

SERVOSTAR[®] S- and CD-series Serial Communications Protocol

Serial communications is a very important operational aspect of the SERVOSTAR[®] product line. The link between drive and host (terminal, PC, or high-level controller) enables this digital drive family to communicate using ASCII-coded messages transmitted over an asynchronous, multi-drop line. The drive language includes messages that not only command and interrogate, but also set and query. This gives you maximum manipulation of the drive's operation.

This document describes the protocol Danaher Motion has established to communicate between the SERVOSTAR and its host. It provides information pertaining to drive requirements, message handling, and data transfer.

Message Control

The SERVOSTAR and the host must use the same format for serial data. When setting up a host for serial transmission, you must consider the type of duplex transmission, the number of bits per character, whether to use a parity check, the number of stop and start bits, and the baud rate. To properly communicate with the SERVOSTAR, the drive variables corresponding to the transmission issues must configure to the following:

- Full-duplex
- bits per character
- No parity
- Start bit
- Stop bit
- BAUD rates supported: 9600, 19200
- Hardware supported: RS-232-C, RS-485

Addressing

The SERVOSTAR has the flexibility to be addressed and controlled as a single-line RS-232-A drive (C1 connector), a daisy-chained RS-232-A line (C1/C7 connectors), or an RS-485 multi-drop line (C1 connector). The 10-position DIP switch located on the top of the drive sets the binary address number for that particular unit. Switch 6 of the same DIP switch determines the baud rate which can be set for either 9600 BPS or 19200 BPS.

Single-Line (RS-232-A)

In the single-line RS-232-A configuration, the drive is set for a binary 0 address. In the daisy chain configuration, the drive can be set for any binary address from 1 to 31 by configuring the first 5 switches in the 10-position DIP switch located on the top of the drive. All drives in this configuration must be daisy-chained through the C7 connector. You can communicate with any or all drives within the daisy chain configuration from any one RS-232-A port on that chain.

Configuration Switch		
Switch #	Function	Settings
1	Daisy Chain Addressing	Bit 0 of Multidrop Address (LSB)
2		Bit 1 of Multidrop Address
3		Bit 2 of Multidrop Address
4		Bit 3 of Multidrop Address
5		Bit 4 of Multidrop Address (MSB)
6	Serial Port Baud Rate	0 = 9600 1 = 19200
7	HOLD Mode Switch	0 = Hold Mode Inactive 1 = Hold Mode Active
8	Drive Enable / Disable	0 = Drive Enable 1 = Drive Disable
9	SERCOS Transmit Power	0 = Low Power 1 = High Power
10	Factory Reserved	Must Be set to 0

To communicate with individual drives in a daisy chain, at the prompt, type `\ x <cr>`, where $x = 0..9, A..V, *, \$, corresponding to the address setting of that desired drive. To globally address all drives on the chain, `\ * <cr>` is typed at the prompt. When globally addressing the drives, no character echo to the terminal occurs. When the switch setting is set for binary 0, the drive assumes a single-line configuration.



The SERVOSTAR cannot be configured for RS-232-A and RS-485 at the same time.

Multi-drop (RS-485)

In the multi-drop configuration, up to 31 units can be addressed either individually or all at one time (global addressing) (described above). The one exception is that you cannot daisy-chain across the C7 connector. Multi-dropping must occur across C1 in the standard RS-485 fashion. You must supply the termination hardware.

Variables and Commands

Communication is always interactive in the **SERVOSTAR**. There are two ways to interact with the drive - through variables and commands. This interaction controls and configures the drive as well as monitors its operational status. Commands tell the drive to do something. Variables store numerical values used by the drive.

Variables can be set or read and come in two forms:

- 1) Read-only variables - used strictly for monitoring purposes. These variables are set or calculated by the drive.
- 2) Read/Write variables - used for both monitoring and controlling.

To read a variable, type the name followed by a <CR>. The drive prompts with the variable's present value. To set the control variable, enter the variable name, followed by optional white spaces (blanks, tab, "=" etc.), and then the desired value. The term, Switch Variable, refers to variables with predetermined multiple-choice selections.

The **SERVOSTAR** maintains two copies of all variables. One set of variables is stored in EEPROM (nonvolatile storage) and copied into RAM upon initialization or the issuing of the LOAD command. The set stored in RAM is used interactively. When a variable is changed, only the content of RAM is affected. To permanently store any changed variable (variables in EEPROM), you must issue a SAVE command.

Data Control

The **SERVOSTAR** is capable of processing about 1 character each 500 uS (assuming 19200 baud rate). The operating system obeys backspaces and resets upon receipt of an ESC character. The following software switches allow you to configure communication responses between drive and host.

