



EDD-3701x Firmware Command Reference
Revision: 1740

12/06/17

©2017 Galil Motion Control

Table of Content

Table of Content	2
Legend	4
\$ Hexadecimal	5
; Semicolon (Command Delimiter)	6
^R^S Master Reset	7
^R^V Revision Information	8
_ Operand Overview	9
_LF Forward Limit Switch Operand	10
_LR Reverse Limit Switch Operand	11
AB Abort	12
AC Acceleration	13
AF Analog Feedback Select	14
AG Amplifier Gain	15
AL Arm Latch	16
AO Analog Output	17
AQ Analog Input Configuration	18
AU Set amplifier current loop	19
BA Brushless Axis	20
BB Brushless Phase Begins	21
BC Brushless Calibration	22
BD Brushless Degrees	23
BG Begin	24
BI Brushless Inputs	25
BL Reverse Software Limit	26
BM Brushless Modulo	27
BN Burn	28
BO Brushless Offset	29
BQ Brushless Offset dual DAC	30
BR Brush Axis	31
BW Brake Wait	32
BX Sine Amp Initialization	33
BZ Brushless Zero	35
CB Clear Bit	37
CE Configure Encoder	38
CN Configure	39
CW Copyright information and Data Adjustment bit on/off	41
DC Deceleration	42
DE Dual (Auxiliary) Encoder Position	43
DP Define Position	44
DV Dual Velocity (Dual Loop)	45
ER Error Limit	46
FA Acceleration Feedforward	47
FL Forward Software Limit	48
FV Velocity Feedforward	49
ID Identify	50
IL Integrator Limit	51
IT Independent Time Constant - Smoothing Function	52
JG Jog	53

KD	Derivative Constant	54
KI	Integrator	55
KP	Proportional Constant	56
LC	Low Current Stepper Mode	57
LD	Limit Disable	58
MG	Message	59
MO	Motor Off	61
MT	Motor Type	62
NB	Notch Bandwidth	63
NF	Notch Frequency	64
NZ	Notch Zero	65
OA	Off on encoder failure	66
OB	Output Bit	67
OC	Output Compare	68
OE	Off-on-Error	70
OF	Offset	71
OP	Output Port	72
OT	Off on encoder failure time	73
OV	Off on encoder failure voltage	74
PL	Pole	75
PR	Position Relative	76
QH	Query Hall State	77
QP	Query Parameters	78
RL	Report Latched Position	79
RP	Reference Position	80
RS	Reset	81
SB	Set Bit	82
SC	Stop Code	83
SH	Servo Here	84
SP	Speed	85
ST	Stop	86
TA	Tell amplifier error status	87
TB	Tell Status Byte	88
TC	Tell Error Code	89
TD	Tell Dual Encoder	90
TE	Tell Error	91
TI	Tell Inputs	92
TIME	Time Operand	93
TK	Peak Torque Limit	94
TL	Torque Limit	95
TP	Tell Position	96
TS	Tell Switches	97
TT	Tell Torque	98
TV	Tell Velocity	99
YA	Step Drive Resolution	100
YB	Step Motor Resolution	101
YC	Encoder Resolution	102
YR	Error Correction	103
YS	Stepper Position Maintenance Mode Enable, Status	104

Legend

Burnable **Not Burnable**

Description

Commands with the "burnable" icon can be saved into memory with the BN command. If a reset is issued, the value of the command with this icon will persist if it has been burned into memory.

Scaled By TM

Description

Any command with the "scaled by TM" icon will be automatically adjusted whenever a change is made to the TM setting. Commands with this icon are dependent on the sample rate.

Trippoint

Description

A command with the "trippoint" icon will halt further program execution until the trippoint's condition is satisfied. Most trippoints cannot be issued as discrete commands, and are only valid in programs.

Valid In Program **Not Valid In Program**

Description

Commands with the "valid in program" icon can be used inside of a DMC program that is run locally on the controller. Certain commands may not be used in the program space, and can only be issued as discrete command from an external source such as a terminal.

Valid In Terminal **Not Valid In Terminal**

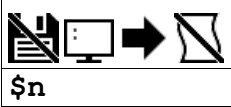
Description

When communicating with a controller externally, only commands which are "valid in terminal" may be sent to the controller as discrete commands. Some commands are only valid when executed in a DMC program and cannot be issued independently.

Valid In Motion **Not Valid In Motion**

Description

If a command is "valid in motion" then it may be executed while an axis is in motion. Some commands may not be executed while certain axes are in motion, and can only be executed when the associated axis is stopped.

\$ Hexadecimal**Description**

The \$ operator denotes that the following string is in hexadecimal notation.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
n	\$80000000.0000	\$7FFFFFFF.FFFF	N/A	\$0.0001	Value of hexadecimal number	32 bits of integer and 16 bits of fraction in total

Remarks

- None

Examples

```
'Galil DMC Code Example
OP $05;' Set outputs 1,3 and clear the others
```

\$ applies to DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

; Semicolon (Command Delimiter)



Description

The semicolon operator allows multiple Galil commands to exist on a single line.

Arguments

arg represents any valid Galil command

Remarks

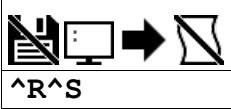
- The semicolon operator is used for the following reasons:
 - To put comments on the same line as the command (STX ;'stop)
 - To compress DMC programs to fit within the program line limit (Note: use a compression utility to do this. Do not program this way because it is hard to read.)
 - To give higher priority to a thread. All commands on a line are executed before the thread scheduler switches to the next thread.

Examples

```
'Galil DMC Code Example
MG @IN[4];AO1,2.3;' multiple commands separated by semicolons
```

; applies to DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,RIO47xxx,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

^R^S Master Reset

Usage	^R^S	Command takes no arguments
--------------	------	----------------------------

Description

The Master Reset command resets the controller to factory default settings and erases EEPROM. A master reset can also be performed by installing a jumper at the location labeled MRST and resetting the board (power cycle or pressing the reset button). Remove the jumper after this procedure.

Arguments

^R^S has no parameters

Remarks

- Sending a ^R^S over an Ethernet connection will cause the IP address to be cleared from the controller and will result in a timeout.

ASCII Values

Char	Dec	Hex
^R	18	12
^S	19	13

Examples

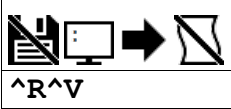
```
'Galil DMC Code Example
REM Example burns-in a non-default value for AQ, does a standard reset with
REM the RS command, then performs a master reset with ^R^S.

:MG_AQ1
2.0000
:AQ 1,3
:MG_AQ1
3.0000
:BN
:RS
:MG_AQ1
3.0000
:^R^S
:MG_AQ1
2.0000
```

^R^S applies to

DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,RIO574x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

^R^V *Revision Information*

Usage	<code>^R^V</code>	Command takes no arguments
--------------	-------------------	----------------------------

Description

The Revision Information command causes the controller to return the firmware revision information.

Arguments

`^R^V` has no arguments

Remarks

- Do not use ^ symbols to send `^R^V` command. ^ symbols denote using the control (Ctrl) key when pressing the characters.

ASCII Values

Char	Dec	Hex
<code>^R</code>	18	12
<code>^V</code>	22	16

Examples

`^R^V` applies to

DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,RIO574x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

Operand Overview



Operand Usage

Operands allow motion or status parameters of the controller to be incorporated into programmable variables and expressions. Most DMC commands have an equivalent operand - which are designated by adding an underscore (_) prior to the DMC command. An operand typically contains the value of the command associated with it, for instance `_TPA` contains the current position of axis A. Below is an example of proper and improper usage for an operand.

Example Usage

'Galil DMC Code Example

'Correct usage

```
MG _TPA; ' Message the A Axis' current position.
```

```
err = _TC; ' Save the current error code to a variable, err.
```

'Incorrect usage

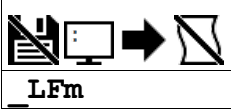
```
_TPA; ' Sending this to the controller will result in an error, as operands are not valid commands on their own.
```

Special Operands

The majority of DMC operands return information directly related to their command. However, there are a few operands which provide access to internal variables that are not accessible by standard DMC commands. Below is a list of special operands which contain information not stored in a typical DMC command.

For more details on the content of these operands, see their associated command page.

Special Operand	Description
<code>_BN</code>	Contains the controller's serial number.
<code>_DA</code>	Contains the number of array space left in the controller's memory.
<code>_DM</code>	Contains the number of array space left in the controller's memory.
<code>_RS</code>	Contains a bitmask of checksum errors.
<code>TIME</code>	Contains the current value of the controller's free running clock.

_LF Forward Limit Switch Operand**_LFm**

Usage	variable= _LF	Holds a value
Operands	_LFm	Operand has special meaning, see Remarks

Description

The _LF operand contains the state of the forward limit.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis of forward limit switch	

Remarks

- _LF is an operand only with the following output:
 - _LFm = 1 when the limit switch state will allow motion in the positive direction.
 - _LFm = 0 when the limit switch state will not allow motion in the positive direction.
- This operand is not a direct readout of the digital input and is affected by the command CN.
- See Connecting Hardware in User Manual for active/inactive state

Values of _LF

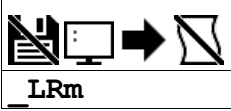
Digital Input activation	_LF value for CN-1	_LF value for CN1
On. Grounded for TTL, or sufficient activation current flowing for optos.	0 (forward motion prohibited)	1 (forward motion allowed)
Off. Pullup for TTL, or insufficient activation current flowing for optos.	1 (forward motion allowed)	0 (forward motion prohibited)

Examples

```
'Galil DMC Code Example
MG _LFA;' Display the status of the A axis forward limit switch
```

_LF applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

_LR Reverse Limit Switch Operand

Usage	variable= _LR	Holds a value
Operands	_LRm	Operand has special meaning, see Remarks

Description

The _LR operand contains the state of the reverse limit.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis of reverse limit switch	

Remarks

- _LR is an operand with the following output
 - _LRm= 1 when the limit switch state will allow motion in the reverse direction.
 - _LRm= 0 when the limit switch state will not allow motion in the reverse direction.
- This operand is not a direct readout of the digital input and is affected by the command CN.
- See Connecting Hardware in User Manual for active/inactive state

Values of _LR

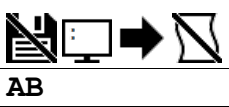
Digital input activation	_LR value for CN-1	_LR value for CN1
On. Grounded for TTL, or sufficient activation current flowing for optos.	0 (reverse motion prohibited)	1 (reverse motion allowed)
Off. Pullup for TTL, or insufficient activation current flowing for optos.	1 (reverse motion allowed)	0 (reverse motion prohibited)

Examples

```
'Galil DMC Code Example
MG _LRA;' Display the status of the A axis reverse limit switch
```

_LR applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

AB *Abort*

AB

Usage	AB n ...	Arguments specified with an implicit, comma-separated order
Operands	_AB	Operand has special meaning, see Remarks

Description

The AB command is a command to issue an abort to controller operation.

AB (Abort) stops motion instantly without a controlled deceleration. The command, AB, will shut off the motors for any axis in which the off on error function is enabled (see command "OE").

Arguments

AB is a command with no arguments.

Remarks

- _AB gives state of Abort Input, 1 inactive and 0 active.

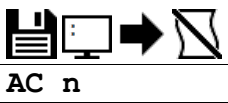
Examples

```
'Galil DMC Code Example
:AB;'      Stops motion
:OE 1;'    Enable off on error on axes
:AB;'      Shuts off motor command and stops motion
```

AB applies to DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,RIO47xxx,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

AC Acceleration



AC n

Usage	ACm= n	Arguments specified with a single axis mask and an assignment (=)
	AC n ...	Arguments specified with an implicit, comma-separated order
Operands	_ACm	Operand holds the value last set by the command

Description

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

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	1,024	1,073,740,800	256,000	1,024	Acceleration rate	At TM 1000. Resolution and Min depend on TM, see remarks

Remarks

- The AC command is used to designate acceleration
- Specify realistic acceleration rates based on physical system parameters such as:
 - motor torque rating
 - loads
 - amplifier current rating
- Specifying an excessive acceleration will cause a large following error during acceleration and the motor will not follow the commanded profile
- The acceleration feedforward command (FA) will help minimize the error for aggressive accelerations

Examples

```
'Galil DMC Code Example
REM Set A-axis acceleration to 150000
AC 150000
MG _ACA;' Prints the A acceleration
```

AC applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

AF Analog Feedback Select

AFm= n
AF n, n, n, n, n, n, n, n, n

Usage	AFm= n	Arguments specified with a single axis mask and an assignment (=)
	AF n ...	Arguments specified with an implicit, comma-separated order
Operands	_AFm	Operand holds the value last set by the command

Description

The AF command configures analog feedback mode for the PID filter.

The controller ADC can be used as position feedback for the axis control law. The analog input used for feedback is fixed and uses the input that corresponds with the axis letter. For example, Analog input 1 is used for the A axis.

Sinusoidal feedback encoders are also configured by the AF command.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	1	0	1	Use the controller ADC as servo feedback	1= analog, 0= digital feedback
	5	12	0	1	Sinusoidal encoder input used with 2^n interpolation counts per encoder cycle	DMC-31xxx required to use sine feedback

Remarks

- Below is the feedback in counts decoded by the controller hardware when reading in analog feedback for certain analog input ranges.

	12 Bit ADC	16 Bit ADC
+/-5 V, +/-10 V	-2048 to 2047 counts	-32768 to 32767 counts
0-5 V, 0-10 V	0 to 4095 counts	0 to 65535 counts

- Voltage/count range with AQ command
 - DMC-30xxx has 0-5v, 12 bit ADC. AQ supported for differential setting only.
 - DMC-31xxx has a 16 bit ADC. See AQ for analog voltage range settings.
- Axis must be in MO state prior to issuing the AF command.
- Differential encoder inputs must be used when using digital encoders with the DMC-31xxx. Consult the factory for single-ended use.
- When using Sin/Cos encoders (AF5-12)
 - The encoder must be connected to the controller prior to issuing the AF command.
 - TP will provide position resolution of $2^{(AFm)}$ counts per cycle. One cycle is four quadrature counts.
 - For example, if an encoder shows a change in TP of 8000 counts with AF0. The same distance at AF 5 would be give by $8000/4 * 2^5 = 64000$

Examples


```
'Galil DMC Code Example
AF 1;'           Analog feedback on A axis
v1= _AFA;'      Assign feedback type to variable
KP 1;'          Assigns PID's for motor using analog feedback on A-axis
KD 10;'
KI 0.5;'
```

```
'Galil DMC Code Example
AF 12;'         Sets sine/cosine feedback to 2^12= 4096 counts/period
AF 8;'          Sets sine/cosine feedback to 2^8= 256 counts/period
```

AF applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

AG Amplifier Gain


AG n, n, n, n, n, n, n, n, n
AGm=n

Usage	AGm= n	Arguments specified with a single axis mask and an assignment (=)
	AG n ...	Arguments specified with an implicit, comma-separated order
Operands	_AGm	Operand holds the value last set by the command

Description

The AG command sets the amplifier current/voltage gain for the internal amplifier. Note: some Galil internal amplifiers have fixed gains. Please reference the manual or data-sheet for more details.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	3	1	1	Gain setting	See table in Remarks for gain settings
	0.5	1.4	0.5	0.0074	Amps per phase	DMC-30016 Gain Range

Remarks

Gain settings by Amplifier (Amps/Volt)

Gain Setting, n=	0	1	2	3	Notes
DMC-3xx12	0.4	0.8	1.6	N/A	
DMC-3xx17	0.4	0.8	1.6	N/A	MT must be set 1 or -1

Gain settings by Amplifier (Amps per phase)

Gain Setting, n=	0	1	2	3	Notes
DMC-3xx17	0.75	1.5	3	6	MT must be set -2 or -2.5

- The axis must be in the motor off state (MO) before setting AG
- The MT command must be issued prior to the AG command to set the proper range

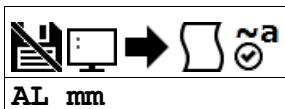
Examples

```
'Galil DMC Code Example
ST ;' Stop any motion
AM ;' wait for motion to decel and stop
MO ;' Turn motor off
MT 1;' Set the A axis as a servo
AG 2;' Sets the highest amplifier gain for A axis on servo amplifier
BN;' Save AG setting to EEPROM
```

AG applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

AL Arm Latch



AL mm

Usage	AL mm	Argument is an axis mask
Operands	_ALm	Operand has special meaning, see Remarks

Description

The AL command enables the latch function (high speed main or auxiliary position capture) of the controller. When the position latch is armed, the main or auxiliary encoder position will be captured upon a rising or falling edge on the specified digital input. Use the CN command to configure the edge that the latch input will trigger on.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	N/A	Multi-Axis Mask	Encoder to latch	Latch main encoder
mm	SA	SA	N/A	Multi-Axis Mask	Encoder to latch	Latch aux encoder
mm	TA	TA	N/A	Multi-Axis Mask	Index input to trigger latch	Main encoder is latched from the index pulse instead of a digital input

Remarks

- Input 1 is used to latch the A axis
- The command RL returns the latched position
- _ALm contains the state of the specified latch. 0 = not armed, 1 = armed
- The CN command can be used to change the edge which causes the latch to trigger.
- The latch function is available on incremental quadrature encoder inputs only. For other position capture methods contact Galil.

Examples

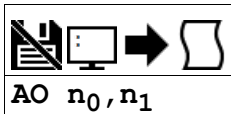
```
'Galil DMC Code Example
#start
AL A;'           Arm A-axis latch
JG 50000;'       Set up jog at 50000 counts/sec
BG A;'           Begin the move
#loop;'          Loop until latch has occurred
JP #loop,(_ALA=1)
RL A;'           Transmit the latched position
EN;'             End of program
```

```
'Galil DMC Code Example
REM Homing routine using the AL command to detect the Motor's index position
#start
AL TA;'         Arm A-axis latch. Latch will trigger off the index pulse
JG 50000;'       Set up jog at 50000 counts/sec
BG A;'           Begin the move
#loop;'          Loop until latch has occurred
JP #loop,(_ALA=1)
STA;'           Stop the jog
AMA
PAA=_RLA;'       Set up a move to return to the latched position
BGA
AMA
WT100;'         Allow for settling.
REM Checking that KI has eliminated error (TE) would be more thorough
DPO;'           Zero position
MG "A Homed";'   Report status
EN;'             End of program
```

AL applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

AO Analog Output



AO n_0, n_1

Usage	AO n ...	Arguments specified with an implicit, comma-separated order
--------------	----------	---

Description

The AO command sets the analog outputs on the Galil or for a Modbus Slave.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
n_0	1	2	N/A	1	Set Local Analog Output	See "AO1 functions" in Remarks
n_0	1,000	6,999	N/A	1	Set Analog Output on Modbus Slave	See "Using AO with a Modbus Slave" in Remarks
n_1	-9.9998	9.9998	N/A	20/65,536	Analog Output Voltage	

Remarks

AO1 Functions

- $n_0 = 1$ for analog output 1. This is available only when a sine drive is being used.
- $n_0 = 2$ for analog output 2. This is always available regardless of the hardware configuration.
- Analog Output 1 can be used as a general purpose Analog Output, or as the Motor Command output to an external Drive. The table below indicates the settings required for the 2 modes.

