

SERVOSTAR® 600 and 300 Serial Communications Protocol

Serial communication is a very important operational aspect of both the SERVOSTAR® 600 and 300 product lines. The link between drive and host (terminal, PC, or high-level controller) enables this digital drive to communicate using ASCII-coded messages transmitted over an asynchronous line. The drive language includes messages that command and interrogate as well as set and query. This gives you maximum manipulation of the drive's operation.

This application note describes the protocol that Danaher Motion has established to communicate between the SERVOSTAR® 600 and its host. It provides information pertaining to drive requirements, message handling, and data transfer. For a full listing of parameters that can be used through serial communication, see the S600 and S300 ASCII.chm parameter listings.

Message Control

Requirements

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 either the SERVOSTAR 600 or 300, the drive parameters corresponding to the previously mentioned transmission issues must configure to:

- Full-duplex
- 8 bits per character
- No parity
- One Start bit
- BAUD rates supported: 9,600 (S600)
38,400 (S300)
- Hardware supported: RS-232

Addressing

The SERVOSTAR 600 and 300 support serial communication from a PC or PLC to *one* drive through an RS-232 cable. To communicate, connect the PC to S600 Connector X6 (RS232 pins). See the *Installation Manual* for either the S600 or S300 for full details on pin-out wiring.

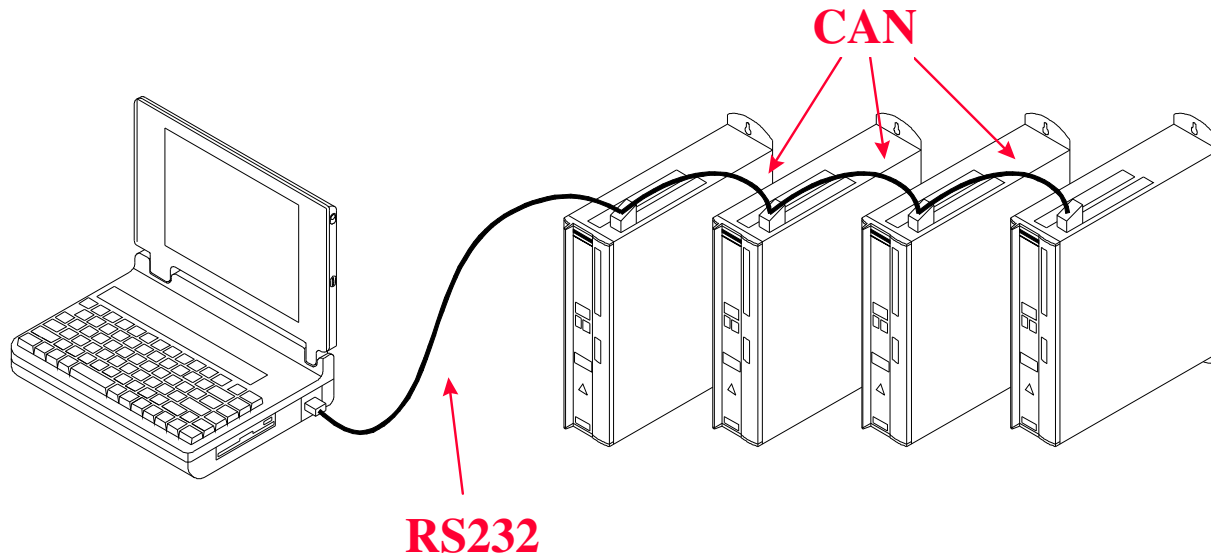
The SERVOSTAR 600 only supports serial communication to multiple S600 drives through one serial connection to the first drive and then uses the CAN link to transfer the serial communication to other axes. This is accomplished by connecting the PC to the first drive using connector X6 (RS232 pins). Then, connecting additional drives using the connector X6 CANOPEN connections on each drive.



NOTE

There is no communication between drives, only communication from a “master” PC to one drive at a time. The first and last drive on the CAN network require a termination resistor. See the installation manual for more details.

When setting up a multi-drop configuration, each axis must have a unique address. The drive's address can be set up in the Drive.exe Basic Setup screen. Here is example of a PC communicating to (4) S600 drives.



Multi-drop communications – S600 only

When more than 1 drive is used, the ASCII command, **SCAN**, looks for the connected drives in combination with RS-232 and CAN applications.