ECHO Enables and disables character-by-character response to the host. If the feature is enabled and a the host sends a character, the drive receives the character and responds (or echos) the character back to the host. This allows the host to check the validity of the information sent to the drive. When an error is detected, the message being transmitted is aborted and the host indicates to you that an echo error (see ACKMODE and Asynchronous Error Control) was detected. The message is aborted.



This switch must be enabled for MOTIONLINK® to operate properly.

PROMPT Enables and disables the serial prompt (- ->). This feature indicates to the host that the drive is ready for instruction.



This switch must be enabled for MOTIONLINK to operate properly.

ACKMODE Sets the communication safety level of the drive (0, 1, 2). These levels of error control provide information back to the host describing the validity of the message just sent. The SERVOSTAR provides this error control only from the host to the drive. Active echoing may be enabled for all levels.

Safety Level 1 No error control is provided. The host receives no feedback regarding the validity of the commands or parameters it sends to the drive.

Safety Level 2 Error control is provided by causing the drive to respond to messages sent by the host with ACK or NAK ASCII characters. The ACK or NAK response indicates the message's executability. ACK indicates positive acknowledgment (executable) of the message and NAK indicates an invalid message (non-executable). A NAK could be returned to the host because the message sent either had incorrect commands or parameters. You can query the status word to determine the exact cause of the negative acknowledgment.

Safety Level 3 Error control uses the ACK or NAK message acknowledgment of Level 2 after verifying the message sent against its checksum. The message sent by the host to the drive has a Block Check Character (BCC) appended to it. This BCC is implemented as a one-byte checksum and is transmitted using an ASCII-HEX format as two characters. This higher-level error control checks the message against its BCC, providing added security of message receipt. Therefore, a NAK could be returned to the host not only because of incorrect commands or parameters but also because of an invalid checksum. The status word indicates which of the events caused the negative acknowledgment.



Safety Level 3:

- 1) **To turn off: ACKMODE=2 (turns on)
ACKMODE=061 (turns off;where 61 is the BCC)**
- 2) **This level is case sensitive,(i.e., enD3 <cr> verses EN93 <cr>).**

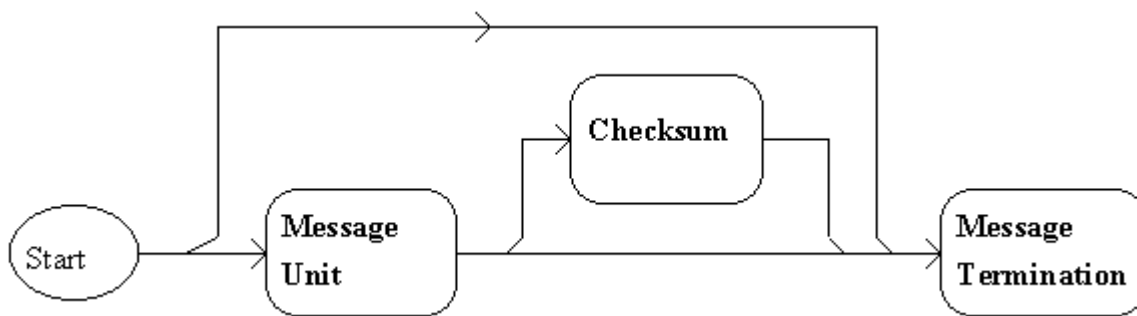
Message Processing

The general message format defines the structure by which the SERVOSTAR processes ASCII-coded messages. Messages from the host to the drive are used to send commands, to set variables, or to request information from the drive. The messages from the drive to the host contains the response to requests for information from the host.

This message format consists of two main elements. The first is called the Message Unit that includes a header (called mnemonic) and a set of parameters. The header defines the context of the parameters that follow it. Messages sent from the host to the drives always have headers. Except in specific cases, information sent from the drive to the host generally do not include a header. The other element is called the message termination. The following diagram shows the message format.