Controller Configuration	MT Setting	BR Setting	Analog Output 1 Mode
Brushless Motor with Internal Drive (DMC-30012 and DMC-30017)	1 or -1	0	General Purpose Analog Output (AO 1, n_1 is Valid)
Brushed Motor with Internal Drive (DMC-30012 and DMC-30017)	1 or -1	1	General Purpose Analog Output (AO 1, n_1 is Valid)
External Servo Amplifier	1, -1	-1 (DMC-30012 and DMC-30017)	+/-10V Motor Command Output
Stepper Motor on Internal Drive (DMC-30017)	-2 or -2.5	0	Not available as General Purpose Analog Output or Motor Command Output
External Stepper Driver	2, -2, 2.5 or -2.5	-1	Not available as General Purpose Analog Output or Motor Command Output

Using AO with a Modbus Slave

- RIO as Modbus Slave
- 3rd Party Modbus Slave Device
- n_0 is the I/O number calculated using the following equations:
- $n_0 = (\text{HandleNum} * 1000) + ((\text{Module} - 1) * 4) + (\text{Bitnum} - 1)$
 - HandleNum is the handle specifier from A to H.
 - Handle must be assigned to port 502 for Modbus comms (See IH)
 - Module is the position of the module in the rack from 1 to 16.
 - BitNum is the I/O point in the module from 1 to 4

Examples

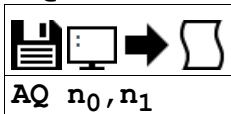
```
'Galil DMC Code Example
:AO 3005,3.2;'           Outputs 3.2 Volts on Channel 5 of the Device connected to Handle C
```

```
'Galil DMC Code Example
:AO 2,1.324;'           Outputs 1.324 Volts on Channel 2
```

AO applies to DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,RIO574x0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

AQ Analog Input Configuration



AQ n_0, n_1

Usage	AQ n ...	Arguments specified with an implicit, comma-separated order
Operands	_AQ1 _AQ2	Operand has special meaning, see Remarks

Description

The AQ command is used to set the behavior of the analog inputs. This command will set the analog range and operation for the specified input.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
n₀	1	2	N/A	1	Analog input channel	$n_0=2$ is not valid for DMC-30xxx
n₁	0	1	0	1	Single Ended or Differential Input	DMC-30xxx 0-5v only, 0=Single ended, 1=differential (AI2 is the complement for AI1)
	1	4	2	1	Set Analog range	DMC-31xxx, See Table Below
	-4	-1	N/A	1	Specify analog input is differential	DMC-31xxx, valid only for $n_0=1$ (AI2 is complement for AI1)

Remarks

- AQ is a configuration command which must be set at the beginning of application code.
- The usage of this command depends on the type of analog inputs present on the particular RIO model, check the ID command to determine the hardware configuration.

Configurable Analog Input Settings

Argument	Value	Description	Notes
n₁	1	-5 to +5 VDC	
	2	-10 to +10 VDC	Default
	3	0 to 5 VDC	
	4	0 to 10 VDC	

- Default resolution for analog inputs is 12bits. 16 bit is optional (31xxx).
- Operands _AQ1 and _AQ2 return the setting for the specified input.

Position Range when in Analog Feedback by AQ

Argument	Value	Analog Range	Position Range (12 bit)	Position Range (16 bit)
n₁	1	+5 to -5 VDC	-2048 to 2047	-32,768 to 32767
	2	+10 to -10 VDC	-2048 to 2047	-32,768 to 32767
	3	0 to 5 VDC	0 to 4095	0 to 65535
	4	0 to 10 VDC	0 to 4095	0 to 65535


Examples

```
'Galil DMC Code Example
:AQ 1,1;'   Sets Analog input 1 and 2 to be a differential input for the DMC-30xxx
:AQ 1,4;'   Sets Analog input 1 to 0-10v range for the DMC-31xxx
```

AQ applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC52xx0,RIO574x0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

AU Set amplifier current loop



AUm= n

AU n, n, n, n, n, n, n, n, n

Usage	AUm= n	Arguments specified with a single axis mask and an assignment (=)
	AU n ...	Arguments specified with an implicit, comma-separated order
Operands	_AUm	Operand holds the value last set by the command

Description

The AU command sets the amplifier current loop gain for internal amplifiers.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	4	1	1	Set amplifier current loop gain setting	See table below for settings

DMC-30012

Argument	Value	(24VDC Bus) Current loop setting	(48VDC Bus) Current loop setting	Notes
n	0	Minimum Current Loop Gain	Minimum Current Loop Gain	
	1	For inductance < 1mH	For inductance < 2.4mH	Default
	2	For inductance > 1mH and < 2.3mH	For inductance > 2.4mH and < 4.2mH	
	3	For inductance > 2.3mH and < 4.2mH	For inductance > 4.2mH and < 7mH	
	4	For inductance > 4.2mH	For inductance > 7mH	

Remarks

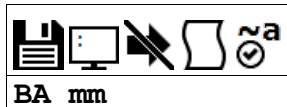
- The AU settings for Galil sine drives are only recommended values for the given bus voltages. For other bus voltages and their recommended settings, contact Galil.

Examples

```
'Galil DMC Code Example
'setting for sine amps
'BLM inductance = 2.6mH
:AU3;' Sets A-axis for standard Galil BLM motor at 24V
:MG_AUA;' Query A axis current loop gain
3
:AU2;' Sets A axis for standard Galil BLM motor at 48V
```

AU applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

BA *Brushless Axis*

BA mm

Usage	BA mm	Argument is an axis mask
Operands	_BAm	Operand has special meaning, see Remarks

Description

BA is used to configure the controller for sinusoidal operation.

The BA command must be issued to initialize the DMC-3x012, DMC-3x014, DMC-3x017, or DMC-3x012-2PB sine amplifier.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	N/A	Multi-Axis Mask	Initialize axis for internal sine amp	
	N	N	N/A	Multi-Axis Mask	Disable sine amp initialization for axs.	

Remarks

- The DMC-30017 must be set to MT1 or -1 for sine drive operation.
- _BAm will contain a 1 if the BA command has been issued for the specified axis, or a 0 if it has not.
- _BAm indicates the axis number of the auxiliary DAC used for the second phase of the selected sinusoidal axis. The axis numbers start with zero for the A axis DAC. If the motor is configured as standard servo or stepper motor, _BAm contains 0.

Examples


```
'Galil DMC Code Example
REM setup DMC-3x017-2PB for drive 2-phased brushless servo motor
MO A; ' Motor must be off to set MT
MT 1; ' Set MT 1 for servo
BA A; ' Designate sinusoidal commutation
BM 200; ' Length of electrical cycle in counts--required setting for commutation
BZ 3<1000; ' Commutate motor using 3 V and timeout after 1000 msec
SH A; ' Enable motor, ready for commands
EN
```

```
'Galil DMC Code Example
BA A; ' Configure axis A for sine amp
BM 200; ' Length of electrical cycle in counts--required setting for commutation
BZ 3<1000; ' Commutate motor with BZ method using 3V and timeout after 1000 msec
SH A; ' Enable motor, ready for commands
EN
```

BA applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BB *Brushless Phase Begins*


BBm= n
BB n, n, n, n, n, n, n, n, n

Usage	BBm= n	Arguments specified with a single axis mask and an assignment (=)
	BB n ...	Arguments specified with an implicit, comma-separated order
Operands	_BBm	Operand holds the value last set by the command

Description

The BB function describes the position offset between the Hall transition point and theta = 0, for a sinusoidally commutated motor. This is used when doing hall initialization of a sine commutated drive.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-359.98	359.98	0	1/32	Phase offset of hall sensors	

Remarks

- This command must be saved in non-volatile memory to be effective upon reset.

Examples

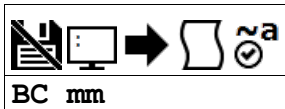
```
'Galil DMC Code Example
BB,30,,60;' The offsets for the Y and W axes are 30 and 60 respectively
```

```
'Galil DMC Code Example
BB 30;' set offset of 30 degrees for A axis
```

BB applies to DMC500x0,DMC40x0,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BC *Brushless Calibration*



BC mm

Usage	BC mm	Argument is an axis mask
Operands	_BCm	Operand has special meaning, see Remarks

Description

The BC command is used to initialize a motor for sine commutation using hall sensors.

The function BC monitors the status of the Hall sensors of a sinusoidally commutated motor, and resets the commutation phase upon detecting the first hall sensor. This procedure replaces the estimated commutation phase value with a more precise value determined by the hall sensors.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axes to initialize with hall commutation	

- BC initialization is valid with the following internal amplifiers:
 - DMC-3x012
 - DMC-3x014
 - DMC-3x017 (with MT 1 for servo operation)
 - BC is not valid with the 2PB option. See BZ.

Remarks

- The BC command is one of several ways to initialize a Galil sine drive. The table below lists the various methods:

Commutation of a Galil Sine Drive

Command	Description
BC/BI	Uses hall sensors to commutate until a hall transition is encountered. Drive then commutates sinusoidally.
BX	Uses an algorithm to determine phase angle with minimal motion.
BZ	Drives the motor to a known magnetic phase. Drive then commutates sinusoidally.

Steps for BC sine initialization

- Specify the axis/axes for initialization with the BA command
- Specify the number of encoder counts per magnetic phase of the motor with the BM command (see command for examples)
- Issue BI to select the inputs to use as hall inputs.
- Servo the motor and verify it holds position
 - If the motor will not servo, verify encoder is functional. If it is, then re-verify hall wiring
- Issue the BC command, then issue a small jog until a hall transition occurs.
- The motor is now fully commutated based off of the hall sensor feedback.
- (Optional) Use the BB command to correct for hall offsets from true magnetic 0 of the motor.

Operand Usage

- _BCm contains the state of the Hall sensor inputs. This value should be between 1 and 6. 0 and 7 are invalid hall states.

Examples


```
'Galil DMC Code Example
BAA;' Enable sine drive
BMA=2000;' Set brushless modulus to 2000 cnts
BIA=4;' Hall inputs on IN4,5,and 6
MG_BCA;' Read hall state
EN
```

```
'Galil DMC Code Example
REM Example for use with internal sine amp
#EX
BAA
BMA=2000
BIA=-1;' use hall sensor inputs on the Galil
BCA;' enable brushless calibration
bc=_BCA;' store hall state
SHA;' enable amplifier
JGA=500
BGA;' begin jog
#hall;JP#hall,_BCA=bc;'wait for a hall transition
STA
MG"Commutation Complete"
EN
```

BC applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

BD *Brushless Degrees*


BDm= n
BD n, n, n, n, n, n, n, n, n

Usage	BDm= n	Arguments specified with a single axis mask and an assignment (=)
	BD n ...	Arguments specified with an implicit, comma-separated order
Operands	_BDm	Operand has special meaning, see Remarks

Description

The BD command sets the commutation phase of a sinusoidally commutated motor manually. When using hall effect sensors, a more accurate value for this parameter can be set by using the command, BC. This command should not be used except when the user is creating a specialized phase initialization procedure.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	360	6	1/32	Brushless motor angle in degrees	

Remarks

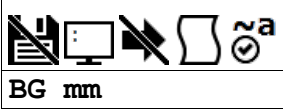
- Using BD to set a brushless degree overrides the current brushless degrees set by the BZ/BX/BI initialization routines.
- Once initialized, BD is updated by the firmware to the current brushless degree value.
- n = ? queries the current brushless degrees
- _BDm contains the commutation phase of the specified axis.

Examples

```
'Galil DMC Code Example
BDA=100;' Set Brushless degrees for A axis to 100
MG_BDA;' Report the brushless degrees for A axis
```

BD applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BG *Begin*

BG mm

Usage	BG mm	Argument is an axis mask
Operands	_BGm	Operand has special meaning, see Remarks

Description

The BG command starts a motion on the specified axis or sequence.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axes to begin motion	DMC-30000 is a single axis controller, mm=A
	S	S	N/A	Multi-Axis Mask	Vector plane to begin motion	
	N	N	N/A	Multi-Axis Mask	Virtual axis to begin motion	

Remarks

- Any combination of Axes, Vector Planes, and Virtual Axes may be mixed to begin motion
- A BG command cannot be executed for any axis in which motion has not completed
 - Slaving to a master in gearing mode is an exception. Gearing does not require the axis to profile a motion and therefore Independent moves may be superimposed on top of gearing.
- Use the AM trippoint to wait for motion complete between moves from embedded code.
- From host code, use one of the following methods to determine motion is complete
 - Poll MG_BGm
 - Use the data record (DR/QR)
 - Use interrupts (EI), if available

Operands

- _BGm contains a '0' if motion complete on the specified axis or coordinate system, otherwise contains a '1'
 - _BGm can be used from host programs to determine if motion is complete by polling the axes of interest

Examples

```
'Galil DMC Code Example
PR 2000,3000,,5000;' Set up for a relative move
BG ;' Start the A,B and D motors moving
```

```
'Galil DMC Code Example
HM ;' Set up for the homing
BG A;' Start only the A-axis moving
```

```
'Galil DMC Code Example
JG 1000,4000;' Set up for jog
BG B;' Start only the B-axis moving
```


```
'Galil DMC Code Example
bstate= _BGB;' Assign a 1 to bstate if the B-axis is performing a move
```

```
'Galil DMC Code Example
VM AB;' Vector Mode
VP 1000,2000;' Specify vector position
VS 20000;' Specify vector velocity
BG S;' Begin coordinated sequence
VP 4000,-1000;' Specify vector position
VE;' Vector End
```

BG applies to DMC40x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,DMC42x0,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

BI *Brushless Inputs*


BI m= n
BI n, n, n, n, n, n, n, n, n, n

Usage	BI m= n	Arguments specified with a single axis mask and an assignment (=)
	BI n ...	Arguments specified with an implicit, comma-separated order
Operands	_BI m	Operand holds the value last set by the command

Description

The BI command is used to define the inputs which are used when Hall sensors have been wired for sinusoidally commutated motors. See the BC command for more information about initialization of sine amplifiers via hall inputs

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-1	0	0	1	Select starting General input for hall sensor use	n = -1 uses dedicated hall inputs. n = 0 clears configuration.

Remarks

- The Hall A, Hall B and Hall C inputs on the Encoder connector may be specified by setting the BI command to -1.


Examples

```
'Galil DMC Code Example
REM Example for use with internal amp
#EX
BAA
BMA=2000
BIA=-1;' use hall sensor inputs on the Galil
BCA;' enable brushless calibration
bc=_BCA;' store hall state
JGA=500
SHA;' enable servo
BGA;' begin jog
#hall;JP#hall,_BCA=bc;'wait for a hall transition
STA
MG"Commutation Complete"
EN
```

BI applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BL Reverse Software Limit


BLm= n
BL n, n, n, n, n, n, n, n, n, n

Usage	BLm= n	Arguments specified with a single axis mask and an assignment (=)
	BL n ...	Arguments specified with an implicit, comma-separated order
Operands	_BLm	Operand holds the value last set by the command

Description

The BL command sets the reverse software limit. If this limit is exceeded during motion, motion on that axis will decelerate to a stop. Reverse motion beyond this limit is not permitted.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-2,147,483,648	2,147,483,647	-2,147,483,648	1	Position for reverse soft limit	

Remarks

- The reverse limit is activated at the position n-1. n = -2147483648 effectively disables the reverse soft limit
- The software limit is specified in counts for a servo system or in microsteps for a stepper system.
- When the reverse software limit is activated, the automatic subroutine #LIMSWI will be executed if it is included in the program.
- If motion is commanded when the axis is already passed the BL value, the axis will profile a small move before the software limit is again detected.
 - This is typically encountered when commanding motion in loops, such as a jog loop.
 - In these scenarios it is recommended to use the #LIMSWI routine to stop the loop when the BL limit has been exceeded.

Examples


```
'Galil DMC Code Example
#TEST: ' Test Program
AC 1000000: ' Acceleration Rate
DC 1000000: ' Deceleration Rate
BL -15000: ' Set Reverse Limit
JG -5000: ' Jog Reverse
BGA: ' Begin Motion
AMA: ' After Motion (limit occurred)
TPA: ' Tell Position
EN: ' End Program
'
```

'Galil Controllers also provide hardware limits.'

BL applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BM *Brushless Modulo*


BMm = n
BM n, n, n, n, n, n, n, n, n

Usage	BMm= n	Arguments specified with a single axis mask and an assignment (=)
	BM n ...	Arguments specified with an implicit, comma-separated order
Operands	_BMm	Operand holds the value last set by the command

Description

The BM command defines the length of the magnetic cycle in encoder counts.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	1	10,000,000	2,000	1/65,536	Encoder counts per magnetic cycle	

Remarks

- For rotary motors, the magnetic cycle (BM value) is calculated by:
 - BM = encoder counts per revolution / # of pole pairs
- The issuance of BM is required for commutation using one of the following methods:
 - BX
 - BZ
 - BI/BC

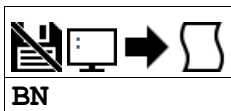
Examples

```
'Galil DMC Code Example
REM setup DMC-3x017-2PB for drive 2-phased brushless servo motor
MO A; ' Motor must be off to set MT
MT 1; ' Set MT 1 for servo
BA A; ' Designate sinusoidal commutation
BM 200; ' Length of electrical cycle in counts--required setting for commutation
BZ 3<1000; ' Commutate motor using 3 V and timeout after 1000 msec
SH A; ' Enable motor, ready for commands
EN
```

```
'Galil DMC Code Example
REM Using Galil's BLM motor specifications as an example
cts= 4000; ' 4000 encoder counts per revolution
pole= 2; ' 2 pole pairs (4 poles total)
BA A
BMA= cts/pole; ' Calculation of BM
BZA= 3.5; ' Commutate using BZ method and 3.5V
SH A
MG "Commutation complete."
EN
```

BM applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BN Burn

Usage	BN	Command takes no arguments
Operands	_BN	Operand has special meaning, see Remarks

Description

The BN command saves certain board parameters in non-volatile EEPROM memory. Once written to the memory, all parameters which can be burned will persist through a software reset (RS command), hardware reset (reset button) or power cycle. This command typically takes 1 second to execute and must not be interrupted. The controller returns a colon (:) when the Burn is complete. All parameters which have been burned into memory can be restored to their factory defaults through a master reset.

