Automation Modules, Inc.

S100

TECHNICAL REFERENCE MANUAL

Revision 2.0

AUTOMATION MODULES, INC.
870 10TH LANE
FOX ISLAND, WA 98333

PHONE: 1-253-549-4868 / FAX: 1-253-549-4866

Table of Contents

1. Introduction

The S100 is a one axis stand-alone integrated controller / driver, with input / output (I/O) capabilities, designed primarily for the control of DC brush type motors or actuators with it's integrated driver.

The S100 implements a mnemonic type command instruction set via a standard RS-232 serial communications interface. These commands can be executed directly or used to create command macros which are stored in the onboard nonvolatile RAM (NVRAM).

The S100 can interface to the real world via the onboard DC motor driver, a quadrature type encoder interface, 8 channels of HCT TTL digital input and 8 channels of HCT TTL type digital output, with additional TTL inputs serving for limit, home and fault functions, 4 channels of 10-bit analog to digital (A/D) conversion (1 of which is reserved for monitoring amplifier output current), and an RS-232 serial communications link. A proprietary RS-422 interface is provided for I/O expansion modules.

1.1 Specifications

Description	Stand-Alone 1 Axis Servo Motor Controller / Driver
Operating Modes	Position, Velocity, Torque
Filter Algorithm	PID
Max. Servo Loop Rate	200 µS
Trajectory Generator	Trapezoidal
Servo Position Feedback	Incremental Encoder with Index
Output (Standard)	PWM Motor Drive, 3 Amps Cont. and 6 Amps Peak at 50 VDC Max.
PWM Frequency	Approximately 19.531 KHz
Encoder and Index Input	Single-ended or Differential
Encoder Supply Voltage	5 VDC
Encoder Input Voltage	5.5 VDC Max., -0.1 VDC Min.
Encoder Count Rate	2 Million Quadrature Counts per Second
Position Range	31 Bits
Velocity Range	31 Bits
Acceleration Range	31 Bits
General Purpose Digital I/O	8 HCT TTL Inputs, 8 HCT TTL Outputs
Dedicated Digital Inputs	Limit+, Limit-, Home and Fault, all TTL
Analog Inputs	4 Channels With 10-Bit Resolution, 3 are user accessible
Expansion I/O	Optional Expansion to 64 I/O
Communication Interface	RS-232 Serial Interface, Adjustable Baud Rate, 8 Bits, 1 Stop Bit, No Parity, XON/XOFF Handshake
Supply Voltage	+11 To +50 VDC
Motor Voltage	+12 To +48 VDC
Dimensions	Approximately 5.0" Long by 3.3" Wide by 1.1" Thick
Weight	Approximately 1 Lb.

Table 1. Specifications.

1.2 Digital I/O Interface

The S100 includes 8 channels of HCT TTL general purpose digital input and 8 channels of HCT TTL general purpose digital output. Additionally, there are four channels of TTL dedicated digital input. The general purpose I/O are buffered with 74HCT541. An expansion interface allows for optional expansion of I/O to 64 channels.

1.2.1 Dedicated Digital Inputs

The S100's dedicated digital inputs are Limit+, Limit-, Home and Fault. Figure 1. S100 Dedicated Input. illustrates one of these inputs. All of **these inputs are active low** and have 2.7K ohm pull up resistors. To activate one of these inputs, the user need only connect that input to the controller's ground.

The Limit inputs are intended for signaling the S100 that an axis has reached it's end of travel. When such an event occurs, the S100 can ignore the event or stop the servo in some controlled fashion. The Home input is for detecting some sort of "home position" sensor. This can be used with the encoder Index input to implement a very accurate homing method. A typical use for the Fault input is for an external device to signal a fault condition such as motor/actuator over-temperature.

Note: The external Fault input is tied to the internal over-temperature signal from the onboard driver. When a fault condition occurs, that is either the internal over-temperature signal or external Fault signal go active, the 16-bit internal variable FCNT (see Internal Variables) begins to increment at a rate of once per millisecond. If the fault condition clears, then the FCNT variable is also cleared. If the fault condition remains present long enough for the FCNT variable to count up to the value assigned to the FCMP variable, then the Fault bit in the status word will be set and the servo will be disabled (assuming the Fault interrupt has not been enabled). The default value for FCMP is 10000 which will give a 10 second delay before causing the Fault bit to be set.

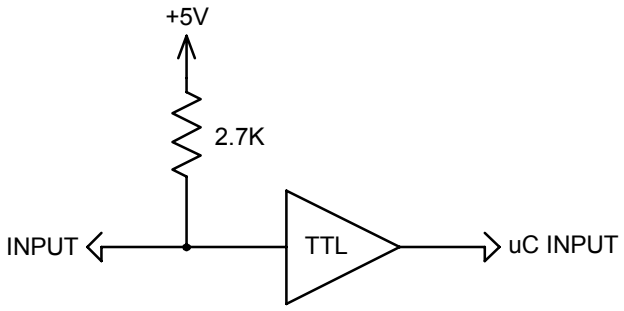


Figure 1. S100 Dedicated Input.

1.2.2 General Purpose Digital Inputs

Figure 2 illustrates one of the S100's general purpose inputs. These inputs are of the type HCT TTL. Each of these inputs has a 15K pull up resistor.