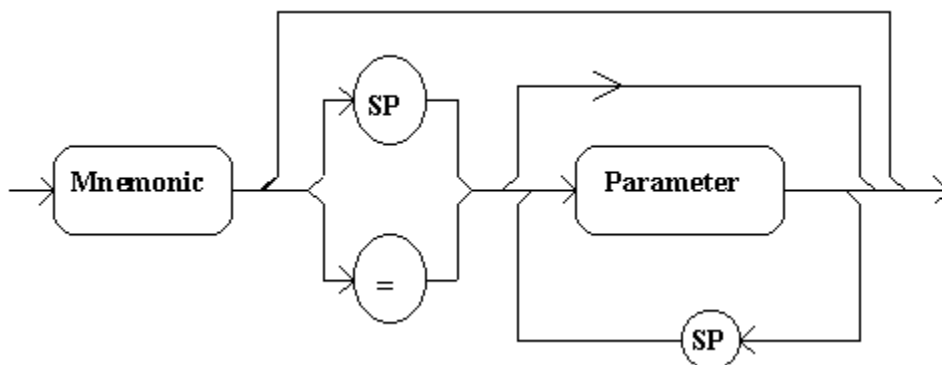
'Start' has no symbolic significance other than to represent the drive waiting for the host to send a message.



The drive also accepts (and ignores) a message termination sent without any additional informatio. The checksum is optional (valid only in Safety Level 3).

Message Unit

Message Unit refers to a block of information being transmitted on the communications medium. In general, Message Units are composed of a mnemonic and one or more optional parameters. The SERVOSTAR receives only one single Message Unit in a Message Format. The basic message unit is depicted in the following diagram.



When parameters are used, they are separated from the mnemonic by either a space or an assignment operator. Multiple parameters within one mnemonic must be separated by spaces.

Data Formats

Different Message Units require parameters represented in different data formats (e.g., recorded data). The format chosen is determined using the GETMODE (0,1,2) variable. The three data formats are:

1. **ASCII format (0)** - suitable for sending decimal integers. The data is arranged in columns defining the <Number of elements> of ASCII data. The data is separated by commas and each row is terminated by <CR><LF>. The number of rows corresponds to <Number of samples>.

```
<X1>,<Y1>,<Z1><CR><LF>
<X2>,<Y2>,<Z2><CR><LF>
<X3>,<Y3>,<Z3><CR><LF>
.
.
.
<Xn>,<Yn>,<Zn><CR><LF>
```



This format must be chosen for proper operation of MOTIONLINK.

2. **ASCII-HEX format (1)** - useful for sending data that requires less processing overhead on the part of the receiver of the message. This format is differentiated from ASCII numeric data with the #H prefix. The characters of the parameter may be any hexadecimal value from 0 through F. Example: ACC H64 corresponds to ACC 100.

The data is arranged in columns defining the <Number of elements> of ASCII-HEX data. The data is not separated and each row is terminated by <CR><LF>. The number of rows corresponds to <Number of samples>. Each data point is comprised of 4 bytes representing a 16 bit value.

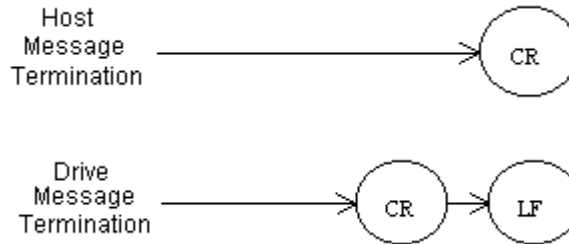
```
<X1><Y1><Z1><CR><LF>
<X2><Y2><Z2><CR><LF>
<X3><Y3><Z3><CR><LF>
.
.
.
<Xn><Yn><Zn><CR><LF>
```

3. **Binary data (2)** - used only for uploading recorded data. The binary data is transmitted in the same order as the ASCII and ASCII-HEX data. The data is without comma or <CR><LF> separators. The transmission is terminated by a <CR><LF> suffix.

```
<X1><Y1><Z1><X2><Y2><Z2><X3><Y3><Z3>...<Xn><Yn><Zn><CR><LF>
```

Message Termination

Message Termination refers to the end of the message being sent. Messages transmitted by the host are terminated by the Carriage Return (CR) character (ASCII 0DH). Messages transmitted by the drive are terminated by the Carriage Return/Line Feed (CR/LF) combination (ASCII 0DH/0AH).