This command reference will denote commands that can and cannot be burned with BN with the following usage icons.



Burnable with BN icon



Not burnable with BN icon

Arguments

The BN command has no arguments

Remarks

- Issuing this command will pause the output of the Data Record until the command is completed.
- The following table shows the commands that have their parameters saved with the BN command:

Parameters saved during burn

AC	BO	EO	IK	MO	OT	TM
AF	BR	ER	IL	MT	OV	TR
AG	BW	FA	IT	MU	PF	VA
AQ	CB	FL	KD	NB	PL	VD
AU	CE	FV	KI	NF	PW	VF
BA	CN	GA	KP	NZ	SB	VS
BB	CW	GM	KS	OA	SM	YA
BI	DC	GR	LC	OE	SP	YB
BL	DH	HV	LD	OF	TK	YC
BM	DV	IA	LZ	OP	TL	

- Board rev C and firmware 1.1B added support for high-performance parameter and variable burning. For serial numbers under BZ-579, and for all firmware earlier than 1.1B, the following limitations apply
 - during a BN the servo will disable for a brief period, roughly 40ms. This can be prevented by first issuing an MO. This is not applicable to stepper motors.
 - BN is not valid while running and will cause a TC code of "7 Command not valid while running".

Operand Usage

- _BN contains the serial number of the processor board.

Examples


```
'Galil DMC Code Example
SB1; ' Set bit 1
CB2; ' Clear bit 2
CW1; ' Set data adjustment bit
BN; ' Burn all parameter states
```

BN applies to

DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,RIO574x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BO *Brushless Offset*


BOm= n
BO n, n, n, n, n, n, n, n, n

Usage	BOm= n	Arguments specified with a single axis mask and an assignment (=)
	BO n ...	Arguments specified with an implicit, comma-separated order
Operands	_BOm	Operand holds the value last set by the command

Description

The BO command sets a fixed offset on the command signal for sinusoidally commutated motors. This may be used to offset any bias in the amplifier, or can be used for phase initialization.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-5	5	0	20/65,536	Offset applied to DAC output in volts	

Remarks

- When using an internal Galil sine drive, each axis has two DACs (Digital to Analog Converter). BO sets the first DAC offset. BQ sets the second.


Examples

```
'Galil DMC Code Example
'Assume internal sine drive
BO 1 ;'set A axis first DAC to 1v offset
BQ 2 ;'set the A axis second DAC to 2v offset
```

BO applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BQ *Brushless Offset dual DAC*


BQm= n
BQ n, n, n, n, n, n, n, n, n

Usage	BQm= n	Arguments specified with a single axis mask and an assignment (=)
	BQ n ...	Arguments specified with an implicit, comma-separated order
Operands	_BQm	Operand holds the value last set by the command

Description

The BQ command sets a fixed offset on the command signal for sinusoidally commutated motors when using an internal Galil sine drive. This may be used to offset any bias in the amplifier, or can be used for phase initialization.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-5	5	0	20/65,536	Offset applied to DAC output in volts.	

Remarks

- When using an internal Galil sine drive, each axis has two DACs (Digital to Analog Converter). BO sets the first DAC offset. BQ sets the second.


Examples

```
'Galil DMC Code Example
'Assume internal sine drive
BO 1;' set A axis first DAC to 1v offset
BQ 2;' set the A axis second DAC to 2v offset
```

BQ applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BR *Brush Axis*


BRm= n
BR n, n, n, n, n, n, n, n, n, n

Usage	BRm= n	Arguments specified with a single axis mask and an assignment (=)
	BR n ...	Arguments specified with an implicit, comma-separated order
Operands	_BRm	Operand holds the value last set by the command

Description

The BR command configures the motor configuration and type for an axis.

The BR command is used with internal Galil amplifiers to enable which axes will be set as brush-type servos or to configure the firmware to use external drives instead of the internal channel.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	

Argument	Value	Description	Notes
n	-1	Configured for external drive	Default for DMC-30010 and DMC-30011
	0	Configured for Brushless servo	Default for DMC-30000 with internal amplifiers (ie DMC-30012)
	1	Configured for Brush-type servo	

Remarks

- If an axis has Off-On-Error(OE) set to 1, an amplifier error will occur on an axis if there are no halls and BR is set to 0. Set BR to 1 to avoid an amplifier error state.
 - The hall error bits cannot cause #AMPERR events if an axis is configured as brush-type.
- With BR1, the hall inputs are available for general use via the QH command.
- Set BR-1 to configure a DMC-30000 controller that has an internal amplifier (ie DMC-30012) to interface with an external amplifier.
- Note: If the controller has been previously configured with the BA command for sinusoidal commutation with a Galil internal amplifier, the command "BA N" must be issued prior to setting the axis to brushed mode.


Examples

```
'Galil' DMC Code Example
BR 1;'      Set to brush type, ignore hall errors
BR -1;'     Set to external amp
```

BR applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

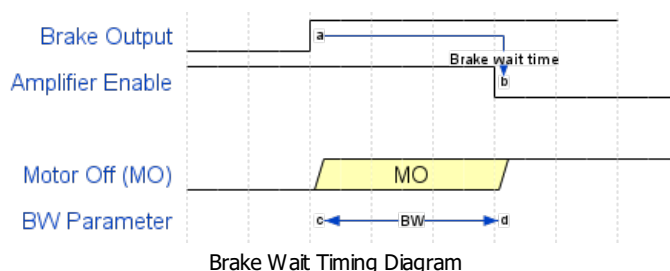
BW Brake Wait


BWm= n
BW n, n, n, n, n, n, n, n, n

Usage	BWm= n	Arguments specified with a single axis mask and an assignment (=)
	BW n ...	Arguments specified with an implicit, comma-separated order
Operands	_BWm	Operand holds the value last set by the command

Description

The BW command sets the delay between when the brake is turned on and when the amp is turned off. When the controller goes into a motor-off (MO) state, this is the time (in samples) between when the brake digital output changes state and when the amp enable digital output changes state. The brake is actuated immediately upon MO and the delay is to account for the time it takes for the brake to engage mechanically once it is energized electrically. The brake is released immediately upon SH.

Brake Wait Timing**Arguments**

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	4,096	0	1	Specify brake wait time, in samples.	0 = Turn brake function off

Remarks

- The Brake Wait does not apply when the motor is shut off due to OE1 (Off on Error). In this case (position error exceeded or Abort triggered) the motor off and brake output will be applied simultaneously.
- SB, CB and OP have no effect on outputs mapped to BW. In order to toggle brake outputs without engaging the servo (e.g. for maintenance), set BWm=0 and then use SB and CB as necessary.
- The state of the output configured as a brake cannot be queried with the @OUT[] command.
- Output 1 is the brake output.
- When using the brake outputs, it is recommended to order the controller with 500mA sourcing output option (HSRC).
- During a program download the servo will disable for a brief period (roughly 300ms) without applying the brake. This can be prevented by first issuing an MO.

Examples

```
'Galil DMC Code Example
BW 100;' Set brake delay to 100 ms (TM1000) for the A axis
```

BW applies to DMC40x0, DMC42x0, DMC41x3, DMC30010, DMC500x0, DMC52xx0, EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BX Sine Amp Initialization

BXm= n
BX n, n, n, n, n, n, n, n, n
BX< o

Usage	BXm= n	Arguments specified with a single axis mask and an assignment (=)
	BX n ...	Arguments specified with an implicit, comma-separated order
Operands	_BXm	Operand has special meaning, see Remarks

Description

The BX command uses a method to initialize an axis with limited movement of the hardware. The BX uses a limited motion algorithm to determine the proper location of the motor within the magnetic cycle. It is expected to move no greater than 10 degrees of the magnetic cycle. The last stage of the BX command will lock the motor into the nearest 15 degree increment.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-4.998	4.998	0	20/65,536	Voltage to be applied during amp initialization	-n = end BX with SH. +n = end BX with MO
o	100	5,000	1,000	1	Number of samples for BX to hold final torque pulse.	o should be set before BXm= n command.

Remarks

- _BXm contains 0 if axis m is not a Galil sine amp axis, contains 1 if axis m is an uninitialized sine amp axis, and contains 3 if axis m is an initialized sine amp axis
- An axis with a Galil sine amp powers up in MO state and SH will generate an error for that axis until it is initialized.
- While the BX command is executing, communication to and from the controller will be halted. This may result in a timeout if the BX command is sent from the host*. Embedded code execution will also pause during BX operation.
 - The long timeout (-I) for GalilTools 1.5.0 has been increased to prevent a timeout while using the BX command.
- If the BX command fails to initialize an axis, it will return an error code of 160. TC1 will return "160 BX Command Failure".
- Issuing this command will pause the output of the Data Record until the command is completed.
- BX initialization is valid with the following internal amplifiers:
 - DMC-30012
 - DMC-3x014
 - DMC-30017 (with MT 1 for servo operation)
 - BX is not valid with the 2PB option. See BZ.
- There are several methods to initialize a motor with the Galil sine amplifier. They are listed below:

Commutation of a Galil Sine Drive

Command	Description
BC/BI	Uses hall sensors to commutate until a hall transition is encountered. Drive then commutates sinusoidally.
BX	Uses an algorithm to determine phase angle with minimal motion.
BZ	Drives the motor to a known magnetic phase. Drive then commutates sinusoidally.

BX Initialization Steps

- Set axes enabled for sine amp with the BA command
- Set motor modulu with the BM command.
- Set OE1 for motor runaway.
- Issue BX1 to test at smaller voltage
 - If error code 160 occurs, try a larger voltage. If motion is occurring, then check that the encoder is working. Ensure that the timeout time is long enough for BX (BX<o). If increasing 'o' doesn't help, invert the encoder direction with CE.
- If BX is successful, issue SH. Ensure the motor holds position.
- Attempt a jog. If the motor jogs, then the initialization is complete.
 - If the motor shuts off due to position error, retry BX. Invert the encoder direction with CE if that hasn't been attempted.

Examples

```
'Galil DMC Code Example
REM Simple Example
BAA
BMA=2000
BXA=-3
#bxa;JP#bxa,_BXA<3
ENDIF
```

```
'Galil DMC Code Example
REM Detailed Example
#COM
```

```

~a=0;'0 = A axis, 1 = B axis . . .
BA~a;'enable brushless mode
BM~a=2000;'must be set per individual motor specifications
BX<1000;'set pulse duration to 1000 samples
bx_i=0;'number of tries for the BX command
#COM_H
tc=0;'response from TC command if an error occurs
MO~a;'start in motor off state
#tv;JP#tv,_TV~a>500;'make sure axis is not moving
BX~a=-3;'command the BX command
REM loop until BX passes or error occurs
#LOOP;JP#LOOP,((~BX~a<>3)&(tc=0))
REM try again if an error occurred and the number of tries < 5
JP#COM_H,((tc<>0)&(bx_i<5))
REM if the number of tries is < 5 then BX passed
REM else, try BZ command
IF (bx_i<5)
  MG "Commutation complete"
ELSE
  MG "BX failed to complete"
  MG "attempting BZ command"
  tc=0;BZ~a=-3
  IF tc=0
    MG "BZ command complete"
  ELSE
    MG "BZ command failed"
    MG "check motor and encoder wiring"
    MG "try setting CE 2 or swapping 2 motor leads"
  ENDF
ENDIF
EN


#CMDERR
tc=_TC
TC1
REM if 160 error, increase BX
IF tc=160
  MG "Retry BX"
  bx_i=bx_i+1
  BX<(bx_i*1000);'increase pulse time on failure
ENDIF
RE

```

BX applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

BZ *Brushless Zero*


BZm= n
BZ n, n, n, n, n, n, n, n, n
BZ <o
BZ >p
BZ <o>p

Usage	BZm= n	Arguments specified with a single axis mask and an assignment (=)
	BZ n ...	Arguments specified with an implicit, comma-separated order
Operands	_BZm	Operand has special meaning, see Remarks

Description

The BZ command is used to initialize axes which use sinusoidal commutation. To do this, the command drives the motor to two different magnetic positions and sets the magnetic position.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-4.998	4.998	0	20/65,536	Voltage to be applied during amp initialization	-n = end BZ with SH. +n = end BZ with MO
o	100	32,767	200	1	Time in milliseconds for BZ to hold at second magnetic position	See Remarks
p	100	32,767	100	1	Time in milliseconds for BZ to hold at first magnetic position	See Remarks

BZ Initialization Sequence:

The *n* parameter is used for setting the current during the two stage initialization process:

- During the first stage, the motor is driven to one magnetic position for *p* milliseconds and the encoder is monitored for movement.
- During the second stage, the motor is driven to a second magnetic position for *o* milliseconds and the encoder is monitored for movement.
 - The second stage may be repeated multiple times. This depends on motor motion during initialization.

Remarks

- *_BZm* contains the un-signed distance in encoder counts from the motor's current position to the position of magnetic zero for the specified axis.
 - The value is only valid after successfully initializing with BZ.
- The BZ hold time must be set prior to initialization.
- The BZ hold times can be lengthened to ensure that any oscillations introduced by the BZ command fully settle for accurate magnetic positioning.
 - The *o* and *p* parameters can be interrogated with BZ <? and BZ >? respectively.
- The BZ command may be given when the motor is off.
- The motor must be positioned so that it can freely move a distance of BM in either direction.
- It is recommended that the *n* parameter is chosen large enough to reliably drive the motor to both magnetic positions but below the continuous current rating of the motor.
 - Use the continuous current rating of the motor (I_m in A) and AG current gain (G in A/V) to determine the maximum *n* parameter. $n_{max} = (I_m)/(2 * G)$ in V.
 - A conservative starting point is $0.5 * n_{max}$ but may be increased up to n_{max} as needed.
- Always use the off-on-error function (OE command) to avoid motor runaway whenever testing sinusoidal commutation.
- The -2PB option is required on the DMC-3x017 to drive a 2-phase brushless servo motor (also known as closed-loop stepper). BZ is the only valid commutation method with this option.
- BZ Initialization is valid with the following internal amplifiers:
 - DMC-3x012
 - DMC-3x014
 - DMC-3x017
 - DMC-3x017-2PB
- Issuing this command will pause the output of the Data Record until the command is completed.
- There are several methods to initialize Galil's internal amplifiers for sinusoidal commutation. They are listed below:

Commutation options with a Galil sine drive

Command	Description	Recommended use
BC/BI	Uses hall sensors to commutate until a hall transition is encountered.	Non-linear or gravitational loads, motors must have hall sensors.
BX	Uses an algorithm to determine phase angle with minimal motion.	Sensitive instrumentation or payload requiring minimal movement.
BZ	Drive the motor to a known magnetic phase.	Easiest to use but with the most motion.

BZ Initialization Steps

1. Set axis being enabled for sine amp with the BA command.
2. Set brushless modu with the BM command.

3. Enable off-on-error (OE) to prevent motor runaway. ER should be \geq BM.
4. Set the desired hold times $BZ < o > p$.
5. Initialize using BZ command with $n \leq n_{max}$.

See user manual for details on troubleshooting BZ.

Examples

```
'Galil DMC Code Example
REM Initialize A axis for internal sine commutation.
BA A;'      Configure axis A for sine amp
BMA = 2000;' Length of electrical cycle in counts--required setting for commutation
OEA = 1;'   Enable off-on-error for A axis.
ERA= _BMA;' Set error limit to brushless modulus.
BZ <200>100;' Set first and second hold times to 100 and 200 ms respectively.
BZA = 3;'   Initialize the motor with 3V motor command.
SH A;'     Enable motor, ready for commands
BZ >200;'   Change first hold time to 200 ms
BZ <400;'   Change second hold time to 400 ms
EN
```

```
'Galil DMC Code Example
REM Initialize A axis for internal sine commutation.
BA A;'      Configure axis A for sine amp
BMA = 2000;' Length of electrical cycle in counts--required setting for commutation
OEA = 1;'   Enable off-on-error for A axis.
ERA= _BMA;' Set error limit to brushless modulus.
BZ <200;'    Set second hold time to 200 samples.
BZA = 3;'   Initialize the motor with 3V motor command.
SH A;'     Enable motor, ready for commands
EN
```

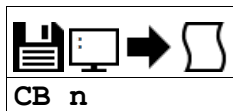
```
'Galil DMC Code Example
REM setup DMC-3x017-2PB for drive 2-phased brushless servo motor.
MO A;'      Motor must be off to set MT.
MT 1;'      Set MT 1 for servo.
BA A;'      Designate sinusoidal commutation.
BM 200;'    Length of electrical cycle in counts--required setting for commutation.
BZ <200>100;' Set first and second hold times to 100 and 200 ms respectively.
BZ 3;'      Initialize amplifier with BZ method using 3V.
SH A;'     Enable motor, ready for commands.
BZ >200;'   Change first hold time to 200 ms
BZ <400;'   Change second hold time to 400 ms
EN
```

```
'Galil DMC Code Example
:BZ 2;'     Drive A axis to electrical zero with 2V torque, and end with Motor off.
```

BZ applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

CB Clear Bit



Usage	CB n ...	Arguments specified with an implicit, comma-separated order
-------	----------	---

Description

The CB command clears a particular digital output. The SB and CB (Clear Bit) instructions can be used to control the state of output lines.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
n	1	4	N/A	1	General output bit to be cleared	
n	1,000	6,999	N/A	1	Clear Modbus slave bit	See "CB via Modbus Slave" in Remarks

Remarks

- The state of the output can be read with the @OUT[] command

Using CB with a Modbus Slave

- $n = (\text{SlaveAddress} * 10000) + (\text{HandleNum} * 1000) + ((\text{Module}-1) * 4) + (\text{Bitnum}-1)$
 - Slave Address is used when the ModBus device has slave devices connected to it and specified as Addresses 0 to 255. Please note that the use of slave devices for modbus are very rare and this number will usually be 0.
 - HandleNum is the handle specifier where A is 1, B is 2 and so on.
 - Module is the position of the module in the rack from 1 to 16.
 - BitNum is the I/O point in the module from 1 to 4

Examples

```
'Galil DMC Code Example
#main
SB 5;'      Set digital output 5
SB 1;'      Set digital output 1
CB 5;'      Clear digital output 5
CB 1;'      Clear digital output 1
EN
```

```
'Galil DMC Code Example
#modbus
REM connect to modbus slave at IP address 192.168.1.50
IHH=192,168,1,50<502>2
WT100
SB 8001;'set bit 1 on modbus slave
WT 10
CB 8003;'clear bit 3 on modbus slave
EN
```

```
'Galil DMC Code Example
:SB 18;'    Set digital output 18
:SB 21;'    Set digital output 21
:CB 18;'    Clear digital output 18
:CB 21;'    Clear digital output 21
```

For detailed information on connecting to a Modbus slave, see:


<http://www.galil.com/news/dmc-programming-io-control/setting-rio-pocket-plc-or-generic-modbus-slave-extended-io>

CB applies to

DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,RIO574x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

CE *Configure Encoder*


CEm= n
CE n, n, n, n, n, n, n, n, n

Usage	CEm= n	Arguments specified with a single axis mask and an assignment (=)
	CE n ...	Arguments specified with an implicit, comma-separated order
Operands	_CEm	Operand holds the value last set by the command

Description

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

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	15	0	1	Encoder configuration setting	n is the sum of 2 integers M and N which configure main and auxiliary encoders. See table below for configuration description.