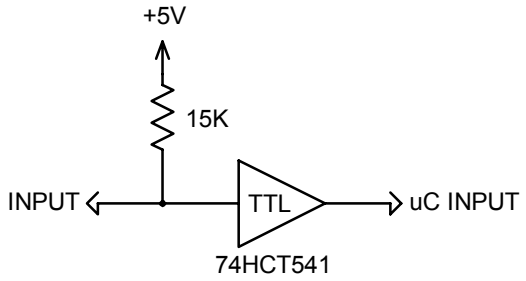


Figure 2. S100 General Purpose Input.

1.2.3 General Purpose Digital Outputs

Figure 3 illustrates one of the S100's general purpose outputs. These outputs are of the HCT TTL type.

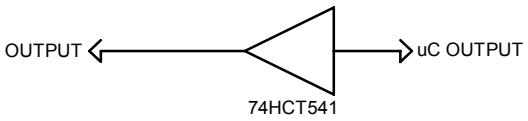


Figure 3. S100 General Purpose Output.

1.2.4 Digital I/O "States"

There are several commands that deal with controlling the general purpose digital I/O.

The Channel High (CH) and Channel Low (CL) commands provide the user with the ability to determine whether a channel is ON in the HIGH state (CH) or ON in the LOW state (CL). This is analogous to a switch and to whether it is normally open or normally closed. The Channel On (CN) and Channel Off (CF) commands do exactly as they imply in that they will turn a given output either ON or OFF, which will make that output either HIGH or LOW depending on the CH and CL commands as stated previously.

The (CH) command causes the following interpretation of the inputs and outputs:

- A HIGH output is considered to be ON (e.g., Channel On “CN” command).
- A LOW output is considered to be OFF (e.g., Channel Off “CF” command).
- A HIGH input is considered to be ON (e.g., Do If On “DN” command).
- A LOW input is considered to be OFF (e.g., Do If Off “DF” command).

The (CL) command causes the following interpretation of the inputs and outputs:

- A HIGH output is considered to be OFF (e.g., Channel Off “CF” command).
- A LOW output is considered to be ON (e.g., Channel On “CN” command).
- A HIGH input is considered to be OFF (e.g., Do If Off “DF” command).
- A LOW input is considered to be ON (e.g., Do If On “DN” command).

Input Voltage	“CH”	“CL”
HIGH	ON	OFF
LOW	OFF	ON

Output Voltage	“CH”	“CL”
HIGH	ON	OFF
LOW	OFF	ON

Table 2. I/O States.

Another feature of the digital input system is the ability for software input debouncing. All of the general purpose digital inputs are automatically sampled once every millisecond. Depending on the debounce delay set by the Input Debounce (ID) command, a given input must remain in the same state during one or more samplings before it is considered valid. If an input were to be found in a changed state during a sampling, the input would become invalid and the debounce delay would be restarted. If no or "0" debounce delay is used, then no input debouncing is performed. For example: if a "ID5" command has been issued, then a given input must remain in the same state for 5 samplings or for 5 milliseconds before it is considered valid and the change is detectable.

1.2.5 I/O Technical Specifications

1.2.5.1 General Purpose I/O Nominal Specifications.

Symbol	Parameter	Conditions	Typ.	Units
V_{IH}	Minimum High Level Input Voltage		2.0	V
V_{IL}	Maximum Low Level Input Voltage		0.8	V
V_{OH}	Minimum High Level Output Voltage	$\|I_{OUT}\|$ <= 6.0mA	3.84	V
V_{OL}	Maximum Low Level Output Voltage	$\|I_{OUT}\|$ <= 6.0mA	0.33	V
I_{IN}	Maximum Input Current	V_{IN} = HIGH or LOW	$\pm$0.1	uA

Table 3: General Purpose I/O Specifications

1.2.5.2 Dedicated I/O Nominal Specifications.

Symbol	Parameter	Conditions	Typ.	Units
V_{IH}	Minimum High Level Input Voltage		1.9	V
V_{IL}	Maximum Low Level Input Voltage		0.9	V
I_{IN}	Maximum Input Current	V_{IN} = HIGH or LOW	$\pm$0.5	uA

Table 4: Dedicated I/O Specifications

1.3 Encoder Interface

The S100 has one channel of quadrature type encoder interface with optional index signal input and the ability to supply +5 VDC at a minimum of 50 mA (or greater depending on other demands put on the internal 5 VDC power supply). The phase A+ and phase B+ inputs are pulled up to +5 VDC with 2.7K resistors, and the phase A- and phase B- inputs are biased at +2.5 VDC with 2.7K resistors. This arrangement which will accommodate both open collector and totem pole single-ended output encoders or differential output encoders. The phasing of the channels as well as the index signal sense can be changed via program command.

1.4 Output Driver Interface

The S100 onboard output driver is a PWM switching amplifier capable of supplying 3 Amps continuous and 6 Amps peak (for 200 mS minimum) at a switching frequency of approximately 19.531 KHz. This driver is intended for driving DC brush type motors or actuators. The peak voltage output to the motor will be nearly that of the main power supply.