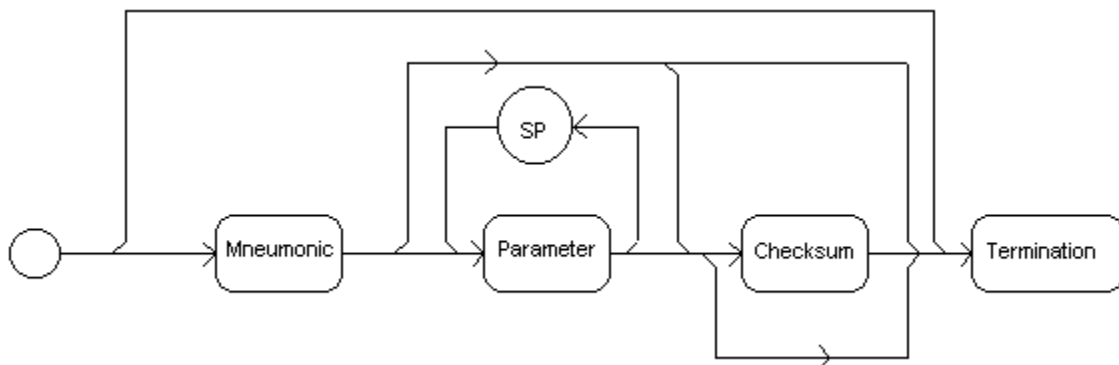


Complete Format

The general message format is shown in the following diagram.



The checksum block is used only when Safety Level 3 is selected.



Asynchronous Error Messages

The Asynchronous Error Message function is enabled through the MSG variable (1 or 0). If an error or fault occurs, the drive transmits a brief error message to the host. This feature works with the error control feature in ACKMODE (although not exclusive to this command). If this feature is enabled and ACKMODE = 1, an invalid command sent to the drive returns a NAK with a message that includes not only the error number but also a brief description.

An error can also occur through a fault in the system operation (e.g., motor overheats and trips the thermostat) where no command was sent to the drive. In this case, the drive returns the same type of error message, but without the NAK. The error message format for both is:

<CTRL-G>ERR xx <Error message string><CR><LF>

where **<CTRL-G>** indicates the BELL character (ASCII 07H).

If echoing is disabled, the error message transmits after the **<CR>** message termination is detected by the drive. If echoing is enabled, the characters in the message are all echoed prior to the error message being transmitted. Even though an error has occurred and its message returned to the host, the drive still accepts new incoming messages and attempts to execute them.



The most recent error/fault recorded by the drive (ERR variable) is stored in a variable that is NOT of the read-and-clear type. That is, reading the error does not clear it. The drive must detect a new error or fault before transmitting the error message.

Controlling the Drive

You control communications through a PC running **MOTIONLINK** for Windows software, your custom program, or a dumb terminal. Any of these three methods allow for the issuing of commands and the reading and changing of variables in the **SERVOSTAR**. The dumb terminal is the most basic method of controlling the drive. The most user-friendly way of controlling the **SERVOSTAR** is through **MOTIONLINK** for Windows. This package provides a robust set of control menus, function buttons, graphical displays, and dialog boxes to detail the operation of the drive. **MOTIONLINK** also features its own terminal mode of operation, offering you a friendly communication tool that far exceeds other terminal packages. For simplicity's sake, this document assumes that a dumb terminal is being used to control the drive.

General Information

The following specifications, special character protocol, and summary provide the information needed to understand the command section that follows.

SPECIFICATIONS

Communications Port: RS-232C
 Baud Rate: 9600 or 19,200 BPS
 Cable Length: 15 Meters Maximum (~50 feet)
 Data Type: Start (1) + Data (8) + Stop (1)
 Parity: None
 H/W Handshake: None
 S/W Handshake: None
 Character: ASCII code
 Data Error Check: 8-bit Checksum (optional ACKMODE switch)

CONTROL CODE DEFINITIONS				
Name	Symbol	Control	Hex	Decimal
Bell	<Bell>	^G	07h	7
Line Feed	<Lf>	^J	0Ah	10
Carriage Return	<Cr>	^M	0Dh	13
Space	<Sp>	--	20h	32
Ack Message	<Ack>	^F	06h	6
Nak Message	<Nak>	^U	15h	21
DELAY	<Dly>	--	--	--
Error Message	<ErrMsg>	--	--	--
Error Number	<##>	--	--	--

Communication Summary		
Drive-to-Terminal Transmission	Terminal-to-Drive Transmission	Protocol Flags (variables)
<ul style="list-style-type: none"> - echoes characters - transmit prompt - Ack/Nak messages - variable values - error/fault messages 	<ul style="list-style-type: none"> - commands - variable values - variable queries 	ACKMODE ECHO PROMPT MSG

Issuing a Command

The following is a set of examples demonstrating the SERVOSTAR serial protocol between the drive and a host.