Example

Three S600 are connected together. Drive 1 connects to the PC through RS-232. Drive 1 connects to drive 2 through the CAN connection. Drive 2 connects to drive 3 through the CAN connection.

- DRIVE1 has ADDR= 1
- DRIVE2 has ADDR= 2
- DRIVE3 has ADDR= 5

The command, **SCAN**, gives back the address of each additional drive to the first.

Using A Terminal Emulator

It is possible to communicate to one or more drives (but only one at a time) using a terminal emulator program, such as HINT or HYPERTERMINAL.



The SERVOSTAR 600 or 300 cannot be configured for RS-485 communication with the host.

Parameters and Commands

Communication is always interactive in the SERVOSTAR 600. This interaction controls and configures the drive as well as monitors its operational status. Commands tell the drive to do something. Parameters store numerical values used by the drive and/or the user. Parameters are set or read and come in two forms:

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

To read a parameter, type the name followed by a carriage return (<CR>). The drive prompts with the parameter's present value. To set a control parameter, enter the parameter name followed by optional white spaces (blanks, tab, "=" etc.), and then the desired value.

The SERVOSTAR 300 and 600 maintains two copies of all parameters. One set of parameters is stored in EEPROM (nonvolatile storage) and copied into RAM at the time of initialization or the issuing of a LOAD command. This set, stored in RAM, is used interactively. When a parameter is changed, only the content of RAM is affected. To permanently store any changed parameters (parameters in EEPROM), issue a SAVE command.

The SERVOSTAR 300 and 600 store motion tasks in RAM (volatile) and Flash memory (non-volatile). The S600 stores motion tasks 0 and 192 to 255 in RAM and motion task 1 to 180 in Flash. The S300 stores motion tasks 0 and 201 to 300 in RAM and motion task 1 to 200 in Flash. Tasks stored in flash can only be changed when the drive is disabled.

Data Control

The SERVOSTAR 600 is capable of processing about 1 character each 1 ms (assuming 9600 baud rate). The operating system obeys backspaces and resets upon receipt of an <ESC> character. The PROMPT command allows the user to configure communication responses between drive and host:

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

PROMPT 0 = no character response to the host.

PROMPT 1 = character by character response to the host

PROMPT 2 = reserved

PROMPT 3 = character by character response with checksum.

Example: PROMPT= 3

All characters of a command are added by modulo 256. This checksum is transmitted as a two digit Hex number at the end of the command.

Command: ADDR 1

Checksum: A = 0x41 (H)

D = 0x44 (H)

D = 0x44 (H)

R = 0x52 (H)

<Sp>= 0x20 (H)

1 = 0x31 (H)

Total: 0x16c (H)

Modulo 256 for 16C(H)= 6C(H)

The first character of checksum: 6(H)+ 30(H) = 3D(H) or "6" ASCII character

The second character of checksum: C(H) + 30(H) = 3F(H) or "<" ASCII character

The command with checksum is: ADDR 16<

For a correct checksum the sign ACK is given. For an incorrect checksum the sign NACK is given. The SERVOSTAR 600 and SERVOSTAR S/CD series utilize the same method to calculate the checksum.

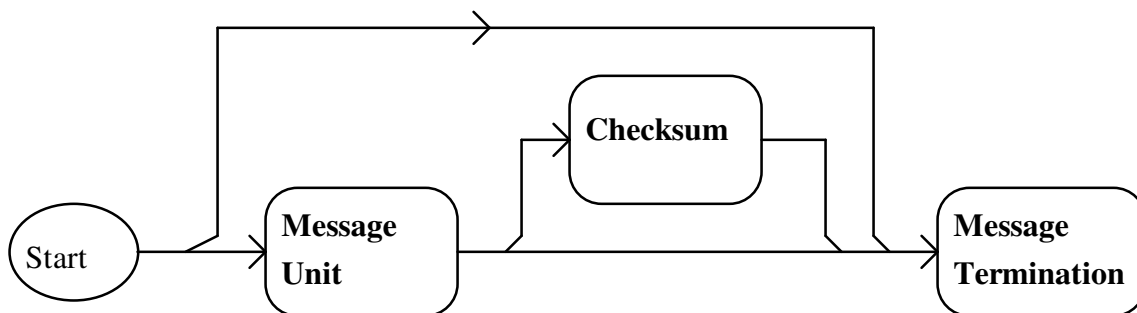
Message Processing

General Message Format

The general message format defines the structure by which the SERVOSTAR 600 or 300 processes ASCII-coded messages. Messages from the host to the drive are used to send commands, to set parameters, or to request information from the drive. The messages from the drive to the host contain the response to requests for information from the host.

