1	0
PHb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: PA, PL,

PL AUX BIT LOW

The PL command will turn the currently active axis auxiliary port bit ‘off’ (low). This command disables power automatic (PA) mode.

Example: Set aux bit of axis T to low.

Enter: AT;
PL;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PLb;	Single Axis	1	0
PLb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: PA, PH,

?PM REPORT HOLD STATE

The ?PM command reports whether the PID for the current servo axis is enabled or whether position maintenance hold mode for the current closed-loop stepper is enabled.

Example: A limit switch was hit by servo axis Y. See if the PID is still enabled for that axis.

Enter:
```
AY;
?PM
```

Response:
```
=off<LF>
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?PM	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: HF, HN

PN PARABOLIC ACCELERATION ON

The PN command sets all axes to truncated parabolic ramps. This acceleration profile starts at 100% of the programmed acceleration and decreases in steps of 10% of the initial acceleration down to as low as 10%. The parameter supplied selects the number of steps. It must be in the range of 3 to 10 corresponding to 70% and 10% acceleration at the peak respectively. A parameter out of this range or no parameter supplied defaults to 70% or 3 steps. Note that the parameter is the number of steps, not the acceleration values. The larger number is a lower acceleration at the peak. This command should not be given while an axis is in motion or the results may not be predictable. This command affects all axes even if issued in the single axis mode. The PF command is used to return to the default linear motion profiles. See the APP command to preserve the PN setting as the Power up/Reset ramp.

See PR for single axis.

RANGE: 3 ≤ PN ≤ 10

Example: Set the board to be in the smoothest parabolic acceleration ramp.

Enter: PN10;

Response: None

QUEUE REQUIREMENTS				
FORMAT	**MODE**	**AXIS RAMP TYPE**	**COMMAND**	**ARGUMENT**
PN#;	Single Axis	Short Form Parabolic (PN0)	5	10
PN#;	Single Axis	All Other Parabolic (PN,PR)	5	24
PN#,#,#,#,#,#,#,#;	AA-AM	Short Form Parabolic (PN0)	5	10
PN#,#,#,#,#,#,#,#;	AA-AM	All Other Parabolic (PN)	5	24
-	AA/CD	Not Valid		

Related commands: CN, LA, PF, PR, ?RT, SC, SR

PR PARABOLIC RAMP PER AXIS

The PR command defines parabolic acceleration/deceleration ramps for use with one or more axes. This command is similar to the PN command except that only the specified axes are affected. The APP command can be used to store the settings of PR as the power-up/reset defaults.

RANGE: 3 ≤ PR ≤ 10

Example: Select a 10 step parabolic ramp for the T axis.

Enter:
```
AT;
PR10;
```

Response: None

Example: Select a 10 step parabolic ramp for the Y axis and an 8 step parabolic ramp for the T axis and the R axis.

Enter:
```
AA;
PR,10,,8,,,8;
```

Response: None

QUEUE REQUIREMENTS				
FORMAT	**MODE**	**AXIS RAMP TYPE**	**COMMAND**	**ARGUMENT**
PR#;	Single Axis	Short Form Parabolic (PN0;)	5	10
PR#;	Single Axis	All Other Parabolic (PN,PR)	5	24
PR#,#,#,#,#,#,#,#;	AA-AM	Short Form Parabolic (PN0)	5	10
PR#,#,#,#,#,#,#,#;	AA-AM	All Other Parabolic (PN)	5	24
-	AA/CD	Not Valid		

Related commands: CN, LA, PF, PN, ?RT, SC, SR

?PS REPORT MOTOR TYPE

The ?PS command reports the type(s) of the active axes. In single axis mode it reports the motor type of the current axis. In AA mode it reports the types of all motors. The indicator reported is interpreted as follows:

O = Stepper Only
E = Stepper with Encoder
M = Servo

?PS is an immediate command and is active in single axis and AA/AM mode.

Example: Report the motor types of all motors.

Enter: AA;
?PS

Response: M,M,M,M,O,O,E,E<LF>
This means axes X, Y, Z and T are servo motors, axes U and V are open-loop stepper motors, and axes R and S are stepper motors with encoders.

Example: Report the motor type of X axis

Enter: AX;
?PS

Response: =M<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?PS	Single Axis	Immediate	
?PS	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: PS

RB REPORT USER DEFINED I/O BIT DIRECTION

The RB command reports the bit direction of the general I/O bits.

Example: Report direction of all I/O's. In this example I/O bits 0-3 are set as outputs and I/O bits 4-15 are set as inputs.

Enter: RB

Response: 000F<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	Single Axis	Immediate	
RB	AA-AM	Immediate	

Related commands: BD

RQ REPORT CONTOUR QUEUE SIZE

The RQ command reports the number of available entries in the contour queue. This command is only valid while in the contour definition mode.

MAX Value = 32763

Example: Report the command queue space remaining in contour mode.

Enter: RQ

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
-	Single Axis	Not Valid	
-	AA-AM	Not Valid	
RQ	AA/CD	Immediate	

Related commands: None.

?RT REPORT RAMP TYPE

The ?RT command reports the current acceleration ramp assigned to the active axis. Possible responses are:

LA	Default linear ramp
SSn	S-curve ramp where n specifies the S-curve profile number

Example:	Make sure custom ramp #3 was assigned to the Y axis
Enter:	AY; ?RT
Response:	=SR3<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?RT	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: LA, SS

SC COSINE RAMP PER AXIS

The SC command specifies that the standard cosine acceleration ramp is to be used by the selected axis/axes. This command is similar to the CN command except that only the selected axis or axes are affected. The APP command will store the settings of SC as the power-up/reset defaults.

Example: Select the cosine ramp for the X axis.

Enter:
```
AX;
SC;
```

Response: None.

Example: Select the cosine ramp for the Y, T, and R axes.

Enter:
```
AA;
SC,1,,1,,,1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SC;	Single Axis	5	24
SCb,b,b,b,b,b,b,b;	AA-AM	5	24
-	AA/CD	Not Valid	