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=0, ECHO=1, PROMPT=1, MSG=1

CASE 1: VARIABLE R/O

EXAMPLE - MAXIMUM CURRENT (IMAX) Example Value: 50000

Sequence #	1	2	3	4	5	6	7
User Enters:	I		M		A		X
Drive Returns:		I		M		A	

Sequence #	8	9	10	11	12	13	14
User Enters:		<Cr>					
Drive Returns:	X		<Cr>	<Lf>	<Dly>	50000	<Cr>

Sequence #	15	16	17	18
User Enters:				
Drive Returns:	<Lf>	-	-	>

Other Commands: ANIN, DICONT, DIP, DIPEAK, ENCINITST, ENCOUT, ERR, HALLS, I, IA, IC, ICMD, IMAX, MLIST, PFB, PRD, RDRES, RECDONE, STAT, TRUN, V, VCMD, VE, VER

CASE 2: VARIABLE R/W

WRITE EXAMPLE - ACCELERATION (ACC) Example Value: 50000

Sequence #	1	2	3	4	5	6	7
User Enters:	A		C		C		=
Drive Returns:		A		C		C	

Sequence #	8	9	10	11	12	13	14
User Enters:		5		0		0	
Drive Returns:	=		5		0		0

Sequence #	15	16	17	18	19	20	21
User Enters:	0		0		<Cr>		
Drive Returns:		0		0		<Cr>	<Lf>

Sequence #	22	23	24	25
User Enters:				
Drive Returns:	<Dly>	-	-	>

READ EXAMPLE - See Case 1: Variable R/O

Other Commands: ACC, ACKMODE, ANDB, ANOFF, AVGTIME, BW, COMPMODE, DEC, DECSTOP, FILTMODE, FOLDMODE, GETMODE, GV, GVI, ICONT, ILIM, ISCALE, IZERO, KV, KVFR, KVI, LMJR, LPFHZ1, LPFHZ2, MBEMF, MENCOFF, MENCRES, MENCATYPE, MLGAINP, MLGAINC, MLGAINZ, MLMAX, MLMIN, MPHASE, MPOLES, MRES, MSPEED, MTANGLC, MTANGLP, MVANGLF, MVANGLH, OPMODE, TF, THERMODE, VBUS, VLIM, VOSPD, VSCALE

CASE 3: COMMANDS MULTI-LINE VALUES (R/O)
 EXAMPLE - DUMP



These commands normally have a longer delay because of the amount of data that is output to the screen.

Sequence #	1	2	3	4	5	6	7
User Enters:	D		U		M		P
Drive Returns:		D		U		M	

Sequence #	8	9	10	11	12	13	14
User Enters:		<Cr>					
Drive Returns:	P		<Cr>	<Lf>	<Dly>	<VAR1>	<VAL1>

Sequence #	15	16	17	18	19	20	21
User Enters:							
Drive Returns:	<Cr>	<Lf>	<VAR2>	<VAL2>	<Cr>	<Lf>	<VARn>

Sequence #	22	23	24	25	26	27
User Enters:						
Drive Returns:	<VALn>	<Cr>	<Lf>	-	-	>

Other Commands: DUMP, FLTHIST, GET, LIST, MLIST

CASE 4: COMMANDS (R/W)
 EXAMPLE - EN

Sequence #	1	2	3	4	5	6	7
User Enters:	E		N		<Cr>		
Drive Returns:		E		N		<Cr>	<Lf>

Sequence #	8	9	10	11
User Enters:				
Drive Returns:	<Dly>	-	-	>

Other Commands: ANZERO, CLEEPROM, CONFIG, DIS, EN, ENCINIT, ENCSTART, ERR, FLTCLR, J, K, RECOFF, RECORD, REFRESH, RSTVAR, S, SAVE, STEP, STOP, T, TESTLED

CASE 5: COMMANDS LONG DELAY

EXAMPLE - TUNE

Sequence #	1	2	3	4	5	6	7
User Enters:	T		U		N		E
Drive Returns:		T		U		N	

Sequence #	8	9	10	11	12	13	14
User Enters:		<Cr>					
Drive Returns:	E		<Cr>	<Lf>	Extended <Dly>	-	-

Sequence #	15
User Enters:	
Drive Returns:	>

Other Commands: LOAD, TUNE

CASE 6: VARIABLES/COMMANDS MULTIPLE VALUES (R/W)

READ EXAMPLE - JOG (J)

Sequence #	1	2	3	4	5	6	7
User Enters:	J		<Cr>				
Drive Returns:		J		<Cr>	<Lf>	<Dly>	<VAL1>

Sequence #	8	9	10	11	12	13	14
User Enters:							
Drive Returns:	<Sp>	<VAL 2>	<Cr>	<Lf>	-	-	>

WRITE EXAMPLE - JOG (J)