This message format consists of two main elements. The first is called the message unit which 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 do not include a header. The other element is called the message termination.

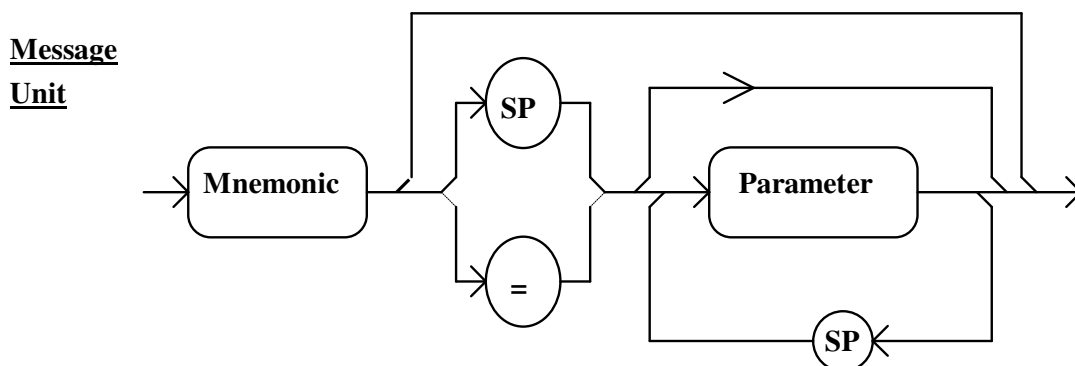
The following diagram shows the message format. The *Start* wording has no symbolic significance other than to represent the drive waiting for the host to send a message.



This diagram shows that the drive also accepts (and ignores) a message termination sent without any additional information and that the checksum is optional (only valid in Safety Level 3).

Message Unit

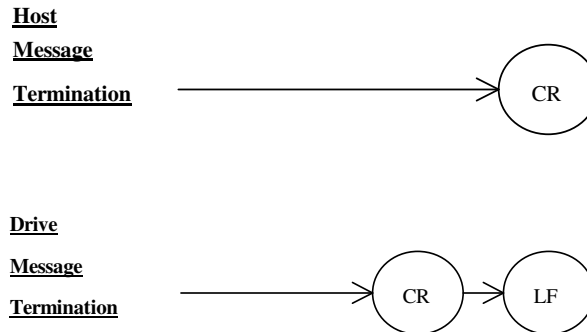
The term, **message unit**, refers to a block of information being transmitted on the communication medium. In general, message units are composed of a mnemonic and one or more optional parameters. The SERVOSTAR 600 or 300 can only receive a single message unit in a message format. The basic message unit can be shown by 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.

Message Termination

Message termination refers to the end of the message being sent. Messages transmitted by the host are terminated by a 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).



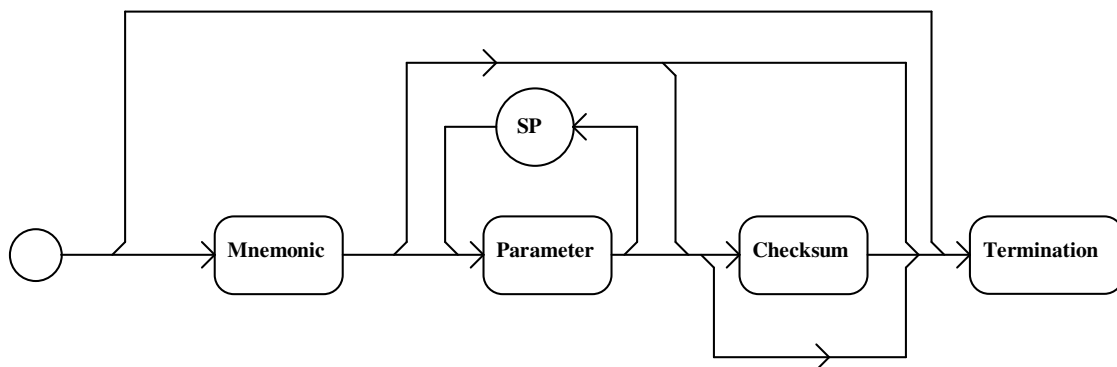
Complete Format

The general message format is shown by the following diagram.