Related commands: CN, LA, PF, PN, PR, ?RT, SR

?SD REPORT SERVO DAC VOLTS

The ?SD command reports the servo DAC output in volts.

NOTE: The voltage reported is the theoretical DAC output. The actual DAC voltage output may vary slightly due to internal DAC offset.

Example: Report the current Y axis servo DAC voltage output.

Enter: AY;
?SD

Response: =2.345

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?SD;	Single Axis	Immediate	
SD;	AA-AM	Immediate	

Related commands: KO, SVI, SVN

?SE REPORT AUX POWER AUTOMATIC SETTLING TIME

The ?SE command reports the settling time setting (SE) used with power automatic mode (PA) for the current axis.

Example: Report the current settling time for axis X

Enter:
```
AX;
?SE
```

Response:
```
=250<LF>
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?SE	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: ?PA, SE

SF SOFT LIMIT OFF

The SF command restores the default operation of the limit switches; i.e. causes the affected axis or axes to abruptly halt when a limit switch is encountered. If soft limits have been made the power-up default, the APP command must be used to store hard limit operation as the default.

Example: Set up a board to make the X axis stop immediately when a limit is encountered.

Enter: AX;
SF;
Response: None.

Example: Set up a board to make the Y and T axes to stop immediately when a limit is encountered.

Enter: AA;
SF,1,,1;
Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SF;	Single Axis	1	0
SFb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: LF, LH, LL, LN, ?SL, SL, TL

?SK REPORT AXIS SLIP KILL MODE SELECTION

The ?SK command reports whether slip kill mode is currently enabled for the active axis.

Example: Find out whether encoder slip kill mode is enabled for axis Z

Enter:

```
AZ;
?SK
```

Response: =on<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?SK	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: TF, TN

SL SOFT LIMIT ON

The SL command enables the MAX to ramp an axis to a stop rather than abruptly killing the motion when a limit switch is encountered on that axis. When a limit condition is encountered in soft limit mode, the axis decelerates to a stop, the limit interrupt is activated to notify the host, and in firmware 1.30 and greater, the axis queue is flushed. Soft limits can be made the power-up default via the APP command.

Example: Set up a board to allow the X axis to ramp to a stop when a limit is encountered.

Enter:
AX;
SL;

Response: None.

Example: Set up a board I/O to allow the Y and T axes to ramp to a stop when a limit is encountered.

Enter:
AA;
SL,1,,1;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SL;	Single Axis	1	0
SLb,b,b,b,b,b,b,b;	AA-AM	1	0

Related commands: LF, LH, LL, LN, SF, ?SL, TL

?SL REPORT SOFT LIMIT STATUS

The ?SL command reports whether soft limits are currently enabled for the active axis.

Example: Find out whether soft limits are enabled for axis Z

Enter: AZ;
?SL

Response: =on<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?SL	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: SF, SL

?SO REPORT SERVO ANALOG OUTPUT MODE

The ?SO command reports whether the analog output type for the current servo axis is bipolar or unipolar. The possible responses are BI and UN, the same commands used to set one mode or the other.

Example: The Y axis should be setup with unipolar outputs. Use ?SO to make sure.

Enter:
```
AY
?SO
```

Response:
```
=un<LF>
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?SO	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: BI, UN

SP STOP AT POSITION

The SP command will cause the controller to attempt to stop at the specified destination. If there is insufficient distance to stop at the previously specified deceleration when the command is received, the controller will stop as soon as possible at that deceleration. This command is not compatible with the JG command.