EXAMPLE VALUE: 100 100

Sequence #	1	2	3	4	5	6	7
User Enters:	J		<Sp>		1		0
Drive Returns:		J		<Sp>		1	

Sequence #	8	9	10	11	12	13	14
User Enters:		<Sp>		1		0	
Drive Returns:	0		<Sp>		1		0

Sequence #	15	16	17	18	19	20	21
User Enters:	0		<Cr>				
Drive Returns:		0		<Cr>	<Lf>	<Dly>	-

Sequence #	22	23
User Enters:		
Drive Returns:	-	>

Other Commands: D, H, J, R, STEP

CASE 7: FLAGS R/O

READ EXAMPLE - REMOTE

Sequence #	1	2	3	4	5	6	7
User Enters:	R		E		M		O
Drive Returns:		R		E		M	

Sequence #	8	9	10	11	12	13	14
User Enters:		T		E		<Cr>	
Drive Returns:	O		T		E		<Cr>

Sequence #	15	16	17	18	19	20	21
User Enters:							
Drive Returns:	<Lf>	<Dly>	0	<Cr>	<Lf>	-	-

Sequence #	22
User Enters:	
Drive Returns:	>

Other Commands: ACTIVE, CCWLIM, CWLIM, DIPEN, DRIVEOK, FOLD, READY, RECDONE, RECING, RECRDY, RELAY, REMOTE, SWEN, THERM

CASE 8: FLAGS R/W

WRITE EXAMPLE - LIMDIS

Sequence #	1	2	3	4	5	6	7
User Enters:	L		I		M		D
Drive Returns:		L		I		M	

Sequence #	8	9	10	11	12	13	14
User Enters:		I		S		<Sp>	
Drive Returns:	D		I		S		<Sp>

Sequence #	15	16	17	18	19	20	21
User Enters:	1		<Cr>				
Drive Returns:		1		<Cr>	<Lf>	<Dly>	-

Sequence #	22	23
User Enters:		
Drive Returns:	-	>

READ EXAMPLE - SEE CASE 7: FLAGS R/O

Other Commands: ACTFAULT, DIR, ECHO, HOLD, LIMDIS, MHINVA, MHINVB, MHINVC, MSG, PROFMODE, PROMPT, RELAYMODE, UVRECOVERZERO

CASE 9: STRING VALUES
EXAMPLE - MOTOR



"<MOTOR>" RESULTS IN AN ALPHA/NUMERIC RESULT, DESCRIBING THE MOTOR.

Sequence #	1	2	3	4	5	6	7
User Enters:	M		O		T		O
Drive Returns:		M		O		T	

Sequence #	8	9	10	11	12	13	14
User Enters:		R		<Cr>			
Drive Returns:	O		R		<Cr>	<Lf>	<Dly>-

Sequence #	15	16	17	18	19	20
User Enters:						
Drive Returns:	<MOTOR>	<Cr>	<Lf>	-	-	>

Other Commands: MOTOR, SERIALNO

SERIAL ACK/NAK MODE - NORMAL

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=1, ECHO=1, PROMPT=1, MSG=1

CASE 1: VARIABLE R/O

EXAMPLE - MAXIMUM CURRENT (IMAX) Example Value: 50000



With ACKMODE=1, the Drive responds with ACK or NAK after every message. Shown below is an example of each, ACK and NAK

Sequence #	1	2	3	4	5	6	7
User Enters:	I		M		A		X
Drive Returns:		I		M		A	

Sequence #	8	9	10	11	12	13	14
User Enters:		<Cr>					
Drive Returns:	X		<Cr>	<Lf>	50000	<Cr>	<Lf>

Sequence #	15	16	17	18	19
User Enters:					
Drive Returns:	<Dly>	<ACK>	-	-	>

CASE 2: INCORRECT COMMAND

EXAMPLE - JOG (J)

Sequence #	1	2	3	4	5	6	7
User Enters:	J		G		<Cr>		
Drive Returns:		J		G		<Cr>	<Lf>

Sequence #	8	9	10	11	12	13	14
User Enters:							
Drive Returns:	<Dly>	<NAK>	<Bell>	<Error>	<Cr>	<Lf>	-

Sequence #	15	16
User Enters:		
Drive Returns:	-	>

SERIAL ACK/NAK MODE - NO ECHO

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=1, ECHO=0, PROMPT=1, MSG=1

CASE 1: VARIABLE R/O

EXAMPLE - MAXIMUM CURRENT (IMAX) Example Value: 50000



With ACKMODE=1, the drive responds with ACK or NAK after every message. Shown below is an example of each, ACK and NAK. However, the ECHO is turned off.

