

KOLLMORGEN

[www.DanaherMotion.com](http://www.DanaherMotion.com)

**SERVOSTAR<sup>®</sup> CD**

**VarCom Reference Guide**

**M-SS-017-04**

**Rev. M**

**Firmware Version 6.3.3 and before**



## Version History

Revision	Date	Description of Revision
1	3/31/99	Transition from Quick Ref format to VarCom .PDF doc
2	12/20/99	Addition of Variables
3	1/20/00	Correction of HOMESTATE variable, added AUTONULL
4	06/30/00	Added HSAVE and ZPOS. Removed documentation history.
5	03/30/01	Updated commands for firmware 3.6.
6	06/30/01	Added information for firmware 4.1 and corrected errors.
7	09/12/01	Added information for firmware version 4.1.8.
8	03/01/02	Updated HOMETYPE, INxMODE, and PFBOFF
9	06/14/02	Added information on slow moves
10	12/16/02	Updated ISCALE , OPMODE, and O1MODE
11	04/18/03	Added Series 5 UCB2 digital board commands
L	09/07/04	Separated SERVOSTAR-S from SERVOSTAR-CD, updated corporate identity, and added Series 5 functionality.
M	09/16/05	Updated MOTORTYPE

© Copyright 2000 - 2005 Danaher Motion - All rights reserved  
Printed in the United States of America.

DANAHER MOTION is a registered trademark of Danaher Corporation. Danaher Motion makes every attempt to ensure accuracy and reliability of the specifications in this publication. Specifications are subject to change without notice. Danaher Motion provides this information "AS IS" and disclaims all warranties, express or implied, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose. It is the responsibility of the product user to determine the suitability of this product for a specific application.



***Warning Alerts users to potential physical danger or harm. Failure to follow warning notices could result in personal injury or death.***



***Caution Directs attention to general precautions, which if not followed, could result in personal injury and/or equipment damage.***



***Note Highlights information critical to your understanding or use of the product.***

## Table of Contents

<b>1</b>	<b>Commands and Variables by Function .....</b>	<b>1</b>
1.1	Analog Input-Related .....	1
1.2	Communications.....	1
1.3	Configurable I/O .....	1
1.4	Current Variables And Commands.....	1
1.5	Drive And Motor Status .....	1
1.6	Drive Configuration And Modes.....	2
1.7	Drive Enabling And Disabling .....	2
1.8	Drive Parameters .....	2
1.9	Feedback Related .....	2
1.10	Gearing-Related Parameters.....	2
1.11	Loop Compensation And Gains.....	3
1.12	Motion Control Parameters.....	3
1.13	Motor Variables And Commands.....	3
1.14	Position Variables And Commands .....	4
1.15	Read And Write Switch Variables.....	4
1.16	Read And Write Switch Mode Variables.....	4
1.17	Read And Write Variables .....	5
1.18	Read-Only Switch Mode Variables .....	5
1.19	Read-Only Switch Variables.....	5
1.20	Read-Only Variables .....	6
1.21	Variable Recording And Playing.....	6
1.22	Variable Setting And Clearing.....	6
1.23	Velocity Variables And Commands .....	6
<b>2</b>	<b>VARIABLE AND COMMAND LIST.....</b>	<b>7</b>
2.1	Format.....	7
2.1.1.	Command.....	7
2.1.2.	Variable.....	7
2.2	Variable And Command Set .....	9
	ACC .....	9
	ACKMODE.....	9
	ACTFAULT .....	9
	ACTIVE .....	10
	ADDR .....	10
	ANDB .....	10
	ANDG .....	10
	ANIN .....	11
	ANLPHZ.....	11
	ANOFF .....	11
	ANOUT .....	11
	ANZERO .....	12
	AUTOHOME.....	12
	AUTONULL.....	12
	AVGTIME13 .....	
	BW .....	13

CCWLIM .....	13
CLREEPROM .....	14
COMPFLT .....	14
COMPMODE .....	14
CONFIG .....	15
CONVERT .....	15
CWLIM .....	15
DEC .....	15
DECSTOP .....	16
DICONT .....	16
DIP .....	16
DIPEAK .....	17
DIPEN .....	17
DIR .....	17
DIS .....	18
DISSPEED .....	19
DISTIME .....	19
DIVISIONS .....	19
DRIVEOK .....	19
DUMP .....	20
DUALFB .....	20
ECHO .....	20
EN .....	20
ENCINIT .....	21
ENCINITST .....	21
ENCOUT .....	21
ENCOUTMOD .....	22
ENCOUTO .....	22
ENCSTART .....	22
ERR .....	23
FILTMODE .....	23
FLTCLR .....	23
FLTHIST .....	23
FOLD .....	24
FOLDMODE .....	24
FOLDTIME .....	24
GEAR .....	24
GEARI .....	25
GEARMODE .....	25
GEARO .....	26
GET .....	26
GETMODE .....	26
GP .....	27
GPAFR .....	27
GPAFR2 .....	27
GPD .....	27
GPI .....	28
GPISATIN .....	28
GPISATOUT .....	28
GPVFR .....	28
GV .....	29
GVI .....	29
HALLS .....	29
HCMODE .....	29

<i>HC1</i>	30
<i>HC2</i>	30
<i>HC3</i>	31
<i>HOLD</i>	31
<i>HOMESPD</i>	31
<i>HOMESTATE</i>	31
<i>HOMETYPE</i>	32
<i>HSAVE</i>	33
<i>HWPOS</i>	33
<i>I</i>	34
<i>IA</i>	34
<i>IAOFF</i>	34
<i>IC</i>	34
<i>ICOFF</i>	34
<i>ICMD</i>	35
<i>ICONT</i>	35
<i>IENCSTART</i>	35
<i>IFRIC</i>	35
<i>IGRAV</i>	35
<i>ILIM</i>	36
<i>ILIM2</i>	36
<i>ILSBMODE</i>	36
<i>IMAX</i>	36
<i>IN</i>	37
<i>IN1</i>	37
<i>IN1MODE</i>	37
<i>IN2</i>	39
<i>IN2MODE</i>	39
<i>IN3</i>	41
<i>IN3MODE</i>	41
<i>IN4</i>	43
<i>IN4MODE</i>	43
<i>IN5</i>	44
<i>IN5MODE</i>	44
<i>IN6</i>	45
<i>IN6MODE</i>	45
<i>INITMODE</i>	46
<i>ININV4</i>	46
<i>ININV5</i>	47
<i>ININV6</i>	47
<i>INDEXPOS</i>	47
<i>ININV1</i>	47
<i>ININV2</i>	47
<i>ININV3</i>	48
<i>INITGAIN</i>	48
<i>INITMODE</i>	48
<i>INPOS</i>	48
<i>ISCALE</i>	49
<i>ISTOP</i>	49
<i>IZERO</i>	49
<i>J</i>	50
<i>K</i>	50
<i>KV</i>	50
<i>KVFR</i>	50

<i>KVI</i>	51
<i>LIMDIS</i>	51
<i>LIST</i>	51
<i>LMJR</i>	51
<i>LOAD</i>	51
<i>LPFHZ1</i>	52
<i>LPFHZ2</i>	52
<i>MA</i>	52
<i>MAPOS</i>	53
<i>MASPEED</i>	53
<i>MBEMF</i>	53
<i>MBEMFCOMP</i>	54
<i>MENCOFF</i>	54
<i>MENCRES</i>	54
<i>MENCTYPE</i>	55
<i>MFBDIR</i>	56
<i>MFOLD</i>	56
<i>MFOLDD</i>	57
<i>MFOLDDIS</i>	57
<i>MFOLDER</i>	57
<i>MFOLDT</i>	57
<i>MH</i>	58
<i>MHINVA</i>	59
<i>MHINVB</i>	59
<i>MHINVC</i>	59
<i>MI</i>	59
<i>MICONT</i>	60
<i>MIDIST0</i>	60
<i>MIDIST1</i>	61
<i>MIDIST2</i>	61
<i>MIDIST3</i>	61
<i>MIPEAK</i>	62
<i>MISPEED0</i>	62
<i>MISPEED1</i>	63
<i>MISPEED2</i>	63
<i>MISPEED3</i>	64
<i>MJ</i>	65
<i>MLGAINC</i>	65
<i>MLGAINP65</i>	
<i>MLGAINZ</i>	66
<i>MLIST</i>	66
<i>MLMIN</i>	66
<i>MODMODE</i>	66
<i>MOTOR</i>	67
<i>MOTORTYPE</i>	67
<i>MPHASE</i>	67
<i>MPITCH</i>	67
<i>MPOLES</i>	68
<i>MRESPOLES</i>	68
<i>MSG</i>	68
<i>MSININT</i>	69
<i>MSINFRQ</i>	69
<i>MSPEED</i>	69
<i>MTANGLC</i>	69

MTANGLP.....	70
MVANGLF.....	70
MVANGLH.....	70
NOTCHBW.....	70
NOTCHHZ.....	71
O1.....	71
O1MODE UCB1 CD Series 2.....	71
O1MODE UCB2 CD Series 5.....	72
O1RST.....	73
O1TRIG.....	73
OPMODE.....	74
PCMD.....	74
PCMDMODE.....	74
PE.....	75
PEINPOS75	
PEMAX.....	75
PEXT.....	75
PEXTHW.....	75
PEXTOFF.....	76
PFB.....	76
PFBOFF.....	76
PLIM.....	76
PMAX.....	77
PMIN.....	77
PRD.....	77
PROFMODE.....	77
PROFSCRV.....	78
PROMPT.....	78
PROTARY.....	78
PSCALE.....	79
RDRES.....	79
READY.....	79
RECDONE.....	79
RECING.....	80
RECOFF.....	80
RECORD.....	80
RECRDY.....	81
RECTRIG81	
REFRESH.....	83
RELAY.....	83
RELAYMODE.....	83
REMOTE.....	83
RESBW.....	84
RSTVAR.....	84
S.....	85
SAVE.....	85
SERIALNO.....	85
SININIT.....	85
SININTOUT.....	85
SININITST.....	86
SINPARAM.....	86
STAT.....	87
STATUS.....	87
STATUS2.....	89

STATUS2 .....	91
STEP .....	92
STOP .....	93
STOPMODE .....	93
STOPPED .....	94
SWEN .....	94
T .....	94
TESTLED .....	94
TF .....	95
THERM .....	95
THERMODE .....	95
THERMTIME .....	95
THERMTYPE .....	96
TRUN .....	96
TUNE .....	96
UNITS .....	97
UVMODE .....	98
UVRECOVER .....	98
UVTIME .....	98
V .....	98
VBUS .....	99
VCMD .....	99
VD .....	99
VE .....	100
VER .....	100
VEXT .....	100
VF .....	100
VFI .....	101
VH .....	101
VLIM .....	101
VMAX .....	102
VOSPD .....	102
VOSPDENS .....	102
VR .....	102
VSCALE .....	103
XENCDIR .....	103
XENCDIR .....	103
XENCRES (UCB1 Series 2) .....	103
XENCRES (UCB2 Series 5) .....	104
ZERO .....	104
ZPOS .....	104



# 1 COMMANDS AND VARIABLES BY FUNCTION

## 1.1 ANALOG INPUT-RELATED

ANDB	ANDG	ANIN	ANOFF	ANZERO
GEARI	GEARO	ISCALE	PMIN	VSCALE

## 1.2 COMMUNICATIONS

ACKMODE	ADDR	ECHO	MSG	PROMPT
---------	------	------	-----	--------

## 1.3 CONFIGURABLE I/O

ANOUT	ENCOUT	ENCOUTO	IN	IN1
IN1MODE	IN2	IN2MODE	IN3	IN3MODE
IN4	IN4MODE	IN5	IN5MODE	IN6
IN6MODE	O1	O1MODE	O1RST	O1TRIG
MSINFRQ	SININTOUT			

## 1.4 CURRENT VARIABLES AND COMMANDS

CONFIG	DICONT	DIPEAK	FOLD
FOLDMODE	FOLDTIME	I	IA
IC	ICMD	ICONT	IENCSTART
IFRIC	IGRAV	ILIM	ILIM2
IMAX	ISCALE	ISTOP	IZERO
MFOLD	MFOLDD	MFOLDDIS	MFOLDR
MFOLDT	MICONT	MIPEAK	T

## 1.5 DRIVE AND MOTOR STATUS

ACTIVE	CCWLIM	CWLIM	DIP
ERR	FLTCLR	FLTHIST	DRIVEOK
READY	RELAY	SERIALNO	STAT
STATUS	THERM	TRUN	VER

## 1.6 *DRIVE CONFIGURATION AND MODES*

ACTFAULT	COMPmode	DIP	DIR
FILTMODE	GEAR	GEARI	GEARMODE
GEARO	HOLD	LIMDIS	OPMODE
PCMDMODE	PROFMODE	RELAYMODE	STOPMODE
THERMODE	THERMTIME	THERMTYPE	UNITS
UVMODE	UVRECOVER	UVTIME	ZERO

## 1.7 *DRIVE ENABLING AND DISABLING*

ACTIVE	DIPEN	DRIVEOK	DIS	EN
K	READY	REMOTE	S	STOP
SWEN				

## 1.8 *DRIVE PARAMETERS*

DICONT	DIPEAK	ERR	FLTCLR	FLTHIST
VBUS				

## 1.9 *FEEDBACK RELATED*

DUALFB	ENCINIT	ENCINITST	ENCOUT
ENCOUTO	ENCSTART	HALLS	HWPOS
IENCSTART	MENCOFF	MENCRES	MENCTYPE
MFBDIR	MHINVA	MHINVB	MHINVC
MPHASE	MRESPOLES	MSINFRQ	MSININT
PFB	PFOFF	PRD	RDRES
RESBW	SINPARAM	STAT	XENCRES

## 1.10 *GEARING-RELATED PARAMENTERS*

GEAR	GEARI	GEARMODE	GEARO	PCMD
PE	PEXT	PEXTOFF		

## 1. 11 LOOP COMPENSATION AND GAINS

ANDG	BW	COMPFLT	COMPMODE
FILTMODE	GP	GPAFR	GPD
GPI	GPISATIN	GPISATOUT	GPVFR
GV	GVI	KV	KVI
KVFR	LMJR	LPFHZ1	LPFHZ2
MJ	MLGAINP	MLGAINZ	MTANGLC
MTANGLP	MVANGLF	MVANGLH	NOTCHBW
NOTCHHZ	REFRESH	TF	TUNE
VD	VEXT	VF	VH
VR			

## 1. 12 MOTION CONTROL PARAMETERS

ACC	CCWLIM	CWLIM	DEC
DECSTOP	DIR	DISSPEED	DISTIME
INPOS	LIMDIS	OPMODE	PEINPOS
PROFMODE	PROFSCRV	STOPMODE	

## 1. 13 MOTOR VARIABLES AND COMMANDS

MBEMF	MBEMFCOMP	MENCOFF	MENCRES
MENCTYPE	MHINVA	MHINVB	MHINVC
MICONT	MIPEAK	MJ	MLGAINC
MLGAINP	MLGAINZ	MLIST	MLMIN
MOTOR	MOTORTYPE	MPHASE	MPITCH
MPOLES	MRESPOLES	MSININT	MSPEED
MTANGLC	MTANGLP	MVANGLF	MVANGLH

## 1. 14 POSITION VARIABLES AND COMMANDS

DUALFB	HOMESPD	HOMESTATE	HOMETYPE
HWPOS	INPOS	MA	MAPOS
MASPEED	MH	MI	MIDIST0
MIDIST1	MIDIST2	MIDIST3	MISPEED0
MISPEED1	MISPEED2	MISPEED3	PCMD
PCMDMODE	PE	PEINPOS	PEMAX
PFB	PLIM	PMAX	PMIN
PRD	PSCALE	STOPPED	

## 1. 15 READ AND WRITE SWITCH VARIABLES

ACTFAULT	COMPFLT	DIR	ECHO
GEAR	HOLD	LIMDIS	MFOLDDIS
MHINVA	MHINVB	MHINVC	MSG
O1	PCMDMODE	PLIM	PROMPT
RELAYMODE	THERMTYPE	UNITS	UVRECOVER
ZERO			

## 1. 16 READ AND WRITE SWITCH MODE VARIABLES

ACKMODE	ANOUT	AVGTIME	COMPMODE
ENCOUT	ENCOUTO	FILTMODE	FOLDMODE
GEARMODE	GETMODE	HOMETYPE	IN1MODE
IN2MODE	IN3MODE	MENCTYPE	MFBDIR
MOTORTYPE	MPOLES	MRESPOLES	MSINFRQ
O1MODE	OPMODE	PCMDMODE	PROFMODE
STAT	STOPMODE	THERMODE	UVMODE

## 1. 17 READ AND WRITE VARIABLES

ACC	ANDB	ANDG	ANOFF
BW	DEC	DECSTOP	DISSPEED
DISTIME	FOLDTIME	GEAR	GEARI
GEARO	GP	GPAFR	GPAFR2
GPD	GPI	GPISATIN	GPISATOUT
GPVFR	GV	GVI	HOMESPD
ICONT	IENSTART	IFRIC	IGRAV
ILIM	ILIM2	IN	ISCALE
ISTOP	IZERO	KV	KVFR
KVI	LMJR	LPFHZ1	LPFHZ2
MAPOS	MASPEED	MBEMF	MBEMFCOMP
MENCOFF	MENCRES	MFOLDD	MFOLDR
MFOLDT	MICONT	MIDIST0	MIDIST1
MIDIST2	MIDIST3	MIPEAK	MISPEED0
MISPEED1	MISPEED2	MISPEED3	MJ
MLGAINP	MLGAINZ	MLMIN	MOTOR
MPHASE	MPITCH	MSININT	MSPEED
MTANGLC	MTANGLP	MVANGLF	MVANGLH
NOTCHBW	O1RST	O1TRIG	PEINPOS
PEMAX	PEXTOFF	PFBOFF	PMAX
PMIN	PROFSCRV	PSCALE	RECTRIG
TF	THERMTIME	UVTIME	VBUS
VD	VF	VH	VLIM
VOSPD	VR	VSCALE	XENCRES

## 1. 18 READ-ONLY SWITCH MODE VARIABLES

DIP	ENCINITST	HALLS	HOMESTATE	ILSBMODE
IN	RDRES			

## 1. 19 READ-ONLY SWITCH VARIABLES

CCWLIM	CWLIM	DIPEN	DRIVEOK	FOLD
IN1	IN2	IN3	MFOLD	READY
RECDONE	RECING	RECRDY	RELAY	REMOTE
STOPPED	SWEN	THERM		

**1. 20** *READ-ONLY VARIABLES*

ADDR	ANIN	DICONT	DIPEAK	HWPOS
I	IA	IC	ICMD	IMAX
PCMD	PE	PEXT	PFB	PRD
STAT	STATUS	TRUN	V	VCMD
VE	VER	VEXT	VMAX	

**1. 21** *VARIABLE RECORDING AND PLAYING*

AVGTIME	GET	GETMODE	RECDONE	RECING
RECOFF	RECORD	RECRDY	RECTRIG	STATUS

**1. 22** *VARIABLE SETTING AND CLEARING*

CLREEPROM	DUMP	LIST	LOAD
MLIST	RESBW	SAVE	

**1. 23** *VELOCITY VARIABLES AND COMMANDS*

ACC	DEC	DECSTOP	ILSBMODE	J
MSPEED	PROFSCRV	S	STOP	V
VCMD	VE	VEXT	VLIM	VMAX
VOSPD	VSCALE			

## 2 VARIABLE AND COMMAND LIST

### 2.1 *FORMAT*

The command and variable descriptions presented here are in alphabetical order. Command and variable descriptions use different formats, as described below. Section 2 contains commands applicable to CD Series 2 and CD Series 5. Appendix B continues with command variables applicable to CD Series 5 only.

#### 2.1.1. **COMMAND**

This is the format of a command description.

**Firmware Versions:**                      **Command Syntax:**  
**Opmodes:**                                      **Drive Status:**

**Firmware Versions:** tells in what firmware versions the command is implemented. To check your drive's firmware version, use the VER command.

**Command Syntax:** gives the exact syntax of the command. Any optional or required parameters are also listed. If parameters are used with the command, the units and ranges of those parameters are given.

**Opmodes:** (0-4, 8) indicates in which operational modes (opmodes) the command can be used (see the description for the variable *OPMODE*).

**Drive Status:** (en, dis, or EN/DIS) indicates the drive state in which the command can be used (en = enabled; dis = disabled; EN/DIS = either).

#### 2.1.2. **VARIABLE**

This is the format that a variable description will take.

**Firmware Versions:**                      **Type:**  
**Range:**  
**Units:**                                      **Default:**  
**Opmodes:**                                      **Drive Status:**  
**EEPROM:**

You can set variable values by typing in the name of the variable followed by a new value for the variable, separated by an equals sign or one or more spaces. Just typing in the name of a variable without a new value will cause the SERVOSTAR to output the current value of that variable to the serial port.

**Firmware Versions:** tells what firmware versions the variable is implemented in. To check your drive's firmware version, use the VER command.

**Type:** switch variable, switch mode variable, (standard) variable, or vector variable. Switch variables can be toggled between two different states (0/1, on/off, etc.). Switch mode variables are state variables used to select one of more than two states (for example, *opmode* is a switch mode variable that selects one of 6 Opmodes: 0-4, 8). Standard variables are set to an integer value within a given range.

Vector variables are special variables that are used for the Advanced Pole Placement compensator (COMPmode=3) and require the use of the REFRESH command when changed.

All variables are classified as read-only (R) or read/write (R/W).

**Range:** defines the range of valid values for the variable.

**Units:** defines the units of the variable. Note that to get the final value of the variable, including its units, you multiply its value by its units. Example: the units of MICONt are “amps \* 0.1.” If MICONt = 200, then its value is  $200 * \text{amps} * 0.1 = 20 \text{ amps}$ .

**Default:** defines the default value of the variable. If this field says “motor data,” then the default value is entered from a motor data file (using MOTIONLINK’s Motor Configuration Screen) or a motor data sheet. Most variables are reset to their defaults by using the RSTVAR command.

**Opmodes:** (0-4, 8) indicates in which operational modes (opmodes) the variable is used (see the description for the variable *OPMODE*). Most variables can be set in any opmode, but they only have an effect in the opmodes listed here.

**Drive Status:** (en, dis, or EN/DIS) indicates the drive state in which the command can be used (en = enabled; dis = disabled; EN/DIS = either).

**EEPROM:** (Yes or no) specifies whether or not a variable can be stored in non-volatile memory (EEPROM). If a variable is stored in EEPROM, it is “remembered” by the SERVOSTAR when the drive is powered down and back up. See the SAVE command for more information.



## 2.2 VARIABLE AND COMMAND SET

### ACC

Sets the drive acceleration rate. This variable is only asserted when linear ramp control is selected (PROFMODE = 1 and OPMODE = 0, 1, 4, or 8). For firmware versions (**VER**) prior to 3.1.0, the range of this variable was 1 to 399,987.

**Firmware Versions:** All

**Range:** 10 to 400,000

**Opmodes:** 0, 1, 4, 8

**Type:** variable (R/W)

**Default:** 400,000

**Drive Status:** EN/DIS

**Units:** rotary: rpm / sec  
linear: mm/sec/sec

**EEPROM:** Yes

### ACKMODE

Sets the communication safety level of the drive. The range of value is 0 through 2.

**0** = No safety procedures or error messages

**1** = Drive responds with ACK or NAK after every message

**2** = Same as 1 with an added Block Check Character (BCC or checksum) attached to the end of every message



#### NOTE

**ACKMODE must be 0 for MOTIONLINK to function properly.**

**Firmware Versions:** All

**Range:** 0 - 2

**Opmodes:** All

**EEPROM:** Yes

**Type:** switch mode (R/W)

**Default:** 400,000

**Drive Status:** EN/DIS

**Units:** N/A

### ACTFAULT

Defines how to handle the DISABLE procedure when a fault occurs.

**0** = disable the drive immediately

**1** = follow an Active Disable procedure (similar to the "S" cmd. See also *DECSTOP*, *DISSPEED*, *DISTIME*, and *O1MODE=5*)



#### NOTE

**The drive is always disabled immediately in the event of a feedback loss fault to prevent the drive from running away.**

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## ACTIVE

Displays if the drive is enabled and power is applied to the motor. This flag is the overall readiness indicator of the drive.

**0** = drive is inactive

**1** = drive is active and ready to operate

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## ADDR

Displays the position of the drive address switches (switches 1-4 or 1-5, depending upon firmware version) of the DIP switch located on the top of the drive.

**Firmware Versions:** All

**Range:** 0 to 15 (firmware versions prior to 2.0.0)

0 to 31 (firmware versions 2.0.0 and later)

**Default:** hardware defined

**Type:** variable (R)

**Units:** N/A

**Opmodes:** All

**Drive Status:** EN/DIS

**EEPROM:** No

## ANDB

Sets the dead band of the analog input signal. If the absolute value of the analog input signal is less than this value, no analog command signal is generated.

**Firmware Versions:** All

**Range:** 0 to 10,000

**Opmodes:** 1,3,8

**Units:** mV

**Type:** variable (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** No

## ANDG

Enables the drive's dual gain algorithm. The dual gain algorithm effectively increases the resolution of the command input from 14 to 15 bits under 4 V of input.

**0** - No dual gain

**1** - Dual gain hysteresis algorithm

**2** - Dual gain linear combination algorithm

**Firmware Versions:** 2.1.0 and later

**Range:** 0 to 2

**Opmodes:** 1,3,8

**Units:** N/A

**Type:** variable (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## ANIN

Displays the analog input value after being filtered by **ANOFF** and **ANDB**. The **AVGTIME** variable affects the time averaging of this variable.

**Firmware Versions:** All  
**Range:** -22,500 to 22,500  
**Opmodes:** All  
**Units:** mV

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** No

## ANLPFHZ

Sets a filter rate (corner frequency) for the analog input filter. This is a simple single pole filter that is always present. The filter rate adjusts automatically as the analog input sampling rate changes for different operational modes. A value of 10,000 = unity gain (no filter).

**Firmware Versions:** 2.1.0 and later  
**Range:** 1 to 10,000  
**Opmodes:** 1,3,8  
**Units:** Hz

**Type:** variable (R/W)  
**Default:** 10,000  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## ANOFF

Sets the analog offset that is added to the analog input command to the drive. This is used to compensate for the analog input signal (**ANIN**) offset or drift.

**Firmware Versions:** All  
**Range:** -10,000 to 10,000  
**Opmodes:** 1,3,8  
**Units:** mV

**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## ANOUT

Sets the source for the analog output feature at user connector C3 pin 13.