Configure Encoder Types. Add value from Column 1 and Column 2 to make n

Column 1	Main Encoder Type	Column 2	Auxiliary Encoder Type
0	Normal quadrature	0	Normal quadrature
1	Normal pulse and direction	4	Normal pulse and direction
2	Reversed quadrature	8	Reversed quadrature
3	Reversed pulse and direction	12	Reversed pulse and direction

For example: n = 10 implies 2 + 8, thus both encoders are reversed quadrature.

Remarks

- When using a servo motor, changing the CE type can cause the motor to run away.
- When the MT command is configured for a stepper motor, the auxiliary encoder (used to count stepper pulses) will be forced to pulse and direction.
- When using pulse and direction encoders, the pulse signal is connected to CHA and the direction signal is connected to CHB.
- Axis must be in MO state prior to issuing the CE command.

Examples

```
'Galil DMC Code Example
:CE 2;'      Configure main encoder for reverse quad
:CE ?;'     Interrogate configuration
2
:V= _CEA;'  Assign configuration to a variable
:V= ?
2
```

CE applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

CN Configure



CN $n_0, n_1, n_2, n_3, n_4, n_5$

Usage	CN n ...	Arguments specified with an implicit, comma-separated order
Operands	_CN0 _CN1 _CN2 _CN3 _CN4 _CN5	Operand holds the value last set by the command

Description

The CN command configures the polarity of the limit switches, home switches, latch inputs, the selective abort function, and the program termination behavior of the abort input.

CN can also be used to map the amplifier enable signal to digital output 4.

Arguments

Argument	Value	Description	Notes
n_0	1	Limit switches active high	
	-1	Limit switches active low	Default
n_1	1	_HM is 1 when grounded (or active-opto), and 0 when pull-up (non-active opto). Affects direction of travel for HM and FE.	See HM and FE commands
	-1	_HM is 0 when grounded (or active-opto), and 1 when pull-up (non-active opto). Affects direction of travel for HM and FE.	Default
n_2	1	Latch input triggers on rising edge	
	-1	Latch input triggers on falling edge	Default
n_3	1	Configures inputs 5,6,7,8,13,14,15,16 as selective abort inputs for axes A,B,C,D,E,F,G,and H respectively.	Will also trigger #POSERR automatic subroutine if program is running.
	0	Inputs 5,6,7,8,13,14,15,16 are configured as general use inputs	Default
n_4	1	Abort input will not terminate program execution	
	0	Abort input will terminate program execution	Default
n_5	0	Digital output 4 is a general purpose output	Default
	1	Digital output 4 will operate as the amplifier enable signal	

Remarks

- n_0 is useful for testing the operation of the #LIMSWI automatic subroutine. See example below.
- When using n_5
 - For safety, ensure that the amplifier disables whenever the controller's reset button is pressed. If the amplifier enables during a reset, then a different output configuration is required. Changes must be performed at the factory. See Chapter 3 in the user manual for available output configurations.
 - The amplifier enable signal will behave the same regardless of the n_5 setting. $n_5=1$ will cause output 4 to also switch with the MO/SH state
 - When using $n_5=1$, output calls (SB,CB,OP,OB) for output 4 are ignored
 - n_5 requires that the hardware be programmed with MCB Rev 3 or higher. Rev 3 started shipping in July, 2013. See ID for the revision.

```
'Galil DMC Code Example
:ID
FW, DMC31010 Rev 1.2a
HW, DMC31016
MCB, rev 3, SPI Flash
IOB, rev 0, LSNK, SER, SINE, 16bit ADC
AMP, rev 0
:'This controller is capable of using the n5 feature
```

Examples

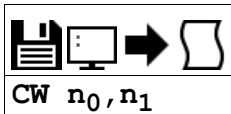
```
'Galil DMC Code Example
CN 1,1;' Sets limit and home switches to active high
CN ,, -1;' Sets input latch active low
```

```
'Galil DMC Code Example
REM n0 is useful for testing the operation of the #LIMSWI automatic subroutine
#test
CN -1;' switches are active low
JGA= 100
BG A;' Start a slow jog move
WT1000
CN 1;' Cause a limit fault by inverting the limit polarity
EN
```

```
#LIMSWI;' Automatic sub will automatically launch on limit detection
MG "Limit Switch Routine"
WT100
CN -1;' Return to correct polarity
RE
```

CN applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

CW Copyright information and Data Adjustment bit on/offCW n₀, n₁

Usage	CW n ...	Arguments specified with an implicit, comma-separated order
--------------	----------	---

Description

The CW command will return the copyright information when the argument, n, is 0 or is omitted. Otherwise, the CW command is used as a communications enhancement for use by the Galil terminal software programs. When turned on, the most significant bit of unsolicited ASCII characters is set to 1. Unsolicited ASCII characters are characters that are returned from a program running on the controller (usually from the MG command). This command does not affect solicited characters, which are characters that are returned as a response to a command sent from a host PC (e.g. TP).

Arguments

Argument	Value	Description	Notes
n₀	0	Causes controller to return a copyright information string	Equivalent to n ₀ = ?
	1	Controller will set the MSB of unsolicited message characters	
	2	Controller will not set the MSB of unsolicited message characters	Default. Must be set when viewing unsolicited messages from non-Galil software
n₁	0	Pause for unreceived data on RS232 port	If the host doesn't receive via hardware handshake within ~500ms, TC will be set to "131 Serial port hardware handshake timeout," data will be discarded, and the program will continue
	1	Throw away unreceived data on RS232 port	Default. Bytes sent to the serial port that are not serviced by the host are overwritten

Remarks

- Galil software packages automatically sends CW 1 during connection to a controller.
 - If reading unsolicited data through a non-Galil software (eg. Hyperterminal), issue CW 2

Operand Usage

- _CW contains the value set for n₀
- _CW4 contains the value set for n₁

Examples

```
'Galil DMC Code Example
CW1;' Set CW to Galil Driver mode (MSB set on unsolicited characters)

' The CW command can cause garbled (non-ASCII) characters to be returned
' by the controller when using third-party software. Use CW2.
CW2;' Set CW to third-party device mode (normal ASCII on unsolicited characters)
```

CW applies to DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,RIO47xxx,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

DC Deceleration

DCm= n

DC n, n, n, n, n, n, n, n, n

Usage	DCm= n	Arguments specified with a single axis mask and an assignment (=)
	DC n ...	Arguments specified with an implicit, comma-separated order
Operands	_DCm	Operand holds the value last set by the command

Description

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

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
	M	N	N/A	Axis	Virtual axis to assign value	
n	1,024	1,073,740,800	256,000	1,024	Deceleration rate	At TM 1000. See Remarks for resolution details.

Remarks

- The AC command is used to designate acceleration
- Specify realistic deceleration rates based on physical system parameters such as:
 - motor torque rating
 - loads
 - amplifier current rating
- Specifying an excessive deceleration will cause a large following error during deceleration and the motor will not follow the commanded profile
- DC may be changed during a move in Jog mode, but not in a PA or PR move
 - However, directly following an axis stop (ST m or a limit switch, #LIMSWI), the DC value of a PA or PR move may be changed while the axis is still decelerating

Resolution


- The resolution of the DC command is dependent on the sampling period of the control loop (TM). With the default rate of TM 1000 the resolution is 1024 counts/second². The equation to calculate the resolution of the DC command is:
 - resolution = min = 1024*(1000/TM)²
 - Example:
 - With TM 500 the minimum DC setting and resolution is 4096 counts/second².
 - resolution = 1024*(1000/500)² = 4096

Examples

```
'Galil DMC Code Example
PR 10000;' Specify position
AC 2000000;' Specify acceleration rate
DC 1000000;' Specify deceleration rate
SP 5000;' Specify slew speed
BG;' Begin motion
```

DC applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

DE Dual (Auxiliary) Encoder Position


DEm= n
DE n, n, n, n, n, n, n, n, n

Usage	DEm= n	Arguments specified with a single axis mask and an assignment (=)
	DE n ...	Arguments specified with an implicit, comma-separated order
Operands	_DEm	Operand has special meaning, see Remarks

Description

The DE command defines the position of the auxiliary (dual) encoders.

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

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-2,147,483,648	2,147,483,647	0	1	Position set for auxiliary encoders	For MT 1,-1,1.5,-1.5
	-2,147,483,648	2,147,483,647	0	1	Position set for main encoders	For MT 2,-2,2.5,-2.5

Remarks

- When using stepper motors, the DE command defines the main encoder position.
- The auxiliary encoders are not available for the stepper axis or for any axis where output compare is active.
- The operand _DEm, as well as _TDm, holds the current aux encoder position.
- n=? will return the encoder position, as returned by TD.


Examples

```
'Galil' DMC Code Example
DE 0;'      Set the current auxiliary encoder position to 0 on A axis
DE ?;'     Return auxiliary encoder positions
duala=_DEA;' Assign auxiliary encoder position of A-axis to the variable duala
```

DE applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

DP Define Position


DPm= n
DP n, n, n, n, n, n, n, n, n

Usage	DPm= n	Arguments specified with a single axis mask and an assignment (=)
	DP n ...	Arguments specified with an implicit, comma-separated order
Operands	_DPm	Operand has special meaning, see Remarks

Description

The DP command sets the current motor position and current command positions to a user specified value. The units are in quadrature counts. This command will set both the TP and RP values.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
	N	N	N/A	Axis	Virtual axis to assign value	
n	-2,147,483,648	2,147,483,647	0	1	Value assigned to motor/commanded position (RP and TD registers)	For MT 1,-1,1.5,-1.5
	-2,147,483,648	2,147,483,647	0	1	Value assigned to step/commanded position (RP and TD registers)	For MT 2,-2,2.5,-2.5

Remarks

- The DP command sets the commanded reference position for axes configured as steppers. The units are in steps.
 - Example: "DP 0" This will set the registers for TD and RP to zero, but will not effect the TP register value. When equipped with an encoder, use the DE command to set the encoder position for stepper mode.
- The DP command is useful to redefine the absolute position.
 - For example, you can manually position the motor by hand using the Motor Off command, MO. Turn the servo motors back on with SH and then use DP0 to redefine the new position as your absolute zero.
- The operand _DPm, as well as _TPm, holds the current main encoder position.
- n=? will return the encoder position, as returned by TP.


Examples

```
'Galil DMC Code Example
:DP 0;' Sets the current position of the A-axis to 0
:DP -50000;' Sets the current position of A-axis to -50000.
:DP ?;' Interrogate the position of A
-50000
```

DP applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

DV Dual Velocity (Dual Loop)



DVm= n

DV n, n, n, n, n, n, n, n, n

Usage	DVm= n	Arguments specified with a single axis mask and an assignment (=)
	DV n ...	Arguments specified with an implicit, comma-separated order
Operands	_DVm	Operand holds the value last set by the command

Description

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

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	1	0	1	State of dual loop mode	n = 0 disables Dual loop. n = 1 enables Dual loop

Remarks

- The DV command is useful in backlash and resonance compensation.
- DV must be set properly for commutation to be successful with internal sine drives.
 - When DVm=0, the firmware will use the main encoder for sine drive commutation.
 - When DVm=1, the firmware will use the aux encoder for sine drive commutation.

Correcting for Positive Feedback

- With motor off (MO) check the motor encoder with TD and load encoder with TP. Manually move the motor/load and reissue the TD and TP commands to confirm both encoders count in the same direction.
- If the encoders count in opposing directions, change the polarity of one encoder using the CE command or by changing the wiring. Consult user manual.
- If positive feedback still persists, switch the motor polarity or reverse the direction of both encoders.
 - Off on error (OE) and error limits (ER) can be used to shut down the motor in the event of a runaway.

Using DV with Large motor/load encoder ratio

- When using Dual Loop mode with a large motor:load ratio and/or running at high velocities where low position error at speed is required, FV should be used to compensate for the derivative contribution from the higher resolution motor encoder.
 - The estimated FV setting required to compensate for the derivative contribution can be calculated by the equation:
 - $FV = (KD/4) * (motor/load)$
 - motor/load = effective motor to load ratio
 - For example: KD = 200, motor encoder changes 5000 counts per 1000 counts of load encoder (motor/load = 5/1)
 - $FV = (200/4) * (5/1) = 250$


Examples

```
'Galil DMC Code Example
DV 0;' Disables DV on A axis
DV 1;' Enables dual loop on A axis.
MG_DVA;' Returns state of dual velocity mode for A axis
```

DV applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

ER Error Limit



ERm= n

ER n, n, n, n, n, n, n, n, n, n

Usage	ERm= n	Arguments specified with a single axis mask and an assignment (=)
	ER n ...	Arguments specified with an implicit, comma-separated order
Operands	_ERm	Operand holds the value last set by the command

Description

The ER command sets the magnitude of the position errors for each axis that will trigger an error condition. When the limit is exceeded, the Error output will go low (true) and the controller's red light will be turned on. If the Off On Error (OE1) command is active, the motors will be disabled.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-1	2,147,483,647	16,384	1	Set the position error limit in counts	n=0 enables Error output. n=-1 disables Error output.

Remarks

- The error limit specified by ER should be high enough as not to be reached during normal operation.
 - Examples of exceeding the error limit would be a mechanical jam, or a fault in a system component such as encoder or amplifier
- For debugging purposes, ER0 and ER-1 can be used to turn the red LED on and off.


Examples

```
'Galil DMC Code Example
:ER 200;' Set the A-axis error limit to 200
:ER ?;' Return A value
200
:V1=_ERA;' Assigns V1 value of ERA
:MG V1;' Returns V1
200
```

ER applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

FA Acceleration Feedforward


FAm= n
FA n, n, n, n, n, n, n, n, n

Usage	FAm= n	Arguments specified with a single axis mask and an assignment (=)
	FA n ...	Arguments specified with an implicit, comma-separated order
Operands	_FAm	Operand holds the value last set by the command

Description

The FA command sets the acceleration feedforward coefficient. This coefficient is scaled by the set acceleration and adds a torque bias voltage during the acceleration phase and subtracts the bias during the deceleration phase of a motion.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	8,191	0	1/4	Value of proportional term	

Remarks

- The Feedforward Bias product is limited to 10 Volts.
- If the feedforward coefficient is changed during a move, then the change will not take effect until the next move.
- FA operates on PA, PR, IP, JG and PVT mode.
- FA does not operate in:
 - Contour Mode (CM)
 - Axis is Gearing or ECAM slave
 - Coordinated motion (LM, VM)
- Acceleration Feedforward Bias = $FA * AC * (1.5 \cdot 10^{-7}) * ((TM/1000)^2)$
- Deceleration Feedforward Bias = $FA * DC * (1.5 \cdot 10^{-7}) * ((TM/1000)^2)$
- FA is enabled during the PVT mode of motion.

Examples


```
'Galil DMC Code Example
'Set feedforward coefficient to 10 the
'effective bias will be 0.75v

:AC 500000
:FA 10
:MG _FAA
10
```

FA applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

FL Forward Software Limit


FLm= n
FL n, n, n, n, n, n, n, n, n

Usage	FLm= n	Arguments specified with a single axis mask and an assignment (=)
	FL n ...	Arguments specified with an implicit, comma-separated order
Operands	_FLm	Operand has special meaning, see Remarks

Description

The FL command sets the forward software position limit. If this limit is exceeded during motion, motion on that axis will decelerate to a stop. Forward motion beyond this limit is not permitted.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-2,147,483,648	2,147,483,647	2,147,483,647	1	Value of software forward limit	2147483647 turns off forward limit

Remarks

- The forward limit is activated at n+1. n = 2147483647 effectively disables the forward soft limit.
- The software limit is specified in counts for a servo system or in microsteps for a stepper system.
- When the forward software limit is activated, the automatic subroutine #LIMSWI will be executed if it is included in the program.
- If motion is commanded when the axis is already passed the FL value, the axis will profile a small move before the software limit is again detected.
 - This is typically encountered when commanding motion in loops, such as a jog loop.
 - In these scenarios it is recommended to use the #LIMSWI routine to stop the loop when the FL limit has been exceeded.

Examples


```
'Galil DMC Code Example
#TEST;' Test Program
AC 1000000;' Acceleration Rate
DC 1000000;' Deceleration Rate
FL 15000;' Forward Limit
JG 5000;' Jog Forward
BGA;' Begin
AMA;' After Limit
RPA;' Tell Position
EN;' End

'Hint: Galil controllers also provide hardware limits.
```

FL applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

FV Velocity Feedforward



FVm= n

FV n, n, n, n, n, n, n, n, n

Usage	FVm= n	Arguments specified with a single axis mask and an assignment (=)
	FV n ...	Arguments specified with an implicit, comma-separated order
Operands	_FVm	Operand holds the value last set by the command

Description

The FV command sets the velocity feedforward coefficient. This coefficient generates an output bias signal in proportions to the sample to sample change in reference position (RP).

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	8,191.75	0	0.25	Value of proportional term	
	0	8,191	0	1	Value of proportional term	-CER firmware only.

Remarks

- FV also applies to Contour Mode (CM) and in gearing when an axis is a slave
- Velocity feedforward bias = $FV * (Velocity [cts/s]) * (1.20 \cdot 10^{-6}) * (TM/1000)$
 - With FVA=10, TM 1000 and the velocity is 200,000 count/s, the velocity feedforward bias equals 2.40 volts

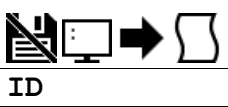
Examples

```
'Galil DMC Code Example
'Set feedforward coefficient to 10
'This effective bias will be 0.366 volts

:FV 10
:JG 30000
:MG _FVA
10
```

FV applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

ID Identify

Usage	ID	Command takes no arguments
--------------	----	----------------------------

Description

The ID command is used to query the controller for the hardware configuration and factory programming.