The output driver includes an over-temperature sensor. If this sensor determines that the amplifier's temperature is greater than 140° C, the amplifier will then be disabled and the Over-Temperature bit will be set in the status word.

1.5 Analog to Digital Conversion (A/D) Interface

The S100 provides a 4 channel, 10 bit A/D conversion interface with a +5 VDC reference and analog ground. **For reverse compatibility purposes the A/D interface is actually ten channels but the user is only given access to channels 7, 8 and 9 while channel 0 is used internally for monitoring the output current of the onboard driver. The other channels are unavailable and should be ignored.**

Whenever a Tell Analog "TA" or Get Analog "GA" command is issued, the specified A/D channel is converted and the result is either reported or stored for access by the user. Also, whenever the servo loop is executed, the "current monitoring" channel is converted and the result is stored for later access.

1.6 Serial Interface

The S100 communicates with a host computer or a "dumb" terminal via an RS-232 serial interface. The baud rate is user selectable from 300 to 19,200 baud with **9600 baud being the default.** Characters are fixed at 8 bits in length with 1 stop bit and no parity. Software XON / XOFF handshaking is provided. Hardware handshaking is not supported at this time.

2. Macro Interrupt System

The S100 employs a "Macro Interrupt System" to provide additional versatility in programming the S100. This system comprises 12 interrupt sources with corresponding vectors. When an interrupt's source is enabled for operation and then becomes active, the current macro being executed is saved to a so called macro stack and execution of the macro specified by that interrupt's vector table entry begins. This happens to be similar procedure to that which the Macro Call (MC) command follows.

2.1 The Interrupt Vector Table

The Interrupt Vector Table consists of an entry for each interrupt source and each entry will correspond to that interrupt's level (level 0 = entry 0, level 1 = entry 1, etc.). A particular table entry must be loaded with the number of a valid macro to be executed should that interrupt source become active. The method for loading a vector table entry is provided by the Load Vector (LV) command. The user must first use the Accumulator Load (AL) command to set the number of the macro for a vector. The LV command is then used to transfer the low 8-bits of the accumulator to the vector table entry specified by the LV command. If an interrupt is generated and that vector table entry has not been defined (equal to 0) then the interrupt will not be executed. Note that this implies that macro "0" cannot be used as an interrupt macro. If an interrupt is generated and it's vector table entry has been defined but the macro it specifies has not, then an error will be reported.

2.2 Enabling and Disabling Interrupts

Loading a vector table entry will not enable an interrupt for operation. The Enable Vector (EV) command must be used for this purpose. When the EV command is used, it will enable the interrupt source (specified with the command) to function. In the event that it is necessary to disable an interrupt source, there is a Disable Vector (DV) command that functions in a similar manner as the EV command.

In order to prevent multiple or continuous interrupts, **as an interrupt is taken it is automatically disabled.** This means that the user must re-enable that interrupt using the EV command before it will occur again.

2.3 Interrupt Sources

The following table lists all the possible interrupt sources.

Interrupt Source	Level / Vector	Interrupt Source	Level/Vector
Axis Error	31	Reserved	15
Reserved	30	Reserved	14
Reserved	29	Reserved	13
Reserved	28	Reserved	12
Axis Fault	27	Reserved	11
Reserved	26	Reserved	10
Reserved	25	Reserved	9
Reserved	24	Reserved	8
Axis Limit	23	Digital Input 7	7
Reserved	22	Digital Input 6	6
Reserved	21	Digital Input 5	5
Reserved	20	Digital Input 4	4
Axis IP/IR	19	Digital Input 3	3
Reserved	18	Digital Input 2	2
Reserved	17	Digital Input 1	1
Reserved	16	Digital Input 0	0

Table 5. Macro Interrupt Sources.

The Axis Error interrupt indicates that the position following error for the axis has exceeded the limit set by the Set Error (SE) command. Normally, when this limit is exceeded, the servo is disabled and the "Error" bit in that axis' status word is set. **If the interrupt for this condition is enabled, the "Error" bit will still be set but the servo will not be disabled.**

The Axis Fault interrupt indicates that a fault condition (usually an over-temperature condition) has arisen. Normally, when this condition is detected, the servo is disabled and the "Fault" bit in the status word is set. **If the interrupt for this condition is enabled, the "Fault" bit will still be set but the servo will not be disabled.**

The Axis Limit interrupt indicates that either a Limit+ or Limit- condition for the axis has been detected. Whether or not a limit input will be recognized is determined by the Limit On (LN) and Limit Off (LF) commands. The action taken is determined by the Limit Mode (LM) command.

Digital Inputs 00 - 07 provide 8 levels of undedicated, user definable interrupts. The interrupt for a given input will be active when that input is active.

2.4 Interrupt Priority

If more than one interrupt source becomes active at the same time, then the source with the higher level will be executed first. Level/vector 31 has the highest priority and level/vector 0 has the lowest priority.

2.5 Interrupt Completion

Once an interrupt macro (or set of macros) has finished executing, a Return from Call (RC) command or an undefined macro may be used to cause a return from the interrupting macro back to the interrupted macro where command execution will continue from where it was interrupted (see MS command). In cases where it is undesirable to return to the interrupted macro, the Unpush Macro (UM) command can be used to remove the previously pushed macro from the macro stack. This command can also be used to completely reset the macro stack in order that the user program can be restarted.