NOTE

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 parameter (2,1,or 0). If an error or fault occurs, the drive transmits a brief error message to the host. The error message format for both instances mentioned is:

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

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

If echoing is disabled (PROMPT= 0) the error message transmits only after the <CR> message termination has been 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.



NOTE

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

Controlling the Drive

The user controls communication through a PC running **MOTIONLINK**® for Windows software, a user's custom program, or a dumb terminal. Any of these three methods allow you to issue of commands and the read and change parameters in the **SERVOSTAR 600**. The dumb terminal is the most basic method of controlling the drive. The most friendly way of controlling the **SERVOSTAR 600** is through **MOTIONLINK**. This package provides a robust set of control menus, function buttons, graphical displays, and dialog boxes to detail the operation of the drive. **MOTIONLINK 600** also features its own terminal mode of operation, offering the user a friendly communication tool that far exceeds other terminal packages. For simplicity sake, this section 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 issuing section that follows.

SPECIFICATIONS

Communication Port: RS-232C
 Baud Rate: 9600 BPS - S600
 38,400 BPS - S300
 Cable Length: 15 Meters Maximum (~50 feet)
 Data Type: Data (8) + Stop (1)
 Parity: None
 Handshake: None
 Character: ASCII code
 Data Error Check: Checksum (optional with PROMPT=3)

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 (parameters)
<ul style="list-style-type: none"> - echoes characters - transmit prompt - Ack/Nak messages - parameter values - error/fault messages 	<ul style="list-style-type: none"> - commands - parameter values - parameter queries 	ACKMODE MSG

The following is a set of examples demonstrating the SERVOSTAR 600 serial protocol between the drive and a host. To obtain a more complete demonstration, please contact the factory at the number listed at the end of this application note.

BASIC COMMAND/PARAMETER SETUP:

ADDR=0, PROMPT=1, MSG=1

PARAMETER R/O

EXAMPLE 1-READ MAXIMUM SYSTEM CURRENT (IPEAK) Example Value: 5

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

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

Sequence #			15	16	17	18
User Enters:						
Drive Returns:	5	<Cr>	<Lf>	-	-	>

PARAMETER R/W

EXAMPLE 2: WRITE 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>	-	-	>

COMMANDS with MULTI-LINE VALUES (R/O)

EXAMPLE 3 - DUMP Command



NOTE

These commands normally have a longer delay because of the amount of data 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>	-	-	>

COMMANDS (R/W)

EXAMPLE 4- EN Command

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>	-	-	>

FLAGS R/O

EXAMPLE 7: READ - 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:	>

STRING VALUES

EXAMPLE 8: LDUMP



NOTE

<MOTOR> results in an alpha/numeric result describing the motor.

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

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

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

SERIAL CHECKSUM

ADDR=0, PROMPT=3, MSG=1

EXAMPLE 1 - ACC Example Value: 25001



NOTE

The last two characters of the hex value sum is entered by the user before the carriage return.

CHARACTER	HEX VALUE	DEC VALUE
A	41	65
C	43	67
C	43	67
<sp>	20	32
2	32	50
5	35	53
0	30	48
0	30	48
1	31	49
SUM=	1DF (H)	479 (D)

Modulo 256 for **1DF(H) = DF(H)**

The first character of checksum: **D(H) + 30(H) = 3D(H)** or “=” ASCII character

The second character of checksum: **F(H) + 30(H) = 3F(H)** or “?” ASCII character

The command with checksum is: **ACC 25001=?**

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

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

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

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

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

MULTIDROP MODE (S600 only)

BASIC COMMAND/PARAMETER SETUP: ADDR=3, PROMPT=1, MSG=1



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

EXAMPLE 1 - \ Command

A value other than 0, puts the system in Multidrop Mode and changes 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>	-	-	>

CASE 1: PARAMETER R/W

EXAMPLE 2 - 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>	-	-	>

Drive /Fault Error

BASIC COMMAND/PARAMETER SETUP: ADDR=0, PROMPT=1, MSG=2

EXAMPLE 1 - Thermostat Trip



NOTE

<Trip> is a condition in which there is no response from the drive or there is a misconnection among the wires.