- 0** - Tachometer (velocity feedback **V**) scaled identical to **VSCALE** with an update rate of 250  $\mu$ s.
- 1** - I monitor (equivalent current) scaled identical to **ISCALE** with an update rate of 10 ms (this variable is calculated in background loop).
- 2** - Velocity Error, **VE**, scaled identical to **VSCALE** with an update rate of 250  $\mu$ s.
- 3** - Torque Command Output Scaled to **ISCALE** with an update rate of 500  $\mu$ s.
- 4** - Reserved, no update rate.
- 5** - Position following error, **PE**, scaled to **PSCALE** with an update rate of 500  $\mu$ s.
- 6** - not used, with a minimum update rate of 500  $\mu$ s.
- 7** - not used, with an update rate of 62.5  $\mu$ s.

**8** - Position feedback, **PFB**, scaled to **PSCALE** with an update rate of 500  $\mu$ s.

**9** - Velocity controller output (before injecting **PRB/HC**), scaled to **VSCALE** with an update rate of 250  $\mu$ s.

**Firmware Versions:** 2.1.0 and later

**Range:** 0 to 5

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## ANZERO

Causes the drive to zero the analog offset. A sample of the motor analog input command is averaged over 64 samples, and the value of **ANOFF** is set to zero out the analog input command. This command may need to be executed more than once to achieve zero offset, and **ANOFF** will probably be modified.

**Firmware Versions:** All

**Opmodes:** All

**Command Syntax:** ANZERO

**Drive Status:** EN/DIS

## AUTOHOME

Defines the homing behavior on power-up.

**0** = No homing on power-up (default).

**1** = Try to perform homing on power-up. Does not continue if the homing cannot be executed.

**2** = Try to perform homing on power-up. Continue trying until homing process is executed.

**Firmware Versions:** 3.7.0 and later

**Range:** 0 to 2

**EEPROM:** Yes

**Opmodes:** 8

**Default:** 0

## AUTONULL

Automatic calibration of the current loop at drive enable.

**AUTONULL = 1:** The drive applies zero volts to the motor for 50 ms.

The motor must be at or near rest or the results can be unpredictable.

**AUTONULL = 0:** The drive reads current after the drive has been disabled. A series of reads over many seconds allows the average reading to be calculated and used as current sensor offset.

**AUTONULL=2:** Cancels all automatic nulling and allows the user to manually adjust the offset using **IAOFF** and **ICOFF**.

All offsets pertain to **IA** and **IC** current sensors. The offsets, whether calculated internally or set manually can be queried with **IAOFF** and **ICOFF**.

**Firmware Versions:** 3.4.2

**Range:** 0 (Old method)

1 (Zero on enable),

2 User Defined offsets with **IAOFF** and **ICOFF**

**EEPROM:** Yes

**Default:** 0

## AVGTIME

Sets the variable averaging time period. This variable is expressed as multiples of the servo update period (Ts), which is 500  $\mu$ s. A value of 0 for **AVGTIME** causes requested variable values to be returned as instantaneous values. **AVGTIME** affects the time averaging of **ANIN**, **I**, **ICMD**, **V**, **VCMD**.



### NOTE

***AVGTIME has no effect on variables that are sampled using the RECORD command and MOTIONLINK's PC Scope Screen.***

**Firmware Versions:** All  
**Range:** 0, 2, 4, 8, 16, 32, or 64  
**Opmodes:** All  
**Units:** Ts (500  $\mu$ s)

**Type:** switch mode (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## BW

Sets the desired velocity control loop bandwidth. This variable only affects the system when using the Standard Pole Placement controller in velocity mode (COMPMODE = 2 or 4 and OPMODE = 0 or 1). With COMPMODE=2, BW is limited to 200 Hz; with COMPMODE=4, **BW** can extend to 400 Hz.



### NOTE

***COMPMODE=4 is only available in firmware versions 2.1.0 and later.***

**Firmware Versions:** All  
**Range:** 10 to 200 (COMPMODE=2)  
 10 to 400 (COMPMODE=4)  
**Type:** variable (R/W)

**Units:** Hz  
**Default:** 20  
**Opmodes:** 0,1,4,8  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## CCWLIM

Displays the state of the external counter clockwise (CCW) limit switch input.

- 0** = switch closed, CCW limit not reached
- 1** = switch open, CCW limit reached

**Firmware Versions:** All  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch(R)  
**Default:** hardware defined  
**Drive Status:** EN/DIS  
**EEPROM:** No

# CLREEPROM

Clears the non-volatile memory (EEPROM) in the drive. The drive null's the EEPROM and recovers from a NVRAM error and assumes a no-comp state. A complete drive configuration procedure then has to be initiated before resuming drive operation. This command is the only method of resetting the run time clock.

**Firmware Versions:** All  
**Opmodes:** All

**Command Syntax:** CLREEPROM  
**Drive Status:** DIS

# COMPFLT

**COMPFLT** is a switch variable that enables and disables a 400 Hz low pass filter in the velocity feedback loop. The filter will automatically be disabled if **COMPmode** = 4. **COMPFLT** retains its value regardless of whether the **COMPmode** setting is enabling and disabling the filter.

**Firmware Versions:** 2.1.0 and later  
**Range:** 0 (OFF), 1 (ON)  
**Opmodes:** 0,1,4,8  
**Units:** N/A

**Type:** switch(R/W)  
**Default:** 1  
**Drive Status:** DIS  
**EEPROM:** No

# COMPmode

Sets the velocity controller type for OPMODE 0 or 1 according to the following table.

COMPmode	Controller Type	Loop Variables
0	PI	GV, GVI
1	PDFF	KV, KVI, KVFR
2	Standard Pole Placement (low-frequency)	BW, MJ, LMJR, TF
3	Advanced Pole Placement	VD, VF, VH, VR
4	Standard Pole Placement (high-frequency)	BW, MJ, LMJR, TF



## NOTE

**COMPmode 3 is not available in version 1 firmware.**  
**COMPmode 4 is available in firmware versions 2.1.0 and later.**

**Firmware Versions:** see Note  
**Range:** 0 to 4  
**Opmodes:** 0,1,4,8  
**Units:** N/A

**Type:** switch mode (R/W)  
**Default:** 2  
**Drive Status:** DIS  
**EEPROM:** Yes

## CONFIG

Configures the current control loops after motor data has been entered. Executing this command tells the drive that all motor data parameters have been entered and that it is time for the drive to configure its control loops using the motor data.

When certain drive or motor variables are entered, they will cause the drive to enter a non-compensated (no-comp) state. The LED display will flash a minus sign. In this case, a **CONFIG** is required. This also occurs when **CLREEPROM** is executed.

**Firmware Versions:** All  
**Opmodes:** All

**Command Syntax:** CONFIG  
**Drive Status:** DIS

## CONVERT

Converts the internal velocity structure, designed at any of the **COMP MODES**, to the external structure variables (**VD**, **VH**, **VR**, **VF**, **VFI**), overriding previous values.

**Firmware Versions:** 3.4 and later  
**Opmodes:** 0,1,4,8

**Command Syntax:** CONVERT  
**Drive Status:** EN/DIS

## CWLIM

Displays the state of the external clockwise (CW) limit switch input.

**0** = switch closed, CW limit not reached

**1** = switch open, CW limit reached

**Firmware Versions:** All  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R)  
**Default:** hardware defined  
**Drive Status:** EN/DIS  
**EEPROM:** No

## DEC

Sets the deceleration rate of the drive. This variable only affects the drive when linear ramp control and velocity mode are selected (PROFMODE = 1 and OPMODE = 0, 1, 4, or 8). For firmware versions (**VER**) prior to 3.1.0, the range of this variable was 1 to 399,987.

**Firmware Versions:** All  
**Units:** rotary: rpm / sec  
linear: mm/sec/sec  
**Opmodes:** 0,1,4,8

**Type:** variable (R/W)  
**Range:** 10 to 400,000  
**Default:** 20  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## DECSTOP

**DECSTOP** is a deceleration used by drive commands that require a faster than usual stop. This **DECSTOP** value is used instead of **DEC** in the following instances: end-travel limits, HOLD, S, and fault occurrences with ACTFAULT=1.

**Firmware Versions:** All

**Units:** rotary: rpm \* 1000/sec  
linear: mm/sec/sec

**Opmodes:** All

**EEPROM:** Yes

**Type:** variable (R/W)

**Range:** 1 to 32767

**Default:** 5000

**Drive Status:** EN/DIS

## DICONT

Defines the continuous rated current for the drive (sinusoidal RMS). This is a hardware-defined read-only variable automatically detected by the drive.

**DICONT** is usually 50% of **DIPEAK**, the peak current of the drive but may vary in many cases with the **SERVOSTAR CD**. In a given application, the drive may be configured to a lower rating than **DICONT** by setting the value of **ICONT** to the desired rating.

**Firmware Versions:** All

**Range:** 10 to 1100

**Opmodes:** All

**Units:** amps \* 0.1

**Type:** variable (R)

**Default:** hardware/user defined

**Drive Status:** EN/DIS

**EEPROM:** Yes

## DIP

Displays the settings of the DIP switches located on top of the drive. This variable returns a series of 1s and 0s for each of the switches, with a comma inserted in the middle for clarity. Switch 10 is the leftmost digit, and switch 1 is the rightmost.



### NOTE

*In firmware versions prior to 2.0.0, there were only 8 DIP switches and no comma was printed out.*

**Firmware Versions:** All

**Range:** 00000000-11111111 (firmware versions prior to 2.0.0)

00000,00000-11111,11111  
(firmware 2.0.0 and later)

**Units:** 1=ON, 0=OFF

**Type:** switch mode (R)

**Default:** 20

**Opmodes:** All

**Drive Status:** EN/DIS

**EEPROM:** No



## DIPEAK

Defines the peak rated current of the drive (sinusoidal RMS). This is a hardware-defined read-only variable. DIPEAK sets the 100% reference for many other current variables.

**Firmware Versions:** All  
**Range:** 20 to 2200  
**Opmodes:** All  
**Units:** amps \* 0.1

**Type:** variable (R)  
**Default:** DICON \* 2  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## DIPEN

Displays the state of the Dip Switch Enable status (switch number 8 of the DIP switches on top of the drive). This variable has to be set=1 (switch set OFF) to allow the drive to be enabled.

**Firmware Versions:** All  
**Range:** 0 (disabled), 1 (enabled)  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R)  
**Default:** hardware defined  
**Drive Status:** EN/DIS  
**EEPROM:** No

## DIR

Sets the direction (inversion) of many different parameters with respect to the rotation of the motor; as viewed looking at the end of the shaft.

**Variable Syntax:** DIR <first parameter> <second parameter> <third parameter>



### NOTE

*This variable switch was greatly expanded for SERCOS use with the introduction of firmware version 3.4.0. Historically, it was a two-position switch with the following definition:*

*0 = positive motion is counter-clockwise (CCW)  
 1 = positive motion is clockwise (CW)*

### For Non-SERCOS Users:

Only the first parameter should be used to affect performance; and even then, only the parameters 0 or 1. Therefore, the standard user need only configure the first parameter (0 for CCW and 1 for CW) to control all three loops simultaneously. If you query the variable, all three parameters are communicated even though only one was changed. Just leave the second and third at the default settings.

Parameter	Position	Velocity	Torque
0	1	1	1
1	0	0	0

**For SERCOS Users:**

The SERCOS interface only affects the position command and position feedback polarities.

**First Parameter:** sets the direction (inversion) of the Position command, Velocity command, and Torque command.

Parameter	Position	Velocity	Torque
0	1	1	1
1	0	0	0
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	0	0	1

**Second Parameter:** sets the direction (inversion) of the External Position feedback, Motor Position feedback, Velocity feedback, Torque feedback.

Parameter	External	Motor	Velocity	Torque
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

**Third Parameter:** sets the direction (inversion) of the Position Additive command, Velocity Additive command, Torque Additive command. This parameter is currently not supported.

**Firmware Versions:** All

**Range:** Refer to tables

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 1, 0, 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## DIS

Disables the drive. Software servo loops are halted and power is disconnected from the motor. The function is immediate and the motor may coast.

**Firmware Versions:** All

**Opmodes:** All

**Command Syntax:** DIS

**Drive Status:** EN/DIS

## DISSPEED

Sets the speed window for the Active Disable function. The Active Disable function ramps the motor to zero speed using **DECSTOP**. **DISSPEED** is compared to the actual motor speed, and if the speed is less than this value, the active disable timer (**DISTIME**) will begin timing. Once the timer times out, the drive disables..

**Firmware Versions:** 2.1.0 and later

**Units:** rotary: rpm  
linear: mm/sec/sec

**Opmodes:** All

**EEPROM:** Yes

**Type:** variable (R/W)

**Range:** 0 to 14,999

**Default:** 50 rpm

**Drive Status:** EN/DIS

## DISTIME

Sets the amount of time to wait after motor speed goes below **DISSPEED** before the drive is disabled in the Active Disable process. Once motor speed goes below **DISSPEED**, the drive waits for the time period specified by **DISTIME**, and then disables the drive..

**Firmware Versions:** 2.1.0 and later

**Range:** 0 to 65535

**Opmodes:** All

**Units:** millisecond \* 0.1

**Type:** variable (R/W)

**Default:** 100

**Drive Status:** EN/DIS

**EEPROM:** Yes

## DIVISIONS

Sets the number of divisions used for indexing (divides **PROTARY**).



*Only used when MODMODE = 1*

### NOTE

**Firmware Versions:** 3.7.0 and later

**Default:** 4

**EEPROM:** Yes

**Range:** 2 to 32767

-2 to -32767

## DRIVEOK

Displays the status of the drive faults.

**0** = faults exist

**1** = no faults exist

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## DUMP

Transmits all variables and their settings to the serial port terminal. This command actually outputs the EEPROM contents of the drive to the serial port, where the variables can then be reviewed or saved to a variable file (\*.SSV).

**Firmware Versions:** All  
**Opmodes:** All

**Command Syntax:** DUMP  
**Drive Status:** EN/DIS

## DUALFB

Enables/disables the reading of an external feedback signal through the C8 connector.

**0** = no dual loop

**1** = dual loop without checking for external feedback fault

**2** = dual loop with checking for external feedback fault



### NOTE

*The fault options above can relate to a line break. However, the motor must be in motion for this detection to occur. It also indicates a wrong XENDIR parameter.*

**Firmware Versions:** 3.3 and later

**Range:** 0, 1, 2

**Opmodes:** 8

**Units:** N/A

**Type:** switch (R/W)

**Default:** 0

**Drive Status:** DIS

**EEPROM:** Yes

## ECHO

Enables/disables serial port character echo. If echo is enabled, characters received via the serial port are echoed back to the serial port and displayed on the **MOTIONLINK** monitor or terminal.

**0** = serial port echo disabled

**1** = serial port echo enabled



### NOTE

*ECHO = 1 is needed for proper operation of MOTIONLINK.*

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 1

**Drive Status:** EN/DIS

**EEPROM:** Yes

## EN

Initiates a software enable of the drive. **EN** first attempts to reset any existing fault conditions, then sets **SWEN** to 1. If **READY**, **REMOTE**, and **DIPEN** are equal to one, the drive becomes ACTIVE. Checking the value of ACTIVE allows you to see if **EN** successfully enabled the drive.

**Firmware Versions:** All

**Opmodes:** All

**Command Syntax:** EN

**Drive Status:** EN/DIS

## ENCINIT

Triggers the encoder initialization process for type 0-2 and type 7 encoders. The initialization process requires rotating the motor until the encoder index is found. Then, the drive sets the value of **MENCOFF**.

If the drive is enabled and in Opmode=0, **JOG** is used to rotate the motor. When the encoder index is encountered, the drive sets the value of **MENCOFF**. In this case, the status of **ENCINIT** is observed using **ENCINITST**.

**Firmware Versions:** All  
**Opmodes:** All

**Command Syntax:** ENCINIT  
**Drive Status:** DIS

## ENCINITST

Displays the status of the encoder initialization function. This variable is reset to 0 when manually set to the index position.

**0** = initialization process has not begun

**1** = encoder initialization is in progress

**2** = encoder initialization has been completed

**Firmware Versions:** All  
**Range:** 0, 1, 2  
**Opmodes:** All  
**Units:** N/A

**Type:** switch mode (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** No

## ENCOUT

Sets the resolution (number of lines) of the encoder equivalent output channel **only for resolver based systems**. For encoder-based systems, this variable is read-only and is set equal to **MENCRES** (for firmware versions prior to 3.3.0) or **MENCRES / ENCOUTO** (for firmware versions 3.3.0 and later). For sine encoder-based systems, this variable is read-only and is set equal to **MENCRES \* SININTOUT / ENCOUTO**.



### NOTE

***ENCOUT values of 2048 and 4096 are available only when VLIM < 6100 rpm (RDRES = 14 or RDRES = 16). ENCOUT values of 8192 and 16384 are available only when VLIM <= 1500 rpm (RDRES = 16). In the majority of resolvers, one electrical revolution = one mechanical revolution.***

**Firmware Versions:** All  
**Units:** lines per electrical revolution of the resolver  
**Opmodes:** All  
**EEPROM:** Yes

**Type:** switch mode (R/W)  
**Range:** 512, 1024, 2048, 4096, 8192, or 16384  
**Default:** 1024  
**Drive Status:** DIS

## ENCOUTMOD

Queries or sets the encoder simulation signal format. When set to 0, the encoder simulation signal has the A-quad-B format, with A leading B for positive rotation. When set to the value 1, the encoder simulation signal has a pulse up/down format. When motion is positive, an encoder pulse train is generated on the A signal and no signal is generated on the B signal line. When motion is negative, an encoder pulse train is generated on the B signal and the A signal line is forced low.

**Syntax for set:** ENCOUTMOD <value>                      **Default:** 0  
**Syntax for query:** ENCOUTMOD                              **Opmodes:** All  
**Type:** Switch variable (R/W)                              **Drive Status:** Disabled  
**Range:** 0: Encoder simulation has A-quad-B format  
           1: Encoder simulation has pulse up/down format  
**Firmware:** Version 6.2.0 and later                      **EEPROM:** Yes  
**Limitations:** Not yet supported by **MOTIONLINK**

## ENCOUTO

Sets the value of a scale-down factor only for the encoder equivalent output channel (**ENCOUT**) for encoder and sine encoder-based systems. For encoder-based systems with firmware versions 3.3.0 or later,  $ENCOUT = MENCRES / ENCOUTO$ . For sine encoder-based systems,  $ENCOUT = MENCRES * SININTOUT / ENCOUTO$ .

**Firmware Versions:** All                                      **Type:** switch mode (R)  
**Range:** 1, 2, 4, 8, 16                                      **Default:** 1  
**Opmodes:** All    **Drive Status:** DIS  
**Units:** N/A    **EEPROM:** Yes

## ENCSTART

Triggers the encoder initialization process for encoder types 1-4 and 6. In the initialization process, the drive rotates the motor to a known electrical position by placing **IENCSTART** current from the motor B terminal to the motor C terminal. If the encoder index is encountered (for type 1 and 2 encoders), the process terminates immediately. The **ENCSTART** process is initiated by doing the following:

1. With the drive disabled, type **ENCSTART**.
2. Enable the drive. The current is placed on the motor terminals and the initialization process is completed after the drive enable occurs.

Enable is inhibited until this command is executed (for encoder types 1 and 3). The status display flashes the current **OPMODE** at 3 Hz as a visual indicator that the encoder is not initialized yet.

**Firmware Versions:** All                                      **Command Syntax:** ENCSTART  
**Opmodes:** All    **Drive Status:** DIS

## ERR

Displays the last error detected by the drive. A numeric code and a short explanatory string are output to the serial port if MSG = 1. The error buffer is cleared when the drive undergoes a transition from disabled (**DIS**) to enabled (**EN**).

**Firmware Versions:** All  
**Opmodes:** All

**Command Syntax:** ERR  
**Drive Status:** EN/DIS

## FILTMODE

Sets the velocity loop filter mode.

- 0** No LPF.
- 1** A single first order filter. Cutoff frequency is **LPFHZ1**.
- 2** Two first order filters. Cutoff frequencies are **LPFHZ1** and **LPFHZ2**.
- 3** Notch filter. Frequency **NOTCHHZ**, bandwidth **NOTCHBW**. Available only if firmware version (**VER**) is 3.1.0 or greater.

The filters affect the **PI**, **PDFF**, and standard pole placement controllers (COMPMODES 0-2 and 4), and are ignored in the advanced pole placement controller (COMPMODE=3).

**Firmware Versions:** All  
**Range:** 0, 1, 2  
**Opmodes:** 0,1,4,8  
**Units:** N/A

**Type:** switch mode (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## FLTCLR

Clears the fault history buffer, (contains up to 10 faults).

**Firmware Versions:** All  
**Opmodes:** All

**Command Syntax:** FLTCLR  
**Drive Status:** EN/DIS

## FLTHIST

Causes the drive to transmit the fault history buffer to the serial port. The most recent fault is sent first. Up to 10 fault messages are output by the drive, with each fault message followed by a CR-LF. A time stamp in the format of *hours:minutes* is displayed along with each fault, indicating the time at which the fault occurred.

**Firmware Versions:** All  
**Opmodes:** All

**Command Syntax:** FLTHIST  
**Drive Status:** EN/DIS

## FOLD

Displays the status of the drive foldback circuit. When the system current level exceeds **ICONT** for too long, the drive enters foldback mode, **FOLD** changes from 0 to 1, and the drive current is limited gradually (in exponential fashion) to the value of **ICONT**.

**0** = drive foldback OFF

**1** = drive foldback ON

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** No

## FOLDMODE

Sets the mode for drive current foldback and motor current foldback operation.

**0** = normal foldback from **ILIM** to **ICONT**

**1** = foldback to **ICONT** and issue fault after **FOLDTIME**

**2** = issue fault immediately upon detection

**Firmware Versions:** All

**Range:** 0, 1, 2

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## FOLDTIME

Sets the time since foldback detection to foldback fault latch (for FOLDMODE=1 only).

**0** = normal foldback from **ILIM** to **ICONT**.

**Firmware Versions:** 2.1.0 and later

**Range:** 1 to 300

**Opmodes:** All

**Units:** seconds

**Type:** variable (R/W)

**Default:** 30

**Drive Status:** EN/DIS

**EEPROM:** Yes

## GEAR

Enables and disables electronic gearing. **GEAR** is an immediate command (causes immediate gearing) unless **INxMODE** is selected to allow hardware control. If **INxMODE** is selected to control this function, this serial command must be 1 and the **INx** input must be on to enable the gear function. **GEAR** is a velocity-lock function so any bits lost during unlock time or ramp-to-speed are lost.

**Firmware Versions:** 2.1.0 and later

**Range:** 0 (OFF), 1 (ON)

**Opmodes:** 4

**Units:** N/A

**Type:** switch (R/W)

**Default:** 1

**Drive Status:** EN/DIS

**EEPROM:** Yes



# GEARI

Specifies the number of teeth on the input *gear* for the Gearing mode.

**Firmware Versions:** 2.1.0 and later

**Range:** -32767 to + 32767

**Opmodes:** 4

**Units:** N/A

**Type:** variable (R/W)

**Default:** 1

**Drive Status:** EN/DIS

**EEPROM:** Yes

# GEARMODE

This is a switch mode variable that specifies the operation of electronic gearing for **OPMODE 4**:

## **GEARMODE = 0 - Encoder Follower, Flex I/O (Connector C3)**

**Inputs:** The encoder input channel is decoded as a quadrature input, scaled through **GEARI / GEARO**, and becomes the position command for the motor. The digital I/O (Flex I/O) inputs serve as the encoder input channel (see note below for clarification).

## **GEARMODE = 1 - Pulse and Direction, Flex I/O (Connector C3)**

**Inputs:** The encoder input channel A counts positive edges and becomes the position command. The encoder input channel B level dictates if the counter will count up or down. Channel B low drives motor in CW direction. The digital I/O (Flex I/O) inputs serve as the encoder input channel (see note below for clarification).

## **GEARMODE = 2 - Up/Down Mode, Flex I/O (Connector C3) Inputs:**

The encoder input channel is configured as a counter to command the motor's position. Positive edges on the A channel increments the counter (motor CW) while positive edges on the encoder input channel B decrements the counter (motor CCW). The digital I/O (Flex I/O) inputs serve as the encoder input channel (see note below for clarification).

## **GEARMODE = 3 - Encoder Follower, Remote Encoder (Connector C8) Inputs:**

The encoder input channel is decoded as a quadrature input, scaled through **GEARI / GEARO**, and becomes the position command for the motor. The remote encoder (handwheel) inputs serve as the encoder input channel (see note below for clarification).

## **GEARMODE = 4 - Pulse and Direction, Remote Encoder (Connector C8) Inputs:**

The encoder input channel A counts positive edges and becomes the position command. The encoder input channel B level dictates if the counter will count up or down. Channel B low drives motor in CW direction. The remote encoder (handwheel) inputs serve as the encoder input channel (see note below for clarification).

## **GEARMODE = 5 - Up/Down Mode, Remote Encoder (Connector C8) Inputs:**

The encoder input channel is configured as a counter to command the motor's position. Positive edges on the A channel increments the counter (motor CW) while positive edges on the encoder input channel B decrements the counter (motor CCW). The remote encoder (handwheel) inputs serve as the encoder input channel (see note below for clarification).

**NOTE**

*If GEARMODE = 0-2, encoder A/B inputs are received via the digital Flex I/O inputs on connector C3. If GEARMODE = 3-5, encoder A/B inputs are received via the remote encoder (handwheel) inputs on connector C8.*

*Each of these modes are subject to: GEARI, GEARO, GEAR, and DIR.*

**Firmware Versions:** 2.1.0 and later

**Range:** 0 to 5

**Opmodes:** 4

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 3

**Drive Status:** DIS

**EEPROM:** Yes

## GEARO

Specifies the number of teeth on the output *gear* for the Gearing mode.

**Firmware Versions:** 2.1.0 and later

**Range:** 1 to 32767

**Opmodes:** 4

**Units:** teeth

**Type:** variable (R/W)

**Default:** 1

**Drive Status:** EN/DIS

**EEPROM:** Yes

## GET

Causes all recorded variables to be transmitted to the serial port for use with PC Scope. The data format is defined by **GETMODE**.

**Firmware Versions:** All

**Opmodes:** All

**Command Syntax:** GET

**Drive Status:** EN/DIS

## GETMODE

Sets the mode of data transfer from the drive to the host when using **GET**.

**0** = ASCII data transfer format

**1** = ASCII-HEX data transfer format

**2** = BINARY data transfer format (fastest)

**NOTE**

*GETMODE=0 is needed for operation of MOTIONLINK.*

**Firmware Versions:** All

**Range:** 0, 1, 2

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## GP

Sets the proportional gain for the position loop. Executing **TUNE** successfully may change the value of this parameter.