2.6 Interrupt Latency

Interrupt sources are sampled before each command in a macro is executed. This means that the amount of time that an interrupt is held off before execution (also known as interrupt latency) depends on how long it takes the previous command to complete. For most commands this delay will be imperceptible to the user.

Commands such as Wait (WA), Wait for Edge (WE), Wait for Stop (WS), Wait for Off (WF), Wait for On (WN) and Wait for Index (WI) would normally be a source of unacceptable delay in that they can quite often be indeterminate in length. This problem has been avoided by making these instructions interruptible. For example, if a WA10000 command (a 10 second delay) is currently in progress and an interrupt comes along, the remaining delay period will be saved and then returned to after the interrupt has completed. If the interrupt were to take 3 seconds to execute, then the total wait time of the WA10000 command would be extended to 13 seconds.

The Position Mode (PM), Torque Mode (QM), Velocity Mode (VM), Wait for Position Absolute (WP) and Wait for Position Relative (WR) commands and any command that uses the serial communications link are all commands that could cause unacceptable interrupt latency. Therefore, their usage should be carefully considered where interrupts are possible.

3. Entering Commands

Immediately after power-up, the S100 is ready to accept commands. To verify this, you can hit the ESC key. If everything is working properly, this should cause a greater than sign (">") prompt to appear on your display. If not, you need to verify that the power and communications connections are correct and verify the compatibility of the communications protocol.

Commands are entered via a "dumb" terminal or host computer such as a PC compatible. Commands sent to the S100 should consist of standard ASCII characters, and the command lines should be followed by a carriage return. Linefeeds are not necessary since they are used for formatting and therefore they are ignored. As characters are entered at the keyboard, they should be echoed on your display. If your display echoes its own transmitted characters, you will want to issue the Echo Off (EF) command; otherwise, the Echo On (EN) command (which is the default mode) should be issued. If you enter an invalid command, the S100 will respond with a question mark "?" and space " " followed by a code indicating the type of error. These codes are listed in Appendix A, S100 Error Code Definitions. Also, the "Status" LED on the controller will begin to flash.

If you make a mistake when entering a command, you can backspace to correct the error. If you are entering commands and change your mind, hitting the ESC key will cancel the line and give a new ">" prompt.

Once a command line has been entered and has finished executing, **hitting the RETURN key will cause the same command line to be re-executed.** While a set of commands are executing, hitting the space bar will cause command execution to pause until the space bar is hit again. Also, **if the ESC key is hit during execution or pause, command execution will be terminated,** and you will receive a new ">" prompt.

Command instructions are intended for use with the following syntax:

```
Command[Argument]<CR>
```

or...

```
Command[Argument],Command[Argument],...etc.
```

The numerical range of an argument will vary depending on the command with which it is used. The mathematical interpretation of the argument will depend on whether the Decimal Mode (DM) or Hexadecimal Mode (HM) was the last issued (DM is the power on default). Both decimal and hexadecimal numbers less than zero should be entered with a preceding minus "-" sign. If no argument is given, then it will be assumed as "0". The exceptions to this are the Macro Define (MD), Macro Jump (MJ), Macro Call (MC),Macro Sequence (MS), Reset Macro (RM) and Tell Macro (TM), commands. It should be noted that commands can be strung together by using commas, up to a maximum line length of 127 characters.

If a command line is ended by a ";" and a comment, i.e...

```
>SG1000,SD5000 ; Set filter gains.<CR>
```

then the ";" and anything following it to the end of the line will be ignored. This feature is not particularly useful if you are entering commands manually, as comments are not retained by the S100. However, if commands are downloaded to the S100 from a host computer, the ability for line comments can make program documentation possible and desirable.

3.1 Downloading Commands

In many cases, it is more convenient to enter commands using a text editor on a host computer and then download that text file to the S100 using a communications program such as ProComm® or the Microsoft® Windows™ Terminal program. Whatever communications software is used, it must have the ability to provide a short delay (approx. 100 mS) after transmitting each line to give the S100 time to interpret and store the commands that were just sent.

4. Introduction to Commands

The S100 command instructions are varied and consist of several categories of purpose. The command descriptions will be detailed by these categories.

4.1 Parameter Commands

The parameter setting commands are considered to be those for setting the operating conditions of the servo system (i.e. PID filter gains, velocity, acceleration and etc.).

Command: DBn -- Dead-Band --

Argument: 0 <= n <= 16383
Default: 0

This command sets the position following error dead-band. The purpose for the DB command is to allow an acceptable static position error for which there will be no restoring force. This has the affect of reducing or eliminating "hunting" which is the continuous movement at or about a position in trying to seek that position. This is useful for applications that cannot tolerate this condition. Please note that the DB command is only in effect when the servo is not in motion (when the Trajectory Complete bit is set in the servo status word).

Related Commands: TF

Command: FAn -- Feed-forward, Acceleration --

Argument: 0 <= n <= 32767
Default: 0

This command allows for the adjustment of the PID digital filter acceleration feed-forward term for the servo.