Sequence #	1	2	3	4	5	6	7
User Enters:	I		M		A		X
Drive Returns:							

Sequence #	8	9	10	11	12	13	14
User Enters:		<Cr>					
Drive Returns:					<Dly>	<Ack>	50000

Sequence #	15	16	17		19
User Enters:					
Drive Returns:	<Cr>	<Lf>	-	-	>

CASE 2: INCORRECT COMMAND

EXAMPLE - JOG (J)

Sequence #	1	2	3	4	5	6	7
User Enters:	J		G		<Cr>		
Drive Returns:							

Sequence #	8	9	10	11	12	13	14
User Enters:							
Drive Returns:	<Dly>	<NAK>	<Bell>	ERR	<Sp>	<##>	<Sp>

Sequence #	15	16	17	18	19	20
User Enters:						
Drive Returns:	<ErrMsg>	<Cr>	<Lf>	-	-	>

SERIAL ACK/NAK MODE - NO ECHO, NO PROMPT

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=1, ECHO=0, PROMPT=0, MSG=1

CASE 1: VARIABLE R/O

EXAMPLE - MAXIMUM CURRENT (IMAX) Example Value: 50000



With ACKMODE=1, the drive responds with ACK or NAK after every message. Shown below is an example of each, ACK and NAK. However, the ECHO is turned off.

Sequence #	1	2	3	4	5	6	7
User Enters:	I		M		A		X
Drive Returns:							

Sequence #	8	9	10	11	12	13	14
User Enters:		<Cr>					
Drive Returns:					<Dly>	<Ack>	50000

Sequence #	15	16	17	19
User Enters:				
Drive Returns:	<Cr>	<Lf>		

CASE 2: INCORRECT COMMAND

EXAMPLE - JOG (J)

Sequence #	1	2	3	4	5	6	7
User Enters:	J		G		<Cr>		
Drive Returns:							

Sequence #	8	9	10	11	12	13	14
User Enters:							
Drive Returns:	<Dly>	<NAK>	<Bell>	ERR	<Sp>	<##>	<Sp>

Sequence #	15	16	17	18	19	20
User Enters:						
Drive Returns:	<ErrMsg>	<Cr>	<Lf>			

SERIAL CHECKSUM - NO ECHO

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=2, ECHO=0, PROMPT=1, MSG=1

CASE 1: VARIABLE R/W

EXAMPLE - ACC Example Value: 25000

Sequence #	1	2	3	4	5	6	7
User Enters:	A		C		C		=
Drive Returns:							

Sequence #	8	9	10	11	12	13	14
User Enters:		2		5		0	
Drive Returns:							

Sequence #	15	16	17	18	19	20	21
User Enters:	0		0		0		F
Drive Returns:							

Sequence #	22	23	24	25	26	27	28
User Enters:		B		<Cr>			
Drive Returns:							<Dly>

Sequence #	29	30	31	32
User Enters:				
Drive Returns:	<ACK>	-	-	>



Enter the last two characters of the HEX VALUE sum before the Carriage Return.

CHARACTER	HEX VALUE	ASCII VALUE
A	41	65
C	43	67
C	43	67
=	3D	61
2	32	50
5	35	53
0	30	48
0	30	48
0	30	48
SUM=	14B	507

MULTIDROP MODE - NORMAL

BASIC COMMAND/VARIABLE SETUP

ADDR=3, ACKMODE=1, ECHO=1, PROMPT=1, MSG=1



Range of Values for ADDR (0-9, A-F)

A value other than 0, puts the system in Multidrop Mode, thus changing the prompt.

Sequence #	1	2	3	4	5	6	7
User Enters:	\		3		<Cr>		
Drive Returns:		\		3		<Cr>	<Lf>

Sequence #	8	9	10	11
User Enters:				
Drive Returns:	<Dly>	3	-	>

CASE 1: VARIABLE R/W

EXAMPLE - IMAX

Sequence #	1	2	3	4	5	6	7
User Enters:	I		M		A		X
Drive Returns:		I		M		A	

Sequence #	8	9	10	11	12	13	14
User Enters:		<Cr>					
Drive Returns:	X		<Cr>	<Lf>	50000	<Cr>	<Lf>

Sequence #	15	16	17	18
User Enters:				
Drive Returns:	<Dly>	3	-	>

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=1, ECHO=1, PROMPT=1, MSG=0

CASE 1: INCORRECT COMMAND (syntax error)

EXAMPLE - JOG (J)

Sequence #	1	2	3	4	5	6	7
User Enters:	J		G		<Cr>		
Drive Returns:		J		G		<Cr>	<Lf>