Arguments

ID is a command with no arguments

Remarks

- Refer to the Examples section for actual controller responses
- The following are descriptions of the ID response

CPU rev [number] [cpu_options]

ICM rev [number] [icm_options]

AMP rev [number] [amp_model]

where

[number] = Revision number for the given hardware type

[cpu_options] = Listed options for the CPU. Valid strings include:

'SPI Flash' : Shown if CPU board is revision C or newer, and firmware revision is 1.1B or newer

'Real Time Clock' : Shown if -RTU option ordered

[icm_options] = Listed options for the ICM. Valid strings include:

'Low Power Sinking Outputs' : Shown if -LSNK option ordered

'High Power Sourcing Outputs' : Shown if -HSRC option ordered

[amp_model] = Listed order for the AMP. Valid strings include:

'SINE' : Shown if DMC-30012 ordered

'STEPPER' : Shown if DMC-30017 ordered


Examples

```
'Galil DMC Code Example
:ID
FW, DMC31010 Rev 1.2e
HW, DMC31012
MCB, rev 3, SPI Flash
IOB, rev 0, LSNK, SER, SINE, 16bit ADC
AMP, rev 1
```

ID applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,RIO47xxx,DMC52xx0,RIO574x0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallmc.com

IL Integrator Limit


ILm= n
IL n, n, n, n, n, n, n, n, n, n

Usage	ILm= n	Arguments specified with a single axis mask and an assignment (=)
	IL n ...	Arguments specified with an implicit, comma-separated order
Operands	_ILm	Operand holds the value last set by the command

Description

The IL command limits the effect of the integrator gain in the filter to a certain voltage.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-9.9982	9.9982	9.9982	20/65,536	Value of Integrator limit in volts	n < 0 (negative value) freezes the effect of the integrator during the move

Remarks

- IL is the absolute value of the integrator limit. For example:
 - ILA= 2 limits the output of the integrator of the A-axis to the +/-2 Volt range.
 - KD and KP terms remain active in any case. The output from the KD and KP terms is not affected.
- A negative parameter will freeze the effect of the integrator during the move. For Example:
 - ILA= -3 limits the integrator output of the A axis to +/-3V but freezes the contribution of the Integrator loop during motion.
- If, at the start of the motion, the integrator output is 1.6 Volts, that level will be maintained through the move and the integrator will not accumulate during the move.
- Once the profiled move has completed (RP has reached final commanded position), the integrator loop will be enabled.

Examples

```
'Galil' DMC Code Example
KI 2;' Integrator constant
IL 3;' Integrator limit
```

IL applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

IT *Independent Time Constant - Smoothing Function*

ITm= n

IT n, n, n, n, n, n, n, n, n, n

Usage	ITm= n	Arguments specified with a single axis mask and an assignment (=)
	IT n ...	Arguments specified with an implicit, comma-separated order
Operands	_ITm	Operand holds the value last set by the command

Description

The IT command filters the acceleration and deceleration functions of independent moves such as JG, PR, PA to produce a smooth velocity profile. The resulting profile, known as smoothing, has continuous acceleration and results in reduced mechanical vibrations. IT sets the bandwidth of the filter where 1 means no filtering and 0.004 means maximum filtering.

The IT command also filters the individual axes during Vector Mode (VM) and Linear Interpolation Mode (LM).

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0.004	1	1	1/256	Value of independent smoothing function	1 = no filtering, 0.004 = maximum filtering

Remarks

- The IT filtering results in longer motion time.
- The use of IT will not effect the trippoints AR and AD.
 - The trippoints AR & AD monitor the profile prior to the IT filter and therefore can be satisfied before the actual distance has been reached if IT is NOT 1.
- Details on the IT filtering can be found in Application Note #3412
 - [<http://www.galil.com/learn/application-notes>]

Examples

```
'Galil DMC Code Example
:IT 0.8, 0.6, 0.9, 0.1;' set independent time constants for a,b,c,d axes
:IT ?;' Return independent time constant for A-axis
0.8000
```

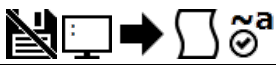
```
'Galil DMC Code Example
REM example showing increased time due to IT filtering
#move
IT 1
t=TIME;'store time reference
PR 1000
BGA;AMA
MG TIME-t;'display move time
IT 0.01
t=TIME;'store time reference
PR 1000
BGA;AMA
MG TIME-t;'display move time
EN

:'program execution output
:xQ
:
508.0000
1112.0000
```

IT applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

JG Jog


JGm= n
JG n, n, n, n, n, n, n, n, n

Usage	JGm= n	Arguments specified with a single axis mask and an assignment (=)
	JG n ...	Arguments specified with an implicit, comma-separated order
Operands	_JGm	Operand has special meaning, see Remarks

Description

The JG command sets the jog mode and the jog slew speed of the axes.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
	N	N	N/A	Axis	Virtual axis to assign value	
n	-15,000,000	15,000,000	25,000	2	Value of jog speed in cnts/second	For MT settings of 1,-1,1.5 and -1.5 (Servos)
	-3,000,000	3,000,000	25,000	2	Value of jog speed in cnts/second	For MT settings of 2,-2,2.5 and -2.5 (Steppers)
	-50,000,000	50,000,000	25,000	2	Value of jog speed in cnts/second	DMC-31000 set to AF>=5

Remarks

- When jogging, the motion controller profiles a continuous move at the commanded speed.
- To stop the motion, use the ST command.
- JG 2 is the minimum non-zero speed
- _JGm contains the absolute value of the jog speed for the specified axis.
- The JG command will set the SP register with the absolute value of the 'n' value.

Resolution

- The resolution of the JG command is dependent upon the update rate setting (TM).
 - With the default rate of TM 1000 the resolution is 2 cnts/second.
 - The equation to calculate the resolution of the JG command is:
 - resolution = 2*(1000/TM)
 - example:
 - With TM 250 the resolution of the JG command is 8 cnts/second
 - resolution = 2*(1000/250) = 8


Examples

```
'Galil DMC Code Example
#jg
REM Sets for jog mode with a slew speed of 100 counts/sec
JG 100
BG;' Begin Motion
WT 1000;' wait one second
JG -2000;' change to slew in the negative direction at -2000 counts/sec
EN
```

JG applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

KD *Derivative Constant*


KDm= n
KD n, n, n, n, n, n, n, n, n

Usage	KDm= n	Arguments specified with a single axis mask and an assignment (=)
	KD n ...	Arguments specified with an implicit, comma-separated order
Operands	_KDm	Operand holds the value last set by the command

Description

KD designates the derivative constant in the control filter. The derivative gain outputs a voltage based on the rate of change of the error. The filter transfer function follows:

$$D(z) = KP + KD \frac{z-1}{z} + KI \frac{z}{z-1}$$

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	4,095.875	64	1/8	Value of derivative term	

Remarks

- n=? will return the currently set value of KD
- m=* will set the KD value for all axes/channels
- For further details see the section "Theory of Operation" in the controller user manual.

Examples


```
'Galil DMC Code Example
:KDA= 8;'           Explicit notation to set A
:KD 8;'            Implicit notation to set A
:KD ?;'           Returns A value
8
:KDA= ?;'         Return A value
8
:MG _KDA;'        Message the operand for the A axis
8
```

```
'Galil DMC Code Example
REM Zeroing the PID filter allows the
REM motor command signal to be
REM used as a programmable DAC
KI*= 0;'          Zero KI
KP*= 0;'          Zero KP
KD*= 0;'          Zero KD
ER -1;'          Turn off position error limit
OF 1;'           Set one volt on A-axis
EN
```

KD applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

KI Integrator


KIm= n
KI n, n, n, n, n, n, n, n

Usage	KIm= n	Arguments specified with a single axis mask and an assignment (=)
	KI n ...	Arguments specified with an implicit, comma-separated order
Operands	_KIm	Operand holds the value last set by the command

Description

The KI command sets the integral gain of the control loop. The integrator term will reduce the position error at rest to zero. It fits in the control equation as follows:

$$D(z) = KP + KD \frac{z-1}{z} + KI \frac{z}{z-1}$$

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	255.999	0	1/1,024	Value of integral term	

Remarks

- n=? will return the currently set value of KD
- m=* will set the KD value for all axes/channels

Examples


```
'Galil DMC Code Example
:KIA= 8;'           Explicit notation to set A
:KI 8;'           Implicit notation to set A
:KI ?;'           Returns A value
8
:KIA= ?;'         Return A value
8
:MG _KIA;'        Message the operand for the A axis
8
```

```
'Galil DMC Code Example
REM Zeroing the PID filter allows the
REM motor command signal to be
REM used as a programmable DAC
KI*= 0;'          Zero KI
KP*= 0;'          Zero KP
KD*= 0;'          Zero KD
OF 1;'           Set one volt on A-axis
EN
```

KI applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,CLS,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

KP *Proportional Constant*


KPm= n
KP n, n, n, n, n, n, n, n, n, n

Usage	KPm= n	Arguments specified with a single axis mask and an assignment (=)
	KP n ...	Arguments specified with an implicit, comma-separated order
Operands	_KPm	Operand holds the value last set by the command

Description

KP designates the proportional constant in the controller filter. The proportional gain outputs a control signal proportional to the amount of error. The filter transfer function follows.

$$D(z) = KP + KD \frac{z-1}{z} + KI \frac{z}{z-1}$$

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	1,023.875	6	1/8	Value of proportional term	

Remarks

- n=? will return the currently set value of KP
- For further details see the section "Theory of Operation" in the controller user manual.

Examples


```
'Galil DMC Code Example
:KP 12;'      Implicit notation
:KPA= 8;'    Explicit notation
:KP ?;'      Return value
8
:MG _KPA;'   Message the operand for the A axis
8
```

```
'Galil DMC Code Example
REM Zeroing the PID filter allows the
REM motor command signal to be
REM used as a programmable DAC
KI*= 0;'     Zero KI
KP*= 0;'     Zero KP
KD*= 0;'     Zero KD
OF 1,2;'     Set one volt on A and two volts on B
EN
```

KP applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,CLS,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

LC Low Current Stepper Mode



LCm= n

LC n, n, n, n, n, n, n, n, n

Usage	LCm= n	Arguments specified with a single axis mask and an assignment (=)
	LC n ...	Arguments specified with an implicit, comma-separated order
Operands	_LCm	Operand holds the value last set by the command

Description

The LC command enables low current mode for stepper motors. Low current mode reduces the holding torque of the stepper motors while at rest.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	1	0	1	n=0, Stepper drive provides 100% torque at rest; n >= 1, 25% holding torque when motor at rest.	
	2	32,767	0	1	Specifies "n" samples after move before going to 25% holding current	

Remarks

- Using LC with an internal Galil Stepper drive (SDM)
 - Using LC will reduce current consumption, but there will be a reduction of holding torque at rest
 - Consult the user manual for more details regarding your specific amplifier
- Using LC with external amplifiers
 - When using external amplifiers low current mode will simply disable the motors by toggling the amplifier enable line during rest
 - Using LC will reduce current consumption, but there will be no holding torque at rest


Examples

```
'Galil DMC Code Example
#ex
MTA=-2;'specify stepper mode for A axis
LCA=15;'specify motor to go to low current
' 15 samples after motion has completed
EN
```

LC applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

LD *Limit Disable*


LDm= n
LD n, n, n, n, n, n, n, n, n

Usage	LDm= n	Arguments specified with a single axis mask and an assignment (=)
	LD n ...	Arguments specified with an implicit, comma-separated order
Operands	_LDm	Operand holds the value last set by the command

Description

Allows user to disables forward and/or reverse limit switches.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	4	0	1	Sets limit disable state	See table below for details

Argument	Value	Description	Notes
n	0	Both limit switches are enabled	Default
	1	Forward limit switch disabled	
	2	Reverse limit switch disabled	
	3	Both limit switches disabled	

Remarks

- n = ? will return the current setting of LD
- When this feature should be used:
 - To gain additional digital inputs if limit switches are not being utilized.
 - To prevent noise from causing the limit switches conditions even though no limit switches are connected.
- LD does not disable software limits set by BL and FL.

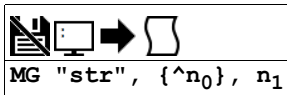
Examples

```
'Galil DMC Code Example
REM use forward limit switch as an extra I/O point
#io
LDA=1;'disable forward limit switch
io=_LFA;'set state of limit switch to variable "io"
'Use "io" in an IF statement
IF io=1
  MG "Input On"
ELSE
  MG "Input Off"
ENDIF
EN
```

LD applies to DMC40x0,DMC42x0,DMC41x3,DMC18x6,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

MG Message



MG "str", {^n₀}, n₁

Usage MG n ... Arguments specified with an implicit, comma-separated order

Description

The MG command is used to send strings, operands, variables, and array values to a specified destination.

Arguments

Argument	Value	Description	Notes
str	String	A string including alphanumeric characters to be displayed	Limited to 76 characters
n ₀	ASCII character in decimal	Allows users to print ASCII characters	Range of 0-255
n ₁	Numeric value	Prints the numeric value specified	See Examples for valid uses of n ₁ .
	Variable name	Prints the numeric value stored by the variable	
	Operand	Prints the numeric value stored by the operand	
	Array element	Prints the numeric value stored by the array element	
	Mathematical expression	Prints the numeric value of the solved equation	

Argument	Value	Description	Notes
n	Operand	Prints the numeric value stored by the operand	

Remarks

- Multiple strings, variables, and ASCII characters may be used; each must be separated by a comma.
- Solicited Messages
 - From a host terminal, application code, or device, sending the MG command will return with the requested information. This is known as a solicited command, because the host sends the command and expects a response.
- Unsolicited Messages
 - From embedded DMC code, the MG command will send an unsolicited, asynchronous message from the controller to the host. This can be used to alert an operator, send instructions, or return a variable value. This is known as an unsolicited command because the host is not explicitly requesting it.
 - The CW command controls the ASCII format of all unsolicited messages.
 - Unsolicited messages can go to any of the Ethernet handles or serial ports.
 - The CF command sets the default communication port for routing unsolicited messages.

Formatting

- Formatters can be placed after each argument in to modify how it is printed.
 - {Fm.n} Display variable in decimal format with m digits to left of decimal and n to the right.
 - {Zm.n} Same as {Fm.n} but suppresses leading zeros.
 - {\$m.n} Display variable in hexadecimal format with m digits to left of decimal and n to the right.
 - {Sn} Display variable as a string of length n, where n is 1 through 6. If n is greater than the length of the string stored in the variable, null chars (0x00) will be inserted at the end of the string.
 - {N} Suppress carriage return at the end of the message.

Message Routing

MG can override the default CF setting by using the following modifiers at the beginning of the message, right after MG.

- {Pn} Sends the message out the Serial port n, where n is 1 or 2 denoting Main or Auxiliary (where equipped).
- {Ex} Sends the message out the Ethernet handle x, where x is A,B,C,D,E, or F

Examples

Valid uses of n₁ argument

```

Galil DMC Code Example
:Values
:MG 1234.5678
1234.5678
:
:Variables
:var= 12345678.9101
:MG var
12345678.9101
:
:Operands
:MG @AN[1]
0.0121
:
:Array Elements
:DM arr[3]
:arr[0]=0
:arr[1]=1
:arr[2]=2
:MG arr[0],arr[1],arr[2]
0.0000 1.0000 2.0000
:
:Mathematical Expressions
:MG 1+2
3.0000
:MG arr[2]+var
12345680.9101
    
```

General Use

```

Galil DMC Code Example
:MG "Good Morning";           'Message command displays ASCII string
Good Morning
:total= 1234.5322;           'Assigns variable total with the value 1234.5322
:MG "The answer is...",total{F4.2}; 'Will print the message and the value of variable total formatted with 4 integer digits and 2 fractional digits
The answer is... 1234.53
:MG {\A13}, {\A10}, {\A48}, {\A055}; 'Specifies carriage return, line feed, and the characters 0 and 7 in ASCII decimal values

07
:MG TIME;                   'Messages the operand TIME
261928200.0000
:variable= 10;              'Sets the variable equal to 10
:MG variable+5;             'Messages out variable + 5
    
```

```
15.0000
:MG _TI0;
255.0000
```

'Messages the value stored in the operand _TI0

'Galil DMC Code Example

```
CF A; 'Messages configured to go out Ethernet handle A
```

```
MG {EB}var; 'Override CF and send the value of variable var to B handle
```

MG applies to

DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,RIO574x0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

MO *Motor Off*

MO mm

Usage	MO mm	Argument is an axis mask
Operands	_MOm	Operand has special meaning, see Remarks

Description

The MO command turns off the motor command line and toggles the amplifier enable signal.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Specifies axis to turn off	

Remarks

- The controller will continue to monitor the motor position
 - See the TP command for more details
- To turn the motor back on use the SH (Servo Here) command.
- The MO command is useful for positioning the motors by hand.
- _MOm contains 1.000 if the axis is in the motor off state or 0.000 if the axes is in the servo here state.


Examples

```
'Galil DMC Code Example
MO A;      'Turns off motor A
SH A;      'Turns motor A on
axis= _MOA; 'Sets variable axis equal to the motor servo status
```

MO applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

MT *Motor Type*


MTm= n
MT n, n, n, n, n, n, n, n, n

Usage	MTm= n	Arguments specified with a single axis mask and an assignment (=)
	MT n ...	Arguments specified with an implicit, comma-separated order
Operands	_MTm	Operand holds the value last set by the command

Description

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

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	

Argument	Value	Description	Notes
n	1	Servo motor	Default for DMC-3x010/11/12 models. Special use for 2PB option, see Remarks.
	-1	Servo motor with reversed polarity	Setting invalid for Galil sine drives
	1.5	PWM/Sign servo drive	
	-1.5	PWM/Sign servo drive with reversed polarity	
	2	Step motor with active low step pulses	
	-2	Step motor with active high step pulses	Default setting for DMC-3x017/16
	2.5	Step motor with reversed direction and active low step pulses	
	-2.5	Step motor with reversed direction and active high step pulses	Valid setting for DMC-3x017/16

Remarks

- n = ? will return the value of the motor type for the specified axis.
- For step and direction modes (n=2,-2,2.5,-2.5), the auxiliary encoder input for the axis is no longer available.
- Axis must be in MO state prior to issuing the MT command.
- For interfacing to external drives, BR -1 must be set on the DMC-30000 controller. This is the default setting for any DMC-30000 that does NOT contain internal Galil amplifiers (ex DMC-30010, DMC-30011, DMC-31010 ...).
- MT 1 must be set on the DMC-3x017-2PB in order to drive a 2-phased brushless servo motor (also known as a closed-loop stepper) using the internal amplifier.