Example: (see MV command on page 364)

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SP#;	Single Axis	2*	1*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

* If the axis is a stepper and encoder or servo axis add 1 to the command queue

* If PA (power automatic) mode is active add 1 to the command queue

* If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: FP, MM, MP, MV

SS SELECT CUSTOM S-CURVE PROFILE

The SS command selects a custom S-curve profile to use for an axis or axes. See the APP and APB commands for archiving the controller’s parameters in non-volatile FLASH.

RANGE: 1 ≤ Ramp Number ≤ 8

Example: Selects custom S-curve profile #1 for Y axis.

Enter: AY;
SS1;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SS#:	Single Axis	10	107
SS#,#,#,#,#,#,#,#;	AA-AM	11	107*

* = per axis

Related commands: AJ, ?RT

?SV REPORT SERVO VOLTAGE INVERSION STATE

The ?SV command reports the current logical direction for the current servo axis. The state is set with the SVI and SVN commands. The possible responses to the ?SV command are =n for normal and =i for inverted.

Example: Report whether servo voltage is positive for positive moves on axis X

Enter:

```
AX;
?SV
```

Response: =n <LF> (voltage is normal; i.e. positive for positive moves)

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?SV	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: SVI, SVN

SVI INVERT SERVO VOLTAGE

The SVI command inverts the voltage output for the current axis. After receiving this command, the MAX will produce a negative voltage for positive motion and a positive voltage for negative motion. To cancel this command, issue an SVN command. To make inverted servo outputs the default at power up or reset, use the APP command.

Example: The Y axis encoder is counting opposite the expected direction. Setup the Y axis to produce a negative voltage when moving positive instead of a positive voltage to correct the problem.

Enter:

```
AY
SVI;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SVI;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: BI, DBI, DBN, ?SV, SVN, UN

SVN NORMALIZE SERVO VOLTAGE

The SVN command normalizes the voltage output for the current axis, negating the effects of the SVI command. After receiving this command, the MAX will produce a positive voltage for positive motion and a negative voltage for negative motion, the default behavior. To make this the default behavior (if it has been changed via SVI/APP), use the APP command. (SVN is the factory default setting.)

Example: The Y axis encoder was rewired and now counts in the correct direction. Return the Y axis servo output to normal; i.e. output positive voltage for positive motion.

Enter:

```
AY;
SVN;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
SVN;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: BI, DBI, DBN, ?SV, SVI, UN

TF TURN OFF SLIP KILL MODE

The TF command disables slip kill mode (enabled with TN). In AA/AM modes, any parameter for an axis will disable slip kill. A missing parameter will leave the axis as is. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

Example: Slip kill mode is enabled but a move needs to be performed where slip is likely and not important for this move. Disable slip kill mode.

Enter: TF;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
TF;	Single Axis	1	1
TF#,#,#,#,#,#,#,#;	AA-AM	1*	0

* = per axis

Related commands: ES, IS, RL, TN

?TL REPORT SOFTWARE OVERTRAVEL LIMITS

The ?TL command reports the software travel limits for the current axis assigned via the TL command. The first value returned is the upper (or "positive") limit and the second value is the lower (or "negative") limit. These are not physical limits but rather positional limits that an axis should not exceed. If one of these limits is exceeded, the MAX will set the current axis' limit flag and notify the host computer of the condition as though the axis encountered a hard limit.

Example: Find out what the software limits of the Y axis are currently set to.

Enter:
```
AY;
?TL
```

Response: `=101000,-1000<LF>`

Example: Find out what the software limits of the T axis are currently set to.