Sequence #	8	9	10	11	12	13	14
User Enters:							
Drive Returns:	<Dly>	<NAK>		<Cr>	<Lf>	-	-

Sequence #	8
User Enters:	
Drive Returns:	>

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=1, ECHO=1, PROMPT=1, MSG=1

CASE 2: INCORRECT COMMAND (syntax error)

EXAMPLE - JOG (J)

Sequence #	1	2	3	4	5	6	7
User Enters:	J		G		<Cr>		
Drive Returns:		J		G		<Cr>	<Lf>

Sequence #	8	9	10	11	12	13	14
User Enters:							
Drive Returns:	<Dly>	<NAK>	<Bell>	ERR	<Sp>	<##>	<Sp>

Sequence #	15	16	17	18	19	20
User Enters:						
Drive Returns:	<ErrMsg>	<Cr>	<Lf>	-	-	>

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=0, ECHO=1, PROMPT=1, MSG=1

CASE 3: DRIVE FAULT ERROR

EXAMPLE - Thermostat Trip



<Trip> is meant to be a condition in which there is no response from the drive or there is a wire misconnection.

Sequence #	1	2	3	4	5	6	7
User Enters:	<Trip>						
Drive Returns:		<Dly>	<Cr>	<Lf>	<Bell>	ERR	<Sp>

Sequence #	8	9	10	11	12	13	14
User Enters:							
Drive Returns:	<##>	<Sp>	<ErrMsg>	<Cr>	<Lf>	-	-

Sequence #	15
User Enters:	
Drive Returns:	>

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=0, ECHO=0, PROMPT=1, MSG=0

CASE 3: DRIVE FAULT ERROR



The **STAT** command is used to check the drive status and fault.

Sequence #	1	2	3	4	5
User Enters:	<Trip>				
Drive Returns:		<Dly>	-	-	>

BASIC COMMAND/VARIABLE SETUP

ADDR=0, ACKMODE=1, ECHO=0, PROMPT=0, MSG=0

CASE 3: STAT COMMAND



The value that it returns depends on the system (i. e., HXXXX).

Sequence #	1	2	3	4	5	6	7
User Enters:	S		T		A		T
Drive Returns:							

Sequence #	8	9	10	11	12	13	14
User Enters:		<Cr>					
Drive Returns:			<Dly>	<ACK>	H	0	0

Sequence #	15	16	17
User Enters:			
Drive Returns:	1	<Cr>	<Lf>

Variable Definitions

ACKMODE

Sets the communication safety level of the drive. The range of values is 0, 1, or 2.

0 = No safety procedures or error messages

1 = Drive responds with ACK or NAK after every message

2 = Same as 1 with a Block Check Character (BCC or checksum) at the end of every message



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

Firmware Versions: All **Type:** switch mode (R/W) **Units:** N/A
Range: 0 - 2 **Default:** 400,000 **EEPROM:** Yes
Opmodes: All **Drive Status:** EN/DIS

ECHO

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

0 = serial port echo disabled

1 = serial port echo enabled



ECHO = 1 is needed for proper operation of MOTIONLINK.

Firmware Versions: All **Type:** switch (R/W) **Units:** N/A
Range: 0, 1 **Default:** 1 **EEPROM:** Yes
Opmodes: All **Drive Status:** EN/DIS

GETMODE

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

- 0 = ASCII data transfer format
- 1 = ASCII-HEX data transfer format
- 2 = BINARY data transfer format (fastest)



GETMODE=0 is needed for operation of MOTIONLINK.

Firmware Versions: All	Type: switch mode (R/W)	Units: N/A
Range: 0, 1, 2	Default: 0	EEPROM: Yes
Opmodes: All	Drive Status: EN/DIS	

LOAD

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

Firmware Versions: All	Command Syntax: LOAD
Opmodes: All	Drive Status: DIS

PROMPT

Enables and disables the serial port prompt (-->) output by the drive after each message.

- 0 = disable the prompt
- 1 = enable the prompt



PROMPT = 1 is needed for proper operation of MOTIONLINK.

Firmware Versions: All	Type: switch (R/W)	Units: N/A
Range: 0, 1	Default: 1	EEPROM: Yes
Opmodes: All	Drive Status: EN/DIS	

SAVE

Copies all system configuration variables from working RAM to non-volatile memory (EEPROM). This command must be executed in order to retain setting changes during power cycling. The SAVE command takes about 2 seconds to execute.

Firmware Versions: All	Command Syntax: SAVE
Opmodes: 0, 1	Drive Status: EN/DIS