Examples

```
'Galil DMC Code Example
REM setup DMC-30000 for external stepper control.
REM step and direction located on Multi-Function (MF) Pins, see User Manual
#a
BR -1
MT -2
EN


REM setup DMC-30000 with for external servo control
REM motor command output located on Analog Output 1, see User Manual
#b
BR -1
MT 1
EN

REM setup DMC-3x017-2PB for drive 2-phased brushless servo motor
MO A; ' Motor must be off to set MT
MT 1; ' Set MT 1 for servo
BA A; ' Designate sinusoidal commutation
BM 200; ' Length of electrical cycle in counts--required setting for commutation
BZ 3<1000; ' Commutate motor using 3 V and timeout after 1000 msec
SH A; ' Enable motor, ready for commands
EN
```

MT applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

NB Notch Bandwidth


NBm= n
NB n, n, n, n, n, n, n, n, n

Usage	NBm= n	Arguments specified with a single axis mask and an assignment (=)
	NB n ...	Arguments specified with an implicit, comma-separated order
Operands	_NBm	Operand holds the value last set by the command

Description

The NB command sets real part of the notch poles. In other words, the NB controls the range of frequencies that will be attenuated.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	62.5	0.5	1/2	Value of the notch bandwidth in Hz	Max value dependent upon TM setting, see Remarks

Remarks

- _NBm contains the value of the notch bandwidth for the specified axis.
- NB also determines the ratio of NB/NZ which controls the attenuation, or depth, of the notch. See NZ for more details.
- See the NF command for recommendations on choosing NZ, NB, and NF values.
- See Application note #2431 for additional information on setting the NF, NB and NZ commands
 - <http://www.galil.com/learn/application-notes>

Maximum Range

- The maximum n argument is specified in Hz and is calculated by the equation below:

$$\frac{1}{(16 \times TM \times 10^{-6})}$$

- where TM is specified in microseconds.
- The default TM is 1000, therefore default maximum NB value = 1/(16x1000E-6) = 62.5 Hz


Examples

```
'Galil DMC Code Example
NBA= 10;           'Sets the real part of the notch pole to 10/2 Hz
notch = _NBA;     'Sets the variable "notch" equal to the notch bandwidth value for the A axis
```

NB applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

NF Notch Frequency


NFm= n
NF n, n, n, n, n, n, n, n, n

Usage	NFm= n	Arguments specified with a single axis mask and an assignment (=)
	NF n ...	Arguments specified with an implicit, comma-separated order
Operands	_NFm	Operand holds the value last set by the command

Description

The NF command sets the frequency of the notch filter, which is placed in series with the PID compensation.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	250	0	1	Sets the frequency of the notch filter	Max value dependent upon TM setting, see Remarks

Remarks

- n = 0 disables the notch.
- _NFm contains the value of notch filter for the specified axis.
- n = ? Returns the value of the Notch filter for the specified axis.
- See Application note #2431 for additional information on setting the NF, NB and NZ commands
 - <http://www.galil.com/learn/application-notes>

Choosing NF, NB, and NZ

1. A simple way for attaining NF, NB, and NZ parameters is to follow these simple rules:
 1. Estimate the resonance frequency (GalilTools Scope with cursors or Galil's FAS software)
 2. Set NF equal to the resonance frequency
 3. Set NB = 1/2NF
 4. Set NZ between 0 and 5
2. The ratio of NB/NF is extremely important. See the NB command for more details.

Maximum Range

- The maximum n argument is specified in Hz and is calculated by the equation below:

$$\frac{1 \times 10^6}{(4 \times TM)}$$

- Where TM is in microseconds.
 - Default TM is 1000, therefore default maximum value = 1E6/(4*1000) = 250 Hz


Examples

```
'Galil DMC Code Example
NF, 20;' Sets the notch frequency of B axis to 20 Hz
```

NF applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

NZ Notch Zero


NZm= n
NZ n, n, n, n, n, n, n, n, n, n

Usage	NZm= n	Arguments specified with a single axis mask and an assignment (=)
	NZ n ...	Arguments specified with an implicit, comma-separated order
Operands	_NZm	Operand holds the value last set by the command

Description

The NZ command sets the real part of the notch zero. In other words, the NB/NZ ratio controls the amount of attenuation, or depth, of the notch filter.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0.5	62.5	0	0.5	Value of Notch Frequency in Hz	Max value dependent upon TM setting, see Remarks

Remarks

- See the NF command for recommendations on choosing NZ, NB, and NF values.
- The maximum n argument is determined by the following equation

$$\frac{1}{(16 \times TM \times 10^{-6})}$$

- Where TM is in microseconds, the default TM is 1000.
- See Application note #2431 for additional information on setting the NF, NB and NZ commands
 - <http://www.galil.com/learn/application-notes>

The NB/NZ Ratio

- The ratio, NB/NZ controls the amount of attenuation, or depth of the notch.
 - The larger the ratio of NB/NZ, the larger the attenuation, and vice versa.
- If NB/NZ > 1 the signal will amplify the output signal causing a resonance.
- NB = NZ essentially eliminates the notch


Examples

```
'Galil DMC Code Example
NZA = 10;' Sets the real part of the notch pole to 10/2 Hz
```

NZ applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

OA Off on encoder failure


OAm= n
OA n, n, n, n, n, n, n, n, n, n

Usage	OAm= n	Arguments specified with a single axis mask and an assignment (=)
	OA n ...	Arguments specified with an implicit, comma-separated order
Operands	_OAm	Operand holds the value last set by the command

Description

The OA command turns on or off encoder failure detection. The controller can detect a failure on either or both channels of the encoder. This is accomplished by checking on whether motion of less than 4 counts is detected whenever the torque exceeds a preset level (OV) for a specified time (OT).

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	1	0	1	Status of encoder failure detection	1 = enabled, 0 = disabled

Remarks

- The OA command works like the OE command: if OA is set to 1 and an encoder failure occurs, the axis goes into the motor off (MO) state and the stop code (SC) is set to 12 if detected during motion.
- The encoder failure detection will shut the motor off regardless of profiling status, but the stop code is not updated unless the axis is executing a profiled move at the time of the detection of the encoder failure.
- If included in the application program and OA is set to 1, #POSERR will run when an encoder failure is detected for the axis.
 - Note that for this function to work properly it is recommended to have a non-zero value for KL.

Examples

```
'Galil DMC Code Example
OAA= 1;' enable A axis encoder error detection
MG_OAA;'query OA value for A axis
```

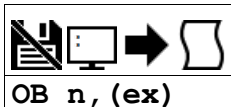
```
'Galil DMC Code Example
#setup
'setup the encoder error detection
OTA=10;' Set time to 10 milliseconds
OVA=5;' Set voltage to 5
OAA=1;' Enable encoder detection feature
EN
```

```
'Galil DMC Code Example
REM #POSERR example for checking to see if encoder failure occurred
REM This procedure is needed because the stop code will only update if
REM the profiler is running at the time the encoder failure is detected.
#POSERR
~a=0
#loop
IF _MC~a=1
IF ((_TE~a<_ER~a)&(_OE~a)&(_OA~a))
MG "possible encoder failure on ",~a{z1.0}," axis"
ENDIF
ENDIF
~a=~a+1
JP#loop,~a<_BV
AI1;' wait for input 1 to go high
SH;' enable all axes
RE
```

OA applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC18x6,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

OB Output Bit



OB n, (ex)

Usage	OB n ...	Arguments specified with an implicit, comma-separated order
-------	----------	---

Description

The OB command allows variable control of an output bit based on logical expressions. The OB n, logical expression command defines output bit i as either 0 or 1 depending on the result from the logical expression.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
n	1	4	0	1	Output bit specified	
n	1,000	6,999	N/A	1	Modbus output bit specified	See Remarks
ex	N/A	N/A	N/A	Expression	Expression that defines status of output	If ex is true/non-zero, set output to 1. If ex is false/zero, set output to 0

Remarks

- An expression is any valid logical expression, variable or array element.
- Any non-zero value of the expression results in a one set to the output bit.

Using OB with a Modbus Slave

- $n = (\text{SlaveAddress} * 10000) + (\text{HandleNum} * 1000) + ((\text{Module}-1) * 4) + (\text{Bitnum}-1)$
 - Slave Address is used when the ModBus device has slave devices connected to it and specified as Addresses 0 to 255. Please note that the use of slave devices for modbus are very rare and this number will usually be 0.
 - HandleNum is the handle specifier where A is 1, B is 2 and so on.
 - Module is the position of the module in the rack from 1 to 16.
 - BitNum is the I/O point in the module from 1 to 4

Examples

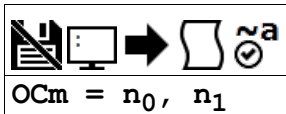
```
'Galil DMC Code Example
OB 1, pos;'          If pos<>0, Bit 1 is high.
                    If pos=0, Bit 1 is low
OB 2, @IN[1]&@IN[2];' If Input 1 and Input 2 are both high, then
                    output 2 is set high
OB 3, count[1];'   If the element 1 in the array is zero, clear bit 3
OB n, count[1];'   If element 1 in the array is zero, clear bit n
```

```
'Galil DMC Code Example
'Toggle digital output 1
OB1,@COM[@OUT[1]] & 1;' read current state of output 1, take the bitwise complement, mask out bits.
```

OB applies to DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,RIO47xxx,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

OC Output Compare



Usage	OC _m = n	Arguments specified with a single axis mask and an assignment (=)
Operands	_OC	Operand has special meaning, see Remarks

Description

The OC command sets up the Output Compare feature, also known as Pulse on Position. The controller has a special digital output which can be configured to pulse on a specified absolute encoder position, and optionally on a delta encoder change after that. These operations are known as one-shot and circular compare, respectively.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to enable output compare	Axes A-D share one output compare, axes E-H share a second output compare output
n₀	-2,147,483,648	2,147,483,647	N/A	1	Absolute encoder position of first pulse	n ₀ must be within 65535 counts of current position
n₁	-65,536	65,535	N/A	1	Incremental encoder distance between pulses	0 indicates single-shot pulse in positive direction, -65536 indicates single shot when moving in the negative direction

Remarks

- For controllers with 5-8 axes, two output compares are available. One for the A-D axes, the other for the E-H axes
- This command is only valid when both n₀ and n₁ are specified.

One shot Compare Mode:

- The output compare signal will go low, and stay low at a specified absolute encoder position.
- This is done by specifying n₁ as 0 for positive motion, and -65536 for negative motion

Circular Compare Mode:

- After the absolute position of the first pulse (n₀), the circular compare can be configured to pulse low at a relative distance thereafter (n₁).
- This is done by specifying n₁ to a non-zero delta position (range of -65535 to 65535)
 - OCA = 0 will disable the Circular Compare function
- The circular compare output is a low-going pulse with a duration of approximately 510 nanoseconds.

Limitations

- The Output Compare function is only valid with incremental encoders.
 - The Output Compare function is not valid with SIN/COS (AF settings of 5-12), standard analog (AF setting of 1), BiSS or SSI feedback (SS or SI commands).
- The OC function cannot work when configured as a stepper.
- The auxiliary encoder input is non-functional during this mode.
 - Dual loop mode (which uses the aux encoder input) will not operate when the OC command is enabled.
- The OC function requires that the main encoder and auxiliary encoders be configured exactly the same (see the command, CE). For example: CE 0, CE 5, CE 10, CE 15.
- OC only requires an encoder, and is independent of axis tuning, and motion profiling.

Operand Usage

- _OC contains the state of the OC function.
 - _OC = 0 : OC function has been enabled but not generated any pulses.
 - _OC = 1: OC function not enabled or has generated the first output pulse.

Examples

```
'Galil DMC Code Example
OCA=300,100;' Select A encoder as position sensor.
REM First pulse at 300. Following pulses at 400, 500, 600 ...
```


```
'Galil DMC Code Example
REM Output compare can be used to create raster scans.
REM By using circular compare on one axis, followed by an index move on a perpendicular axis
REM raster patterns are easily made.
REM The following image shows a rastered "dot matrix" type image easily created
REM with output compare and a laser on a two dimensional stage.
```



OC applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

OE Off-on-Error


OE m= n
OE n, n, n, n, n, n, n, n, n

Usage	OE m= n	Arguments specified with a single axis mask and an assignment (=)
	OE n ...	Arguments specified with an implicit, comma-separated order
Operands	_OE m	Operand holds the value last set by the command

Description

The OE command sets the Off On Error function for the controller. The OE command causes the controller to shut off the motor command if a position error exceeds the limit specified by the ER command, an abort occurs from either the abort input or on AB command, or an amplifier error occurs based on the description of the TA command.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	0	0	0	Disables the Off On Error Function	Default
	1	1	0	0	Motor shut off by position error, amplifier error or abort input	
	2	2	0	0	Motor shut off by hardware limit switch	
	3	3	0	0	Motor shut off by position error, amplifier error, abort input or by hardware limit switch	

Remarks

- For any value of OE <> 0, the axis will be shut off due to amplifier faults on any amplifier axis. See the TA command for conditions of an amplifier fault.
- BR1 must be enabled when internal brushless servo amplifiers are installed but the axis is driven with an external amplifier. BR1 disables hall error checking when OE <> 0

Examples

```
'Galil DMC Code Example
:OE 1;' Enable error checks for abort, pos err
:OE 3;' Enable error checks for limit switches also
:MG _OE A;' Query A axis OE setting
3.0000
```

```
'Galil DMC Code Example
#main
'code to enable the OE command for all error conditions
'and setup the corresponding automatic subroutines
'to display relevent data
'no loop for abort input, as that stops code operation
OE 3
SHABCD
JGA=1000;' jog at 1000
BG A
#loop
'endless loop
WT1000
JP#loop
EN

#AMPERR
MG "amplifier fault"
MG _TA0,_TA1,_TA2,_TA3
EN


#POSERR
MG "position error fault"
MG _TEA
EN

#LIMSWI
MG "limit switch fault"
MG _TSA
EN
```

OE applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

OF Offset


OFm= n
OF n, n, n, n, n, n, n, n

Usage	OFm= n	Arguments specified with a single axis mask and an assignment (=)
	OF n ...	Arguments specified with an implicit, comma-separated order
Operands	_OFm	Operand holds the value last set by the command

Description

The OF command sets a bias voltage in the command output or returns a previously set value.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-9.9982	9.9982	0	20/65,536	Offset voltage applied to MCMD	

Remarks

- This can be used to counteract gravity or an offset in an amplifier.

Examples

```
'Galil DMC Code Example
:OFA= 1;' Set A-axis offset to 1
:OF -3;' Set A-axis offset to -3
:OF ?;' Return A offset
-3.0000
```

OF applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

OP Output Port



OP n

Usage	OP n ...	Arguments specified with an implicit, comma-separated order
Operands	_OP0	Operand holds the value last set by the command

Description

The OP command sets the output ports of the controller in a bank using bitmasks. Arguments to the OP command are bit patterns (decimal or hex) to set entire banks (bytes) of digital outputs. Use SB, CB or OB to set bits individually.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
n	0	15	0	1	Decimal representation: General Outputs 1-4	

Remarks

- Bit patterns for extended I/O banks (where available) configured as inputs have no affect on the IO status.

Output Mapping Examples

Examples	Command issued (Hex version)	Bits Set	Bits Cleared
Set bank high	OP15 (OP\$0F)	1-4	-
Set bank low	OP0 (OP\$00)	-	1-4

Examples


```
'Galil DMC Code Example
OP 0;' Clear Output Port -- all bits
OP $05;' Set outputs 1,3 and clear the others
MG _OP0;' Returns the parameter "n0"
```

OP applies to

DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,RIO574x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

OT *Off on encoder failure time*



OTm= n

OT n, n, n, n, n, n, n, n, n, n

Usage	OTm= n	Arguments specified with a single axis mask and an assignment (=)
	OT n ...	Arguments specified with an implicit, comma-separated order
Operands	_OTm	Operand holds the value last set by the command

Description

The OT command sets the timeout time for the encoder failure routine. The command sets the time in samples that the encoder failure will wait for motion after the OV threshold has been exceeded. The controller can detect a failure on either or both channels of the encoder.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	1	32,000	30	1	Number of samples for error detection	

Remarks

- Encoder error detection is based on whether motion of at least 4 counts is detected whenever the torque exceeds a preset level (OV) for a specified time (OT).
 - Note that for this function to work properly it is necessary to have a non-zero value for KI.
- See the OA command for more details on this error detection mode

Examples


```
'Galil DMC Code Example
#setup
OTA= 10;' Set time to 10 milliseconds
OVA= 5;' Set voltage to 5
OAA= 1;' Enable encoder detection feature
EN
```

```
'Galil DMC Code Example
REM #POSERR example for checking to see if encoder failure occurred
REM This procedure is needed because the stop code will only update if
REM the profiler is running at the time the encoder failure is detected.
#POSERR
~a=0
#loop
IF !_MC~a=1
IF ((_TE~a<_ER~a)&(_OE~a)&(_OA~a))
MG "possible encoder failure on ",~a{z1.0}," axis"
ENDIF
ENDIF
~a=~a+1
JP#loop,~a<_BV
AI1;' wait for input 1 to go high
SH;' enable all axes
RE
```

OT applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC18x6,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

OV Off on encoder failure voltage



OVm= n
OV n, n, n, n, n, n, n, n, n

Usage	OVm= n	Arguments specified with a single axis mask and an assignment (=)
	OV n ...	Arguments specified with an implicit, comma-separated order
Operands	_OVm	Operand holds the value last set by the command

Description

The OV command sets the threshold voltage for detecting an encoder failure. The controller can detect a failure on either or both channels of the encoder.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	9.9982	0.9438	20/65,536	Torque voltage to trigger encoder error detection	

Remarks

- Encoder error detection is accomplished by checking on whether motion of at least 4 counts is detected whenever the torque exceeds a preset level (OV) for a specified time (OT).
 - Note that for this function to work properly it is recommended to have a non-zero value for KI.
- The value of OV should be high enough to guarantee that the motor would overcome any static friction in the system. If it is too low, there will be false triggering of the error condition.
- The OV value may not be higher than the TL value.
- See the OA command for more details on this error detection mode

Examples

```
'Galil DMC Code Example
OV 0.54;' Set A axis encoder detection torque value to 0.54V
```


```
'Galil DMC Code Example
#setup
'setup the encoder error detection
OTA= 10;' Set time to 10 milliseconds
OVA= 5;' Set voltage to 5
OAA= 1;' Enable encoder detection feature
EN
```

```
'Galil DMC Code Example
REM #POSERR example for checking to see if encoder failure occurred
REM This procedure is needed because the stop code will only update if
REM the profiler is running at the time the encoder failure is detected.
#POSERR
~a=0
#loop
IF _MC~a=1
IF ((_TE~a<_ER~a)&(_OE~a)&(_OA~a))
MG "possible encoder failure on ",~a{z1.0}," axis"
ENDIF
ENDIF
~a=~a+1
JP#loop,~a<_BV
AI1;' wait for input 1 to go high
SH;' enable all axes
RE
```

OV applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC18x6,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

PL Pole


PLm= n
PL n, n, n, n, n, n, n, n, n

Usage	PLm= n	Arguments specified with a single axis mask and an assignment (=)
	PL n ...	Arguments specified with an implicit, comma-separated order
Operands	_PLm	Operand holds the value last set by the command

Description

The PL command adds a low-pass filter in series with the PID compensation.