**Firmware Versions:** 2.1.0 and later

**Units:** rotary: 0.01 krpm/rev  
linear: 0.01 m/min/mm

**Drive Status:** EN/DIS

**EEPROM:** Yes

**Type:** variable (R/W)

**Range:** 1 to 7000

**Default:** calculated

**Opmodes:** 4,8

## GPAFR

This is a position loop feedforward acceleration gain term. This term is applied in the position loop and is used to create an acceleration feedforward input to the current loop.

**Firmware Versions:** 3.1.0 and later

**Range:** 0 to 2000

**Opmodes:** 4,8

**Units:** 0.1%

**Type:** variable (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## GPAFR2

This is a second position loop feedforward acceleration gain term. This term is applied in the position loop and is used to create an acceleration feedforward input to the velocity loop.



*Prior to firmware version 3.2.0, the range was 0 to 2000.*

### NOTE

**Firmware Versions:** 3.1.0 and later

**Range:** -10,000 to +10,000

**Opmodes:** 4,8

**Units:** 0.1%

**Type:** variable (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## GPD

Sets the derivative gain for the Proportional-Integral-Derivative (PID) compensator in the position loop.

**Firmware Versions:** 3.1.0 and later

**Range:** 0 to 32767

**Opmodes:** 4, 8

**Units:** N/A (1000=unity gain)

**Type:** variable (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## GPI

Sets the integral gain for the Proportional-Integral-Derivative (PID) compensator in the position loop. Setting this value = 10,000 means that  $GPI=GP$  (expressed mathematically, the internal PID gain used by the drive processor equals  $GP*GPI/10000$ ).

**Firmware Versions:** 3.1.0 and later  
**Range:** 0 to 10000  
**Opmodes:** 4, 8  
**Units:** N/A

**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## GPISATIN

Limits the input of the position loop integrator by setting the input saturation. When used in concert with **GPISATOUT**, this variable enables the operator to make the position loop integrator effective near the target position, whereas far from the target position, the integrator is not dominant in the loop dynamics.

**Firmware Versions:** 3.1.0 and later  
**Range:** 0 to 1,000,000  
**Opmodes:** 4, 8  
**Units:** feedback counts

**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## GPISATOUT

Limits the output of the position loop integrator by setting the output saturation. When used in concert with **GPISATIN**, this variable enables the operator to make the position loop integrator effective near the target position, whereas far from the target position, the integrator is not dominant in the loop dynamics.

**Firmware Versions:** 3.1.0 and later  
**Range:** 0 to 1,000,000  
**Opmodes:** 4, 8  
**Units:** feedback counts

**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## GPVFR

This is a position loop feedforward velocity gain term. This term is applied in the position loop and is used to create a feedforward input to the velocity loop.

**Firmware Versions:** 3.1.0 and later  
**Range:** 0 to 2000  
**Opmodes:** 4, 8  
**Units:** 0.1%

**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## GV

Sets the proportional gain for the Proportional Integral Velocity Control Loop (PI loop; COMPMODE = 0). Executing **TUNE** successfully may change the value of this parameter.

**Firmware Versions:** All

**Range:** 0 to 1,000,000,000 (firmware versions 3.3.0 and later)  
0 to 65,535 (firmware versions 2.2.0 to 3.2.1)  
0 to 32,767 (firmware versions up to 2.2.0)

**Default:** 500

**Type:** variable (R/W)

**Units:** N/A

**EEPROM:** Yes

**Opmodes:** 0,1,4,8

**Drive Status:** EN/DIS

## GVI

Sets the velocity integral gain for the Proportional Integral Velocity Control Loop (PI loop; COMPMODE = 0). Executing **TUNE** successfully may change the value of this parameter.

**Firmware Versions:** All

**Type:** variable (R/W)

**Range:** 0 to 65535

**Default:** 20

**Opmodes:** 0,1,4,8

**Drive Status:** EN/DIS

**Units:** N/A

**EEPROM:** Yes

## HALLS

Returns the hall switch values (encoder feedback option only). The switch values are displayed as a three-bit code in the sequence C-B-A.

**Firmware Versions:** All

**Type:** switch mode (R)

**Range:** 000 to 111

**Default:** N/A

**Opmodes:** 2 and 3

**Drive Status:** EN/DIS

**Units:** bits

**EEPROM:** No

## HCMODE

Selects the harmonic correction mode of operation.

**0** – Disabled

**1** – Harmony HC1 corrects position feedback used for velocity (resolver systems), harmonics HC2, HC3 summed and correct the torque command.

**2** – All harmonies are summed and used for torque correction.

**3** – All harmonies are summed and used for resolver feedback correction (for velocity only.)

**4** – Harmony HC1 and HC2 are summed to correct position feedback used for velocity (resolver systems,) harmony HC3 corrects the torque command.

This command does not operate in torque Opmodes (2 and 3).

**Firmware Versions:** 3.4 and later

**Type:** switch (R/W)

**Range:** 0 to 4

**Default:** 0

**Opmodes:** 0, 1, 4, 5, 6, 7, 8

**Drive Status:** EN/DIS

**Units:** N/A

**EEPROM:** No

# HC1

Defines the properties of correction harmony number 1.

**Syntax:** HC1<harmonic number> <phase offset> <amplitude>  
 <harmonic number> sets the number of cycles per resolver cycle (resolver system) or per motor revolution (encoder system.)  
 <phase offset> relative to resolver zero (resolver system) or PRD zero (encoder system.)  
 <amplitude> specifies the amplitude of the harmony in internal counts at the injection point.

**Units:**

<harmonic number>: N/A

<phase offset>: degrees  
 (PRD cycle = 360°)

<amplitude>: counts

**Type:** variable (R/W)

**EEPROM:** Yes

**Opmodes:** All

**Drive Status:** EN/DIS

**Default:**

<harmonic number>: 2

<phase offset>: 0

<amplitude>: 0

**Range: Versions 3.7.0 and below:**

<harmonic number>: 1 to 1000

<phase offset>: 0 to 359

<amplitude>: 0 to 16384

**Versions 4.0.1 and later:**

<harmonic number>: 1 to 8192

<phase offset>: 0 to 359

<amplitude>: 0 to 16384

# HC2

Defines the properties of correction harmony number 2.

**Syntax:** HC2<harmonic number> <phase offset> <amplitude>  
 <harmonic number> sets the number of cycles per resolver cycle (resolver system) or per motor revolution (encoder system.)  
 <phase offset> relative to resolver zero (resolver system) or PRD zero (encoder system.)  
 <amplitude> specifies the amplitude of the harmony in internal counts at the injection point.

**Units:**

<harmonic number>: N/A

<phase offset>: degrees  
 (PRD cycle = 360°)

<amplitude>: counts

**Type:** variable (R/W)

**EEPROM:** Yes

**Opmodes:** All

**Drive Status:** EN/DIS

**Default:**

<harmonic number>: 2

<phase offset>: 0

<amplitude>: 0

**Range: Versions 3.7.0 and below:**

<harmonic number>: 1 to 1000

<phase offset>: 0 to 359

<amplitude>: 0 to 16384

**Versions 4.0.1 and later:**

<harmonic number>: 1 to 8192

<phase offset>: 0 to 359

<amplitude>: 0 to 16384

## HC3

Defines the properties of correction harmony number 3.

**Units:**

<harmonic number>: N/A  
 <phase offset>: degrees  
 (PRD cycle = 360°)  
 <amplitude>: counts

**Type:** variable (R/W)

**EEPROM:** Yes

**Opmodes:** All

**Drive Status:** EN/DIS

**Range: Versions 3.7.0 and below:**

<harmonic number>: 1 to 1000  
 <phase offset>: 0 to 359  
 <amplitude>: 0 to 16384

**Versions 4.0.1 and later:**

<harmonic number>: 1 to 8192  
 <phase offset>: 0 to 359  
 <amplitude>: 0 to 16384

**Default:**

<harmonic number>: 2  
 <phase offset>: 0  
 <amplitude>: 0

## HOLD

Sets a flag indicating whether or not the drive should enter the position-hold mode. When activated, the motor will decelerate to zero speed at the **DECSTOP** rate and switch modes to hold the motor shaft at its present position. This variable may be set either by serial communication, by asserting both limit switches (**CWLIM** and **CCWLIM**), or by setting DIP switch number 7, or during active disable. The Status Display will flash the current OPMODE as a visual indicator that the drive is in the HOLD mode.

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** No

## HOMESPD

Sets the homing speed and direction (first time initialize) for **INxMODE** 10-12 and **MH**. A positive speed is CW.

**Firmware Versions:** 3.1.0 and later

**Units:** rotary: rpm  
 linear: mm/sec

**Opmodes:** 8

**EEPROM:** Yes

**Type:** variable (R/W)

**Range:** -VMAX to VMAX

**Default:** 100

**Drive Status:** EN/DIS

## HOMESTATE

Returns the status of the homing function. This variable can be polled during homing to track the homing status. A homing process may not go through all of the steps listed below, which are a chronological listing of the steps taken during a full homing procedure that begins when homing is initiated with a home switch that is already pressed. When this process is complete, home or absolute machine zero is known and PFB=0.

**0** = no drive controlled homing has been initiated.

**1** = homing started; moving away from pressed home switch.

**2** = waiting for home switch to clear; drive will stop when it does.

- 3 = home switch has cleared; check if drive is stopped.
- 4 = home search; move towards home switch.
- 5 = waiting for home switch to be pressed; drive will stop.
- 6 = home switch detected; check if motor is stopped.
- 7 = after motion stops, motor will go home.
- 8 = check if motor is at home.
- 9 = motor is at home, waiting for in-position.
- 10 = homing is complete.
- 11 = homing process was interrupted during execution.
- 12 = homing was canceled, waiting for the motor to stop (SERCOS only).

**Firmware Versions:** 3.1.0 to 3.3.0, 3.4.2 and later.

**For Firmware Versions 3.4.0 and 3.4.1, contact Customer Support for a comprehensive definition of HOMESTATE.**

**EEPROM:** N/A

**Units:** N/A

**Drive Status:** EN/DIS

**Opmodes:** 8

**Type:** switch mode (R)

**Default:** N/A

**Range:** 0 to 12

## HOMETYPE

Defines the type of homing function performed. This variable takes on a different meaning depending on the type of positioning selected (using **PCMDMODE**). In the following descriptions, *Home switch* refers to a digital input (**IN1-IN3**) that has been configured as a home switch by setting **INxMODE** = 10. *Marker* refers to an encoder's index pulse or a resolver zero point:

**In Serial Position Mode** (**PCMDMODE** = 0. **PFB** is normalized to zero after the home search):

- 0 homing with home switch and marker (**PFB** = 0).
- 1 homing with home switch only (**PFB** = 0).
- 2 homing with marker only (**PFB** = 0).
- 3 present position is home on the rising edge of **IN1**, **IN2**, or **IN3** with the corresponding **INxMODE** value set to 12 (**PFB**=0).
- 4 homing on marker with every drive enable, **MH**, or on the rising edge of **IN1-IN3** with the corresponding **INxMODE** set equal to 12 (**PFB**=0).
- 5 homing on marker with every **MH** command, or on the rising edge of **IN1-IN3** with the corresponding **INxMODE** set equal to 12 (**PFB**=0).
- 6 homing with home switch and marker (**PFB**=0) on every **MH** (firmware versions 3.3.0 and later).
- 7 homing with home switch only (**PFB**=0) on every **MH** (firmware versions 3.3.0 and later).

**For HOMETYPE 0 to 3, the first execution of the homing command initiates a search for absolute machine zero. Executing successive homing commands causes an absolute move to zero position without searching for machine zero. For HOMETYPE 4 to 7, the drive searches for absolute machine zero on every home command, whether absolute machine zero has previously been detected or not.**



**NOTE**



In Analog Position Mode (PCMDMODE = 1):

**HOMETYPES 0 to 7** are the same as for Serial Position Mode. When homing is complete, **PFB** is normalized (set equal to 0) at the current value of the analog input (**ANIN**). Example: when homing is complete, if 2 volts are present at the analog input, **PFB** is set equal to 0 at 2 volts of input.

**HOMETYPES 50 to 57** are the same as 0-7, except that when homing is complete, **PFB** is *not* normalized at the current value of **ANIN**. Instead, **PFB** is normalized at 0 volts input.



### CAUTION

*If there is a voltage on the analog input other than 0 V and the drive is enabled, the motor will see this as a command to move and move to the position specified by the analog input voltage.*

Firmware revision information: prior to firmware version 3.3.0, only Analog Position Mode **HOMETYPES 0** and **3** were available.

<b>Firmware Versions:</b> 3.1.0 and later	<b>Type:</b> switch mode (R/W)
<b>Range:</b> 0 to 7 (analog and serial position modes) 50-57 (analog position mode only)	
<b>Units:</b> N/A	<b>EEPROM:</b> Yes
<b>Opmodes:</b> 8	<b>Drive Status:</b> EN/DIS
<b>Default:</b> 0	

## HSAVE

Copies **MPHASE**, **PFBOFF** and **ZPOS** into the EnDat encoder memory. Care must be taken when overwriting **MPHASE** as runaway conditions are possible with incorrect values. **HSAVE** is applicable only when MENCTYPE=9.

<b>Firmware Versions:</b> 3.6.0 and later	<b>Command Syntax:</b> HSAVE
<b>Opmodes:</b> All	<b>Drive Status:</b> EN/DIS

## HWPOS

Displays the position feedback directly from the feedback hardware counter. For resolver-based systems, **HWPOS** ranges from 0 to 65,535 per electrical revolution of the resolver (the number of electrical resolver revolutions per each mechanical revolution is calculated by dividing the number of resolver poles by 2). The resolution of **HWPOS** is based on **RDRES**:

RDRES = 12, resolution of HWPOS = 16

RDRES = 14, resolution of HWPOS = 4

RDRES = 16, resolution of HWPOS = 1

For encoder-based systems, **HWPOS** returns the encoder counter content, based on quadrature pulse input and ranges from 0 to 65535, with a resolution of 32.

<b>Firmware Versions:</b> 2.0.0 and later	<b>Type:</b> variable (R)
<b>Range:</b> 0 to 65,535	<b>Default:</b> N/A
<b>Opmodes:</b> All	<b>Drive Status:</b> EN/DIS
<b>Units:</b> counts	<b>EEPROM:</b> No

**I**

Displays the motor current. The **AVGTIME** variable determines the averaging of this variable, except when recorded for graphical display by **MOTIONLINK**, in which case it is not averaged.

**Firmware Versions:** All  
**Range:** 0 to 2000  
**Opmodes:** All  
**Units:** % of DIPEAK \* 0.1

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** No

**IA**

Displays the motor's A phase current. **AVGTIME** does not affect this variable.

**Firmware Versions:** All  
**Range:** -1000 to 1000  
**Opmodes:** All  
**Units:** % of DIPEAK \* 0.1

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** No

**IAOFF**

Displays the motor's A phase current. Offset value or used (with AUTONULL=2) to establish (W\*)the A-phase current offset.

**Firmware Versions:** All  
**Range:** -1000 to 1000  
**Opmodes:** All  
**Units:** internal bits

**Type:** variable (R/W\*)  
**Default:** See AUTONULL  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

**IC**

Displays the motor's C phase current. **AVGTIME** does not affect this variable.

**Firmware Versions:** All  
**Range:** -1000 to 1000  
**Opmodes:** All  
**Units:** % of DIPEAK \* 0.1

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** No

**ICOFF**

Displays the motor's A phase current. Offset value or used (with AUTONULL=2) to establish (W\*)the C-Phase current offset.

**Firmware Versions:** All  
**Range:** -1000 to 1000  
**Opmodes:** All  
**Units:** internal bits

**Type:** variable (R/W\*)  
**Default:** See AUTONULL  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## ICMD

Displays the Current (Torque) command to the current controller. This variable is equivalent to the Analog Input (**ANIN**) in OPMODE 3, to the Torque Command (T) in OPMODE 2, and to the output of the velocity controller in OPMODE 0 or 1. The **AVGTIME** variable affects averaging of this variable, except when recorded for graphical display by **MOTIONLINK**, in which case it is not averaged.

**Firmware Versions:** All  
**Range:** -1000 to 1000  
**Opmodes:** All  
**Units:** % of DIPEAK \* 0.1

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** No

## ICONT

Sets the system continuous current. This variable is used in the foldback algorithm. The default value of this variable is the minimum of **DICONT** (Drive Continuous Current) and **MICONT** (Motor Continuous Current), unless that value exceeds **IMAX**, in which case **ICONT** is set equal to **IMAX**. This variable is reset to its default whenever **DICONT** or **MICONT** is changed. You can override the default.

**Firmware Versions:** All  
**Range:** 0 to IMAX  
**Opmodes:** All  
**Units:** % of DIPEAK \* 0.1

**Type:** variable (R/W)  
**Default:** min of DICONT & MICONT  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## IENCSTART

Sets the maximum current for the **ENCSTART** encoder initialization process.

**Firmware Versions:** All  
**Range:** 1 to 100  
**Opmodes:** All  
**Units:** % of MICONT

**Type:** variable (R/W)  
**Default:** 25  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## IFRIC

This is the Coulomb Friction constant for the current loop.

**Firmware Versions:** 2.1.0 and later  
**Range:** 0 to 500  
**Opmodes:** 8  
**Units:** % of DIPEAK

**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## IGRAV

This is the Gravity constant for the current loop.

**Firmware Versions:** 2.1.0 and later  
**Range:** -500 to 500  
**Opmodes:** 8  
**Units:** % of DIPEAK

**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## ILIM

Sets the application current limit, allowing the user to limit the drive's peak current. This variable limits the current command that will be accepted from the user (using the T command in Opmode 2) or issued by the control loops (in Opmodes 0, 1, 3, and 4). This variable is an independent variable that is not calculated from hardware parameters and is not tied to any other variables. **ILIM** is similar to **VLIM** (which is used in Opmodes 0 and 1) and is used to protect delicate load equipment.

**Firmware Versions:** All  
**Range:** 0 to IMAX  
**Opmodes:** All  
**Units:** % of DIPEAK \* 0.1

**Type:** variable (R/W)  
**Default:** IMAX  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## ILIM2

This variable is used to define a new current limit value for INxMODE 8. This variable functions in similar fashion to **ILIM** if INxMODE = 8 and the corresponding INx input = 1.

**Firmware Versions:** 3.1.0 and later  
**Range:** 0 to IMAX  
**Opmodes:** All  
**Units:** % of DIPEAK \* 0.1

**Type:** variable (R/W)  
**Default:** 0.1 \* IMAX  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## ILSBMODE

(Relevant for resolver feedback only.) Sets the mode of operation of the inter-LSB algorithm, which interpolates feedback between least significant bits (LSB's) of the resolver. Enabling this algorithm will improve performance when the **RDRES** resolution is low (12 bits), **BW** is high, and the commanded velocity is low.

ILSBMODE = 0; algorithm disabled

ILSBMODE = 1; enabled for velocity feedback.

ILSBMODE = 2; enabled for velocity and position feedback

**Firmware Versions:** 2.1.0 and later  
**Range:** 0, 1, 2  
**Opmodes:** 0,1,4,8  
**Units:** N/A

**Type:** switch mode (R/W)  
**Default:** 2  
**Drive Status:** DIS  
**EEPROM:** Yes

## IMAX

Displays the system current maximum for a drive and motor combination. This variable is actually the minimum of the drive Peak Current (**DIPEAK**) and the Motor Peak Current (**MIPEAK**).

**Firmware Versions:** All  
**Range:** 0 to 1000  
**Opmodes:** All  
**Units:** % of DIPEAK \* 0.1

**Type:** variable (R)  
**Default:** min of DIPEAK & MIPEAK  
**Drive Status:** EN/DIS  
**EEPROM:** No

## IN

Returns the state of the three digital inputs (**IN1**, **IN2**, **IN3**) in a three-character string. The leftmost bit represents **IN1** and the rightmost bit represents **IN3**.

**Firmware Versions:** 3.1.0 and later

**Range:** 000 to 111 (0=OFF, 1=ON)

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** N/A

## IN1

This is used to read the state of the hardware input on user connector C3 Pin 9.

**Firmware Versions:** 2.1.0 and later

**Range:** 0 (OFF), 1=(ON)

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** N/A

## IN1MODE

**IN1MODE** sets the functionality of the **IN1** input. The functions are:

**IN1MODE=0:** No function\*

**IN1MODE=1:** CW limit switch\*

**IN1MODE=2:** CCW limit switch\*

**IN1MODE=3:** Gear disable input (all **GEARMODE** values)\*

**IN1MODE=4:** Gear mask input (all **GEARMODE** values)\*

**IN1MODE=5:** Gear A input (**GEARMODE** = 0-2)\*

**IN1MODE=6:** Gear B input (**GEARMODE** = 0-2)\*

**IN1MODE=7:** Trigger incremental move / jog \*\*

**IN1MODE=8:** Use second current limit (**ILIM2**)\*

**IN1MODE=9:** Switch **OPMODE** from 1 to 3\*\*

**IN1MODE=10:** Home switch\*\*

**IN1MODE=11:** Reserved

**IN1MODE=12:** Search for absolute machine zero/Move to absolute machine zero (home)\*\*

**IN1MODE=13:** Trigger absolute move (**MAPOS** at **MASPEED**)\*\*

**IN1MODE=14:** Binary **MIDIST** / **MISPEED** selection code MSB\*\*

**IN1MODE=15:** Binary **MIDIST** / **MISPEED** selection code LSB\*\*

**IN1MODE=16:** Reserved

**IN1MODE=17:** Trigger active disable (see **DISSPEED**)\*

**IN1MODE=18:** Control fault relay\*\* (see notes below)

**IN1MODE=19:** Hold position\*

**IN1MODE=20:** When **OPMODE**=1, an active input switches to **OPMODE**=4 (must be disabled or 0 velocity)

**IN1MODE=21:** Trigger incremental move \*\*\* (see notes below)

**IN1MODE=22:** Triggers incremental move when **MODMODE**=1.  
Distance = (**PROTARY** / **DIVISIONS**)\*\*\*\*

**IN1MODE=23:** Trigger incremental move when **MODMODE**=1.  
Distance = (2 \* **PROTARY** / **DIVISIONS**)\*\*\*\*

**IN1MODE=24:** Change **OPMODE** 8 to **OPMODE** 4 using digital input.\*\*\*\*

**IN1MODE=25:** Upon transition from 0 to 1, the faults are cleared if REMOTE=0 or SWEN=0.\*\*\*\*\*

\* available in firmware versions 2.1.0 and later

\*\* available in firmware versions 3.1.0 and later

\*\*\* available in firmware versions 3.5.0 and later

\*\*\*\* available in firmware versions 4.0.0 and later

\*\*\*\*\* available in firmware versions 4.1.8 and later

**Notes:**

**IN1MODE = 7** operation: If the drive is in OPMODE 8 (positioning), an input with IN1MODE=7 can be used to trigger an incremental move (**MI**), using the variables MIDIST0-3, MISPEED0-3. See the descriptions for those variables, as well as the description of **MH**, for more information. If the drive is in OPMODE 1 (analog velocity), an input with IN1MODE =7 can be used to trigger a jog at a speed entered in MISPEED0-3. See the description of MISPEED0 for more details. IN1MODE 7 can be activated while the motor is in motion. This causes the motor to begin the next index immediately after completing the previous index (called *look-ahead-buffering*).

**IN1MODE = 9** operation: If the drive is in OPMODE 1 (analog velocity) and IN1MODE is 9, then switching the related IN1 input to 1 causes the drive to switch to OPMODE 3 (analog current). Switching the IN1 input back to 0 causes the drive to switch back to OPMODE 1 (analog velocity). The LED display of **OPMODE** changes according to the user input, but the serial response to a prompt for **OPMODE** returns 1. The **OPMODE** change can happen when the drive is enabled, therefore the user must make the switch with zero command.

**IN1MODE = 12** operation: Execute **HOMETYPE** (see note in **HOMETYPE**). Moves are performed at velocity equal to **HOMESPD**.

**IN1MODE = 18:** if input = 0, the fault relay will open. If input = 1, the fault relay operates as normal.

**IN1MODE=21:** The same function as IN1MODE=7 except that IN1MODE=21 ignores input signals until the motor has completed in prior index (comes to rest). There is no look-ahead-buffering.

**IN1MODE=22:** The motion velocity is **MISPEED0**. There is no movement unless in rotary mode (MODMODE=1). The direction depends on the sign of **DIVISIONS** and on the value of **DIR**.

**IN1MODE=23:** The motion velocity is **MISPEED0**. There is no movement unless in rotary mode (MODMODE=1). The direction depends on the sign of **DIVISIONS** and on the value of **DIR**.

**IN1MODE=24:** With the drive starting in the position-mode (OPMODE=8) with the INPUT = Low; a *Low to High* transition of the **INPUT** changes the drive to the gearing-mode (OPMODE=4); with a *High to Low* transition, the drive is changed back to the position-mode (OPMODE=8). These **OPMODE** changes occur when the drive is enabled or disabled.

**Firmware Versions:** see above

**Range:** 0 to 24

**Opmodes:** dependent

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 1

**Drive Status:** EN/DIS

**EEPROM:** N/A

## IN2

This is used to read the state of the hardware input on user connector C3 Pin 10.