During the course of a Position Mode (PM) or Velocity Mode (VM) move, at any point during acceleration or deceleration (with a consistent load), the ideal required value of the servo output is fairly consistent and somewhat predictable.

During acceleration or deceleration:

OUTPUT = (VELOCITY * FV_CONSTANT) + (ACCELERATION * FA_CONSTANT)

During constant velocity:

OUTPUT = (VELOCITY * FV_CONSTANT)

If this value can be dynamically predicted and summed with the output of the PID digital filter, in effect, it reduces the burden of the PID filter to make lead/lag corrections based of the following error, thereby enhancing performance.

Related Commands: FV, OO

Command: FF -- Fail Input Off --

Default: Off

This command has no effect but is retained for backward compatibility purposes.

Command: FN -- Fail Input On --

Default: Off

This command has no effect but is retained for backward compatibility purposes.

Command: FRn -- Set Derivative Sampling Period --

Argument: 0 <= n <= 127
Default: 0

This command allows for the adjustment of the derivative sampling interval for the servo. The period of this interval can be calculated by the following:

T = (n+1) * S * 0.000100

where "T" is the period in seconds, "n" is the FR command argument and "S" is the sample period count specified by the Servo Speed (SS) command. For example, if the value previously set by the SS command is 10 and the value set by the FR command is 1, then the derivative sample period will be:

(1+1) * 10 * 0.000100 = .002000 S or 2 mS

This command is useful in tuning the PID servo loop to the inertial properties of the system.

Related Commands: RI, SS

Command: FVn -- Feed-forward, Velocity --

Argument: 0 <= n <= 32767
Default: 0

This command allows for the adjustment of the PID digital filter velocity feed-forward term for the servo.

During the course of a Position Mode (PM) or Velocity Mode (VM) move, at any point along the way (with a consistent load), the ideal required value of the servo output is fairly consistent and somewhat predictable.

During acceleration or deceleration:

OUTPUT = (VELOCITY * FV_CONSTANT) + (ACCELERATION * FA_CONSTANT)

During constant velocity:

OUTPUT = (VELOCITY * FV_CONSTANT)

If this value can be dynamically predicted and summed with the output of the PID digital filter, in effect, it reduces the burden of the PID filter to make lead/lag corrections based of the following error, thereby enhancing performance.

Related Commands: FA, OO

Command: ILn -- Set Integration Limit --

Argument: 0 <= n <= 16,383
Default: 0

This command clamps the level of influence that the PID digital filter integral term can use to reduce the static position error thereby reducing integral “wind up”. When properly adjusted, this can enhance loop stability and operation. The Integral Limit (IL) and Set Integral Gain (SI) must both be set to a non-zero value in order for the integral term to have any effect.

Related Commands: SI

Command: LFn -- Limit Switch Input Off --

Argument: 0 <= n <= 3
Default: 0

This command disables one or more of the limit switch inputs. The valid arguments to this command determine which inputs will be disabled and are as follows:

n	Limit Switch Inputs Disabled
0, 3 or no argument	Limit+ and Limit-
1	Limit+
2	Limit-

Related Commands: LM, LN

Command: LMn -- Limit Switch Input Mode --

Argument: 0 <= n <= 3
Default: 0

This command is used to select how the S100 will react when a limit switch is activated. The valid arguments for this command are as follows:

n	Action
0	Turn servo off, continue commands
1	Stop abruptly, continue commands
2	Decelerate smoothly, continue commands
3	Interrupt only

In all cases, the Error flag in the status word will be set. This will prevent the S100 from moving the servo until the flag is cleared by issuing the Motor On (MN) command. Before this command will have any effect, the limit switch must be enabled with the Limit Switch On command (LN).

Related Commands: LF, LN

Command: LNn -- Limit Switch Input On --

Argument: 0 <= n <= 3

This command is used to enable one or both of the limit switch inputs. Once enabled, the servo will be stopped or turned off if a limit switch input goes active. At the same time the Limit Switch Tripped and Error Flags will be set in the status word. These flags will remain set until the servo is turned back on with the Motor On (MN) command. Once the servo is turned back on, it can be moved out of the limit switch region with any of the standard motion commands. The argument to this command determines which of the limit switch inputs will be enabled. The coding is as follows:

n	Limit Switch Inputs Enabled
0,3 or no argument	Limit+ and Limit-
1	Limit+
2	Limit-

Related Commands: LF, LM

Command: OOn -- Output Offset --

Argument: -32767 <= n <= 32767
Default: 0

This command allows the user to set a continuous output for the servo. In certain applications, such as an overhanging load, there will be a continuous burden placed upon a servo axis. In cases like these, where there is a predictable load, the OO command can be used to provide a continuous restoring force that will be combined with the output of the PID digital filter. This has the affect of improving the performance of the PID digital filter in that because it is not saturated with static load, it has a better dynamic response to load disturbances.

Related Commands: FA, FV

Command: PHn -- Set Servo Phasing --

Argument: 0 <= n <= 63
Default: 0

This command is used to set the output polarity, encoder phasing, Index input sense, Home input sense, Limit+ and Limit- input sense. The polarity of the output will determine whether the servo is driven in a direction that reduces or increases position error. The encoder phase will determine whether the position count will increase or decrease for a valid encoder input sequence. The Index sense determines what logic edge will cause the Index input to be active. The Limit+, Limit- and Home sense determines whether these signals are active “on” or active “off.