The crossover frequency is entered directly as an argument to PL. To maintain compatibility with earlier versions, a value less than 1 may also be specified.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	250	0	1	Crossover frequency created by the PL command	'Max' is a function of TM. See Remarks
	0	0.9999	0	2/65,536	Value used to generate pole filter crossover frequency	See Remarks for the equation used. n = 0 disables the Pole filter

Remarks

- At lower TM settings, the maximum pole frequency is increased. The maximum value of the PL command is determined by the value of TM according to the following equation
 - Max = (1/4 * 10^6) * (1/TM)
- The digital transfer function of the filter is (1 - n) / (Z - n) and the equivalent continuous filter is A/(S+A) where A is the filter cutoff frequency: A=(1/T) ln (1 / n) rad/sec and T is the sample time.

Calculated Pole

- To convert from the desired crossover (-3 dB) frequency in Hertz to the value given to PL, use the following formula

$$n = e^{-T \cdot f_c \cdot 2\pi}$$

- where
 - n is the argument given to PL (less than 1)
 - T is the controller's servo loop sample time in seconds (TM divided by 1,000,000)
 - Fc is the crossover frequency in Hertz
- Example: Fc=36Hz TM=1000 n=e^(-0.001*36*2*pi) =0.8
- The following shows several example crossover frequencies achieved with various values of PL

n	Fc (Hz)
0	Infinite (off)
0.2	256
0.4	145
0.6	81
0.8	36
0.999	0

Examples

```
'Galil DMC Code Example
'Set A-axis Pole to 0.95
:PL .95
Query Pole value
:PL ?
0.9527
```

PL applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

PR *Position Relative*


PRm= n
PR n, n, n, n, n, n, n, n, n

Usage	PRm= n	Arguments specified with a single axis mask and an assignment (=)
	PR n ...	Arguments specified with an implicit, comma-separated order
Operands	_PRm	Operand holds the value last set by the command

Description

The PR command sets the incremental distance and direction of the next move. The move is referenced with respect to the current position. .

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
	N	N	N/A	Axis	Virtual axis to assign value	
n	-2,147,483,648	2,147,483,647	N/A	1	Incremental distance for independent move	n = ? returns the current incremental distance specified

Remarks

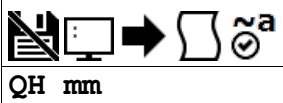
- _PRm contains the current incremental distance for the specified axis.

Examples

```
'Galil DMC Code Example
'using PA/PR, you can query PR for the incremental distance
:DP 10000
:PA 8000
:PR ?
-2000
```

PR applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

QH *Query Hall State*

QH mm

Usage	QH mm	Argument is an axis mask
Operands	_QHm	Operand has special meaning, see Remarks

Description

The QH command transmits the state of the Hall sensor inputs. The value is decimal and represented by a 3 bit value (see Remarks).

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axis to return Hall status	

Remarks

- The 3 bit value returned by QH is defined in the table below:

Bit	Status
07	Undefined (set to 0)
06	Undefined (set to 0)
05	Undefined (set to 0)
04	Undefined (set to 0)
03	Undefined (set to 0)
02	Hall C State
01	Hall B State
00	Hall A State

- QH should return a value from 1 through 6 as valid Hall combinations. A value of 0 or 7 is invalid when using Hall sensors and will generate a Hall error with OE set.
 - The valid sequence for Hall inputs is a greycode output (only one bit changes at a time):
 - 1,3,2,6,4,5 (or 5,4,6,2,3,1)
 - To disable Hall error checking, set the axis to brushed with a BR 1 command.
- When using an internal sine amplifier, the BA command must be issued before QH will report the Hall state status.

Operand Usage

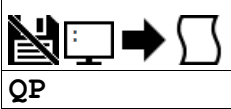
- _QHm Contains the state of the Hall sensor inputs for the specified axis

Examples

```
'Galil DMC Code Example
:QHA;' Query A axis Hall status
7
:TA1;' Check for Hall errors in the amp
1
:'A 1 indicates Hall error on axis A
```

QH applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

QP *Query Parameters*

QP

Usage	QP	Command takes no arguments
--------------	----	----------------------------

Description

Returns memory information for the controller.

Arguments

QP takes no arguments.

Remarks

- Each row of the QP response describes a parameter characteristic of the controller.
- The data is provided in a comma separated list starting with a fixed ID string.

QP response row descriptions

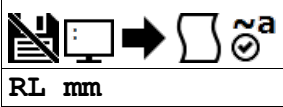
Row ID	Field 1	Field 2	Field 3	Field 4	Description of the row
"PR"	characters per line	number of lines	flash=1, ram=0	N/A	Determines the dimensions of the program and the runtime location of the program.

Examples

```
'Galil DMC Code Example
:QP
PR, 80, 1000, 1
```

QP applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,RIO47xxx,DMC18x6,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

RL Report Latched Position

RL mm

Usage	RL mm	Argument is an axis mask
Operands	_RLm	Operand has special meaning, see Remarks

Description

The RL command will return the last position captured by the latch. The latch must first be armed by the AL command and then the appropriate input must be activated. Each axis uses a specific general input for the latch input; see the AL command for information on latch inputs.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axis to query for latched position	

Remarks

- The armed state of the latch can be configured using the CN command.
- The Latch Function works with the main or auxiliary encoder.
- See the DMC-3x01x user manual for a description of RL with sinusoidal encoders (DMC-31000 only)

Capturing Stepper Position using the Latch

- When working with a stepper motor without an encoder, the latch can be used to capture the stepper position. Follow the steps below to achieve this.
 1. Place a wire from the controller Step (PWM) output into the main encoder input, channel A+.
 2. Connect the Direction (sign) output into the channel B+ input.
 3. Configure the main encoder for Step/Direction using the CE command.
 4. The latch will now capture the stepper position based on the pulses generated by the controller.

Operand Usage

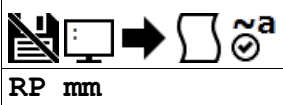
- _RLm contains the latched position of the specified axis.

Examples

```
'Galil DMC Code Example
:JG 5000;' Set up to jog the A-axis
:BG A;' Begin jog
:AL A;' Arm the latch, assume that after about 2 seconds, input goes low
:RL A;' Report the latch
10000
```

RL applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

RP Reference Position

Usage	RP mm	Argument is an axis mask
Operands	_Rpm	Operand has special meaning, see Remarks

Description

The RP command returns the commanded reference position of the motor(s). RP command is useful when operating step motors since it provides the commanded position in steps when operating in stepper mode.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axis to report commanded position	
	N	N	N/A	Multi-Axis Mask	Virtual axis to report commanded position	

Remarks

- The relationship between RP, TP and TE: TEA equals the difference between the reference position, RPA, and the actual position, TPA.
 - TE = RP - TP
- _Rpm contains the commanded reference position for the specified axis.

Examples

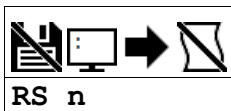
```
'Galil DMC Code Example
'Assume that A axis is commanded to be at the position 200
'The returned units are in quadrature counts.
:PF 7;' Position format of 7
:RP
200
:RPA
200 Return the A motor reference position
:PF-6.0;' Change to hex format
:RP
$0000c8
:position =_RPA;' Assign the variable, position, the value of RPA
```

```
'Galil DMC Code Example
:GAN;' make A axis slave to N imaginary axis
:GR-1;' 1:-1 gearing
:SPN=10000
:PRN=10000
:BGN;' Begin motion
:RPN;' Get master position
10000
:RPA;' Get slave commanded position
-10000
```

RP applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

RS *Reset*



Usage	RS n ...	Arguments specified with an implicit, comma-separated order
Operands	_RS	Operand has special meaning, see Remarks

Description

The RS command resets the state of the processor to its power-on condition. The previously saved state of the hardware, along with parameter values and saved program, are restored.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
n	-1	0	0	1	Set behavior of RS command	n = 0 performs normal reset. n = -1 performs soft master reset. See Remarks.

RS has no arguments.

Remarks

- A soft master reset performed by issuing RS -1 restores factory default settings without erasing the EEPROM. To restore saved EEPROM settings use RS with no arguments, or RS 0.

Operand Usage

- _RS returns the state of the processor on its last power-up condition. The value returned is the decimal equivalent of the 4 bit binary value shown below.
 - Bit 3 For master reset error
 - Bit 2 For program checksum error
 - Bit 1 For parameter checksum error
 - Bit 0 For variable checksum error
- At startup the controller operating system verifies the firmware sector. If there is a checksum error shown by _RS in firmware, it is not loaded and the controller will boot to monitor mode.
 - The #AUTOERR automatic subroutine will run if this error occurs and the subroutine is located in the program space.

Examples

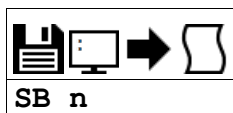
```
'Galil DMC Code Example
:RS;'      Reset the hardware
:RS-1;'    Perform a soft master reset
```

RS applies to

DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,RIO574x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

SB Set Bit



Usage	SB n ...	Arguments specified with an implicit, comma-separated order
-------	----------	---

Description

The SB command sets a particular digital output. The SB and CB (Clear Bit) instructions can be used to control the state of output lines.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
n	1	4	N/A	1	General output bit to be set	
n	1,000	6,999	N/A	1	Set Modbus slave bit	See "SB via Modbus Slave" in Remarks

Remarks

- The state of the output can be read with the @OUT[] command.

Using SB with a Modbus Slave

- $n = (\text{SlaveAddress} * 10000) + (\text{HandleNum} * 1000) + ((\text{Module}-1) * 4) + (\text{Bitnum}-1)$
 - Slave Address is used when the ModBus device has slave devices connected to it and specified as Addresses 0 to 255. Please note that the use of slave devices for modbus are very rare and this number will usually be 0.
 - HandleNum is the handle specifier where A is 1, B is 2 and so on.
 - Module is the position of the module in the rack from 1 to 16.
 - BitNum is the I/O point in the module from 1 to 4

Examples

```
'Galil DMC Code Example
#main
SB 5;'      Set digital output 5
SB 1;'      Set digital output 1
CB 5;'      Clear digital output 5
CB 1;'      Clear digital output 1
EN
```

```
'Galil DMC Code Example
#modbus
REM connect to modbus slave at IP address 192.168.1.50
IHH=192,168,1,50<502>2
WT100
SB 8001;'set bit 1 on modbus slave
WT 10
CB 8003;'set bit 3 on modbus slave
EN
```

For detailed information on connecting to a Modbus slave, see:

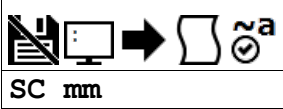
<http://www.galil.com/news/dmc-programming-io-control/setting-rio-pocket-plc-or-generic-modbus-slave-extended-io>

SB applies to

DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,RIO574x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

SC Stop Code



Usage	SC mm	Argument is an axis mask
Operands	_SCm	Operand has special meaning, see Remarks

Description

The Stop Code command returns a number indicating why a motor has stopped.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	N/A	Multi-Axis Mask	Axis to query stop code	Omitting argument shows stop code for A axis

Remarks

- When SC is issued, the controller reponds with a number for the axis queried. The number is interpreted as follows:

Stop Code Table

Stop Code Number	Meaning
0	Motors are running, independent mode
1	Motors decelerating or stopped at commanded independent position
2	Decelerating or stopped by FWD limit switch or soft limit FL
3	Decelerating or stopped by REV limit switch or soft limit BL
4	Decelerating or stopped by Stop Command (ST)
7	Stopped by Abort command (AB)
8	Decelerating or stopped by Off on Error (OE1)
11	Stopped by selective abort input
12	Decelerating or stopped by encoder failure (OA1) (For controllers supporting OA/OV/OT)
15	Amplifier Fault
16	Stepper position maintenance error

- _SCm contains the value of the stop code for the specified axis.

Examples

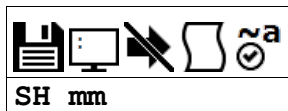
```
'Galil DMC Code Example
tom =_SCA;' Assign the Stop Code of A axis to variable tom
```

```
'Galil DMC Code Example
:JG10000
:BG A
:SC A
0 //Axis is running in independent mode
:ST A
:SC A
4 //Axis is stopped by ST command
```

SC applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

SH *Servo Here*



SH mm

Usage	SH mm	Argument is an axis mask
--------------	-------	--------------------------

Description

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

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axis to enable	

Remarks

- The SH command changes the coordinate system.
 - Therefore, all position commands given prior to SH, must be repeated. Otherwise, the controller produces incorrect motion.
- This command can be useful when the position of a motor has been manually adjusted following a motor off (MO) command.
- During a program download the servo will disable for a brief period (roughly 300ms). This can be prevented by first issuing an MO.

Examples

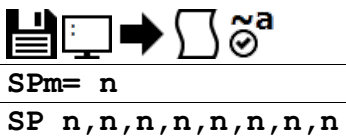
```
'Galil DMC Code Example
SHA;' servo on the A axis
```

```
'Galil DMC Code Example
'show how issuing SH clears position error
'by resetting the coordinate system
:MOA;' disable the A axis
:TEA;' check error on A axis
-12435
:TPA;' Check position
12435
:SHA;' enable A axis, doing so clears the error
:TEA;' check error again
0
:TPA;' confirm position hasn't changed
12435
```

SH applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

SP *Speed*



Usage	SPm= n	Arguments specified with a single axis mask and an assignment (=)
	SP n ...	Arguments specified with an implicit, comma-separated order
Operands	_SPm	Operand holds the value last set by the command

Description

The SP command sets the slow speed of any or all axes for independent moves.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
	N	N	N/A	Axis	Virtual axis to assign value	
n	0	15,000,000	25,000	2	Value of jog speed in cnts/second	For MT settings of 1,-1,1.5 and -1.5 (Servos) - See Remarks for Resolution details
	0	3,000,000	25,000	2	Value of jog speed in cnts/second	For MT settings of 2,-2,2.5 and -2.5 (Steppers) - See Remarks for Resolution details
	0	50,000,000	25,000	2	Value of jog speed in cnts/second	DMC-31000 set to AF>=5 - See Remarks for Resolution details

Remarks

- Negative values will be interpreted as the absolute value

Resolution

- The resolution of the SP command is dependent upon the update rate setting (TM).
 - With the default rate of TM 1000 the resolution is 2 cnts/second.
 - The equation to calculate the resolution of the SP command is:
 - resolution = 2*(1000/TM)
 - example:
 - With TM 250 the resolution of the SP command is 8 cnts/second
 - resolution = 2*(1000/250) = 8

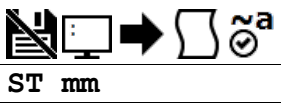
Examples

```
'Galil DMC Code Example
PR 2000,3000,4000,5000;'          Specify a,b,c,d parameter
SP 5000,6000,7000,8000;'        Specify a,b,c,d speeds
BG;'                             Begin motion of all axes
AM C;'                          After C motion is complete
'
'
'For vector moves, use the vector speed command (VS) to change the speed.
'SP is not a "mode" of motion like JOG (JG).
'Note: 2 is the minimum non-zero speed.
```

SP applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

ST Stop



ST mm

Usage	ST mm	Argument is an axis mask
-------	-------	--------------------------

Description

The ST command stops motion on the specified axis. Motors will come to a decelerated stop.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	ANS	A	Multi-Axis Mask	Axes to command to stop motion	

Remarks

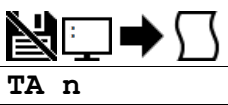
- If ST is sent from the host without an axis specification, program execution will stop in addition to motion.

Examples

```
'Galil DMC Code Example
:ST A;'      Stop motion on the A axis
:SC A;'      Query A axis status
4           Indicates stopped by ST command
:MG _NO;'    Check if code is running
1           Thread 0 running
:ST ;'       General stop
:MG _NO;'    check code again
0           Thread 0 stopped
```

ST applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TA *Tell amplifier error status*

TA n

Usage	TA n ...	Arguments specified with an implicit, comma-separated order
Operands	_TA0 _TA1 _TA2 _TA3	Operand has special meaning, see Remarks

Description

The command returns the amplifier error status. The value is decimal and represents an 8 bit value. Bit 7 is most significant bit, 0 is least.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
n	0	3	N/A	1	Selects amp status byte to return	

Tell Amplifier Error Status Bit Definition

	TA0	TA1	TA2	TA3	
BIT #:	STATUS:	STATUS:	STATUS:	STATUS:	BIT #
7	0	0	0	0	7
6	0	0	0	0	6
5	0	0	0	0	5
4	0	0	0	0	4
3	Under Voltage	0	0	0	3
2	Over Temperature	0	0	0	2
1	Over Voltage	0	0	0	1
0	Over Current	Hall Error A Axis	Peak Current A Axis	ELO Active	0

Remarks

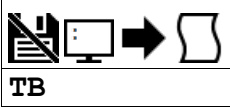
- _TAn Contains the amplifier error status. n = 0,1,2, or 3
- If a brushed-type servo motor is disabling and TA1 shows a Hall error, use the BR command to set the axis as a brushed axis. This causes the controller to ignore invalid Hall states.

Examples

```
'Galil DMC Code Example
:TA1
1 'bit 0 means Hall error for A axis
:TA0
8 'bit 3 means under voltage error for amp
```

TA applies to DMC500x0,DMC40x0,DMC41x3,DMC30010,DMC21x3,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TB *Tell Status Byte*

Usage	TB	Command takes no arguments
Operands	_TB	Operand has special meaning, see Remarks

Description

The TB command returns status information from the controller as a decimal number. Each bit of the status byte denotes an active condition when the bit is set (high):

Arguments

The following table describes the specific conditions reported with each bit of the TB report.

Tell Status Byte Response Bit Description

Bit #	Status
Bit 7	Executing application program
Bit 6	N/A
Bit 5	Contouring
Bit 4	Executing error or limit switch routine
Bit 3	Input Interrupt enabled
Bit 2	Executing input interrupt routine
Bit 1	N/A
Bit 0	Echo on

Remarks

- _TB Contains the status byte reported by the TB command

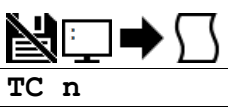
Examples

```
'Galil DMC Code Example
:TB
33'      Contouring on and Echo is on (2^5 + 2^0 = 32 + 1 = 33)
```

```
'Galil DMC Code Example
:TB;'
129'    Tell status information
        Executing program and echo on (2^7 + 2^0 = 128 + 1 = 129)
```

TB applies to DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,RIO47xxx,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TC *Tell Error Code*

Usage	TC n ...	Arguments specified with an implicit, comma-separated order
Operands	_TC	Operand has special meaning, see Remarks

Description

The TC command reports programming or command errors detected by the controller. The TC command returns a number between 1 and 255. This number is a code that reflects why a command was not accepted by the controller. This command is useful when the controller halts execution of a program or when the response to a command is a question mark.