**Firmware Versions:** 2.1.0 and later

**Range:** 0 (OFF), 1=(ON)

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** N/A

## IN2MODE

**IN2MODE** sets the functionality of the **IN2** input. The functions are:

**IN2MODE=0:** No function\*

**IN2MODE=1:** CW limit switch\*

**IN2MODE=2:** CCW limit switch\*

**IN2MODE=3:** Gear disable input (all **GEARMODE** values)\*

**IN2MODE=4:** Gear mask input (all **GEARMODE** values)\*

**IN2MODE=5:** Gear A input (GEARMODE = 0-2)\*

**IN2MODE=6:** Gear B input (GEARMODE = 0-2)\*

**IN2MODE=7:** Trigger incremental move / jog \*\*

**IN2MODE=8:** Use second current limit (**ILIM2**)\*\*

**IN2MODE=9:** Switch **OPMODE** from 1 to 3\*\*

**IN2MODE=10:** Home switch\*\*

**IN2MODE=11:** Reserved

**IN2MODE=12:** Search for absolute machine zero/Move to absolute machine zero (home)\*\*

**IN2MODE=13:** Trigger absolute move (**MAPOS** at **MASPEED**)\*\*

**IN2MODE=14:** Binary **MIDIST** / **MISPEED** selection code MSB\*\*

**IN2MODE=15:** Binary **MIDIST** / **MISPEED** selection code LSB\*\*

**IN2MODE=16:** Reserved

**IN2MODE=17:** Trigger active disable (see **DISSPEED**)\*

**IN2MODE=18:** Control fault relay\*\* (see notes below)

**IN2MODE=19:** Hold position\*

**IN2MODE=20:** When OPMODE=1, an active input switches to OPMODE=4 (must be disabled or 0 velocity)

**IN2MODE=21:** Trigger incremental move \*\*\* (see notes below)

**IN2MODE=22:** Triggers incremental move when MODMODE=1.  
Distance = (**PROTARY** / **DIVISIONS**)\*\*\*\*

- IN2MODE=23:** Trigger incremental move when MODMODE=1.  
Distance = (2 \* **PROTARY / DIVISIONS**)\*\*\*\*
- IN2MODE=24:** Change OPMODE 8 to OPMODE 4 using digital input.\*\*\*\*
- IN2MODE=25:** Upon transition from 0 to 1, the faults are cleared if REMOTE=0 or SWEN=0.\*\*\*\*\*
- \* available in firmware versions 2.1.0 and later  
 \*\* available in firmware versions 3.1.0 and later  
 \*\*\* available in firmware versions 3.5.0 and later  
 \*\*\*\* available in firmware versions 4.0.0 and later  
 \*\*\*\*\* available in firmware versions 4.1.8 and later

### Notes:

- IN2MODE = 7** operation: If the drive is in OPMODE 8 (positioning), an input with IN2MODE=7 can be used to trigger an incremental move (**MI**), using the variables MIDIST0-3, MISPEED0-3. See the descriptions for those variables, as well as the description of **MH**, for more information. If the drive is in OPMODE 1 (analog velocity), an input with IN2MODE =7 can be used to trigger a jog at a speed entered in MISPEED0-3. See the description of **MISPEED0** for more details. IN2MODE 7 can be activated while the motor is in motion. This causes the motor to begin the next index immediately after completing the previous index (called *look-ahead-buffering*).
- IN2MODE = 9** operation: If the drive is in OPMODE 1 (analog velocity) and IN2MODE is 9, then switching the related IN2 input to '1' causes the drive to switch to OPMODE 3 (analog current). Switching the IN2 input back to 0 causes the drive to switch back to OPMODE 1 (analog velocity). The LED display of **OPMODE** changes according to the user input, but the serial response to a prompt for **OPMODE** returns 1. The **OPMODE** change can happen when the drive is enabled, therefore the user must make the switch with zero command.
- IN2MODE = 12** operation: Execute **HOMETYPE** (see note in **HOMETYPE**). Moves are performed at velocity equal to **HOMESPD**.
- IN2MODE = 18:** if input = 0, the fault relay will open. If input = 1, the fault relay operates as normal.
- IN2MODE=21:** The same function as IN2MODE=7 except that IN2MODE=21 ignores input signals until the motor has completed in prior index (comes to rest). There is no look-ahead-buffering.
- IN2MODE=22:** The motion velocity is **MISPEED0**. There is no movement unless in rotary mode (MODMODE=1). The direction depends on the sign of **DIVISIONS** and on the value of **DIR**.



**IN2MODE=23:** The motion velocity is **MISPEED0**. There is no movement unless in rotary mode (MODMODE=1). The direction depends on the sign of **DIVISIONS** and on the value of **DIR**.

**IN1MODE=24:** With the drive starting in the position-mode (OPMODE=8) with the **INPUT** = Low; a *Low to High* transition of the INPUT changes the drive to the gearing-mode (OPMODE=4); with a *High to Low* transition, the drive is changed back to the position-mode (OPMODE=8). These **OPMODE** changes occur when the drive is enabled or disabled.

**Firmware Versions:** see above

**Range:** 0 to 24

**Opmodes:** dependent

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 2

**Drive Status:** EN/DIS

**EEPROM:** N/A

## IN3

This is used to read the state of the hardware input on user connector C3 Pin 11.

**Firmware Versions:** 2.1.0 and later

**Range:** 0 (OFF), 1(ON)

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** N/A

## IN3MODE

**IN3MODE** sets the functionality of the **IN3** input. The functions are:

**IN3MODE=0:** No function\*

**IN3MODE=1:** CW limit switch\*

**IN3MODE=2:** CCW limit switch\*

**IN3MODE=3:** Gear disable input (all **GEARMODE** values)\*

**IN3MODE=4:** Gear mask input (all **GEARMODE** values)\*

**IN3MODE=5:** Gear A input (GEARMODE = 0-2)\*

**IN3MODE=6:** Gear B input (GEARMODE = 0-2)\*

**IN3MODE=7:** Trigger incremental move / jog \*\*

**IN3MODE=8:** Use second current limit (**ILIM2**)\*\*

**IN3MODE=9:** Switch **OPMODE** from 1 to 3\*\*

**IN3MODE=10:** Home switch\*\*

**IN3MODE=11:** Reserved

**IN3MODE=12:** Search for absolute machine zero/Move to absolute machine zero (home)\*\*

**IN3MODE=13:** Trigger absolute move (**MAPOS** at **MASPEED**)\*\*

**IN3MODE=14:** Binary **MIDIST** / **MISPEED** selection code MSB\*\*

**IN3MODE=15:** Binary **MIDIST** / **MISPEED** selection code LSB\*\*

**IN3MODE=16:** Reserved

**IN3MODE=17:** Trigger active disable (see **DISSPEED**)\*

**IN3MODE=18:** Control fault relay\*\* (see notes below)

**IN3MODE=19:** Hold position\*

**IN3MODE=20:** When OPMODE=1, an active input switches to OPMODE=4 (must be disabled or 0 velocity)

- IN3MODE=21:** Trigger incremental move \*\*\* (see notes below)
- IN3MODE=22:** Triggers incremental move when MODMODE=1.  
Distance = (**PROTARY / DIVISIONS**)\*\*\*\*\*
- IN3MODE=23:** Trigger incremental move when MODMODE=1.  
Distance = (2 \* **PROTARY / DIVISIONS**)\*\*\*\*\*
- IN3MODE=24:** Change OPMODE 8 to OPMODE 4 using digital input. \*\*\*\*
- IN3MODE=25:** Upon transition from 0 to 1, the faults are cleared if REMOTE=0 or SWEN=0.\*\*\*\*\*
- \* available in firmware versions 2.1.0 and later  
 \*\* available in firmware versions 3.1.0 and later  
 \*\*\* available in firmware versions 3.5.0 and later  
 \*\*\*\* available in firmware versions 4.0.0 and later  
 \*\*\*\*\* available in firmware versions 4.1.8 and later

### Notes:

- IN3MODE = 7** operation: If the drive is in OPMODE 8 (positioning), an input with IN3MODE=7 can be used to trigger an incremental move (**MI**), using the variables MIDIST0-3, MISPEED0-3. See the descriptions for those variables, as well as the description of **MH**, for more information. If the drive is in OPMODE 1 (analog velocity), an input with IN3MODE =7 can be used to trigger a jog at a speed entered in MISPEED0-3. See the description of MISPEED0 for more details. IN3MODE 7 can be activated while the motor is in motion. This causes the motor to begin the next index immediately after completing the previous index (called *look-ahead-buffering*).
- IN3MODE = 9** operation: If the drive is in OPMODE 1 (analog velocity) and IN3MODE is 9, then switching the related **IN3** input to 1 causes the drive to switch to OPMODE 3 (analog current). Switching the **IN3** input back to 0 causes the drive to switch back to OPMODE 1 (analog velocity). The LED display of **OPMODE** changes according to the user input, but the serial response to a prompt for **OPMODE** returns 1. The **OPMODE** change can happen when the drive is enabled, therefore the user must make the switch with zero command.
- IN3MODE = 12** operation: Execute **HOMETYPE** (see note in **HOMETYPE**). Moves are performed at velocity equal to **HOMESPD**.
- IN3MODE = 18:** if input = 0, the fault relay will open. If input = 1, the fault relay operates as normal.
- IN3MODE=21:** The same function as IN3MODE=7 except that IN3MODE=21 ignores input signals until the motor has completed in prior index (comes to rest). There is no look-ahead-buffering.

**IN3MODE=22:** The motion velocity is **MISPEED0**. There is no movement unless in rotary mode (MODMODE=1). The direction depends on the sign of **DIVISIONS** and on the value of **DIR**.

**IN3MODE=23:** The motion velocity is **MISPEED0**. There is no movement unless in rotary mode (MODMODE=1). The direction depends on the sign of **DIVISIONS** and on the value of **DIR**.

**IN1MODE=24:** With the drive starting in the position-mode (OPMODE=8) with the INPUT = Low; a Low --> High transition of the **INPUT** changes the drive to the gearing-mode (OPMODE=4); with a High --> Low transition, the drive is changed back to the position-mode (OPMODE=8). These **OPMODE** changes occur when the drive is enabled or disabled.

**Firmware Versions:** see above

**Range:** 0 to 24

**Opmodes:** dependent

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 3

**Drive Status:** EN/DIS

**EEPROM:** N/A

## IN4

This is used to read the state of digital input 4 on the extended I/O connector.

**Firmware Versions:** 6.3.3 and later

**Range:** 0 (OFF), 1=(ON)

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## IN4MODE

**IN4MODE** sets the functionality of the **IN4** input. The functions are:

**IN4MODE= 0:** No function

**IN4MODE= 1:** CW limit switch

**IN4MODE= 2:** CCW limit switch

**IN4MODE= 7:** Trigger incremental **MOVE / JOG**

**IN4MODE= 8:** Use second current limit (**ILIM2**)

**IN4MODE= 9:** Switch **OPMODE** from 1 to 3

**IN4MODE= 12:** Search for absolute machine zero/ Move to absolute machine zero (home)

**IN4MODE= 13:** Trigger absolute move (**MAPOS** at **MASPEED**)

**IN4MODE= 14:** Binary **MIDIST / MISPEED** selection code MSB

**IN4MODE= 15:** Binary **MIDIST / MISPEED** selection code LSB

**IN4MODE= 17:** Trigger active disable (see **DISSPEED**)

**IN4MODE= 18:** Control fault relay (see notes below)

**IN4MODE= 19:** Hold position

**IN4MODE= 20:** When OPMODE= 1, an active input switches to OPMODE= 4 (must be disabled or 0 velocity)

**IN4MODE= 21:** Trigger incremental move (see notes below)

**IN4MODE= 22:** Triggers incremental move when MODMODE= 1. Distance = (**PROTARY / DIVISIONS**)



- IN5MODE= 23:** Trigger incremental move when MODMODE= 1.  
Distance = (2 \* **PROTARY / DIVISIONS**)
- IN5MODE= 24:** Change OPMODE 8 to OPMODE 4 using digital input.
- IN5MODE= 25:** Upon transition from 0 to 1, the faults are cleared if REMOTE= 0 or SWEN= 0.
- IN5MODE=26:** Upon transition from 0 to 1, the number of counts represented by **PFB** are output on the Equivalent Encoder Output, interrupting the regular generation of equivalent encoder output pulses. The digital Output (**O1**) is set high if **O1MODE** is set to the value 13. At the end of the generation of the **PFB** pulses, the equivalent encoder output reverts to generation of encoder pulses, and the digital output is set low.

**Syntax:** IN5MODE <value>

**Default:** 0

**Range:** 0, 1,2,7,8,9,12,13,14,15,17,18,19,20,21,22,23,24,25

**Firmware Versions:** 6.3.3 and later

**Type:** switch mode (R/W)

**Opmodes:** dependent

**Drive Status:** EN/DIS

**Units:** N/A

**EEPROM:** No

## IN6

This is used to read the state of digital input 6 on the extended I/O connector.

**Firmware Versions:** 6.3.3 and later

**Type:** switch (R)

**Range:** 0 (OFF), 1=(ON)

**Default:** N/A

**Opmodes:** All

**Drive Status:** EN/DIS

**Units:** N/A

**EEPROM:** No

## IN6MODE

**IN6MODE** sets the functionality of the **IN4** input. The function list is:

**IN6MODE= 0:** No function

**IN6MODE= 1:** CW limit switch

**IN6MODE= 2:** CCW limit switch

**IN6MODE= 7:** Trigger incremental **MOVE / JOG**

**IN6MODE= 8:** Use second current limit (**ILIM2**)

**IN6MODE= 9:** Switch **OPMODE** from 1 to 3

**IN6MODE= 12:** Search for absolute machine zero/ Move to absolute machine zero (home)

**IN6MODE= 13:** Trigger absolute move (**MAPOS** at **MASPEED**)

**IN6MODE= 14:** Binary **MIDIST / MISPEED** selection code MSB

**IN6MODE= 15:** Binary **MIDIST / MISPEED** selection code LSB

**IN6MODE= 17:** Trigger active disable (see **DISSPEED**)

**IN6MODE= 18:** Control fault relay (see notes below)

**IN6MODE= 19:** Hold position

**IN6MODE= 20:** When OPMODE= 1, an active input switches to OPMODE= 4 (must be disabled or 0 velocity)

**IN6MODE= 21:** Trigger incremental move (see notes below)

**IN6MODE= 22:** Triggers incremental move when MODMODE= 1. Distance = (**PROTARY / DIVISIONS**)

- IN6MODE= 23:** Trigger incremental move when MODMODE= 1. Distance = (2 \* **PROTARY / DIVISIONS**)
- IN6MODE= 24:** Change OPMODE 8 to OPMODE 4 using digital input.
- IN6MODE= 25:** Upon transition from 0 to 1, the faults are cleared if REMOTE= 0 or SWEN= 0.
- IN6MODE=26:** Upon transition from 0 to 1, the number of counts represented by **PFB** are output on the Equivalent Encoder Output, interrupting the regular generation of equivalent encoder output pulses. The digital Output (**O1**) is set high if **O1MODE** is set to the value 13. At the end of the generation of the **PFB** pulses, the equivalent encoder output reverts to generation of encoder pulses, and the digital output is set low.

**Syntax:** IN6MODE <value> **Default:** 0  
**Range:** 0, 1,2,7,8,9,12,13,14,15,17,18,19,20,21,22,23,24,25  
**Firmware Versions:** 6.3.3 and later **Type:** switch mode (R/W)  
**Opmodes:** dependent **Drive Status:** EN/DIS  
**Units:** N/A **EEPROM:** No

## INITMODE

Sets the type of initialization for the **ENCSTART** encoder initialization process.

- 0 – soft exponential ramp (motor will rotate CW and CCW several times, fault occurs if unsuccessful)
- 1 – hard step (motor jerks into position, fault occurs if unsuccessful)
- 2 – Initialization with minimal motion



### WARNING

*When using either option, disconnect the load from the motor shaft. Option 0 rotates the load in both directions and may be undesirable for the application. Option 1 suddenly jerks the motor into position and could cause damage to the load or surrounding environment.*

**Syntax for set:** INITMODE <value> **Syntax for query:**  
INITMODE  
**Firmware Versions:** 3.4.0 and later **Type:** variable (R/W)  
**Range:** 0 to 2 **Default:** 0  
**Opmodes:** All **Drive Status:** EN/DIS  
**Units:** N/A **EEPROM:** Yes

## ININV4

Invert digital input 4 on extended I/O.

- 0 – user input not inverted.
- 1 – user input inverted.

**Syntax:** ININV4 <value> **Type:** switch mode (R/W)  
**Firmware Versions:** 6.3.3 and later **Default:** 0  
**Range:** 0, 1 **Drive Status:** EN/DIS  
**Opmodes:** dependent **EEPROM:** No  
**Units:** N/A

## ININV5

Invert digital input 5 on extended I/O.

**0** – user input not inverted.

**1** – user input inverted.

**Syntax:** ININV5 <value>

**Firmware Versions:** 6.3.3 and later

**Range:** 0, 1

**Opmodes:** dependent

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** No

## ININV6

Invert digital input 6 on extended I/O.

**0** – user input not inverted.

**1** – user input inverted.

**Syntax:** ININV6 <value>

**Firmware Versions:** 6.3.3 and later

**Range:** 0, 1

**Opmodes:** dependent

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** No

## INDEXPOS

Determines the position of the marker signal in the encoder simulation for resolver based systems.

**Firmware Versions:** 3.4.0 and later

**Range:** 0, 45, 90, 135, 180, 225, 270, 315

**Opmodes:** All

**Units:** N/A

**Type:** variable

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## ININV1

Invert user input #1 C3 pin 9.

**0** – user input not inverted.

**1** – user input inverted.

**Firmware Versions:** 3.4.0 and later

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0

**Drive Status:**

**EEPROM:** Yes

## ININV2

Invert user input #2 C3 pin 10.

**0** – user input not inverted.

**1** – user input inverted.

**Firmware Versions:** 3.4.0 and later

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0

**Drive Status:**

**EEPROM:** Yes

## ININV3

Invert user input #3 C3 pin 11.  
**0** – user input not inverted.  
**1** – user input inverted.

**Firmware Versions:** 3.4.0 and later  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch mode (R/W)  
**Default:** 0  
**Drive Status:**  
**EEPROM:** Yes

## INITGAIN

Sets the gain for the encoder initialization process controller.

**Syntax for set:** INITGAIN <value>

**Type:** R/W

**Syntax for query:** INITGAIN

**Opmodes:** All

**Range:** 100 – 10000

**Default:** 1000

**Drive Status:** DIS

**EEPROM:** Yes

**Limitations:** Not yet supported by **MOTIONLINK**

## INITMODE

Sets the type of initialization for the **ENCSTART** encoder initialization process.

**0** – soft exponential ramp (motor will rotate CW and CCW several times, fault occurs if unsuccessful)

**1** – hard step (motor jerks into position, fault occurs if unsuccessful)



**WARNING**

*When using either option, disconnect the load from the motor shaft. Option 0 rotates the load in both directions and may be undesirable for the application. Option 1 suddenly jerks the motor into position and could cause damage to the load or surrounding environment.*

**Syntax for set:** INITMODE <value>  
**Firmware Versions:** 3.4.0 and later  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Syntax for query:** INITMODE  
**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## INPOS

Indicates if the actual position (**PFB**) is following the commanded position (**PCMD**) within the following error set by **PEINPOS**.

**0** – not in position

**1** – in position

**Firmware Versions:** 2.1.0 and later  
**Range:** 0, 1  
**Opmodes:** 4,8  
**Units:** N/A

**Type:** switch (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** N/A



## ISCALE

**ISCALE** is an analog current scale factor that scales (1) the analog input **ANIN** for OPMODE 3 (analog torque mode), and (2) the analog output for ANOUT=1 or 3. A 10 V input provides the percentage of **DIPEAK** defined by **ISCALE**. Units for **ISCALE** are 0.1%. This variable may be either higher or lower than 100%, but the actual analog I/O is limited by the application current limit (**ILIM**). If **ISCALE** = 1000 and a 10 V command is given, 100% of Peak Current (or **DIPEAK**) is commanded. The default **ISCALE** values in the SERVOSTAR CD are defined so 8 volts at the input or output is equal to  $2 \times \text{DICONT}$ . This leaves a 25% margin for overshoot at the controller.

### CD10

Where  $\text{DIPEAK} = 2 \times \text{DICONT}$ , the default value of **ISCALE** is 1250. This is so 8 volts (not 10) at the input is equal to  $2 \times \text{DICONT}$  (or **DIPEAK**). The equation is:  
 $\text{ISCALE} = 1250$  then:  
 $8 \text{ V command} \times 1250 / 10 = 1000 = \text{DIPEAK} = 2 \times \text{DICONT}$

### CD03 and CD06

Where  $\text{DIPEAK} = 3 \times \text{DICONT}$ , the default value of **ISCALE** is 833. This is so 8 volts at the input is equal to  $2 \times \text{DICONT}$ . In this case,  $2 \times \text{DICONT}$  is equal to  $2/3 \times \text{DIPEAK}$ . If **ISCALE** = 833 then:  
 $8 \text{ volt command} \times 833 / 10 = 666 = 2/3 \times \text{DIPEAK} = 2 \times \text{DICONT}$   
 If **ISCALE** = 1250 then:  
 $8 \text{ volt command} \times 1250 / 10 = 1000 = \text{DIPEAK} = 3 \times \text{DICONT}$

<b>Firmware Versions:</b> All	<b>Type:</b> variable (R/W)
<b>Range:</b> 100 to 10,000	<b>Default:</b> 1250 for the SERVOSTAR S
<b>Opmodes:</b> 3	833 for the SERVOSTAR CD
<b>Units:</b> (% $\text{DIPEAK} \times 0.1$ ) / 10V	<b>EEPROM:</b> Yes
<b>Drive Status:</b> EN/DIS	

## ISTOP

Sets the current command for the braking function. See **STOPMODE**.

<b>Firmware Versions:</b> 2.1.0 and later	<b>Type:</b> variable (R/W)
<b>Range:</b> 0 to IMAX	<b>Default:</b> DICONT
<b>Opmodes:</b> All	<b>Drive Status:</b> EN/DIS
<b>Units:</b> % of $\text{DIPEAK} \times 0.1$	<b>EEPROM:</b> Yes

## IZERO

Sets the C-B phase current for ZERO Mode (A=0). See **XENCRES**.

<b>Firmware Versions:</b> All	<b>Type:</b> variable (R/W)
<b>Range:</b> 1 to 100	<b>Default:</b> 25
<b>Opmodes:</b> All	<b>Drive Status:</b> EN/DIS
<b>Units:</b> % of MICONT	<b>EEPROM:</b> Yes

## J

Sets the continuous jog speed and initiates motion at that speed if the motor is currently enabled (see **EN** and **REMOTE**) in OPMODE 0. **J** is set to 0 whenever the drive is disabled or enabled, or the operational mode is changed to prevent the motor from moving when enabled. **J** has an optional parameter of *for-time*, in milliseconds. Not available when the drive is in Hold mode.

**Command Syntax:** J [*speed*] {*time*}

*speed* (required) = -VLIM to +VLIM in rpm (rotary) or mm/sec (linear)

*time* (optional) = 0 to 32767 in milliseconds

**Firmware Versions:** All

**Opmodes:** 0

**Drive Status:** EN

## K

**K** functions the same as **DIS**, but provides a one-key hot-button. The drive is disabled and the motor may coast when this command is issued.

**Firmware Versions:** All

**Command Syntax:** K

**Opmodes:** All

**Drive Status:** EN/DIS

## KV

**KV** is a tuning variable which sets the proportional gain for the Pseudo Derivative Feedback with Feed-Forward Velocity Control Loop (**PDFF** loop; COMPMODE = 1). This variable is set manually by the user.

Executing **TUNE** successfully may change the value of this parameter.

**Firmware Versions:** All

**Range:** 0 to 1,000,000,000 (firmware versions 3.3.0 and later)

0 to 65,535 (firmware versions 2.2.0 to 3.2.1)

0 to 32,767 (firmware versions up to and including 2.1.0)

**Drive Status:** EN/DIS

**Opmodes:** 0, 1, 4, 8

**Units:** N/A

**Default:** 1000

**Type:** variable (R/W)

**EEPROM:** Yes

## KVFR

**KVFR** is a tuning variable which sets the feed-forward to feedback gain ratio for the Pseudo Derivative Feedback with Feed-Forward Velocity Control Loop (**PDFF** loop; COMPMODE = 1). This variable is set manually by the user. Executing **TUNE** successfully may change the value of this parameter.

**Firmware Versions:** All

**Type:** variable (R/W)

**Range:** 0 to 1000

**Default:** 0

**Opmodes:** 0,1,4,8

**Drive Status:** EN/DIS

**Units:** % \*0.1

**EEPROM:** Yes

## KVI

**KVI** sets the integral gain for the Pseudo Derivative Feedback with Feed-Forward Velocity Control Loop (**PDFF** loop; **COMP**MODE=1). This variable is set manually. Executing **TUNE** successfully may change the value of this parameter.

**Firmware Versions:** All  
**Range:** 0 to 65535  
**Opmodes:** 0,1,4,8  
**Units:** N/A

**Type:** variable (R/W)  
**Default:** 1000  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## LIMDIS

Enables/disables the End Travel Limit function. This function only pertains to units with the limit switch option.

**0** = limit switch function enabled

**1** = limit switch function disabled; LED decimal point flashes

**Firmware Versions:** All  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## LIST

Dumps a list of valid commands and variables to the serial port. Only the names of variables are transmitted, not values.



### CAUTION

*Some factory variables and commands not intended for use by the user may be printed. Do not use commands and variables that are not described in this guide.*

**Firmware Versions:** All  
**Opmodes:** All

**Command Syntax:** LIST  
**Drive Status:** EN/DIS

## LMJR

Sets the ratio of the estimated Load Moment of Inertia (**LMJ**) relative to the Motor Moment of Inertia (**MJ**). **LMJR** and **MJ** and the required closed loop bandwidth (**BW**) are used for the Velocity Control Loop design in the Standard Pole-Placement controller (**COMP**MODE = 2 or 4). Executing **TUNE** successfully may change the value of this parameter.

**Firmware Versions:** All  
**Range:** 0 to 10,000  
**Opmodes:** All  
**Units:** percent of MJ

**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## LOAD

Loads all variables saved in the EEPROM into system RAM. This command is automatically executed on power-up.

**Firmware Versions:** All  
**Opmodes:** All

**Command Syntax:** LOAD  
**Drive Status:** DIS

## LPFHZ1

Sets the cutoff frequency of the first Low Pass Filter (LPF) in the velocity loop. This variable only affects the system when FILTMODE=1 or 2.

**Firmware Versions:** All  
**Range:** 20 to 800, steps of  
 20 (20, 40, ..., 800)  
**Opmodes:** 0,1,4,8  
**Units:** Hz

**Type:** variable (R/W)  
**Default:** 500  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## LPFHZ2

Sets the cutoff frequency of the second Low Pass Filter (LPF) in the velocity loop. This variable only affects the system when FILTMODE=2.

**Firmware Versions:** All  
**Range:** 20 to 800, steps of  
 20 (20, 40, ..., 800)  
**Opmodes:** 0,1,4,8  
**Units:** Hz

**Type:** variable (R/W)  
**Default:** 500  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## MA

This command (Serial Move Absolute) moves to the specified position at the specified speed. Motion can occur in either direction, depending upon the relationship between the starting position and the commanded position. The current position of the motor can be read using **PFB**.

The optional flag [*in pos ack*] enables the drive to indicate when the commanded move is completed. When this flag is set to 1, the drive, upon completion of the commanded move, outputs an exclamation point (!) over the serial port. The exclamation point is tied to the **STOPPED** flag and is issued when **STOPPED** transitions from 0 to 1.