To determine the argument to be used with the PH command, use the follow table and add the required values together.

	Add to 'n'
Output Phase Normal	0
Output Phase Reversed	1
Encoder Phase Normal	0
Encoder Phase Reversed	2
Index Active Level Low	0
Index Active Level High	4
Home Sense Active "ON"	0
Home Sense Active "OFF"	8
Limit+ Sense Active "ON"	0
Limit+ Sense Active "OFF"	16
Limit- Sense Active "ON"	0
Limit- Sense Active "OFF"	32

For example, if it were necessary to reverse the encoder phasing and to set the Limit+ and Limit- inputs to active "OFF", then 'n' would be (2 + 16 + 32) or 50. The default phasing and sense is equivalent to issuing this command with a argument of 0.

Command: RIn -- Sampling Rate of Integral --

Argument: 0 <= n <= 127
Default: 0

This command allows for the adjustment of the PID digital filter integral sampling interval for the servo. The period of this interval can be calculated by the following:

T = (n+1) * S * 0.000100

where "T" is the period in seconds, "n" is the RI command argument and "S" is the sample period count specified by the Servo Speed (SS) command. For example, if the value previously set by the SS command is 10 and the value set by the RI command is 1, then the integral sample period will be:

(1+1) * 10 * 0.000100 = .002000 S or 2 mS

This command is useful in tuning the PID servo loop to the inertial properties of the system.

Related Commands: FR

Command: SAn -- Set Acceleration --

Argument: 0 <= n < 1,073,741,823
Default: 0

This command sets the acceleration rate for the servo. The 32 bit argument to this command is scaled by 65536. This number determines how much the servo's velocity will be altered by each servo loop interval (determined by the Servo Speed "SS" command) while it is accelerating or decelerating. If this command is executed during a Position Mode move, it will be ignored.

Example:

Encoder: 500 lines or 2000 Counts/Rev
Desired Acceleration: 75 Rev/Sec²
Servo Loop Interval: 1,000 Hz

9830.4 = ((75 Rev/Sec * 2000 Counts/Rev) / 1000 Hz²) * 65536

To achieve an acceleration of 75 Rev/Sec², the command SA9830 would be issued. A simpler way to calculate the acceleration argument would be to determine a constant for your application by which to calculate desired acceleration.

131.072 = K = ((1 Rev/Sec * 2000 Counts/Rev) / 1000 Hz²) * 65536

9830.4 = 75 Rev/Sec * K

Please note that if the Set Acceleration (SA) command is used with an argument of "0", then you have commanded the velocity to change in steps of zero which means if the servo is stopped it will not be able to move, and if the servo is moving it will not be able to change velocity.

Related Commands: SS, SV

Command: SCn -- Set Current Mode Gain --

Argument: 0 <= n <= 32,767
Default: 0

This command sets the "current mode" gain used by Torque Mode (QM1 only, see QM command). This allows the response of the current error integrator to be adjusted to suit a given application. A low SC value will provide slow response to load changes while a high SC value will provide quick response to load changes. If SC is set is too low then inadequate control may occur. If the SC setting is too high then the system may be unstable. A good "general" value for SC is about 8000.

Related Commands: QM, SQ

Command: SDn -- Set Derivative Gain --

Argument: 0 <= n <= 32,767
Default: 0

This command sets the derivative gain term of the PID digital filter loop for the servo.

Related Commands: SG, SI, IL

Command: SGn -- Set Proportional Gain --

Argument: 0 <= n <= 32,767
Default: 0

This command sets the proportional gain term of the PID digital filter loop for the servo.

Related Commands: SI, SD, IL

Command: SIn -- Set Integral Gain --

Argument: 0 <= n <= 32,767
Default: 0

This command sets the integral gain term of the PID digital filter loop for the servo. The Set Integral Gain (SI) and Integral Limit (IL) must both be set to a non-zero value in order for the integral term to have any effect.

Related Commands: SG, SD, IL

Command: SQn -- Set [Maximum] Torque Level --

Argument: -32,767 <= n <= 32,767 in Torque Mode 0 (QM0)
-1,023 <= n <= 1,023 in Torque Mode 1 (QM1)
0 <= n <= 32,767 in Position Mode (PM)
0 <= n <= 32,767 in Velocity Mode (VM)
Default: 32,767

This command sets the maximum output level when in Position Mode (PM) or Velocity Mode (VM) and sets the desired output level when in Torque Mode (QM). When this command is issued in Position Mode (PM) or Velocity Mode (VM), its limiting effect will remain, in all modes of operation, until again changed. Note that an argument less than zero is allowed only in Torque Mode (QM).