Arguments

Argument	Value	Description	Notes
n	0	Return the numerical code only	Default
	1	Return the numerical code and human-readable message	

TC Error Code List

Tell Code Number	Description	Notes
1	Unrecognized command	
4	Operand error	
6	Number out of range	
7	Command not valid while running	
8	Command not valid while not running	
14	EEPROM check sum error	
15	EEPROM write error	
20	Begin not valid with motor off	
21	Begin not valid while running	
22	Begin not possible due to Limit Switch	
51	Question mark not valid	
58	Bad command response	i.e._GNX
83	Not a valid number	
105	EB1 command must be given first	
110	No hall effect sensors detected	
111	Must be made brushless by BA command	
112	BZ command timeout	
113	No movement in BZ command	
114	BZ command runaway	
119	Not valid for axis configured as stepper	
134	All motors must be in MO for this command	
135	Motor must be in MO	
140	Serial encoder missing	Valid for BiSS support
144	Incompatible with encoder type	
160	BX failure	
161	Sine amp axis not initialized	

Remarks

- TC command accepts ? as a query. This is equivalent to TC or TC 0
- After TC has been read, the error code is set to zero.
- _TC contains the value of the error code. Use of the operand does not clear the error code.

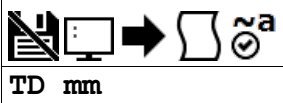
Examples

```
'Galil DMC Code Example
:GF32;' Bad command
?
:TC1;' Tell error code
1
Unrecognized command
```

TC applies to

DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,DMC52xx0,RIO574x0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TD *Tell Dual Encoder*

Usage	TD mm	Argument is an axis mask
Operands	_TDm	Operand has special meaning, see Remarks

Description

The TD command returns the current position of the dual (auxiliary) encoder input. When operating with stepper motors, the TD command returns the number of counts that have been output by the controller.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axis to report dual (auxiliary) encoder position.	

Remarks

- Auxiliary encoders are not available for a stepper axis or for the axis where output compare is used.

Operand Usage

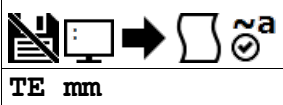
- _TDm reports the dual encoder position for the specified axis.

Examples

```
'Galil DMC Code Example
:TDA;' Return the A motor Dual encoder
200
:DUAL=_TDA;' Assign the variable, DUAL, the value of TDA
```

TD applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TE Tell Error

TE mm

Usage	TE mm	Argument is an axis mask
Operands	_TEm	Operand has special meaning, see Remarks

Description

The TE command returns the current error in the control loop.

The command returns the position error of the motor(s), which is the difference between commanded (RP) and actual (TP) position.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axis to report position error	

Remarks

- Under normal operating conditions with servo control, the position error should be small. The position error is typically largest during acceleration and deceleration.
- The Tell Error command is not valid for step motors since they operate open-loop.

Operand Usage

- _TEm contains the current position error value for the specified axis.

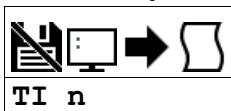
Examples

```
'Galil DMC Code Example
:TE;' Return all position errors
5, -2, 0, 6
:TEA;' Return the A motor position error
5
:TEB;' Return the B motor position error
-2
:Error =_TEA;' Sets the variable, Error, with the A-axis position error
```

```
'Galil DMC Code Example
:TEA;' Return the A motor position error
5
```

TE applies to DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,RIO47xxx,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TI Tell Inputs

TI n

Usage	TI n ...	Arguments specified with an implicit, comma-separated order
Operands	_TI0 _TI10	Operand has special meaning, see Remarks

Description

The TI command returns the state of the inputs in banks of 8 bits, or 1 byte. The value returned by this command is decimal and represents an 8 bit value (decimal value ranges from 0 to 255). Each bit represents one input where the LSB is the lowest input number and the MSB is the highest input bit.

Arguments

Argument	Value	Description	Notes
n	0	Report status of Inputs 1-8	Default
	10	Report status of Inputs 81-82	Auxiliary encoder inputs. See Remarks

Remarks

- For n = 10, the auxiliary encoder channels A and B can be used as additional IO. Only 2 inputs are available, resulting in a value of 255-252 reported by TI10.
 - See the User manual for more details.

Operand Usage

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

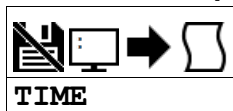
Examples

```
'Galil DMC Code Example
:TI0          Tell input state on bank 0
8            Bit 3 is high, others low
:input=_TI0;' Sets the variable, input, with the TI0 value
:input=?
8.0000
```

TI applies to

DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,RIO47xxx,DMC18x6,DMC18x2,RIO574x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TIME *Time Operand***TIME**

Usage	variable= TIME	Holds a value
Operands	TIME	Operand has special meaning, see Remarks

Description

The TIME operand returns the value of the internal free running, real time clock.

The returned value represents the number of servo loop updates and is based on the TM command. The default value for the TM command is 1000. With this update rate, the operand TIME will increase by 1 count every update of approximately 1000usec. The clock is reset to 0 with a standard reset or a master reset.

Arguments

TIME is an operand and has no parameters

Remarks

- The keyword, TIME, does not require an underscore (_) as with the other operands.
- TIME will increment up to +2,147,483,647 before rolling over to -2,147,483,648 and continuing to count up.
 - TIME rollover occurs after ~24-25 days of on-time at TM 1000 with no reset.
- TM 1000 will actually set an update rate of 976 microseconds. Thus the value returned by the TIME operand will be off by 2.4% of the actual time.


Examples

```
'Galil DMC Code Example
MG TIME;'  Display the value of the internal clock
t1=TIME;'  Sets the variable t1 to the TIME value
```

TIME applies to DMC500x0,DMC52xx0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,RIO47xxx,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallmc.com

TK Peak Torque Limit



TKm= n

TK n, n, n, n, n, n, n, n, n

Usage	TKm= n	Arguments specified with a single axis mask and an assignment (=)
	TK n ...	Arguments specified with an implicit, comma-separated order
Operands	_TKm	Operand holds the value last set by the command

Description

The TK command sets the peak torque limit on the motor command output. This command works with the TL command which sets the continuous torque limit. When the average torque is below TL, the motor command signal can go up to the TK (Peak Torque) limit for a short amount of time.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	9.9982	0	20/65,536	Value of peak torque limit	n = 0 disables the peak torque limit

Remarks

- TK provides the absolute value of the peak torque limit for +/- torque outputs
- Peak torque can be achieved for approximately 1000 samples upon initial command from 0V torque.
- If TK is set lower than TL, then TL is the maximum command output under all circumstances.


Examples

```
'Galil DMC Code Example
TLA= 7;' Limit A-axis to a 7 volt average torque output
TKA= 9.99;' Limit A-axis to a 9.99 volt peak torque output
```

TK applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TL Torque Limit


TLm= n
TL n, n, n, n, n, n, n, n, n

Usage	TLm= n	Arguments specified with a single axis mask and an assignment (=)
	TL n ...	Arguments specified with an implicit, comma-separated order
Operands	_TLm	Operand holds the value last set by the command

Description

The TL command sets the limit on the motor command output. This limit is designed to prevent over current to motors with lower current rating than the drive.

TL works along with the TK (Peak torque) command to control output current to the motor.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	9.9982	9.9982	20/65,536	Value of torque limit	

Remarks

- TL sets the absolute torque maximum for negative and positive torque
 - For example, TL of 5 limits the motor command output to 5 volts maximum and -5 volts minimum

TL With Internal Drives

- The maximum torque limit is different for certain amplifier configurations at a specific AG setting. These cases are listed below.

Amplifier	AG setting (current rating)	TL limit
DMC-3x012	2 (1.6 A/V)	6.2500
DMC-3x014	N/A (0.2 A/V)	5.0000
DMC-3x017 (MT 1,or-1)	2 (1.6 A/V)	6.2500

Examples

```
'Galil DMC Code Example
:TL 1;' Limit A-axis to 1 volt
:TL ?;' Return limit for A axis
1.0000
```

TL applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

TP *Tell Position*

TP mm

Usage	TP mm	Argument is an axis mask
Operands	_TPm	Operand has special meaning, see Remarks

Description

The TP command returns the current position of the motor.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axis to report motor position	

Remarks

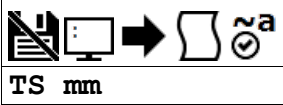
- _TPm contains the current position value for the specified axis.
- Omitting mm returns the position of all axes

Examples

```
'Galil DMC Code Example
'Assume the A-axis is at the position 200 (decimal)
'The returned parameter units are in quadrature counts.
:PF 7;' Position format of 7
:TP;' Return A position
200
:PF-6.0;' Change to hex format
:TPA;' Return A in hex
$0000c8
:position =_TPA;' Assign the variable, position, the value of TPA
```

TP applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TS Tell Switches

TS mm

Usage	TS mm	Argument is an axis mask
Operands	_Tsm	Operand has special meaning, see Remarks

Description

The TS command returns information including axis-specific IO status, error conditions, motor condition and state. The value returned by this command is decimal and represents an 8 bit value (decimal value ranges from 0 to 255).

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axes to report axis switches	

Remarks

- Each bit of the TS response represents the following status information when the bit is set (1).

Bit #	Status
Bit 7	Axis in motion
Bit 6	Position error exceeds error limit
Bit 5	Motor off
Bit 4	Reserved (0)
Bit 3	Forward Limit switch inactive
Bit 2	Reverse Limit switch inactive
Bit 1	Home switch status
Bit 0	Position Latch has occurred

- For active high or active low configuration (CN command), the limit switch bits are '1' when the switch is inactive and '0' when active.

Operand Usage

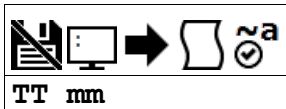
- _Tsm contains the current status of the switches for the specified axis.

Examples

```
'Galil DMC Code Example
:v1= _TSA;' Assigns value of TSA to the variable v1
:v1= ?;' Interrogate value of variable v1
15 (returned value) Decimal value corresponding to bit pattern 00001111
A axis not in motion (bit 7 - has a value of 0)
A axis error limit not exceeded (bit 6 has a value of 0)
A axis motor is on (bit 5 has a value of 0)
A axis forward limit is inactive (bit 3 has a value of 1)
A axis reverse limit is inactive (bit 2 has a value of 1)
A axis home switch is high (bit 1 has a value of 1)
A axis latch is not armed (bit 0 has a value of 1)
```

TS applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TT *Tell Torque*

TT mm

Usage	TT mm	Argument is an axis mask
Operands	_TTm	Operand has special meaning, see Remarks

Description

The TT command reports the value of the analog output signal, which is a number between -9.9982 and 9.9982 volts.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axis to report output torque command	

Remarks

- Torque output is limited by the value set for the TL command.
- _TTm contains the value of the torque for the specified axis.

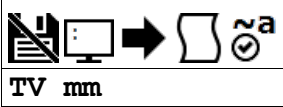
Examples

```
'Galil DMC Code Example
:v1=_TTA;' Assigns value of TTA to variable, v1
:TTA;' Report torque on A
-0.2843
```

TT applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

TV *Tell Velocity*



TV mm

Usage	TV mm	Argument is an axis mask
Operands	_TVm	Operand has special meaning, see Remarks

Description

The TV command returns the actual velocity of the axes in units of encoder count/s. The value returned includes the sign bit for direction.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
mm	A	A	A	Multi-Axis Mask	Axis to report velocity	

Remarks

- The TV command is computed using a special averaging filter (over approximately 0.25 sec for TM1000). Therefore, TV will return average velocity, not instantaneous velocity.
- _TVm contains the value of the velocity for the specified axis.


Examples

```
'Galil DMC Code Example
:vela=_TVA;'      Assigns value of A-axis velocity to the variable VELA
:TVA;'           Returns the A-axis velocity
3420
```

TV applies to DMC40x0,DMC42x0,DMC41x3,DMC21x3,DMC18x6,DMC18x2,DMC30010,DMC500x0,DMC52xx0,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

YA Step Drive Resolution


YAm= n
YA n, n, n, n, n, n, n, n, n, n

Usage	YAm= n	Arguments specified with a single axis mask and an assignment (=)
	YA n ...	Arguments specified with an implicit, comma-separated order
Operands	_YAm	Operand holds the value last set by the command

Description

Specifies the microstepping resolution of the step drive in microsteps per full motor step. Consult your drive documentation to determine its microstepping setting. See the table below for internal Galil stepper drives.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	9,999	16	1	Drive resolution in step counts/motor step for SPM mode	YA has special functionality for certain hardware configurations. See the rest of the notes in this table.
	1	16	16	see Notes	Valid settings for DMC-30016	1,2,4 and 16 set the step resolution of the DMC-30016 to full, half, 1/4th and 1/16th microstepping respectively. When full stepping (n=1) on the 30016, the max gain will be 70% of value set with AG. Max current is available for any microstepping mode.
	256	256	256	0	Valid setting for DMC-30017	The DMC-30017 is always configured for 256th microstepping, YA must be set to 256 for SPM mode

Remarks

YA Settings for Galil Stepper Drives

Stepper Drive Hardware	YA Setting	Notes
DMC-30016	1	Drive set to single step (70% current max)
DMC-30016	2	Drive set to half step
DMC-30016	4	Drive set to quarter step
DMC-30016	16	Drive set to 1/16 step
DMC-30017	256	Drive fixed at 1/256 step


Examples

```
'Galil DMC Code Example
'Set the step drive resolution for a 1/256 Microstepping Drive:
:YA 256
:'Query the A axis value
:MG_YAA;' Response shows A axis step drive resolution
256.0000
::
```

YA applies to DMC500x0, DMC40x0, DMC42x0, DMC41x3, DMC30010, DMC21x3, DMC18x6, DMC18x2, EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

YB Step Motor Resolution


YBm= n
YB n, n, n, n, n, n, n, n, n

Usage	YBm= n	Arguments specified with a single axis mask and an assignment (=)
	YB n ...	Arguments specified with an implicit, comma-separated order
Operands	_YBm	Operand holds the value last set by the command

Description

The YB command specifies the resolution of the step motor, in full steps per full revolution, for Stepper Position Maintenance (SPM) mode.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	9,999	200	1	Motor resolution in full steps/revolution	

Remarks

- This command is only required if using SPM mode with stepper motors with an attached encoder.
- A 1.8 degree step motor is 200 steps/revolution.


Examples

```
'Galil DMC Code Example
'Set the step motor resolution of the A axis for a 1.8 degree step motor:
:YBA=200
:'Query the A axis value
:YBA=?
200 Response shows A axis step motor resolution
```

YB applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com

YC Encoder Resolution


YcM= n
YC n, n, n, n, n, n, n, n, n

Usage	YcM= n	Arguments specified with a single axis mask and an assignment (=)
	YC n ...	Arguments specified with an implicit, comma-separated order
Operands	_YcM	Operand holds the value last set by the command

Description

The YC command specifies the resolution of the encoder, in counts per revolution, for Stepper Position Maintenance (SPM) mode.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	32,766	4,000	1	Encoder resolution in counts/revolution	

Remarks

- This command is only required if using SPM mode with stepper motors with an attached encoder.


Examples

```
'Galil DMC Code Example
'Set the encoder resolution of the A axis
:YcA=2000
:'Query the A axis value
:YcA=?
2000
:'Response shows A axis encoder resolution
```

YC applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

YR Error Correction


YRm= n
YR n, n, n, n, n, n, n, n, n

Usage	YRm= n	Arguments specified with a single axis mask and an assignment (=)
	YR n ...	Arguments specified with an implicit, comma-separated order

Description

The YR command allows the user to correct for position error in Stepper Position Maintenance mode. This correction acts like an IP command, moving the axis or axes the specified quantity of step counts. YR will typically be used in conjunction with QS.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	-2,147,483,648	2,147,483,647	0	1	Number of step pulses to increment position by	

Remarks

- Users will typically use the value of QS to increment motor by the number of step pulses of error.
 - EG. YRm = _QSm increments the specified axis by the error magnitude.
- The sign of YR depends on the polarity of the position encoder
 - If the encoder increments when the stepper moves forward (increasing TD), the correction is YRm=_QSm. This is typical.
 - If the encoder decrements when the stepper moves forward, the correction is YRm= -_QSm. See CE to invert the polarity of the position encoder, if desired.


Examples

```
'Galil DMC Code Example
'Query the error of the A axis:
:QSA
253
:' This shows 253 step counts of error
:'Correct for the error:
:YRA=_QSA;' The motor moves _QS step counts to correct for the error
'and YS is set back to 1
```

YR applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@galilmc.com

YS Stepper Position Maintenance Mode Enable, Status


YSm= n
YS n, n, n, n, n, n, n, n, n

Usage	YSm= n	Arguments specified with a single axis mask and an assignment (=)
	YS n ...	Arguments specified with an implicit, comma-separated order
Operands	_YSm	Operand has special meaning, see Remarks

Description

The YS command enables and disables the Stepper Position Maintenance Mode function. YS also reacts to excessive position error condition as defined by the QS command.

Arguments

Argument	Min	Max	Default	Resolution	Description	Notes
m	A	A	N/A	Axis	Axis to assign value	
n	0	1	0	1	Setting of the SPM mode	n = 0 disables SPM mode, n = 1 Enables SPM mode. See Remarks

Remarks

- Both YSm = ? and _YSm contain the value of n. n is 1 when SPM mode is enabled and no error has occurred. If a position error has occurred, n becomes 2.
 - If n = 2, this indicates a position error condition defined as more than 3 full motor steps of position error.
 - Issuing an n = 1 will clear the error

Position Error Limit

Microstep Setting (YA)	Error (QS) Limit
1	3
2	6
16	48
64	192
256	768

Examples

```
'Galil DMC Code Example
'Enable the mode:
:YSA=1
:'Query the value:
:YSA=?
1 Response shows A axis is enabled
```

YS applies to DMC500x0,DMC40x0,DMC42x0,DMC41x3,DMC30010,DMC21x3,DMC18x6,DMC18x2,EDD3701x

©2017 Galil Motion Control. Revision: 1740 . Corrections, Feedback: documentation@gallimc.com