Position is in feedback counts. Issuing an **MA** command while the motor is not stopped causes the command to be buffered. *<velocity>* is always given in rpm or mm/sec and is not affected by units. See **INPOS**, **MI**, **PCMD**, **PEINPOS**, **PFB**, and **STOPPED**.

in pos ack = 0: do not indicate when move is complete

in pos ack = 1: indicate when move is complete

**Command Syntax:** MA *<position>* *<velocity>* [*in pos ack*]

**Firmware Versions:** 3.1.0 and later

**Range:** *<position>* -LONG to LONG (feedback counts)\*  
*<velocity>* 1 to VMAX (rpm or mm/sec)  
 [*in pos ack*] 0 or 1 (optional)

**Opmodes:** 8

**Drive Status:** EN

**Example:**

MA 10000 1000

(Move to absolute position 10,000 at a speed of 1,000 rpm)

MA -5000 100 1

(Move to absolute position -5000 at a speed of 100 rpm; transmit a (!) to the serial port when the move is completed)

**NOTE**

*\*Actual move distance (position to be commanded – present commanded position) must not exceed  $\pm 2,147,482,647$  counts or motion occurs in the opposite direction.*

**NOTE**

*For very slow moves, set VLIM as low as possible before setting <velocity> less than 1 in MA and MI. (This can only be set in firmware version 3.7.0 and higher.)*

## MAPOS

Sets the absolute position for IN1MODE=13, or IN2MODE=13, or IN3MODE=13. Position is in counts, where 65536 counts represent one motor revolution for resolver systems, and 4\*MENCRES counts represents one motor revolution for encoder systems.

When **IN1MODE**, **IN2MODE**, or **IN3MODE** is set equal to 13, and the associated digital input (IN1, IN2, or IN3) goes high, the drive moves to **MAPOS** at a speed of **MASPEED**.

**Firmware Versions:** 3.1.0 and later

**Type:** variable (R/W)

**Range:** - 2,147,483,647 to  
+ 2,147,483,647 \*

**Default:** 0

**Opmodes:** 8

**Drive Status:** EN/DIS

**Units:** counts

**EEPROM:** Yes

**NOTE**

*\*Actual move distance (position about to be commanded – present commanded position) must not exceed  $\pm 2,147,482,647$  counts or the motion occurs in the opposite direction.*

## MASPEED

Sets the move speed (unsigned) for INxMODE=13. When **IN1MODE**, **IN2MODE**, or **IN3MODE** is set equal to 13, and the associated digital input (**IN1**, **IN2**, or **IN3**) goes high, the drive moves to **MAPOS** at a speed of **MASPEED**.

**Firmware Versions:** 3.1.0 and later

**Type:** variable (R/W)

**Range:** 1 to VMAX

**Default:** 0

**Opmodes:** 8

**Drive Status:** EN/DIS

**Units:** rotary: rpm  
linear: mm/sec

**EEPROM:** Yes

## MBEMF

Displays the motor's back EMF constant. This value is used for current loop controller design. This variable requires issuing a **CONFIG** when changed.

**Firmware Versions:** 3.1.0 and later

**Type:** variable (R/W)

**Range:** 1 to 3900

**Default:** motor data

**Opmodes:** All

**Drive Status:** DIS

**Units:** rotary: ( $V_{RMS}$ ) / krpm  
linear: ( $V_{Peak}$ ) / (m/sec)

**EEPROM:** Yes

## MBEMFCOMP

Sets a back EMF compensation percentage value. This variable affects the amount of back EMF compensation that is applied to the motor command.



### NOTE

*For firmware version, 2.0.0, this variable was BEMFCOMP.*

**Firmware Versions:** 2.0.1 and later

**Range:** 1 to 130

**Opmodes:** All

**Units:** percent

**Type:** variable (R/W)

**Default:** 50, or motor data

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MENCOFF

Sets the encoder index position (encoder feedback systems only). This variable is expressed in units of encoder counts after quadrature, and the range is from 0 to (4\*encoder resolution-1), or (4\*MENCRES-1).

This variable is automatically set using **ENCINIT**.

**Firmware Versions:** All

**Range:** 0 to (4\*MENCRES) – 1

**Opmodes:** All

**Units:** encoder counts per  
mechanical motor rev

**Type:** variable (R/W)

**Default:** motor data  
(120 degrees if undefined)

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MENCRES

Displays the resolution of the motor encoder (encoder feedback systems only) in number of lines per revolution of the motor.



### NOTE

*The number of encoder counts per revolution is obtained by multiplying MENCRES by 4.*

This variable requires a **CONFIG** command when changed.



### NOTE

*Prior to firmware version 3.2.0, the lowest valid value for MENCRES was 100.*

**Firmware Versions:** All

**Range:** 1 to 10,000,000

**Opmodes:** All

**Units:** rotary: lines/motor rev  
linear: lines/pitch

**Type:** variable (R/W)

**Default:** motor data

**Drive Status:** DIS

**EEPROM:** Yes

# MENCTYPE

Sets the motor encoder type. When this variable is changed on an encoder-based system, the drive enters a no-comp state, requiring a **CONFIG** command (see **CONFIG**). In version 1 firmware prior to 1.2.0, MENCTYPE=0 is assumed. This variable may take value from 0 to 9.

DIGITAL ENCODER OPTIONS				
MENCTYPE	A/B quad	Marker Pulse	Hall Effects	Firmware Versions
0	X	X	X	All
1	X	X		All
2	X	X		All
3	X			All
4	X			All
5			X	4.1.8 and later
6	X		X	All
SINE ENCODER OPTIONS				
0	X	X	X	3.4.0 and later
7	X	X	C/D Channels	3.4.0 and later
8	X		C/D Channels	3.4.0 and later
9	X		Endat	3.4.0 and later

**MENCTYPE 0 can be used with both the Digital and Sine option. The drive automatically senses the Sine Encoder board and make the appropriate adjustments.**



## NOTE

**All MENCTYPEs applicable for incremental encoders are applicable for sine encoders. This also includes the "wake-and-shake" initialization.**

INITIALIZATION METHODS	
MENCTYPE	Method of Initialization
0	Initialization is automatic upon power-up. Marker pulse location may be performed using <b>ENCINIT</b> to locate the marker.
1	Initialization is required and is triggered by the <b>ENCSTART</b> command. This may optionally be followed by marker pulse location using <b>ENCINIT</b> .
2	Initialization is required and is triggered on power up (on drive enable) or by using <b>ENCSTART</b> . This may optionally be followed by marker pulse location using <b>ENCINIT</b> .
3	Initialization is required and is triggered by <b>ENCSTART</b> .
4	Initialization is required and is triggered on power up (when the drive is enabled) or by using <b>ENCSTART</b> .
5	Halls only (version 4.1.8 and higher).
6	Initialization is automatic upon power-up.
7	Initialization is automatic upon power-up. Marker pulse location may be performed using <b>ENCINIT</b> to locate the marker.
8	Initialization is automatic upon power-up.
9	Initialization is automatic upon power-up.



*All initializations above are completely automatic when using standard Kollmorgen motors.*

*For MENCTYPE 0-2 and 7, perform ENCINIT where MENCOFF is unknown.*

#### NOTE

**Firmware Versions:** All

**Range:** 0 to 9

**Opmodes:** All

**Drive Status:** DIS

**Type:** switch mode (R/W)

**Default:** motor data (0 if undefined)

**Units:** N/A

**EEPROM:** Yes

## MFBDIR

Sets the motor feedback direction. This switch is used during the autoconfig process (**ACONFIG**). If Phase 5 of autoconfig (direction test) fails, use **MFBDIR** to switch the motor feedback direction. The range is:

MFBDIR = 0: normal commutation direction, normal velocity direction.

MFBDIR = 1: inverted commutation direction, normal velocity direction.

MFBDIR = 2: normal commutation direction, inverted velocity direction.

MFBDIR = 3: inverted commutation direction, inverted velocity direction.

If the motor leads/motor stator are phased incorrectly, setting MFBDIR=1 solves the problem. If the feedback device is connected/phased incorrectly, you have to set MFBDIR=3, which changes both the commutation and velocity loop directions.

**Firmware Versions:** 3.3.0 and later

**Range:** 0 to 3

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0

**Drive Status:** DIS

**EEPROM:** Yes

## MFOLD

Displays the status of the motor foldback circuit. When the system current level exceeds **MICONT** for too long, the drive enters motor foldback mode, **MFOLD** changes from 0 to 1, and the drive current is limited gradually (in exponential fashion) to the value of **MICONT**. See also **FOLD**, **FOLDMODE**, **MFOLDD**, **MFOLDDIS**, **MFOLDR**, AND **MFOLDT**.

**0** = motor foldback OFF (inactive)

**1** = motor foldback ON (drive is limiting output current)

**Firmware Versions:** 3.2.0 and later

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** No



## MFOLDD

Sets the delay time for motor foldback. This is the amount of time that the system current can exceed **MICONT** before the drive enters the motor foldback state. The time units assume a worst-case scenario where the drive is applying **MIPEAK** current. A current level of less than **MIPEAK** can be allowed for a longer time. See **FOLD**, **FOLDMODE**, **MFOLD**, **MFOLDDIS**, **MFOLDR**, and **MFOLDT**.

**Firmware Versions:** 3.2.0 and later

**Range:** 1 to 2400

**Opmodes:** All

**Units:** seconds at MIPEAK

**Type:** variable (R/W)

**Default:** 1200

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MFOLDDIS

Enables/disables the motor current foldback function. See **FOLD**, **FOLDMODE**, **MFOLD**, **MFOLDD**, **MFOLDR**, and **MFOLDT**.

**0** = enable motor foldback function

**1** = disable motor foldback function

**Firmware Versions:** 3.2.0 and later

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 1 (disabled)

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MFOLDR

Sets the recovery time for motor foldback. After the drive enters the motor foldback state (MFOLD=1), and the current folds back to **MICONT**, this is the amount of time that the current is held at **MICONT** or below before it is allowed to exceed **MICONT** again. See **FOLD**, **FOLDMODE**, **MFOLD**, **MFOLDD**, **MFOLDDIS**, and **MFOLDT**.

**Firmware Versions:** 3.2.0 and later

**Range:** 900 to 3600

**Opmodes:** All

**Units:** seconds

**Type:** variable (R/W)

**Default:** 1800

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MFOLDT

Sets the time constant for motor foldback. After the drive enters the motor foldback state (MFOLD=1), this variable defines how long it takes the drive to reduce the system current level to **MICONT**. See **FOLD**, **FOLDMODE**, **MFOLD**, **MFOLDD**, **MFOLDDIS**, and **MFOLDR**.

**Firmware Versions:** 3.2.0 and later

**Range:** 1 to 1200

**Opmodes:** All

**Units:** seconds

**Type:** variable (R/W)

**Default:** 600

**Drive Status:** EN/DIS

**EEPROM:** Yes

# MH

This command (Move Home) causes the motor to move to the home position. **HOMESPD** controls the speed. **HOMETYPE** defines the type of homing sequence, while **HOMESTATE** gives the homing status and describes the homing process further.

After power up, or after a feedback loss fault, the first issue of this command causes a search for home. Further issues of this command causes a move to home position (equivalent to MA 0 command).

**HOMESPD** controls the speed and direction of the search. **PFB** is normalized to 0 after completion.

The optional flag [*in pos ack*] enables you to direct the drive to indicate when the commanded **MOVE** is completed. When this flag is set to 1, the drive, upon completion of the commanded **MOVE**, outputs an exclamation point (!) over the serial port. The exclamation point is tied to the **STOPPED** flag and is issued when **STOPPED** transitions from 0 to 1.

## Homing Process Using MH command:

- Issue **MH** via serial port
- If home switch is already pressed (see **IN1MODE = 10**). The drive moves in the opposite homing direction, until home switch is released.
- If home switch is not already pressed, the drive move in the homing direction, until home switch is pressed.
- Drive captures position of first index (or resolver 0) past switch.
- Drive decelerates to stop using **DEC**.
- Drive goes back to home position using **ACC**, **DEC**, and **HOMESPD** (in firmware versions 3.3.0 and later, the speed is **MISPEED0**). Drive resets the absolute position (**PFB**) to 0.

## Homing Process using digital I/O (see descriptions for IN<sub>x</sub>, and IN<sub>x</sub>MODE for IN<sub>x</sub>MODE 10 and 12):

*When Home Input is activated for the first time after power-up:*

- If home switch is already pressed (see **IN<sub>x</sub>MODE 10**), the drive moves in the opposite homing direction, until home switch is released.
- If home switch is not already pressed, drive moves in the homing direction, until home switch is pressed.
- Drive captures position of first index (or resolver 0) past switch.
- Drive decelerates at **DEC** to stop.
- Drive goes back to home position using **ACC**, **DEC**, and **HOMESPD** (in firmware versions 3.3.0 and later, the speed is **MISPEED0**).
- Drive resets the absolute position (**PFB**) to 0

*Otherwise:*

Drive moves to absolute 0 at **HOMESPD**.

in pos ack = 0: do not indicate when move is complete

in pos ack = 1: indicate when move is complete

**Firmware Versions:** 3.1.0 and later

**Opmodes:** 8

**Drive Status:** EN

**Command Syntax:** MH [*in pos ack*]

**Range:** [*in pos ack*] 0 or 1 (optional)

## MHINVA

**MHINVA** applies to encoder-based systems which use hall switches to commutate. This variable inverts the hall sensor A feedback, causing the system to read the A hall channel as inverted data.

MHINVA = 0: do not invert hall A

MHINVA = 1: invert hall A

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 0

**Drive Status:** DIS

**EEPROM:** Yes

## MHINVB

**MHINVB** applies to encoder-based systems which use hall switches to commutate. This variable inverts the hall sensor B feedback, causing the system to read the B hall channel as inverted data.

MHINVB = 0: do not invert hall B

MHINVB = 1: invert hall B

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 0

**Drive Status:** DIS

**EEPROM:** Yes

## MHINVC

**MHINVC** applies to encoder-based systems which use hall switches to commutate. This variable inverts the hall sensor C feedback, causing the system to read the C hall channel as inverted data.

MHINVC = 0: do not invert hall C

MHINVC = 1: invert hall C

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 0

**Drive Status:** DIS

**EEPROM:** Yes

## MI

**MI** (serial Move Incremental) incrementally moves the specified distance at the specified speed. A positive incremental move occurs in the direction determined by the variable **DIR**, and a negative incremental move occurs in the opposite direction.

*[in pos ack]* enables the operator to direct the drive to indicate when the commanded move is completed. When this flag is set to 1, the drive, upon completion of the commanded move, outputs an exclamation point (!) over the serial port. The exclamation point is tied to the **STOPPED** flag and is issued when **STOPPED** transitions from 0 to 1.

Position is in feedback counts. Issuing **MI** while the motor is not stopped causes the command to be buffered for later execution. No more than one command is buffered. *<velocity>* is always given in rpm or mm/sec and is not affected by units. See **INPOS**, **MA**, **PCMD**, **PEINPOS**, **PFB**, and **STOPPED**.

in pos ack = 0: do not indicate when move is complete

in pos ack = 1: indicate when move is complete

**Firmware Versions:** 3.1.0 and later

**Command Syntax:** MI *<pos>* *<vel>* [*in pos ack*]

**Opmodes:** 8                      **Drive Status:** EN

**Range:** *<pos>* LONG to LONG (feedback counts)

*<vel>* 1 to VMAX (rpm or mm/sec)

[*in pos ack*] 0 or 1 (optional)

**Example:**

MI 10000 1000 (move 10,000 counts in the positive direction at a speed of 1,000 rpm)

MI -10000 100 1 (move 10,000 counts in the negative direction at a speed of 100 rpm; transmit a (!) to the serial port when the move is complete)



**NOTE**

*For very slow moves, set VLIM as low as possible before setting <velocity> less than 1 in MA and MI. This can only be set in firmware version 3.7.0 and higher.*

## MICONT

Sets the motor's continuous rated current. When this variable is changed, the drive enters a no-comp state, requiring a **CONFIG**.

**Firmware Versions:** All

**Range:** 1 to 1750

**Opmodes:** All

**Units:** amp<sub>RMS</sub>\*0.1

**Type:** variable (R/W)

**Default:** motor data

**Drive Status:** DIS

**EEPROM:** Yes

## MIDIST0

**MIDIST0**, with **MIDIST1**, **MIDIST2**, and **MIDIST3**, set four possible index distances for triggering via a digital input (**IN1**, **IN2**, or **IN3**) set to **INxMODE 7**. The other two inputs must be configured to **INxMODE 14** and 15.

Selects an **MIDISTx/MISPEEDx** pair using two digital inputs configured to **INxMODEs 14** and 15. An incremental move (**MI**) is triggered for the selected distance and speed by configuring the third digital input to **INxMODE=7** and transitioning that input from low to high. When the **INxMODE** input goes high, the move is initiated.

Position is in counts, where 65536 counts represent one motor revolution for resolver systems, and 4\*MENCRES counts represents on motor revolution for encoder systems. See **MISPEED0**.

**Firmware Versions:** 3.1.0 and later

**Range:** -2,147,483,647 to +2,147,483,647

**Opmodes:** 8

**Units:** counts

**Type:** variable (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MIDIST1

**MIDIST1**, with **MIDIST0**, **MIDIST2**, and **MIDIST3**, set four possible index distances for triggering via a digital input (**IN1**, **IN2**, or **IN3**) set to **INxMODE** 7. The other two inputs must be configured to **INxMODE** 14 and 15.

Select an **MIDISTx/MISPEEDx** pair using two digital inputs configured to **INxMODEs** 14 and 15. An incremental move (**MI**) is triggered for the selected distance and speed by configuring the third digital input to **INxMODE=7** and transitioning that input from low to high. When the **INxMODE** input goes high, the move is initiated.

Position is in counts, where 65536 counts represent one motor revolution for resolver systems, and 4\*MENCRES counts represents one motor revolution for encoder systems. See **MISPEED1**.

**Firmware Versions:** 3.1.0 and later

**Range:** -2,147,483,647 to +2,147,483,647

**Opmodes:** 8

**Units:** counts

**Type:** variable (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MIDIST2

**MIDIST2**, with **MIDIST0**, **MIDIST1**, and **MIDIST3**, set four possible index distances for triggering via a digital input (**IN1**, **IN2**, or **IN3**) set to **INxMODE** 7. The other two inputs must be configured to **INxMODE** 14 and 15.

Select an **MIDISTx/MISPEEDx** pair using two digital inputs configured to **INxMODEs** 14 and 15. An incremental move (**MI**) is triggered for the selected distance and speed by configuring the third digital input to **INxMODE=7** and transitioning that input from low to high. When the **INxMODE** input goes high, the move is initiated.

Position is in counts, where 65536 counts represent one motor revolution for resolver systems, and 4\*MENCRES counts represents one motor revolution for encoder systems. See **MISPEED2**.

**Firmware Versions:** 3.1.0 and later

**Range:** -2,147,483,647 to +2,147,483,647

**Opmodes:** 8

**Units:** counts

**Type:** variable (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MIDIST3

**MIDIST3**, with **MIDIST0**, **MIDIST1**, and **MIDIST2**, set four possible index distances for triggering via a digital input (**IN1**, **IN2**, or **IN3**) set to **INxMODE** 7. The other two inputs must be configured to **INxMODE** 14 and 15.

Select an **MIDISTx/MISPEEDx** pair using two digital inputs configured to **INxMODEs** 14 and 15. An incremental move (**MI**) is triggered for the selected distance and speed by configuring the third digital input to **INxMODE=7** and transitioning that input from low to high. When the **INxMODE** input goes high, the move is initiated.

Position is in counts, where 65536 counts represent one motor revolution for resolver systems, and 4\*MENCRES counts represents one motor revolution for encoder systems. See **MISPEED3**

**Firmware Versions:** 3.1.0 and later

**Type:** variable (R/W)

**Range:** -2,147,483,647 to +2,147,483,647

**Default:** 0

**Opmodes:** 8

**Drive Status:** EN/DIS

**Units:** counts

**EEPROM:** Yes

## MIPEAK

Sets the motor's peak rated current. When this variable changes, the drive enters a no-comp state, requiring a **CONFIG**.

**Firmware Versions:** All

**Type:** variable (R/W)

**Range:** 3 to 3500

**Default:** motor data

**Opmodes:** All

**Drive Status:** DIS

**Units:** amp<sub>RMS</sub>\*0.1

**EEPROM:** Yes

## MISPEED0

**MISPEED0**, with **MISPEED1**, **MISPEED2**, and **MISPEED3**, set four possible index speeds for triggering via a digital input (**IN1**, **IN2**, or **IN3**) set to **INxMODE** 7. The other two inputs must be configured to **INxMODE** 14 and 15.

Opmode 8 allows you to select an **MIDIST0** / **MISPEED0** pair using two digital inputs that are configured to **INxMODEs** 14 and 15. An incremental move (**MI**) is then triggered for the selected distance and speed by configuring the third digital input to **INxMODE=7** and setting that input high. When the **INxMODE** input goes high, the move is initiated.



### NOTE

*In firmware versions 3.3.0 and later, when homing is performed (MH, HOMESPD, HOMESTATE, and HOMETYPE), MISPEED0 sets the velocity at which the drive returns to home after finding home.*

Opmode 1 allows you to select a **MISPEED0** velocity using two digital inputs configured to **INxMODEs** 14 and 15. A **JOG** is then triggered for the selected speed by configuring the third digital input to **INxMODE=7** and setting that input high. When the **INxMODE** input is high, **JOG** is issued, and when the **INxMODE** input is low, the **JOG** is not performed.



### NOTE

*PROFMODE affects JOG in this scenario. If you select MISPEED0 and set MISPEED0=0, an analog input JOG can be given, as is normal for OPMODE 1 operation.*

**Firmware Versions:** 3.1.0 and later

**Type:** variable (R/W)

**Range:** -VLIM to +VLIM

**Default:** 100

**Opmodes:** 1,8

**Drive Status:** EN/DIS

**Units:** rotary: rpm

**EEPROM:** Yes

linear: mm/sec

## MISPEED1

**MISPEED1**, with **MISPEED0**, **MISPEED2**, and **MISPEED3**, set four possible index speeds for triggering via a digital input (**IN1**, **IN2**, or **IN3**) set to **INxMODE** 7. The other two inputs must be configured to **INxMODE** 14 and 15.

Opmode 8 allows you to select an **MIDIST0 / MISPEED0** pair using two digital inputs that are configured to **INxMODEs** 14 and 15. An incremental move (**MI**) is then triggered for the selected distance and speed by configuring the third digital input to **INxMODE=7** and setting that input high. When the **INxMODE** input goes high, the move is initiated.



### NOTE

*In firmware versions 3.3.0 and later, when homing is performed (MH, HOMESPD, HOMESTATE, and HOMETYPE), MISPEED0 sets the velocity at which the drive returns to home after finding home.*

Opmode 1 allows you to select a **MISPEED0** velocity using two digital inputs configured to **INxMODEs** 14 and 15. A **JOG** is then triggered for the selected speed by configuring the third digital input to **INxMODE=7** and setting that input high. When the **INxMODE** input is high, **JOG** is issued, and when the **INxMODE** input is low, the **JOG** is not performed.



### NOTE

*PROFMODE affects JOG in this scenario. If you select MISPEED0 and set MISPEED0=0, an analog input JOG can be given, as is normal for OPMODE 1 operation.*

**Firmware Versions:** 3.1.0 and later  
**Range:** -VLIM to +VLIM  
**Opmodes:** 1,8  
**Units:** rotary: rpm  
 linear: mm/sec

**Type:** variable (R/W)  
**Default:** 100  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## MISPEED2

**MISPEED2**, with **MISPEED0**, **MISPEED1**, and **MISPEED3**, set four possible index speeds for triggering via a digital input (**IN1**, **IN2**, or **IN3**) set to **INxMODE** 7. The other two inputs must be configured to **INxMODE** 14 and 15.

Opmode 8 allows you to select an **MIDIST0 / MISPEED0** pair using two digital inputs that are configured to **INxMODEs** 14 and 15. An incremental move (**MI**) is then triggered for the selected distance and speed by configuring the third digital input to **INxMODE=7** and setting that input high. When the **INxMODE** input goes high, the move is initiated.



### NOTE

*In firmware versions 3.3.0 and later, when homing is performed (MH, HOMESPD, HOMESTATE, and HOMETYPE), MISPEED0 sets the velocity at which the drive returns to home after finding home.*

Opmode 1 allows you to select a **MISPEED0** velocity using two digital inputs configured to **INxMODEs** 14 and 15. A **JOG** is then triggered for the selected speed by configuring the third digital input to **INxMODE=7** and setting that input high. When the **INxMODE** input is high, **JOG** is issued, and when the **INxMODE** input is low, the **JOG** is not performed.

**NOTE**

*PROFMODE affects JOG in this scenario. If you select MISPEED0 and set MISPEED0=0, an analog input JOG can be given, as is normal for OPMODE 1 operation.*

**Firmware Versions:** 3.1.0 and later

**Range:** -VLIM to +VLIM

**Opmodes:** 1,8

**Units:** rotary: rpm

linear: mm/sec

**Type:** variable (R/W)

**Default:** 100

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MISPEED3

**MISPEED3**, with **MISPEED0**, **MISPEED1**, and **MISPEED2**, set four possible index speeds for triggering via a digital input (**IN1**, **IN2**, or **IN3**) set to **INxMODE** 7. The other two inputs must be configured to **INxMODE** 14 and 15.

Opmode 8 allows you to select an **MIDIST0 / MISPEED0** pair using two digital inputs that are configured to **INxMODEs** 14 and 15. An incremental move (**MI**) is then triggered for the selected distance and speed by configuring the third digital input to **INxMODE=7** and setting that input high. When the **INxMODE** input goes high, the move is initiated.

**NOTE**

*In firmware versions 3.3.0 and later, when homing is performed (MH, HOMESPD, HOMESTATE, and HOMETYPE), MISPEED0 sets the velocity at which the drive returns to home after finding home.*

Opmode 1 allows you to select a **MISPEED0** velocity using two digital inputs configured to **INxMODEs** 14 and 15. A **JOG** is then triggered for the selected speed by configuring the third digital input to **INxMODE=7** and setting that input high. When the **INxMODE** input is high, **JOG** is issued, and when the **INxMODE** input is low, the **JOG** is not performed.

**NOTE**

*PROFMODE affects JOG in this scenario. If you select MISPEED0 and set MISPEED0=0, an analog input JOG can be given, as is normal for OPMODE 1 operation.*

**Firmware Versions:** 3.1.0 and later

**Range:** -VLIM to +VLIM

**Opmodes:** 1,8

**Units:** rotary: rpm

linear: mm/sec

**Type:** variable (R/W)

**Default:** 100

**Drive Status:** EN/DIS

**EEPROM:** Yes



## MJ

Sets the motor's rotor inertia (rotary motors) or motor coil mass (linear motors, MOTORTYPE=2). The Motor Rotor Inertia (**MJ**) and the Load Moment of Inertia Ratio (**LMJR**) define the total system moment of inertia. **LMJR**, **MJ** and the required closed loop bandwidth (**BW**) are used for the Velocity Control Loop design in the Standard Pole-Placement controller (COMPmode = 2 or 4).