Examples:

```
>PM,SQ20000<CR>   ; Set maximum output to +-20000.

>VM,SQ10000<CR>   ; Set maximum output to +-10000.

>QM,SQ20000<CR>   ; Set output to forward 20000.

>QM,SQ-10000<CR>  ; Set output reverse 10000.

>PM,SQ10000<CR>   ; Set maximum output to +-10000.
>QM0,SQ15000<CR>  ; Set output to forward 15000 but
                  ; will only achieve output of 10000
                  ; due to previous command.
```

Related Commands: PM, QM, SC, VM

Command: SSn -- Set Servo Speed --

Argument: 1 <= n <= 255
Default: 10

This command sets the servo loop interval. The period of this interval can be calculated by the following:

T = (n) * 0.000100

where "T" is the period in seconds and "n" is the SS command argument. For example, if the value set by the SS command is 10 then the servo loop period will be:

10 * 0.000100 = .001000 S or 1 mS

Although this command will accept a parameter of 0 or 1, it will be clamped to a minimum servo loop rate of 200 uS. The following table shows the minimum servo loop times:

# Axis Enabled	Minimum Loop Time
0	200µS
1	200µS
2	200µS
3	300µS
4	400µS

Using the SS command affects the Set Velocity (SV) and Set Acceleration (SA) commands in that they both are specified in terms of servo loop intervals. For example, if the servo loop rate is doubled, the velocity and acceleration would appear to have been halved. It should also be noted that the "SS" command will most likely affect the required settings of the PID digital filter.

Related Commands: SA, SV

Command: SVn -- Set Maximum Velocity --

Argument: 0 <= n < 1,073,741,823
Default: 0

This command sets the desired velocity (in quadrature counts per servo loop interval) for the servo . The 32 bit argument to this command consists of a 16 bit integer part and a 16 bit fractional part.

Example:

Encoder: 500 lines, 2000 Counts/Rev
Desired Velocity: 40 Rev/Sec
Servo Loop Interval: 1,000 Hz

5242880 = ((40 Rev/Sec * 2000 Counts/Rev) / 1000 Hz) * 65536

To achieve a velocity of 40 Rev/Sec, the command SV5242880 would be issued. A simpler way to calculate the velocity would be to determine a constant for your application by which to calculate desired velocity.

131072 = K = ((1 Rev/Sec * 2000 Counts/Rev) / 1000 Hz) * 65536

5242880 = 40 Rev/Sec * K

Related Commands: SA, SS

4.2 Reporting Commands

The reporting commands output data relevant to the operating status of the S100. Numerical output will be in the mathematical base determined by the Decimal Mode (DM) / Hexadecimal Mode (HM) commands and output will be followed by a carriage return and linefeed.

Command: TAn -- Tell Analog to Digital Channel 'n' --

Argument: 0 <= n <= 9

This command will cause a conversion on A/D channel 'n' and will display the result.

Note: Only inputs 0, 7, 8 and 9 are used.

Related Commands: GA

Command: TB -- Tell Breakpoint --

Default: "None"

This command reports the last programmed breakpoint value specified by the Interrupt on Position Absolute (IP) or the Interrupt on Position Relative (IR) commands. If no breakpoint value has yet been specified then the word "NONE is returned.

Related Commands: IP, IR

Command: TCn -- Tell State of I/O Channel 'n' --

Argument: 0 <= n <= 63
Default: "Off"

This command reports the current state of the I/O channel specified by 'n' and what type of logic the channel is (either active "on" or active "off"). The display is formatted as follows:

/xx = aaa

The presence of the "/" character indicates that a channel is active in the "off" state. A lack of the "/" character indicates that the channel is active in the "on" state. The "xx" indicates the channel number and the "aaa" indicates the channel's current state, either "ON" or "OFF". The following examples show all the possible combinations:

```
 01 = ON  ; Channel 1 is "ON" in the active "ON" state.
/01 = ON  ; Channel 1 is "ON" in the active "OFF" state.
 01 = OFF ; Channel 1 is "OFF" in the active "ON" state.
/01 = OFF ; Channel 1 is "OFF" in the active "OFF" state.
```

Related Commands: CF, CH, CL, CN

Command: TD -- Tell Derivative Gain --

This command reports the derivative gain value of the PID digital filter for the servo.

Related Commands: TG, TI, TL

Command: TE -- Tell Error --

This command reports the last error code caused by any command or macro. If no error has occurred then a "0" will be reported. Once the TE command has been issued, then the error code will be reset to zero. The TE command is useful for determining what error occurred in the case that no display was connected at the time.

Command: TF -- Tell Following Error --

This command reports the following error for the servo. This value is the difference between the current desired temporal position (or that which is reported by the Tell Optimal (TO) command) of the trajectory generator and the servo's current real position (or that which is reported by the Tell Position (TP) command).

Related Commands: DB

Command: TG -- Tell Proportional Gain --

This command reports the proportional gain value of the PID digital filter for the servo.

Related Commands: TI, TD, TL

Command: TI -- Tell Integral Gain --

This command reports the integral gain value of the PID digital filter for the servo.

Related Commands: TG, TD, IL

Command: TKn -- Tell (K) Constants --

Argument: 0 <= n <= 1

This command will display a number of internal settings depending on the value specified by 'n'. If 'n' is "0" or is not specified, then various parametric values for the servo will be displayed. If 'n' is "1", then various system parameters will be displayed.