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:	>

Parameter Definitions

ADDR

Description Drive Address. The entry is the station address (1...63) of the amplifier. This number is required by the fieldbus (CANopen, PROFIBUS DP, SERCOS, etc.) and for the parameter setting of the servo amplifier in multi-axis systems for an unambiguous identification of the servo amplifier within the system. You can also use the keys on the front panel of the servo amplifier to set the station address (see *Installation Manual*).

Type	Standard Parameter (Read / Write)	Units	-
Range	0 to 63	Default	Serial No. & 0x1F
Opmodes	All	Drive Status	-

ERRCODE[*]

Description Activated Fault Messages (1 Number). This parameter returns an error code (corresponds to error displays Exx in the front-panel LED display) and a brief description of the error for any active errors in the servo amplifier.

Type	Command	Units	-
Range	0 to 0xFFFFFFFF (1 Bit is 1 Fault)	Default	-
Opmodes	All	Drive Status	-

LOAD

Description Load Data from the EEPROM to RAM. This command automatically executes on power up (see SAVE).

Type	Command	Opmode	All
-------------	---------	---------------	-----

MSG

Description Enable / Disable All Messages. This flag enables the sending of messages from the drive to the serial port:

If MSG 2 is set, execution of the individual initialization steps is signaled through the serial interface when the amplifier is switched on (initialization phase). This setting should only be used for test purposes (e.g. during commissioning).

Since the drive setup software Drive.exe only works with MSG 1, the MSG 2 setting can only be implemented with the help of a terminal program (not in the terminal window of the operating program).

Type	Switch Parameter (Read / Write)	Units	-
Range	0 - 2	Default	0
Opmodes	All	Drive Status	-

PROMPT

Description Prompt Enable. This parameter configures the serial port prompt put out by the drive:

- 0 – no character response or prompt to the host
- 1 – character response and prompt to the host
- 2 – reserved
- 3 – character response, prompt, and checksum to the host



NOTE

PROMPT=1 is required for proper operation of MOTIONLINK 600.

Type	Switch Parameter (Read / Write)	Units	-
Range	0 - 3	Default	1
Opmode	All	Drive Status	-
Syntax Tr/R	[Data] / <Data>	EEPROM:	No

SCAN (S600 only)

Description Bus Scan for Slave Drives. This command determines the quantity and addresses of all drives connected via CANopen (to the first drive) in a multidrop configuration.

Type Command **Units** -

Range No. (0 to 63) and Addresses of Slaves (1 to 63) **Default** -

Opmode All **Drive Status** -

Syntax Tr/R - \ <Data> **EEPROM** -

For a CAN network with several drives connected, there is an option for using a serial connection to one of the devices (master) to communicate with all the other amplifiers. To do this, initiate SCAN on the master device. This performs an automatic detection of all the connected drives. The response to SCAN contains the total number and a list of the addresses of all the detected drive devices

The time taken to carry out this command is dependent on the baud rate (**CBAUD**) that is has been set for CAN, and is in the range from 1 second (at 1 Mbaud/1 Mbps) to 37 seconds (at 10Kbaud/10 kbps).

Example:

Three S600 are connected by CAN BUS: DRIVE1 has address 5
DRIVE2 has CAN address 7,
DRIVE3 has CAN address 8.

SCAN gives back the address from each station.

```
SCAN {Command: scans the via CAN bus connected drives}
2 7 8 {Response: CAN-Communication to 2 drives with the addresses 7 and 8}
\ 7 {Command: select drive with address 7}
\ {Command: inquiry which drive is connected to the PC}
7 {Response: drive 7 is selected}
\0 {Command: select drive 1, in this case the Servo - Master with original – address 5}
```

Customer Support

Danaher Motion products are available world-wide through an extensive authorized distributor network. These distributors offer literature, technical assistance, and a wide range of models off the shelf for the fastest possible delivery. Danaher Motion sales engineers are conveniently located to provide prompt attention to customer needs. Call the nearest office for ordering and application information and assistance or for the address of the closest authorized distributor. If you do not know who your sales representative is, contact us at:

Danaher Motion
203A West Rock Road
Radford, VA 24141 USA
Phone: 1-540-633-3400
Fax: 1-540-639-4162
Email: customer.support@danahermotion.com
Website: www.DanaherMotion.com