**Firmware Versions:** All

**Range:** 1 to 2,000,000,000

**Opmodes:** All

**Units:** rotary: Kg \*m<sup>2</sup> \*10<sup>-6</sup>  
linear: grams

**Type:** variable (R/W)

**Default:** motor data

**Drive Status:** DIS

**EEPROM:** Yes

## MLGAINC

Sets the current loop adaptive gain value at continuous motor current (**MICONT**). **MLGAINC**, **MLGAINP**, and **MLGAINZ** define the adaptive gain algorithm that is based on motor current.

The current-based adaptive gain algorithm is a gain calculation method that increases current loop stability by reducing the current loop gain as the motor current increases. The current-based adaptive gain algorithm is set up by defining the gains at peak motor current (**MLGAINP**), at continuous motor current (**MLGAINC**), and at zero motor current (**MLGAINZ**). All other gains between zero, continuous, and peak current are interpolated linearly.

**MLGAINC** is typically set to the midpoint of **MLGAINZ** and **MLGAINP**. When this variable changes, the drive enters a no-comp state, requiring a **CONFIG**.

**Firmware Versions:** All

**Range:** 1 to 100

**Opmodes:** All

**Units:** % \*10

**Type:** variable (R/W)

**Default:** 8

**Drive Status:** DIS

**EEPROM:** Yes

## MLGAINP

Sets the current loop adaptive gain value at peak motor current (**MIPEAK**). **MLGAINC**, **MLGAINP**, and **MLGAINZ** define the adaptive gain algorithm that is based on motor current.

The current-based adaptive gain algorithm is a gain calculation method that increases current loop stability by reducing the current loop gain as the motor current increases. The current-based adaptive gain algorithm is set up by defining the gains at peak motor current (**MLGAINP**), at continuous motor current (**MLGAINC**), and at zero motor current (**MLGAINZ**). All other gains between zero, continuous, and peak current are interpolated linearly.

**MLGAINP** is typically set to 4 for motors that do not have a lot of iron in their construction and peak currents within the boundaries of the drive. If the motor is rated for much more than what the drive can deliver or if there is a lot of iron in the motor, saturation has less of an effect, and there may be an opportunity to increase this variable. The range for this variable is typically 4 to 7. When this variable is changed, the drive enters a no-comp state, requiring a **CONFIG**.

**Firmware Versions:** All

**Range:** 1 to 100

**Opmodes:** All

**Units:** % \* 10

**Type:** variable (R/W)

**Default:** 4

**Drive Status:** DIS

**EEPROM:** Yes

## MLGAINZ

Set **MLGAINZ** to 10 as it performs no function.

**Firmware Versions:** All

**Range:** 1 10

**Opmodes:** All

**Units:** % \* 10

**Type:** variable (R/W)

**Default:** 10

**Drive Status:** DIS

**EEPROM:** Yes

## MLIST

Dumps all motor variables and their values to the serial port.

**Firmware Versions:** All

**Opmodes:** All

**Command Syntax:** MLIST

**Drive Status:** EN/DIS

## MLMIN

Sets the motor's minimum line-to-line inductance. This variable is used for current loop controller design and as an input to the Torque Angle Control algorithms. When this variable is changed, the drive enters a no-comp state, requiring a **CONFIG**.

**Firmware Versions:** All

**Range:** 1 to 32767

**Opmodes:** All

**Units:** millihenries \* 10<sup>-2</sup>

**Type:** variable (R/W)

**Default:** motor data

**Drive Status:** DIS

**EEPROM:** Yes

## MODMODE

Causes the rotary mode to be enabled. See the *Rotary Mode Application Note* for further details.

**0** = disabled

**1** = enabled

**Firmware Versions:** 3.6.4 and higher

**Range:** 0, 1

**Opmodes:** 8

**Units:** N/A

**Type:** switch (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MOTOR

**MOTOR** is the name of the motor connected to the drive. The motor string variable **MUST BE PRECEDED BY DOUBLE QUOTES (“)** when entered.

**Firmware Versions:** All

**Range:** 10 characters

**Opmodes:** All

**Units:** N/A

**Type:** string variable (R/W)

**Default:** motor data

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MOTORTYPE

Sets the drive's communication and user unit settings for different motor types as follows:

MOTORTYPE=0: permanent magnet rotary motor with legacy resolver and encoder alignment

MOTORTYPE=1: reserved; do not use

MOTORTYPE=2: permanent magnet linear motor which converts units to linear

MOTORTYPE=3: permanent magnet rotary motor with resolver and encoder alignment for all AKM motors (version 7.0.0 and higher and **MOTIONLINK** 3.4.0 and higher)

**Firmware Versions:** 2.1.0 and later

**Range:** 0 to 3

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0 automatically set as part of comp file loading

**Drive Status:** DIS

**EEPROM:** Yes

## MPHASE

Defines the resolver or encoder phase relative to the *standard* commutation table. This variable is used to compensate for resolver offset and should be set to 0, if there is no resolver offset.



**NOTE**

*Changing MPHASE does not change the value of PRD or HWPOS, nor does it create a physical change in the position of the motor shaft - it merely shifts the internal commutation table.*

**Firmware Versions:** All

**Range:** 0 to 359

**Opmodes:** All

**Units:** Feedback device electrical degrees

**Type:** variable (R/W)

**Default:** motor data

**Drive Status:** DIS

**EEPROM:** Yes

## MPITCH

**MPITCH** is for use with linear motors (MOTORTYPE = 2). It defines the pole-pitch (length in millimeters of one electrical cycle - 360 electrical degrees) of the motor and allows the drive to calculate other variables (such as velocity). The drive assumes a no-comp state after an entry of this parameter and requires a **CONFIG**.

**Firmware Versions:** 2.1.0 and later

**Range:** 1 to 500

**Opmodes:** All

**Units:** mm per 360 electrical degrees

**Type:** variable (R/W)

**Default:** 16

**Drive Status:** DIS

**EEPROM:** Yes

## MPOLES

Sets the number of motor poles. This variable is used for commutation control and represents the number of individual magnetic poles of the motor (not pole pairs). When this variable is changed, the drive enters a no-comp state, requiring a **CONFIG**. When MOTORTYPE=2, this variable is forced to a value of 2.

**Firmware Versions:** All

**Range:** 2, 4, 6, 8, .... , 78,80

**Opmodes:** All

**Units:** poles

**Type:** switch mode (R/W)

**Default:** motor data

**Drive Status:** DIS

**EEPROM:** Yes

## MRESPOLES

Sets the number of individual poles in the feedback device. This variable is used for the commutation function, as well as for velocity feedback scaling and represents the number of individual poles, not pole pairs. When this variable is changed on a resolver system, the drive enters a no-comp state, requiring a **CONFIG**.

**Firmware Versions:** All

**Range:** 2, 4, 6, 8, 12, 14, 16, .... 64 (resolver-based)

0, 2, 4, 6, 8, 12, 14, 16, .... 64 (encoder-based)

**Default:** motor data

**Units:** poles

**Drive Status:** DIS

**Type:** switch mode (R/W)

**Opmodes:** All

**EEPROM:** Yes

## MSG

Enables and disables the sending of error messages from the drive to the serial port.

**0** = disable messages

**1** = enable messages



**MSG = 1 is needed for proper operation of MOTIONLINK.**

### NOTE

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 1

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MSININT

**MSININT** is used with the sine encoder option and sets the interpolation level of the CB drive used with sine encoders. Actual resolution is 4x this number. For example, **MSININT** 256 with **MENCRES** 2048 results in a system resolution of 21 bits or 2,097,152 counts per revolution.

**Firmware Versions:** 3.2.0

(3.3.0 or higher variable is automatically set to 256)

**Range:** 1, 2, 4, 8, 16, 32, 64, 128, 256

(Endat encoders in 3.4 firmware)

**CD Series 5 allows 512 and 1024**

256 only (for all other sine encoders)

**Type:** switch mode (R/W)

**Units:** bits

**Drive Status:** DIS

**Opmodes:** All

**Default:** 256

**EEPROM:** Yes

## MSINFRQ

**MSINFRQ** is used with the sine encoder option and sets the maximum frequency limit of the encoder equivalent output. If the encoder equivalent output exceeds the value set by this variable, it becomes inaccurate and a burst overflow fault is generated.

**0** = ignore burst overflow fault

**1** = 2.5 MHz

**2** = 1.25 MHz

**3** = 625 kHz

**4** = 312 kHz

**Firmware Versions:** 3.3.0 and higher

**Range:** 0 to 4

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 1

**Drive Status:** DIS

**EEPROM:** Yes

## MSPEED

Defines the maximum recommended velocity of the motor. When this variable is changed the drive enters a no-comp state requiring a **CONFIG**.

**Firmware Versions:** All

**Range:** 10 to 32767

**Opmodes:** All

**Units:** rotary: rpm

linear: mm/sec

**Type:** variable (R/W)

**Default:** motor data

**Drive Status:** DIS

**EEPROM:** Yes

## MTANGLC

Sets the value of the torque-related commutation angle advance at the motor's continuous current rating (**MICONT**). This variable helps increase reluctance torque. For surface magnet motors, a typical value is 5. For motors with embedded magnets, a typical value is 8 to 10.

**Firmware Versions:** All

**Range:** 0 to 45

**Opmodes:** All

**Units:** electrical degrees

**Type:** variable (R/W)

**Default:** 10

**Drive Status:** EN/DIS

**EEPROM:** Yes

## MTANGLP

Sets the value of the torque-related commutation angle advance at the motor's peak current (**MIPEAK**). This variable helps increase reluctance torque. For surface magnet motors, a typical value is 10. For motors with embedded magnets, a typical value is 23 to 25.

**Firmware Versions:** All  
**Range:** 0 to 45  
**Opmodes:** All  
**Units:** electrical degrees

**Type:** variable (R/W)  
**Default:** 23  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## MVANGLF

Sets the value of the velocity-related commutation angle advance to be used when the motor is operating at motor max speed (**MSPEED**). Between **MSPEED/2** rpm and **MSPEED**, the angle advance is linearly interpolated based on **MVANGLH** and **MVANGLF**.

When a **CLREEPROM** is issued, **MVANGLF** is set to a value of 10. If a **CONFIG** is then issued, **MVANGLF** is set to a default value based on **MSPEED** and **MPOLES**. Once you enter a value for **MVANGLF**, it keeps that value and is not changed if a **CONFIG** is executed.

**Firmware Versions:** All  
**Range:** 0 to 90  
**Opmodes:** All  
**Units:** electrical degrees

**Type:** variable (R/W)  
**Default:** calculated for motor data  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## MVANGLH

Sets the value of the velocity-related commutation angle advance to be used when the motor is operating at half of the motor max speed (**MSPEED**). Between 0 rpm and half of **MSPEED**, the angle advance is linearly interpolated based on **MVANGLH**.

When a **CLREEPROM** is issued, **MVANGLH** is set to a value of 5. If a **CONFIG** is then issued, **MVANGLH** is set to a default value based on **MSPEED** and **MPOLES**. Once you enter a value for **MVANGLH**, it keeps that value and is not changed if a **CONFIG** is executed.

**Firmware Versions:** All  
**Range:** 0 to 90  
**Opmodes:** All  
**Units:** electrical degrees

**Type:** variable (R/W)  
**Default:** calculated for motor data  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## NOTCHBW

Sets the bandwidth of the notch filter used in the velocity loop. Affects the system only when **FILTMODE=3**.

**Firmware Versions:** 3.1.0 and later  
**Range:** 1 to 100  
**Opmodes:** All  
**Units:** Hz

**Type:** variable (R/W)  
**Default:** 1  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

# NOTCHHZ

Sets the center frequency of the notch filter used in the velocity loop. Affects the system only when **FILTMODE**=3.

**Firmware Versions:** 3.1.0 and later  
**Range:** 30 to 1000  
**Opmodes:** All  
**Units:** Hz

**Type:** variable (R/W)  
**Default:** 500  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## 01

**01** is used to read or write the state of the hardware output on user connector C3 Pin 12. See **O1MODE**.

**0** = OFF  
**1** = ON



### NOTE

*Writing **01** in certain **O1MODE** conditions does not stop the drive from resetting the output according to drive conditions.*

**Firmware Versions:** 2.1.0 and later  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R/W)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** N/A

## O1MODE UCB1 CD SERIES 2

A switch mode variable used to define the function of **O1**:

- 0** - Disabled
- 1** - **O1** goes on when absolute motor speed is above **O1TRIG** speed.
- 2** - **O1** goes on when the absolute actual output current is above **O1TRIG** current.
- 3** - **O1** goes on when drive is in **FOLDBACK**.
- 4** - **O1** goes on when absolute motor speed is above **O1TRIG** but less than **O1RST**.
- 5** - Brake Mode: **O1** is OFF only when (1) drive is disabled or (2) during an active disable sequence, when actual motor speed has dropped below **DISSPEED**, but **DISTIME** timer has not timed out yet. **O1** is ON when the drive is enabled.
- 6** - Motion Complete Output: **O1** is tied to **STOPPED** switch.
- 7** - In Position Output: **O1** is tied to **INPOS**.
- 8** - Zero Speed Detect: **O1** on if absolute motor speed <**O1TRIG**.
- 9** - Programmable Limit Switch: **O1** goes on if **O1TRIG**<**PFB**<**O1RST** (firmware versions 3.2.0 and later). If **O1MODE**=9, the range of **O1TRIG** and **O1RST** is **0** to **2147483647**. Trying to write a negative value to **O1TRIG** or **O1RST** results in an error (out of range) message. When **O1MODE**=9, the output is active if **O1TRIG**< abs(Motor Position Feedback)<**O1RST**. The absolute value of **PFB** is used.

- 10** - Active: **O1** goes on if drive is active (firmware versions 3.1.0 and later).
- 11** - **O1** is 1 after initialization is completed or when the commutation is aligned to the motor. In resolver based systems the output is always 1 if there is no feedback loss fault.
- 12** - The output is active when the SERCOS ring is broken and the drive does not receive data. The output is inactive when the SERCOS telegrams are received by the drive (firmware version 4.1.4 and later).



***O1 at 0 indicates if encoder initialization is incomplete or feedback loss is detected.***

#### NOTE

**Firmware Versions:** 2.1.0 and later  
**Range:** 0 to 12  
**Opmodes:** All  
**Units:** Hz

**Type:** switch mode (R/W)  
**Default:** 6  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## O1MODE UCB2 CD SERIES 5

Firmware version 6.6.0 25/Sep/03:

- O1MODE=0** – **O1** is controlled directly by **O1**.
- O1MODE=1** – **O1** is on when Motor Speed > **O1TRIG**.
- O1MODE=2** – **O1** is on when the equivalent current ('I') is greater than **O1TRIG**.
- O1MODE=3** – **O1** is on during foldback state.
- O1MODE=4** – **O1** is on when **O1TRIG** < Motor Speed < **O1RST**.
- O1MODE=5** – brakes control.
- O1MODE=6** – **O1** is on when motion completed (**STOPPED** bit is set).
- O1MODE=7** – **O1** is on when IN POSITION bit is set.
- O1MODE=8** – **O1** is on when Motor Speed < **O1TRIG**.
- O1MODE=9** – **O1** is on when **O1TRIG** < abs(Motor Position Feedback) < **O1RST**.
- O1MODE=10** – **O1** is on when the drive is enabled (**ACTIVE=1**).
- O1MODE=11** – **O1** goes high when Encoder init was done and low again in the beginning of an Encoder init process.
- O1MODE=12** – **O1** goes high when SERCOS break is detected and low again when SERCOS communication functions well.
- O1MODE=13** – **O1** is on when PFB burst is transmitted via the encoder simulation output.



## O1RST

A variable used to define the reset level for **O1MODE**. Range is dependent on **O1MODE**:

- 0 - N/A
- 1 - N/A
- 2 - N/A
- 3 - N/A
- 4 - Absolute: 0 - 15000 rpm (0 - 250\*MPITCH mm/sec for linear)
- 5 - Absolute: 0 - 15000 rpm (0 - 250\*MPITCH mm/sec for linear)
- 6 - N/A
- 7 - N/A
- 8 - N/A
- 9 - Refer to O1MODE=9
- 10 - N/A

**Firmware Versions:** 2.1.0 and later  
**Range:** 0 to 10 (see above)  
**Opmodes:** All  
**Units:** rpm or mm/sec

**Type:** variable (R/W)  
**Default:** VOSPD  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## O1TRIG

A variable used to define the trip level for **O1MODE**. Range is dependent on **O1MODE**:

- 0 - N/A
- 1 - Absolute: 0 to 15000 rpm (mm/sec for linear motors)
- 2 - Absolute: 0 to 1000 (0.1 percent of DIPEAK)
- 3 - N/A
- 4 - Absolute: 0 to 15000 rpm (mm/sec for linear motors)
- 5 - N/A
- 6 - N/A
- 7 - N/A
- 8 - Absolute: 0 to 15000 rpm (mm/sec for linear motors)
- 9 - Refer to O1MODE=9
- 10 - N/A

**Firmware Versions:** 2.1.0 and later  
**Range:** 0 to 10 (see above)  
**Opmodes:** All  
**Units:** see above

**Type:** variable (R/W)  
**Default:** 1000  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## OPMODE

Sets the operational mode for the drive. The drive can be configured as a velocity, torque loop, pulse following (gearing), or position controller.

Opmode	Description	See Also
0	Serial Velocity	J, COMPMODE, PROFMODE, S, STOP
1	Analog Velocity	VSCALE, COMPMODE, S, PROFMODE
2	Serial Torque	S, STOP, T
3	Analog Torque	ISCALE, S
4	Gearing	GEAR, GEARI, PEXT, GEARMODE, XENCDIR, XENCRES
8	Positioning	MA, MH, MI, PCMDMODE, PROFMODE, S

When you change **OPMODE** via serial communication, the drive must be disabled. Digital input can be used to switch between **OPMODEs**; with the drive enabled.



*It is your responsibility to switch OPMODEs at the proper time (zero velocity for example) for your application.*

### NOTE

The possible **OPMODE** switching by digital inputs are:

OPMODE 1 to OPMODE 3 - use INxMODE 9

OPMODE 1 to OPMODE 4 - use INxMODE 20

OPMODE 4 to OPMODE 8 - use INxMODE 24



*Opmode 4 available only in firmware versions 2.1.0 and later.*

*Opmode 8 available only in firmware versions 3.1.0 and later.*

### NOTE

**Firmware Versions:** All (see table above)

**Range:** 0, 1, 2, 3, 4, 8

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 1

**Drive Status:** DIS

**EEPROM:** Yes

## PCMD

Returns the position command as output by the profile generator. **PCMD** is expressed in counts.

**Firmware Versions:** 3.1.0 and later

**Range:** -2,147,483,647 to +2,147,483,647

**Opmodes:** 4, 8

**Units:** counts

**Type:** variable (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** N/A

## PCMDMODE

A switch mode variable which can change the flow of data in the position loop according to the following arguments:

**0** = Normal operation: Command comes from profile generator, feedback comes from motor. **PCMDMODE** must be set to 0 to generate move commands via the serial port.

**1** = Analog Position Mode: Absolute position command comes from the analog input port, feedback from the motor.

**Firmware Versions:** 3.2.0 and later  
**Range:** 0, 1  
**Opmodes:** 8  
**Units:** N/A

**Type:** switch (R)  
**Default:** 0  
**Drive Status:** DIS  
**EEPROM:** Yes

## PE

Displays the position following error. If this value is greater than **PEMAX**, the drive is disabled. Position is in counts.

**Firmware Versions:** 2.1.0 and later  
**Range:** -2,147,483,647 to +2,147,483,647  
**Opmodes:** 4, 8  
**Units:** N/A

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** N/A

## PEINPOS

Sets the threshold position error for the **INPOS** flag. If **PE** is less than **PEINPOS**, the **INPOS** switch is set, indicating the drive is in position. If **PE > PEINPOS**, the **INPOS** switch is not set. Position is in counts.

**Firmware Versions:** 2.1.0 and later  
**Range:** 0 to 32767  
**Opmodes:** 4, 8  
**Units:** counts

**Type:** variable (R/W)  
**Default:** 100  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## PEMAX

Sets the maximum allowable following error (**OPMODEs** 4 and 8). If the error exceeds this value, the drive is disabled on fault. **PEMAX = 0** disables this function. Position is in counts.

**Firmware Versions:** 2.1.0 and later  
**Range:** 0 to 2,147,483,647  
**Opmodes:** 4, 8  
**Units:** counts

**Type:** variable (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## PEXT

Displays the accumulated position feedback from the external encoder. This variable is similar to **PFB** for the resolver feedback.

**Firmware Versions:** 2.1.0 and later  
**Range:** -2,147,483,647 to +2,147,483,647  
**Opmodes:** All  
**Units:** counts

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** N/A

## PEXTHW

Defines hand wheel connector signals.

- 0 - Differential
- 1 - TTL

## PEXTOFF

An offset that is added to the internal accumulated position feedback from the external encoder to give the value of **PEXT**.

**Firmware Versions:** 2.1.0 and later

**Range:** -2,147,483,647 to +2,147,483,647

**Opmodes:** All

**Units:** counts

**Type:** variable (R/W)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## PFB

Displays the cumulative position feedback from the feedback device.

Prior to firmware version 2.0.0, **PFB** had the same definition as

**HWPOS**. For firmware versions 2.0.0 and later, **PFB** was extended into a cumulative counter.

**Firmware Versions:** All

**Range:** -2,147,483,647 to +2,147,483,647

**Opmodes:** All

**Units:** counts

**Type:** variable (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## PFBOFF

A feedback offset that is added to the internal cumulative position counter to give the value of **PFB**. This offset can be used to offset absolute machine zero.

**Firmware Versions:** 2.0.0 and later

**Range:** -2,147,483,647 to +2,147,483,647

**Opmodes:** All

**Units:** counts

**Type:** variable (R/W)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## PLIM

A switch mode variable that controls operation of the software position limits **PMAX** and **PMIN**:

**0** = Software position limits disabled

**1** = Drive disables when a soft position limit is exceeded *Caution!*  
*Motor may coast*

**2** = Drive decelerates to a stop at **DECSTOP** deceleration when a soft position limit is exceeded. Drive remains enabled and only allows motion in opposite direction.

**Firmware Versions:** 3.1.0 and later

**Range:** 0 to 2

**Opmodes:** All

**Units:** N/A

**Type:** switch (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## PMAX

Sets the maximum allowable position for the motor shaft. Position is expressed in counts and is read using **PFB**. If position exceeds **PMAX** (**PFB** > **PMAX**), an overtravel fault is generated. A fault is generated only if **PLIM** is set to a nonzero value.

**Firmware Versions:** 3.1.0 and later

**Range:** -2,000,000,000 to +2,000,000,000

**Opmodes:** All

**Units:** counts

**Type:** variable (R/W)

**Default:** 2,000,000,000

**Drive Status:** EN/DIS

**EEPROM:** Yes

## PMIN

Sets the minimum allowable position for the motor shaft. Position is expressed in counts and is read using **PFB**. If position goes below **PMIN** (**PFB** < **PMIN**), an overtravel fault is generated. A fault is generated only if **PLIM** is set to a nonzero value.

**Firmware Versions:** 3.1.0 and later

**Range:** -2,000,000,000 to +2,000,000,000

**Opmodes:** All

**Units:** counts

**Type:** variable (R/W)

**Default:** 2,000,000,000

**Drive Status:** EN/DIS

**EEPROM:** Yes

## PRD

Displays the absolute position feedback of the hardware feedback device (for both resolver and encoder based systems). **PRD** increments from 0 to 65,535 throughout the course of one mechanical motor shaft revolution (360 degrees). The range of **PRD** does not change. Its resolution for resolver feedback systems is dependent upon the value of **RDRES**:

RDRES = 12, resolution of PRD = 16.

RDRES = 14, resolution of PRD = 4.

RDRES = 16, resolution of PRD = 1.

For encoder-based systems, until the encoder has been initialized, **PRD** is uninitialized and its value is not useful or meaningful. For information on encoder initialization requirements according to the type of encoder, see **MENCTYPE**, **ENCINIT**, AND **ENCINITST**.

**Firmware Versions:** All

**Range:** 0 to 65,535

**Opmodes:** All

**Units:** counts

**Type:** variable (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## PROFMODE

Selects the acceleration and deceleration algorithm used by the drive (profile mode). **PROFMODE** is associated with **ACC** and **DEC** but may not affect ramping, depending on the values of **ACTFAULT**, **STOP**, and **DECSTOP**.

0 = No acceleration and deceleration ramp limits

1 = Linear acceleration and deceleration ramp limits

2 = S-curve accel/decel in Opmode 8 only (firmware versions 3.3.0 and later - see **PROFSCRV**).

**Firmware Versions:** All  
**Range:** 0 to 2  
**Opmodes:** 0,1,4,8  
**Units:** counts

**Type:** switch mode (R/W)  
**Default:** 0  
**Drive Status:** DIS  
**EEPROM:** Yes

## PROFSCRV

Defines the S-curve acceleration time (when PROFMODE=2) relative to the trapezoidal, or linear, acceleration time of PROFMODE=1. Setting PROFMODE=2 and setting the value of **PROFSCRV** to a non-zero value introduces S-curve acceleration, which is a smoothing of the acceleration that occurs when a positional move is initiated. The tradeoff is that peak accelerations and horsepower requirements are higher when using S-curving than when linear profiling (PROFMODE=1) is used.

**0** = S-curve acceleration time is equal to the trapezoidal acceleration time (**ACC** applies to both acceleration and deceleration).

**100** = S-curve acceleration time is equal to twice the trapezoidal acceleration time.

**Firmware Versions:** 3.3.0 and later  
**Range:** 0 to 100  
**Opmodes:** 8  
**Units:** percent

**Type:** variable (R/W)  
**Default:** 50  
**Drive Status:** DIS  
**EEPROM:** Yes

## PROMPT

Enables and disables the serial port prompt (-->) output by the drive after each message.

**0** = disable the prompt

**1** = enable the prompt



**PROMPT = 1 is needed for proper operation of MOTIONLINK.**

### NOTE

**Firmware Versions:** All  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R/W)  
**Default:** 1  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## PROTARY

Defines the modulo values of **PFB** and **PCMD**. See **DIVISIONS**.

**Firmware Versions:** 3.6.4 and later  
**Range:** 1000 to  $2^{30}$   
**Default:**  $2^{21}$  (2048 Sine Encoder Modulo)  
**EEPROM:** Yes

**Type:** variable (0 to Long)  
**Drive Status:** EN/DIS

**Opmodes:** 8

## PSCALE

A position scale factor that scales the analog output, ANOUT=5 or 8, to **PE** or **PFB**. The value entered is the motor position movement in counts per 10 volts of output.