Enter:
```
AT
?TL
```

Response: `=0,0;<LF>` (software limits for axis T are currently disabled)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?TL	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: TL

TN TURN ON SLIP KILL MODE

The TN command enables slip kill mode. In this mode, if the motor slips beyond the deadband set by the ES command, the MAXp will kill motion on the axis that slipped as though a KL command had been issued to the axis. This mode can be disabled (default) with the TF command. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

Example: X axis is sent on a move. Its encoder cable was not connected to the controller (oops!). The controller issues a KL (Kill) command to the X axis after receiving the slip interrupt. The slip interrupt is generated once the difference between the motor position counts and encoder counts exceed 20.

Enter:

```
AX
ES20;
TN;
IS;
LP0;
MA30;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
TN;	Single axis	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ES, IS, RL, TL

TX TRACK THE X AXIS

Set up the current axis so that it tracks the X axis. The command argument is the tracking ratio specified as a floating point.

NOTE: If this is a negative, then the axis moves in the opposite direction of the X axis. The command is invalid for the X axis.

Example: Set the Z axis to track the X axis.

Enter:
AZ;
TX;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
TX;	Single axis	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related Commands: ET, HF

UN SET SERVO OUTPUT UNIPOLAR MODE

The UN command sets the analog torque outputs of servo axes to unipolar. The analog output will range between 0.0VDC and +10VDC when unipolar is enabled. At maximum positive velocity, the board outputs +10VDC. At maximum negative velocity, the board output approaches 0.0VDC. To maintain position the board outputs 5VDC. This command is valid only in single axis mode. See the APP and APB commands for archiving the controller's parameters in non-volatile FLASH.

Example: Set up servo axis X for unipolar operation.

Enter:
```
AX;
UN;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
UN;	Single Axis	Immediate	
-	AA-AM	Not Valid	

Related commands: BI, DBI, DBN, ?SO, SVI, SVN

?UR REPORT THE CONTROLLER’S MOTOR UPDATE RATE

This command causes the controller to report its motor control update rate.

Example: Requests the controller’s motor update rate.

Enter: ?UR

Response: 1024<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?UR	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: #UR

?UU REPORT AXIS USER UNITS

This command returns the current user units' multiplier as set via the UU command.

Example: Make sure the UU512 command we sent earlier is still current. The command will return the UU value with six digits to the right of the decimal point. If the UU value exceeds six digits for the fractional value, the value will be rounded off to the sixth decimal place.

Enter: ?UU

Response: =512.000000<LF>

If user units are turned off (UF) ?UU returns:

=off<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?UU	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: UF, UU

?VB REPORT BASE VELOCITY

The ?VB command returns the base (starting) velocity setting for the current axis as set by the VB command. Note that the base velocity must be lower than the command velocity.

Example: The acceleration ramp should start at 0 velocity. Make sure we didn't leave it at some other value.

Enter: ?VB

Response: =1500<LF> (Oops! We forgot to set it back to zero)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?VB	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: VB, ?VL

?VL REPORT MAXIMUM VELOCITY SETTING

The ?VL command returns the peak velocity setting for the current axis as set by the VL command.

Example: Make sure our "AX;VL15000;" command worked.

Enter: ?VL

Response: =150000<LF>

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
?VL	Single Axis	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: VB, ?VB, VL

VS VELOCITY STREAMING

The VS command will generate a pulse train without acceleration or deceleration at the rates specified using the X and Y axes. The parameters are time in 1 update rate second sample intervals, X velocity, and Y velocity. This is a slave mode and cannot be mixed or queued with other commands; i.e. no other motions involving the X and Y axes or AA mode may be currently in execution when this command is issued. AX mode is the only valid mode for use with this command. The VS command does not require a GO command to start the motion; motion begins immediately upon receipt of the complete VS command.

RANGES:
1 ≤ Parameter 1 (Time in 1 update rate of a second) ≤ 200,000
0 ≤ Parameter 2 (X Axis Velocity) ≤ 4,194,303
0 ≤ Parameter 3 (Y Axis Velocity) ≤ 4,194,303

Example: Create a stair step profile on the X and Y axes, with the X axis moving in the negative direction and the Y axis in the positive direction. Make each step last 1 second long and increase velocity by 1,000 steps/second, until a velocity of 3,000 steps/second is reached, then step down to 0 steps/second. (Example assumes an update rate of 1024.)

Enter:

```
AX;
VS1024,-1000,1000;
VS1024,-2000,2000;
VS1024,-3000,3000;
VS1024,-2000,2000;
VS1024,-1000,1000;
VS1,0,0;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
VS#,#,#,#,#,#,#,#;	Single axis	2	3
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: MV

This page is intentionally left blank.

INDEX