**Firmware Versions:** 3.2.0 and later

**Range:** 10 to 2,147,483,647

**Opmodes:** 1

**Units:** counts per 10 volts

**Type:** variable (R/W)

**Default:** 2048

**Drive Status:** EN/DIS

**EEPROM:** Yes

## RDRES

Displays the resolver resolution on resolver-based systems. **RDRES** is a read-only variable automatically calculated in order to achieve maximum resolution. The setting is based on **VLIM**, which is the maximum application velocity. The relationship between **VLIM** and **RDRES** is given below:

If  $(VLIM \geq 6101)$  then  $RDRES = 12$

If  $(1501 \leq VLIM \leq 6100)$  then  $RDRES = 14$

If  $(VLIM \leq 1500)$  then  $RDRES = 16$

**Firmware Versions:** All

**Range:** 12, 14, or 16

**Opmodes:** All

**Units:** bits

**Type:** switch mode (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## READY

A flag indicating the status of the software enable. **READY=1** means that there are no faults (**DRIVEOK=1**) and a communication enable request has been commanded (**SWEN=1**). An external Remote Enable (**REMOTE=1**) and a Dip Switch Enable (**DIPEN=1**) are still required to enable the drive (**ACTIVE=1**).

**0** = faults exist or **SWEN = 0**

**1** = no faults exist and **SWEN = 1**

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## RECDONE

Indicates whether or not **RECORD** is complete and data is available.

**0** = recording not finished

**1** = recording done; data available

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** No

## RECING

Indicates if data recording is in progress.

**0** = recording not in progress

**1** = recording in progress

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** No

## RECOFF

Used to cancel/reset a recording process that has been armed but has not triggered.

State before **RECOFF**

RECRDY=0

RECING=1

RECDONE=0.

State after **RECOFF**

RECRDY=1

RECING=0

RECDONE=0.

**Firmware Versions:** All

**Opmodes:** All

**Command Syntax:** RECOFF

**Drive Status:** EN/DIS

## RECORD

Captures realtime variables to memory for retrieval or display using **GET** or the **MOTIONLINK** PC Scope function. Set up **RECORD** before using **RECTRIG**.



**Variables that are recorded using this method are NOT averaged using AVGTIME.**

### NOTE

1024 four-word buffers are available for use by **RECORD**, where one word is defined as 16 bits. Most variables are one word in size, but some are two words. You can record up to three variables, as long as the total size of the three variables does not exceed 4 words. The **SERVOSTAR** records:

Variable	(Size)	Variable	(Size)
ANIN*	(1 word)	PCMD*	(2 words)
I	(2 words)	PE*	(2 words)
IA	(1 word)	PEXT*	(2 words)
IC	(1 word)	PFB*	(2 words)
ICMD	(1 word)	PRD	(1 word)
IN1*	(1 word)	STOPPED*	(1 word)
IN2*	(1 word)	V	(1 word)
IN3*	(1 word)	VCMD	(1 word)
INPOS*	(1 word)	VEXT*	(1 word)
O1*	(1 word)		

\* Added in firmware version 3.2.0

For example, a combination of **V**, **VCMD**, and **VEXT** is valid for recording, because it only takes up 3 words of memory. However, a combination of **PCMD**, **PE**, and **PEXT** (6 words total) cannot be recorded.



**RECORD** also defines the time period between each consecutive recorded data point and the variable names (up to three) being recorded. An additional parameter defines the number of recorded data points for each variable (up to 1024). Once they are recorded, variables are retrieved using **GET**. See **RECTRIG**, **RECDONE**, **RECING**, and **RECRDY**.

**NOTE**

*System variables must be preceded by a double-quote (").*

**Command Syntax:** RECORD [*sample time*] [*num points*] [VAR1] {VAR2} {VAR3}

**Firmware Versions:** All

**Opmodes:** All

**Drive Status:** EN/DIS

**Example:** RECORD 10 100 "VCMD "V "PRD

(record 100 points for VCMD, V, and PRD every 5 milliseconds)

**Range/Units:** [*sample time*]: 1 to 10,000(\* 0.5 milliseconds)

[*num points*]: 1 to 1024 (1, 2, 4, 8, ... 512, 1024)

[VARn]: a system variable, by ASCII (text) name

## RECRDY

Indicates the ready status of the RECORD function. This variable can be polled after a RECORD command is issued to determine if the system is waiting for RECTRIG.

**0** = RECTRIG has been received and record function is armed

**1** = record function is waiting to be armed by RECTRIG command

**Firmware Versions:** All

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** 1

**Drive Status:** EN/DIS

**EEPROM:** No

## RECTRIG

Sets up the trigger mechanism for **RECORD**. **RECORD** must be set up before a **RECTRIG** is issued. Four parameters are required to set up **RECTRIG**: Mode, Level, Location, and Direction.

**MODE** is a string variable that specifies the parameter that will trigger recording. Mode can be a variable name or a triggering condition. Mode determines what other parameters must be entered in order to completely set up the trigger. Mode must be preceded by a double-quote when entered, as shown in the following table, which tells what other parameters are required (LEVel, LOCation, and DIRection) depending upon the selected Mode.

**LEVEL** specifies the value that the variable defined by Mode must reach for recording to begin.

**LOCATION** specifies how many data points to save before the trigger in the Recording buffer (see **RECORD** for a description of the 1024 data points that are available). When recorded data is retrieved and displayed, the trigger point's location in the 1024-point buffer is at the place specified by Location.

**DIRECTION** has two meanings, depending upon the type of Mode parameter that is used. For Mode variables (**PRD**, **IA**, **IC**), it defines the direction the variable value must be changing when it crosses Level in order to trigger recording (1 = increasing, 0 = decreasing). For Mode switch inputs (CW, CCW, etc. - see below) it defines the logic level the input must achieve in order to trigger recording (1 = HI, 0 = LOW).

Required RECTRIG Parameters Based on MODE Parameter				
MODE	DESCRIPTION	LEV	LOC	DIR
"IA	Trigger on Phase A Current	√	√	√
"IC	Trigger on Phase C Current	√	√	√
"ICMD	Trigger on Current Cmd	√	√	√
"PCMD*	Trigger on Position Cmd	√	√	√
"PFB*	Trigger on Position Fdbck	√	√	√
"PRD	Trigger on PRD	√	√	√
"V	Trigger on Velocity	√	√	√
"VCMD	Trigger on Velocity Cmd	√	√	√
"CCW	Trigger on CCWLIM Sw	X	√	√
"CW	Trigger on CWLIM Sw	X	√	√
"IN1*	Trigger on IN1 Input	X	√	√
"IN2*	Trigger on IN1 Input	X	√	√
"IN3*	Trigger on IN1 Input	X	√	√
"O1*	Trigger on O1 Output	X	√	√
"RMT	Trigger on REMOTE Input	X	√	√
"CMD	Trigger on Next Command	X	√	X
"IMM	Trigger Immediately	X	X	X

\* = firmware versions 3.2.0 and later

√ = Required Parameter

X = Don't care. Something must be entered to make the command work, but it does not matter what is entered.

**Syntax:** RECTRIG *[mode]* *[level]* *[location]* *[direction]*

**Firmware Versions:**

All

**Example:** RECORD 10 100 "VCMD "V "PRD

(record 100 points for VCMD, V, and PRD every 5 milliseconds)

**Range/Units:** *[mode]*: see table above

*[level]*: depends upon the mode variable (range of PRD levels is 0-65535. All others are -32768 to 32767)

*[location]*: 0 – 1023

*[direction]*: 0 or 1

**Type:** variable (R/W)

**Default:** *[level]*: 0

*[location]*: 0

*[direction]*: 1

**Opmodes:** All

**EEPROM:** No

**Drive Status:** EN/DIS

## REFRESH

A command used when tuning the drive for COMPMODE 3. With the Advanced Pole Placement algorithm utilized in the drive, the interaction of the variables is too dramatic to allow variables to be changed one by one. Therefore, as pole placement algorithm vector variables (**VD**, **VF**, **VH**, and **VR**) are entered, the new values are buffered without changing the actual values used by the control loops.

Once all desired new values have been entered, enter a **REFRESH** and all vector variables are written to the control loops simultaneously.

**Firmware Versions:** All  
**Opmodes:** 0, 1

**Command Syntax:** REFRESH  
**Drive Status:** EN/DIS

## RELAY

Indicates the status of the Fault / Drive Up Relay.

**0** = relay open  
**1** = relay closed

**Firmware Versions:** All  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R)  
**Default:** hardware defined  
**Drive Status:** EN/DIS  
**EEPROM:** No

## RELAYMODE

Sets the operation of the Drive Up / Drive Ready Relay.

**0** = relay is closed when no faults exist  
**1** = relay is closed when **ACTIVE** equals 1  
**2** = during Active Disable, relay opens when the fault occurs (it does not wait until **DISTIME** times out).

**Firmware Versions:** All  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## REMOTE

Indicates the state of the external hardware enable input line. When **REMOTE=1**, the software is ready (**READY=1**), and Dip Switch 8=OFF (**DIPEN=1**), the drive is enabled (**ACTIVE=1**).

**0** = remote enable input off  
**1** = remote enable input on

**Firmware Versions:** All  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R)  
**Default:** hardware defined  
**Drive Status:** EN/DIS  
**EEPROM:** No

## RESBW

Set the cutoff frequency of the filter.

**Syntax:** RESBW <value> to set the cut-off frequency  
RESBW to read the cut-off frequency

**Type:** switch mode (R/W)

**Range:** 300, 600, 1200

**Units:** Hz

**Default:** 600

**Opmodes:** All

**Drive Status:** DIS

**Firmware:** 6.3.3 and higher

**EEPROM:** No

## RSTVAR

Sets all variables, switch variables, and switch mode variables to their factory default settings. These settings are stated within this document under the variable DEFAULT category. The default values of variables loaded from a motor data file cannot be predicted and are denoted motor data in the DEFAULT category.

### Default values:

ACC = 400,000	ACKMODE = 0	ACTFAULT = 0	ANDB = 0
ANOFF = 0	ANOUT = 0	AVGTIME = 0	BW = 20
ANDG = 0	COMPFLT = 1	COMPMODE = 2	DEC = 400,000
DIR = 1	DISSPEED = 50	DISTIME = 100	ECHO = 1
ENCOUT = 1024	ENCOUTO = 1	FILTMODE = 0	FOLDMODE = 0
FOLDTIME = 30	GEAR = 1	GEARI = 1	GEARMODE = 3
GEARO = 1	GETMODE = 0	GPAFR = 0	GPAFR2 = 0
GPD = 0	GPI = 0	GPISATIN = 0	GPISATOUT = 0
GPVFR = 0	GV = 500	GVI = 20	HOMESPD = 100
HOMETYPE = 0	IENCSTART = 25	IFRIC = 0	IGRAV = 0
ILIM = IMAX	ILIM2 = 100	ILSBMODE = 2	IN1MODE = 1
IN2MODE = 2	IN3MODE = 3	ISCALE = 1250	ISTOP = DICONT
IZERO = 20	KV = 1000	KVI = 1000	KVFR = 0
LIMDIS = 0	LMJR = 0	MPFHZ1 = 500	LPFHZ2 = 500
MAPOS = 0	MASPEED = 0	MFBDIR = 0	MFOLDD = 1200
MFOLDDIS = 0	MFOLDR = 1800	MFOLDT = 600	MIDISTx = 0
MISPEEDx = 100	MSININT = 256	MPHASE = 0	MSG = 1
MSINFRQ = 1	NOTCHBW = 1	NOTCHHZ = 500	O1MODE = 6
O1RST = VOSPD	O1TRIG = 1000	OPMODE = 1	PCMDMODE = 0
PEMAX = 0	PLIM = 0	PMAX = 2,000,000,000	PMIN = 2,000,000,000
PROFMODE = 0	PROFSCRV = 50	PROMPT = 1	RELAYMODE = 0
SININTOUT = 1	STOPPED = 0	TF = 100	THERMODE = 0
THERMTIME = 30	THERMTYPE = 0	UNITS = 0	UVMODE = 0
UVTIME = 30	UVRECOVER = 0	VD = 0	VF = 1,000,000
VH = 0	VLIM = VMAX	VOSPEED = 1.2 * VLIM	VR = 0
XENCDIR = 0	XENCRES = 1024		

**Firmware Versions:** All  
**Opmodes:** 0, 1

**Command Syntax:** RSTVAR  
**Drive Status:** DIS

## S

Stops motor motion in all **OPMODES**. Deceleration ramp control is always used, using the rate specified by **DECSTOP**. After the profile generator reaches 0 speed, the drive waits for the time period specified by **DISTIME** and disables the drive.

This command is a one-key hot button, similar to **K**, but with an active stop function controlled by the drive (no coasting of the motor occurs, as is possible with **K**).

**Firmware Versions:** All  
**Opmodes:** 0, 1

**Command Syntax:** S  
**Drive Status:** EN/DIS

## SAVE

Copies all system configuration variables from working RAM to non-volatile memory (EEPROM). This command must be executed in order to retain setting changes during power cycling. The **SAVE** command takes about 2 seconds to execute.

**Firmware Versions:** All  
**Opmodes:** 0, 1

**Command Syntax:** SAVE  
**Drive Status:** EN/DIS

## SERIALNO

Indicates the serial number of the drive in which the firmware is installed. This variable is password protected. This variable is included in **VER**.

**Firmware Versions:** All  
**Range:** 10 ASCII characters  
**Opmodes:** All  
**Units:** N/A

**Type:** string variable (R)  
**Default:** blanks  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## SININIT

Initialize or reset the sine calibration process.

**Syntax:** SININIT  
**Range:** N/A  
**Opmodes:** All  
**Firmware:** 6.2.0 and higher

**Type:** Command  
**Default:** 0  
**Drive Status:** DIS  
**EEPROM:** No

## SININTOUT

Sets an interpolation factor of the sine encoder board for the equivalent encoder output. For sine encoder systems, the encoder output value is  $(ENCOUT) = MENCRES * SININTOUT / ENCOUTO$ . See **MSINFRQ**.

**Firmware Versions:** 3.3.0 and later  
**Range:** 1, 2, 4, 8, 16, 32, 64, 128  
**Opmodes:** All  
**Units:** N/A

**Type:** switch mode (R/W)  
**Default:** 1  
**Drive Status:** DIS  
**EEPROM:** Yes

## SININITST

Queries the status of the sine calibration process. The following values may be returned by the query:

- 0 – No request
- 1 – Process running
- 2 – Motor velocity too high. The sine frequency must not exceed 250 Hz. When speed is reduced, the status returns to 1 (running).
- 3 – Process completed
- 4 – Defaults loaded. Defaults are loaded when the calibration process has never been performed, such as in a new drive.
- 5 – Internal failure

**Type:** Read-only

**Opmodes:** All

**Firmware:** 6.2.0 and higher

**Range:** 0 .. 5

**Drive Status:** EN/DIS

**EEPROM:** No

## SINPARAM

Queries or sets the sine calibration values. There are four calibration parameters when working with a sine encoder, and six calibration parameters when working with a resolver. All are signed hexadecimal values.

### Sine offset and cosine offset:

Multiply by (10000/32768) to get value in millivolts. This offset is at the ADC terminal (after the sine and cosine signals were amplified).

### Sin to cos match gain and scale:

The algorithm requires that the sine and cosine signals should have the same amplitude. This value equals  $(\text{gain} / 2^{\text{scale}})$ , and represents the amplitude difference of the sine and cosine signals. It should be close to 1. The firmware multiplies the sine signal samples by this value to get the same amplitude for the sine and the cosine signals.

### Full-scale gain and scale (relevant for Resolver ONLY):

The algorithm requires that the sine and cosine signals should be scaled to 32768. The final value equals  $(\text{gain} / 2^{\text{scale}})$ , and it represents the factor to multiply the sine and cosine signals. It should be in the range 1.2 to 1.3. The firmware multiplies the sine and cosine signals samples by this value.

**Syntax for set:** SINPARAM<Sine offset><Cosine offset><Sine to Cosine match gain> <Sine to Cosine match scale> <full-scale gain> <full-scale scale>



**NOTE**

*Last 2 parameters relevant for resolver feedback only.*

**Syntax for query:** SINPARAM

**Range: Sine offset:** -32768 .. 32767

**Cosine offset:** -32768 .. 32767

**Sine to Cosine match gain:** 1 .. 32767

**Sine to Cosine match scale:** 1 .. 15

**full-scale gain:** 1 .. 32767

**full-scale scale:** 1 .. 15

**Default:** 0h 0h 4000h Eh 4000h Eh (*last 2 are for resolver only*)

**Type:** variable (R)

**Units:** Internal

**Opmodes:** All

**Drive Status:** DIS

**Firmware:** 6.3.3 and higher

**EEPROM:** Yes

**Limitations:** Not yet supported by MOTIONLINK

## STAT

Outputs a drive status summary word to the serial port. The summary word is in ASCII-hex format, prefixed by the letter *H*. See **STATUS** for information on how to obtain more detailed drive status information. The format of the STAT word:

Bit #	Function	Convention
0 (LSB)	Disable Status	1 = drive is disabled 0 = drive is enabled
1	Fault Status	1 = fault exists 0 = no fault exists
2	Safety Status	1 = safety feature triggered/inactive* 0 = drive is safe
3	Special Mode Status	1 = Step, Burnin, or Zero is active 0 = normal
4**	Hold Mode Status**	1 = drive is in Hold mode, in position, or stopped 0 = drive is not in Hold mode
5-15	not used	

\*CWLIM=1, CCWLIM=1, LIMDIS=1, THERMODE=1 or 2, or FOLD=1, or (PLIM>0 and PFB>PMAX), or (PLIM>0 and PFB<PMIN).

\*\* Prior to firmware version 2.0.0, Bit 4 was undefined.

**Firmware Versions:** All

**Type:** variable (R)

**Range:** see above

**Default:** N/A

**Opmodes:** All

**Drive Status:** DIS

**Units:** N/A

**EEPROM:** No

## STATUS

Outputs the drive status detail words to the serial port. Five words are transferred in ASCII-HEX format, with each word preceded by the letter *H*. The words are separated by a space.

The following tables break the status words down bit by bit (bit 15 = MSB; bit 0 = LSB; n/u = not used). For all bits, 0=false and 1=true.

<b>STATUS Word 1: Disable Status Word</b>	
<b>If the drive is disabled (Bit 0 of the STAT word = 1), the process(es) which have caused that disable condition will have their bits set to 1 in this word.</b>	
Bit #	Description
0 (LSB)	Remote disable (REMOTE = 0)
1	Software disable (SWEN = 0)
2	DIP switch disable (DIPEN = 0)
3	Fault disable
4	Velocity loop design failure
5	Encoder not initialized
6-15	not used

<b>STATUS Word 2: Fault Status Word</b>	
<b>If a fault exists (Bit 1 of the STAT word = 1), the fault(s) has the corresponding bits set to 1 in this word.</b>	
Bit #	DESCRIPTION
0 (LSB)	Drive over temperature
1	Over voltage condition
2	Over current condition
3	Feedback loss
4	Under voltage condition
5	Motor over temperature
6	Analog supply fault
7	Over speed condition
8	EEPROM fault
9	EEPROM checksum fault
10	No comp (compensation) for the motor
11	Foldback condition
12	not used
13	Overtravel fault
14	Position deviation fault
15	not used

<b>STATUS Word 3: Safety Status Word</b>	
<b>If safety of the drive is compromised (Bit 2 of the STAT word = 1), the condition causing that state has its corresponding bit set to 1 in this word.</b>	
Bit #	Description
0 (LSB)	CWLIM = 1 (motor has reached CW travel limit)
1	CCWLIM = 1 (motor has reached CCW travel limit)
2	LIMDIS = 1 (limit switch function disabled by user)
3	THERMODE = 1, 2, 3 (set to non-zero by the user)
4	FOLD = 1 (drive current foldback mode)
5*	LIMDIS=0 & CW switch not routed (INxMODE 1)
6*	LIMDIS=0 & CCW switch not routed (INxMODE 2)
7***	Positive Overtravel (PFB > PMAX) with PLIM > 0
8***	Negative Overtravel (PFB < PMIN) with PLIM > 0
9****	MFOLD = 1 (motor current foldback mode)
10-15	not used



<b>STATUS Word 4: Special Mode Status Word</b>	
<b>If the drive is in a special operating mode (Bit 3 of the STAT word = 1), the drive's special mode has its corresponding bit set to 1 in this word.</b>	
Bit #	DESCRIPTION
0 (LSB)	Drive is in Step mode (see <b>STATUS2</b> )
1	Drive is in Burnin mode (factory function)
2	Drive is in Zeroing mode (see <b>XENCRES</b> )
3-15	not used

<b>STATUS Word 5: Hold Mode Status Word</b>	
<b>If the drive is in Hold mode (Bit 4 of the STAT word = 1), the condition causing the drive to enter Hold mode has its corresponding bit set to 1 in this word).</b>	
Bit #	Description
0 (LSB)	User request (user set HOLD = 1)
1	DIP switch setting (DIP switch 7 = 1)
2	Drive is in Active disable state
3	Limit switch(es) tripped: 1. velocity command is in direction of tripped switch in opmode 0 or 1 with drive enabled; or 2. both limit switches are activated
4**	User input switch hold (INxMODE=19)
5**	Internal hold request during homing process.
6-10	not used
11++	Analog position hold before homing
12-15	not used

**Firmware Versions:** All, with exceptions:

\*-versions 2.1.0 and later

\*\*-versions 3.0.0 and later

\*\*\* - versions 3.1.0 and later

\*\*\*\* - versions 3.2.0 and later

++ - versions 3.3.0 and later

**Type:** variable (R)

**Default:** N/A

**Opmodes:** All

**EEPROM:** No

**Range:** see above

**Units:** N/A

Drive Status: EN/DIS

## STATUS2

Outputs drive status detail words to the serial port. Four words are transferred in ASCII-HEX format, with each word preceded by the letter *H*.

The following tables break the status words down bit by bit (bit 15 = MSB; bit 0 = LSB; n/u = not used). For all bits, 0=false and 1=true.

<b>STATUS2 Word 1: Feedback Loss Status Word</b>	
<b>If the drive has experienced a feedback loss fault (Bit 3 of STATUS Word 2 = 1), the condition causing the fault has its bit set to 1 in this word.</b>	
Bit #	DESCRIPTION
0 (LSB)	Resolver line break
1	Resolver/Digital Converter Error bit (following err)
2	Sine encoder initialization failed
3	Line break of encoder A/B input
4	Line break of encoder index input
5	Illegal halls state
6	Line break of encoder C/D input (sine encoder)
7	A/B lines out of range (sine encoder)
8	Burst overflow (sine encoder)
9*	External feedback line break
10-15	not used

<b>STATUS2 Word 2: Analog Supply Fault Status Word</b>	
<b>If the drive has experienced an analog supply fault (Bit 6 of STATUS Word 2 = 1), the condition causing the fault has its bit set to 1 in this word.</b>	
Bit #	Description
0 (LSB)	Positive analog supply fault
1	Negative analog supply fault
2-15	not used

<b>STATUS2 Word 3: Position Deviation and Over Travel Fault Status Word</b>	
Bit #	Description
0	Internal Numerical Position Deviation
1	Pos. Error ( <b>PE</b> ) exceeded max <b>PE</b> limit ( <b>PEMAX</b> )
2	Positive Over-travel (PFB > PMAX) with PLIM=1
3	Negative Over-travel (PFB < PMIN) with PLIM=1
4-15	Reserved

<b>STATUS2 Word 4: Limit Switches Status Word</b>	
<b>If the drive has experienced a limit switch fault (Bit 3 of STATUS Word 5 = 1), the condition causing the fault has its bit set to 1 in this word.</b>	
Bit #	Description
0 (LSB)	CW Limit Switch tripped (CWLIM=1)
1	CCW Limit Switch tripped (CCWLIM=1)
2-15	not used

<b>STATUS2 Word 5: Encoder Initialization Status Word (for firmware version 3.3.0 and higher)</b>	
<b>If the drive has experienced a fault during encoder initialization (Bit 5 of STATUS Word 1=1), the condition causing the fault has its bit set to 1 in this word.</b>	
Bit #	Description
0 (LSB)	<b>MENCRES</b> , <b>MPOLES</b> or low <b>IENCSTART</b> failure
1	Phase A current mismatch
2	Phase C current mismatch
3	Limit switch tripped
4-15	not used

<b>STATUS2 Word 6: Over speed Status Word</b> (firmware version 3.3.0 and higher)	
<b>This status word details the cause of over speed fault.s.</b>	
<b>Bit #</b>	<b>Description</b>
0 (LSB)	velocity feedback > VOSPD
1	velocity feedback > 1.8*VLIM
2-15	not used

- firmware version 3.3.0 and later

**Firmware Versions:** 2.1.0 and later

**Range:** see above

**Opmodes:** All

**Units:** N/A

**Type:** variable (R)

**Default:** N/A

**Drive Status:** DIS

**EEPROM:** No

## STATUS2

Returns drive status words to the serial port. Six words are transferred in ASCII-HEX format, with each word preceded by the letter *H*. The encoder initialization status is reported in word 5, which is described below. Bit 2 in the first word indicates if the encoder initialization process has failed.

<b>STATUS2 Word 1: Feedback Loss Status Word</b>	
<b>If the drive has experienced a feedback loss fault (Bit 3 of STATUS Word 2 = 1), the condition causing the fault has its bit set to 1 in this word.</b>	
<b>Bit #</b>	<b>Description</b>
0 (LSB)	Resolver line break
1	Resolver/Digital Converter Error bit (following err)
2	Sine encoder initialization failed
3	Line break of encoder A/B input
4	Line break of encoder index input
5	Illegal halls state
6	Line break of encoder C/D input (sine encoder)
7	A/B lines out of range (sine encoder)
8	Burst overflow (sine encoder)
9*	External feedback line break
10-15	not used

<b>STATUS2 Word 2: Analog Supply Fault Status Word</b>	
<b>If the drive has experienced an analog supply fault (Bit 6 of STATUS Word 2 = 1), the condition causing the fault has its bit set to 1 in this word.</b>	
<b>Bit #</b>	<b>Description</b>
0 (LSB)	Positive analog supply fault
1	Negative analog supply fault
2-15	not used

<b>STATUS2 Word 3: Position Deviation and Over Travel Fault Status Word</b>	
<b>Bit #</b>	<b>Description</b>
0	Internal Numerical Position Deviation
1	Pos. Error ( <b>PE</b> ) exceeded max <b>PE</b> limit ( <b>PEMAX</b> )
2	Positive Over-travel (PFB > P <b>MAX</b> ) with PLIM=1
3	Negative Over-travel (PFB < P <b>MIN</b> ) with PLIM=1
4-15	Reserved

<b>STATUS2 Word 4: Limit Switches Status Word</b>	
<b>If the drive has experienced a limit switch fault (Bit 3 of STATUS Word 5 = 1), the condition causing the fault has its bit set to 1 in this word.</b>	
<b>Bit #</b>	<b>Description</b>
0 (LSB)	CW Limit Switch tripped (CWLIM=1)
1	CCW Limit Switch tripped (CCWLIM=1)
2-15	not used

<b>STATUS2 Word 5: Encoder Initialization Status Word (firmware 6.3.3 and higher)</b>	
<b>If the drive has experienced a fault during encoder initialization (Bit 5 of STATUS Word 1 = 1), the condition causing the fault has its bit set to 1 in this word.</b>	
<b>Bit#</b>	<b>Description</b>
0 (LSB)	<b>MENCRES</b> , <b>MPOLES</b> or low <b>IENCSTART</b> failure
1	Phase A current mismatch
2	Phase C current mismatch
3	Limit switch tripped
4	Internal fault
5	Runaway – algorithm failed to due excessive motion. Either <b>IENCSTART</b> or <b>INITGAIN</b> is too high. Make sure <b>IENCSTART</b> is set correctly. Reduce the value of <b>INITGAIN</b> .
6	Motion profile mismatch. Adjust the value of <b>INITGAIN</b> .
7	Insufficient motion. Increase <b>IENCSTART</b> or <b>INITGAIN</b> .
8-15	Not used

<b>STATUS2 Word 6: Over speed Status Word (firmware version 3.3.0 and higher)</b>	
<b>This status word details the cause of over speed fault.s.</b>	
<b>Bit #</b>	<b>Description</b>
0 (LSB)	velocity feedback > VOSPD
1	velocity feedback > 1.8*VLIM
2-15	not used

- firmware version 3.3.0 and later

**Range:** The bits in status word 5 are described in the following table. Bits 4-7 are relevant for INITMODE 2.

**Default:** 0

**Opmodes:** All

**Drive Status:** DIS

**EEPROM:** Yes

## STEP

Generates a step or square wave velocity command. This command is used to record the drive response after **RECTRIG** has been set up to define the trigger as occurring after the next command. This command takes 2, 3, or 4 parameters.

When 2 parameters are used, the drive is issued a **STEP** command with a specified duration (“duration1”) and velocity (“velocity1”).

When 3 parameters are used, the command becomes a repeating square wave which includes a zero velocity cycle whose duration is specified by the third parameter (“duration2”).

When 4 parameters are used, the square wave command runS for the time specified by “duration1” at the speed specified by “velocity1,” then runS for the time specified by “duration2” at the speed specified by “velocity2.” This motion then repeats.

You can terminate the command by entering **S**, **K**, **DIS**, or Jog (**J**). This command is prohibited while in Hold mode.

**Firmware Versions:** All

**Command Syntax:** STEP [duration1] [velocity1] {<duration2> <velocity2>}

**Range:** [durationN]: 0 to 32767  
[velocityN]: -VLIM to +VLIM

**Units:** [durationN] milliseconds  
[velocityN]: rotary: rpm  
linear: mm/sec

**Opmodes:** 0

**Drive Status:** EN

## STOP

Stops motion in OPMODE 0 (**J** and **STEP**) or 2 (**T**). Unlike **S** and **K**, the drive is not disabled using **STOP**. Deceleration ramp control is used in OPMODE 0, if PROFMODE is set to 1. The deceleration rate is stored in **DEC**. If this command is invoked in Opmode 1 or 3, it is ignored.

**Firmware Versions:** All

**Command Syntax:** STOP

**Opmodes:** 0, 2

**Drive Status:** EN

## STOPMODE

Sets the mode of dynamic braking operation. See **ISTOP**.

**0** = no braking operation (default).

**1** = brake on fault only.

**2** = brake on fault and/or drive disable.



**Faults do not include over-voltage or power stage faults!**

### NOTE

**Firmware Versions:** 2.1.0 and later

**Type:** switch mode (R/W)

**Range:** 0 to 2

**Default:** 0

**Opmodes:** All

**Drive Status:** EN/DIS

**Units:** N/A

**EEPROM:** Yes

## STOPPED

A read-only switch indicating the status of a move command (**MA**, **MI**, or **MH**) issued by the profile generator in Opmode 8. This bit reads 1 when a move is complete and the next move command can be issued. It reads 0 when a move is in progress.

0 = move in progress.

1 = move complete; next move command can be issued.

**Firmware Versions:** 3.1.0 and later

**Type:** switch (R)

**Range:** 0, 1

**Default:** N/A

**Opmodes:** 8

**Drive Status:** EN/DIS

**Units:** N/A

**EEPROM:** N/A

## SWEN

A software enable switch defining the status of the serial port Enable (**EN**) request. If SWEN=1, and there are no faults (DRIVEOK = 1), then READY=1.

0 = software disabled (**DIS**, **K**, or **S** was issued)

1 = software enabled (**EN** was issued)

**Firmware Versions:** 3.1.0 and later

**Type:** switch (R)

**Range:** 0, 1

**Default:** 1 (analog drives)  
0 (SERCOS)

**Opmodes:** All

**Drive Status:** EN/DIS

**Units:** N/A

**EEPROM:** No

## T

Used to set commanded current in OPMODE 2 (Serial Torque Mode). This command is subject to current limits, clamps, and digital filtering, and it is set to zero whenever the drive is enabled or disabled. The range of this value is -1000 to 1000, but the value entered cannot exceed **ILIM**. An **S**, **STOP**, **DIS**, **K**, or change of operating mode zeros the value of **T**.

**COMMAND SYNTAX:** T [*current*] (where  $-1000 \leq \text{current} \leq 1000$ )

**Firmware Versions:** All

**Units:** % of DIPEAK\*0.1

**Range:** -ILIM to +ILIM

**Drive Status:** EN

**Opmodes:** 2

## TESTLED

Used to put the drive into a Status LED test mode. In this test mode, all the segments of the LED illuminate for approximately half a second, then return to normal.

**Firmware Versions:** All

**Command Syntax:** TESTLED

**Opmodes:** All

**Drive Status:** DIS

## TF

Sets the damping factor for the velocity loop when using COMPMODE 2 or COMPMODE 4 (Standard Pole Placement). A value of 100 is backward compatible to all previous firmware.

As **TF** approaches zero, overshoot is diminished while sacrificing some tracking ability. As **TF** approaches 200, the system may overshoot more, but has excellent steady-state tracking ability. Successful execution of **TUNE** may result in this parameter being changed.

**Firmware Versions:** 3.1.0 and later

**Range:** 0 to 200

**Opmodes:** 0,1,4,8

**Units:** N/A

**Type:** switch (R)

**Default:** 100

**Drive Status:** EN/DIS

**EEPROM:** Yes

## THERM

Indicates the state of the motor thermostat input.

**0** = thermostat input closed (normal)

**1** = thermostat input open (overheat condition)

**Firmware Versions:** 3.1.0 and later

**Range:** 0, 1

**Opmodes:** All

**Units:** N/A

**Type:** switch (R)

**Default:** hardware defined

**Drive Status:** EN/DIS

**EEPROM:** No

## THERMODE

Determines drive operation when the Motor Thermostat Input (**THERM**) opens.

**0** = disable drive and open fault relay immediately

**1** = disable drive after 2 minutes; open fault relay immediately

**2** = do not disable drive; open fault relay immediately

**3** = ignore thermostat input

**4** = issue warning; no other action\*

**5** = issue warning, open fault relay after **THERM**TIME elapses\*

\* Firmware versions 3.1.0 and later



*Opening the fault relay sets RELAY=0.*

### NOTE

**Firmware Versions:** All

**Range:** 0 to 5

**Opmodes:** All

**Units:** N/A

**Type:** switch mode (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## THERMTIME

Sets the number of seconds the drive waits after motor over-temperature detection before it opens the fault relay (**THERMODE**=5 only).

**Firmware Versions:** 3.1.0 and later

**Range:** 1 to 300

**Opmodes:** All

**Units:** Seconds

**Type:** variable (R/W)

**Default:** 30

**Drive Status:** EN/DIS

**EEPROM:** Yes

## THERMTYPE

Sets the motor temperature sensor type:

- 0 = PTC (Positive Temperature Coefficient)
- 1 = NTC (Negative Temperature Coefficient)

**Firmware Versions:** 3.1.0 and later

**Range:** 0, 1

**Opmodes:** All

**Units:** Seconds

**Type:** switch (R/W)

**Default:** 0

**Drive Status:** EN/DIS

**EEPROM:** Yes

## TRUN

Provides a relative incremental run time counter. Error log stamps include the value of this counter at the time of the error. The clock is a very coarse counter and is incremented every 15 minutes. It is intended for use by factory quality assurance program personnel. This clock has a resolution of 15 minutes and is reset only when **CLREEPROM** is used.

**Firmware Versions:** All

**Range:**0000:00 to 9999:45

**Opmodes:** All

**Units:** hours: minutes

**Type:** variable (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** Yes

## TUNE

Tunes the system for the given drive and load conditions. Velocity steps are performed in closed loop while maintaining position and velocity constraints in order to capture the system dynamics and set tuning constants accordingly.

Motor rotations are performed in OPMODE 0, with a bandwidth of 10 Hz. Successful termination of this command sets the value of **LMJR** (COMPMODE=2 or 4) and changes the control variables of the **PI** (COMPMODE = 0) and **PDFF** (COMPMODE = 1) controllers accordingly. Successful termination of this command may change **BW**, **FILTMODE**, **GP**, **GV**, **GVI**, **KV**, **KVI**, **KVFR**, **LMJR**, and **TF**.

The command can take a few seconds to execute. This command is not always successful, in which case the tuning variables must be set manually. Unsuccessful termination may result due to current saturation, a motor that cannot rotate, or an unsuccessful controller design. For successful termination, **VLIM** ≥ 500 rpm, and **VMAX** ≥ 1160 rpm.

This command takes three optional parameters: bandwidth, direction, and speed. Bandwidth ranges from 10 to 100 Hz (the default is the current bandwidth **BW**). Direction is equal to 0, 1, or 2 (0 = bi-directional rotation, which is the default, 1 = CW rotation only, 2 = CCW rotation only). Speed must be greater than 350 rpm, and the default is the minimum of 500, (0.7 \* VLIM), and (0.3 \* VMAX).



## Recommendations:

- Use low bandwidth for tuning and increase the bandwidth (using **BW**), if desired, after tuning is successful;
- Execute a **SAVE** after **TUNE** has executed successfully in order to write new gain parameters to EEPROM;
- The higher the **TUNE** speed, the more accurate the process is.

**NOTE**

**This command not available in version 1.0.0 & 1.0.1 firmware.**

**Firmware Versions:** All

**Command Syntax:** TUNE [*bw*] [*dir*] [*speed*]

*bw* = bandwidth in Hz (10 to 100)

*dir* = 0, 1, 2 (0=bidir, 1=CW, 2=CCW)

*speed* = speed used during TUNE in rpm (rotary) or mm/sec (linear) (350 to 0.7 \* VLIM)

**Opmodes:** All

**Drive Status:** EN/DIS

## UNITS

Defines whether physical units or internal bits are used. This variable is relevant for current, velocity, acceleration and analog input variables to allow more precise definitions while using the internal bits of the Integer variables. It is recommended that physical units are used.

**0** = use physical units

**1** = use internal units

The descriptions in this manual use physical units. Variables defined using internal units are listed here with their internal unit ranges and units. UNITS, whether user or internal, do not affect the velocity of MA or MI.

INTERNAL VARIABLE UNITS		
Variable	Range	Internal Units
ANDB	0 to 16383 bits	1 bit = 10V / 16384
ANIN	-16383 to 16383 bits	1 bit = 10V / 16384
ANOFF	-16383 to 16383 bits	1 bit = 10V / 16384
I	0 to 65535 bits	32768 bits = DIPEAK * ( $\sqrt{2}/0.8$ )
IA	-32767 to 32767 bits	32768 bits = DIPEAK * ( $\sqrt{2}/0.8$ )
IC	-32767 to 32767 bits	32768 bits = DIPEAK * ( $\sqrt{2}/0.8$ )
ICMD	-32767 to 32767 bits	32768 bits = DIPEAK * ( $\sqrt{2}/0.8$ )
ICONT	0 to IMAX bits	32768 bits = DIPEAK * ( $\sqrt{2}/0.8$ )
ILIM	0 to IMAX bits	32768 bits = DIPEAK * ( $\sqrt{2}/0.8$ )
IMAX	0 to 32767 bits	32768 bits = DIPEAK * ( $\sqrt{2}/0.8$ )
J <vel> <time>	vel:-16383 to 16383 time:0 to 32767	vel:1 bit = VLIM/16384 time:1 bit = 0.5 ms
STEP<period><speed>	Period :0 to 32767 Speed :-16363 to +16383	period: milliseconds speed: VLIM / 16384
T	-ILIM to ILIM	32768 bits = DIPEAK * ( $\sqrt{2}/0.8$ )
V	-32767 to 32767	1 bit = VLIM / 16384
VCMD	-VLIM to VLIM	1 bit = VLIM / 16384
VE	-16383 to 16383	1 bit = VLIM / 16384

**Firmware Versions:** All  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## UVMODE

Defines how the drive will respond to an under-voltage (UV) fault:

- 0 = latch fault immediately, display flashing “u”.
- 1 = display steady “u”. Warning only, with no fault latch.
- 2 = display steady “u” - after **UVTIME** elapses, latch fault relay.

If UVMODE= 1 or 2, and the drive is disabled, the UV fault is ignored.

**Firmware Versions:** 3.1.0 and later  
**Range:** 0 to 2  
**Opmodes:** All  
**Units:** N/A

**Type:** switch mode (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## UVRECOVER

Defines how the drive will recover from an under-voltage (UV) fault:

- 0 = recover by toggling drive from disable to enable condition after the UV condition clears
- 1 = automatically recover when the UV condition clears

**Firmware Versions:** 3.1.0 and later  
**Range:** 0, 1  
**Opmodes:** All  
**Units:** N/A

**Type:** switch (R/W)  
**Default:** 0  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## UVTIME

Sets the amount of time an under-voltage warning is displayed (u) before it is latched when UVMODE=2.

**Firmware Versions:** 3.1.0 and later  
**Range:** 1 to 300  
**Opmodes:** All  
**Units:** Seconds

**Type:** variable (R/W)  
**Default:** 30  
**Drive Status:** EN/DIS  
**EEPROM:** Yes

## V

Displays the velocity as calculated from the hardware feedback (resolver or encoder). The velocity displayed is subject to averaging by **AVGTIME** except when recorded for graphical display by **MOTIONLINK**, in which case it is not averaged.

**Firmware Versions:** All  
**Range:** -15000 to 15000  
**Opmodes:** All  
**Units:** rotary: rpm  
 linear: mm/sec

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** No

## VBUS

Sets the drive bus voltage. This variable is used for current controller design. **VBUS** affects the value of **VMAX**. When this variable changes, the drive enters a no-comp state and requires a **CONFIG**.

**Firmware Versions:** All

**Range:** 10 to 850

**Opmodes:** All

**Units:** volts

**Type:** variable (R/W)

**Default:** 325

**Drive Status:** DIS

**EEPROM:** Yes

## VCMD

Displays the Velocity command to the velocity controller. This value is equivalent to the Analog Input (**ANIN**) in OPMODE 1, to the Jog command (**J**) in OPMODE 0, and the output of the position controller gearing (OPMODE 4), positioning (OPMODE 8), and Hold Position mode (HOLD=1). This variable is averaged, based on the **AVGTIME**, when requested via the serial port. It is not averaged when it is recorded for graphical display.

**Firmware Versions:** All

**Range:** -VLIM to +VLIM

**Opmodes:** 0,1,4,8

**Units:** rotary: rpm  
linear: mm/sec

**Type:** variable (R)

**Default:** N/A

**Drive Status:** EN/DIS

**EEPROM:** No

## VD

A vector variable that sets the forward path polynomial of the Advanced Pole-Placement velocity controller (COMPMODE = 3). The vector defined by this variable includes five integers that represent the polynomial coefficients and a shift parameter that scales the polynomial. A **REFRESH** is required if this variable changes.



### NOTE

*Prior to firmware version 2.1.0, this command mnemonic was "D".*

**Firmware Versions:** 2.1.0 and later (previously D)

**Syntax:** VD [vector1] [vector2] [vector3] [vector5] [scale]

**Range:** [vectorN] -32768 to 32767  
[scale] 0 to 15

**Opmodes:** 0,1,4,8

**Type:** vector variable (R/W)

**Default:** 0 (all parameters)

**EEPROM:** Yes

**Units:** N/A

**Drive Status:** EN/DIS

**Example:** VD 100 200 300 400 500 1

## VE

Displays the velocity error (difference between the commanded motor velocity (**VCMD**) and the actual motor velocity (**V**). This value is an instantaneous reading.

**Firmware Versions:** All  
**Range:** -32768 to 32767  
**Opmodes:** 0,1,4,8  
**Units:** rotary: rpm  
 linear: mm/sec

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** No

## VER

Indicates the version of the drive firmware in use. This variable also displays other pertinent information such as the drive name, current ratings, **TRUN**, etc. **VER** has two optional parameters: VER 1 returns feedback type encoder or resolver, and VER 2 returns the firmware version.

**Firmware Versions:** All  
**Range:** VER {1 or 2}  
**Opmodes:** All  
**Units:** N/A

**Type:** string variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** No

## VEXT

Displays the instantaneous velocity feedback as calculated from the external encoder input channel. The command uses **XENCRES** to calculate velocity. This variable is similar to **V** for the motor feedback. This variable is subject to **AVGTIME**.

**Firmware Versions:** 2.1.0 and later  
**Range:** -32767 to +32767  
**Opmodes:** All  
**Units:** rotary: rpm  
 linear: mm/sec

**Type:** variable (R)  
**Default:** N/A  
**Drive Status:** EN/DIS  
**EEPROM:** N/A

## VF

A vector variable that defines the filter at the output of the Advanced Pole-Placement velocity controller (COMPMODE = 3). The vector defined by this variable includes five integers that represent the polynomial coefficients, and two shift parameters, one that scales each polynomial. A **REFRESH** is required if this variable changes.

Filter structure:  $(b_0 + b_1*z^{-1} + b_2*z^{-2}) / (1 + a_1*z^{-1} + a_2*z^{-2})$ .

**Syntax:** VF [b0] [b1] [b2] [bshift] [a1] [a2] [ashift]  
**Range:** [aN], [bN] 32767 to -32768  
 [ashift], [bshift] 0 to 32767

**Firmware Versions:** 3.1.0 and later  
**Type:** vector variable (R/W)  
**Default:** 1 0 0 0 0 0  
**Example:** VF 100 200 300 4 500 600 7

**Opmodes:** 0, 1, 4, 8  
**EEPROM:** Yes  
**Units:** N/A  
**Drive Status:** EN/DIS

## VFI

A vector variable that defines the filter at the input of the Advanced Pole-Placement velocity controller (COMPmode = 3). The vector defined by this variable includes five integers that represent the polynomial coefficients, two shift parameters, and one that scales each polynomial. A **REFRESH** is required if this variable changes.

Filter structure:  $(b_0 + b_1*z^{-1} + b_2*z^{-2}) / (1 + a_1*z^{-1} + a_2*z^{-2})$ .

**Syntax:** VFI [b0] [b1] [b2] [b\_shr] [a1] [a2] [a\_shr]

**Range:** [aN], [bN] 32767 to -32768  
[a\_shr], [[b\_shr] 0 to 32767

**Firmware Versions:** 3.2.0 and later

**Type:** vector variable (R/W)

**Default:** 1 0 0 0 0 0

**Example:** VF 100 200 300 4 500 600 7

**Opmodes:** 0,1,4,8

**Units:** N/A

**Drive Status:** EN/DIS

**EEPROM:** Yes

## VH

A vector variable that defines the H (feedback path) polynomial of the Advanced Pole-Placement velocity controller (COMPmode = 3). The vector defined by this variable includes four integers that represent the polynomial coefficients, four shift parameters, and one that scales each polynomial. A **REFRESH** is required if this variable changes.



*Prior to firmware version 2.1.0, this command mnemonic was "H".*

### NOTE

**Syntax:** VH [h0] [h0shift] [h1] [h1shift] [h3] [h3shift]

**Range:** [hN] = -2,147,483,647 to 2,147,483,647  
[hNshift] = 0 to 32767

**Firmware Versions:** 3.2.0 and later

**Type:** vector variable (R/W)

**Default:** 0 (all parameters)

**Example:** VF 100 200 300 4 500 600 7

**Opmodes:** 0,1,4,8

**Units:** N/A

**Drive Status:** EN/DIS

**EEPROM:** Yes

## VLIM

Sets the application velocity limit, allowing you to limit the motor's peak velocity. **VLIM** limits the accepted velocity command (using the J command in Opmode 0) issued either by the user or by the control loops (in Opmode 1). **VLIM** is an independent variable that is not calculated from hardware parameters and is not tied to any other variables. **VLIM** is similar to **ILIM** (used in Opmodes 2 & 3) and can be used to protect delicate load equipment. For rotary motors, VLIM > 6100 only if ENCOUT ≤ 1024, and VLIM > 1500 only if ENCOUT ≤ 4096.

**Firmware Versions:** All

**Range:** 10 to VMAX

**Opmodes:** 0,1,4,8

**Units:** rotary: rpm

linear: mm/sec

**Type:** variable (R/W)

**Default:** VMAX

**Drive Status:** DIS

**EEPROM:** N/A

# VMAX

Displays the system velocity maximum for a drive and motor combination. This variable is based on drive and motor hardware parameters and is set equal to the MINIMUM of the five following values:

- 1.) MSPEED
- 2.)  $(VBUS * 0.707 / MBEMF) * 1000$
- 3.) 24,000
- 4.)  $180,000,000 / MENCRES$  (encoder-feedback systems only)
- 5.)  $192,000 / MRESPOLES$  (resolver system, MRESPOLES>8)



## NOTE

**24,000 is the highest value VMAX can take. VMAX limits VLIM and VOSPD.**

**Firmware Versions:** All  
**Range:** 10 to 24,000  
**Opmodes:** 0,1,4,8  
**Units:** rotary: rpm  
 linear: mm/sec

**Type:** variable (R)  
**Default:** see above  
**Drive Status:** DIS  
**EEPROM:** Yes

# VOSPD

Sets the overspeed trip limit for the motor. The drive is disabled with an error condition when the drive velocity exceeds this limit. The default value of this variable is 20% above the system velocity maximum (VMAX), but can be reduced during regular motor operation for protection.

**Firmware Versions:** All  
**Range:** 10 to  $(1.2 * VMAX)$   
**Opmodes:** All  
**Units:** rotary: rpm  
 linear: mm/sec

**Type:** variable (R/W)  
**Default:**  $VMAX * 1.2$   
**Drive Status:** DIS  
**EEPROM:** Yes

# VOSPDENS

Sets the sensitivity parameter.

**Firmware Versions:** 4.1.8 and later  
**Range:** 1 to 40  
**Opmodes:** All  
**Units:**

**Type:** variable (R/W)  
**Default:** 20  
**Drive Status:**  
**EEPROM:** Yes

# VR

A vector variable that defines the R (feed-forward path) polynomial of the Advanced Pole-Placement velocity controller (COMPmode = 3). This vector includes three integers that represent the polynomial coefficients and three shift parameters, one that scales each polynomial. If this variable is changed, a **REFRESH** is required.



## NOTE

**Prior to firmware version 2.1.0, this command mnemonic was "R".**

**Syntax:** VR [r0] [r0shift] [r1] [r1shift] [r2] [r2shift]

**Range:** [rNvector] = -2,147,483,647 to 2,147,483,647  
[rNshift]= 0 to 32767

**Firmware Versions:** 2.1.0 and later (previously R)

**Opmodes:** 0,1,4,8

**Type:** vector variable (R/W)

**Default:** 0 (all parameters)

**Units:** N/A

**EEPROM:** Yes

**Drive Status:** EN/DIS

**Example:** VR 10000 2 30000 4 50000 6

## VSCALE

An analog velocity scale factor that scales the analog input **ANIN** for OPMODE 1 (analog torque mode), and the analog output for ANOUT=0 or 2. The value entered is the motor velocity per 10 volts of analog input or output. This variable may be either higher or lower than the application velocity limit (**VLIM**), but the actual analog I/O is limited by **VLIM**.

**Firmware Versions:** All

**Type:** variable (R/W)

**Range:** 10 to (1.2 \* VMAX)

**Default:** VLIM / 0.8

**Opmodes:** 1

**Drive Status:** EN/DIS

**Units:** rotary: rpm / 10 V  
linear: mm/sec / 10 V

**EEPROM:** Yes

## XENCDIR

Sets the direction defined as positive rotation for the external encoder input.

**0** = normal

**1** = inverted

**Firmware Versions:** 2.1.0 and later

**Type:** variable (R/W)

**Range:** 0, 1

**Default:** 0

**Opmodes:** All

**Drive Status:** DIS

**Units:** N/A

**EEPROM:** Yes

## XENCDIR

Sets the direction defined as positive rotation for the external encoder input.

**0** = normal

**1** = inverted

**Type:** variable (R/W)

**Range:** 0,1

**Default:** 0

**Opmodes:** all

**Drive Status:** dis

**EEPROM:** yes

## XENCRES (UCB1 SERIES 2)

Sets the resolution of the external encoder input channel and is used to calculate **VEXT**.

**Firmware Versions:** 2.1.0 and later

**Type:** variable (R/W)

**Range:** 100 to 10,000,000

**Default:** 1024

**Opmodes:** All

**Drive Status:** DIS

**Units:** Lines per revolution

**EEPROM:** Yes

## XENCRES (UCB2 SERIES 5)

Sets the resolution of the external encoder input channel and is used to calculate **VEXT**.

**Type:** variable (R/W)

**Range:** 100 to 10,000,000

**Units:** Lines per revolution

**Default:** 1024

**Opmodes:** all

**Drive Status:** dis

**EEPROM:** yes

## ZERO

Enables and disables Resolver/Encoder Zeroing mode. If Zeroing Mode is enabled, the drive rotates the motor to an electrical null by placing **IZERO** current from the motor C terminal to the B terminal.

**0** = zeroing mode disabled

**1** = zeroing mode enabled (puts the drive in OPMODE 2)

**Firmware Versions:** All

**Type:** switch (R/W)

**Range:** 0,1

**Default:** 0

**Opmodes:** All

**Drive Status:** EN/DIS

**Units:** N/A

**EEPROM:** No

## ZPOS

Is applicable only if MENCTYPE=9 (EnDat encoder). It moves the location of the encoder equivalent output marker channel relative to the motor shaft position.



### NOTE

*The range accepts all data, but you must limit it to **MENCRES \* SININTOUT**.*

**Firmware Versions:** 3.6.0 and later

**Type:** variable (R/W)

**Range:** 0 to 2<sup>31</sup>

**Default:** 0

**Opmodes:** All

**Drive Status:** EN/DIS

**EEPROM:** Yes (Encoder must use **HSAVE**)

**Units:** SININTOUT