Example display for the command "TK0":

```
>TK0

Parameter Values for Axis [1]

Proportional Gain ---------- (SG) = 0
Integral Gain ------------- (SI) = 0
Derivative Gain ------------ (SD) = 0
Integral Limit ------------- (IL) = 0
Current Gain --------------- (SC) = 0
Velocity Feed-forward Gain - (FV) = 0
Accel. Feed-forward Gain --- (FA) = 0
Output Offset -------------- (OO) = 0
Position Error Dead-Band --- (DB) = 0
Maximum Following Error ---- (SE) = 16383
Integral Sample Rate ------- (RI) = 0
Derivative Sample Rate ----- (FR) = 0
Phase and Sense Settings --- (PH) = 0
Maximum Velocity ----------- (SV) = 0
Acceleration --------------- (SA) = 0
Desired Direction ---------- (DI) = 0
Torque (output) Limit ------ (SQ) = 32767
Axis Type ------------------ (OM) = 0
```

Example display for the command "TK1":

```
>TK1

System Parameter Settings (group 1).

Axis 1 Enabled ------------- (EA) = Yes
Base 16 Input & Output -- (HM/DM) = Off
Character Echo ---------- (EN/EF) = On
Handshake ---------------- (HN/HF) = Off
Fail --------------------- (FN/FF) = Off
Servo Loop Rate ------------ (SS) = 2
Input Debounce/Delay ------- (ID) = 0
Phase and Sense Settings --- (CV) = 0
Intr. Vector Enable, HIGH (EV/DV) = 0
Intr. Vector Enable, LOW  (EV/DV) = 0
Firmware Revision ---------- (VE) = 1.04
```

Command: TL -- Tell Integration Limit --

This command reports the integration limit value of the PID digital filter for the servo.

Related Commands: TG, TI, TD

Command: TMn -- Tell Macros --

Argument: -2 <= n <= 255

The Tell Macro (TM) command will display the commands which make up any macros that have been defined. If 'n' >= 0 and 'n' <= 255, then the macro specified by 'n' be displayed. If 'n' = -1, then all the macros will be displayed preceded by the individual macro number. If 'n' =-2, then all macros will be displayed and will be preceded by the letters "MD" and the individual macro number. This is useful when downloading programs from the S100 to a computer in that you need not re-enter the Macro Define (MD) command and number at the beginning of each macro.

Related Commands: MD, RM

Command: TO -- Tell Optimal Position --

This command reports the desired position for the servo. This value may be different from the position reported by the TP command if the servo is moving or the controller is unable to drive the servo.

Related Commands: TT, TP

Command: TP - Tell Real Position --

This command reports the absolute position of the servo. It may be used to monitor motion during both the Motor On (MN) and the Motor Off (MF) states.

Related Commands: TT, TO

Command: TQ -- Tell Torque --

This command reports the current output torque being commanded by the servo axis 'a'. This value will be either that which is generated by the PID output or which is set by the user via the Torque Mode (QM).

Command: TRn -- Tell Contents of Register 'n' --

Argument: 0 <= n <= 511

This command reports the value contained by Register 'n'.

Related Commands: Register Commands

Command: TS -- Tell Status Word --

This command reports the operating status word of the servo. The response is coded into a single 32 bit value. The meaning of each bit is listed below:

Bit	Purpose
0	**Servo Enabled.** This bit will be set to "1" when the servo is enabled via the Motor On (MN) command. A "0" indicates that the servo has been disabled via one of the following reasons: a Motor Off (MF) command, a servo error due to excessive following error or over temperature condition, a limit event, an external fault.
1	**Servo Error.** A "1" in this bit position indicates that the maximum servo following error or a limit event has occurred.
2	**Over Temperature / Fault.** A "1" in this bit position indicates that an over-temperature condition has occurred or the external fault input has been activated.
3	**Breakpoint Reached.** A "1" in this bit position indicates that a previously defined breakpoint has been reached. This bit will be reset by any of the following commands: Motor On (MN), Interrupt on Position (IR), Interrupt on Relative Position (IR).
4	**Trajectory Complete.** A "1" in this bit position indicates that the servo has completed a commanded move. A "0" indicates that the servo is busy executing a commanded move.
5	**Servo Stopping.** A "1" in this bit position indicated that the servo has been commanded to stop. Upon stopping, this bit is then cleared.
6	**Current Direction.** This bit indicated the current direction of travel for the servo. (0 = Positive, 1 = Negative)
7	**Desired Direction.** This bit indicates the direction commanded by the Direction (DI) command. (0 = Positive, 1 = Negative)
8	**Reserved.**
9	**Reserved.**
10	**Looking for Index.** A "1" in this bit position indicates that the S100 is currently watching for an index pulse to occur as commanded by the Find Index (FI) command or Capture Index (CI) command.
11	**Looking for Edge.** A "1" in this bit position indicates that the S100 is currently watching for the Coarse Home input to go active as commanded by the Find Edge (FE) command.
12	**Reserved.**
13	**Coarse Home Input Active.** A "1" in this bit position indicates that the Coarse Home input is active.
14	**Capture Index Flag.** A "1" in this bit position indicates that the Find Index (FI) command is trying to capture the position on occurrence of an index pulse as opposed to initializing the position.
15	**Bad Input.** A "1" in this bit position indicates that invalid input has been entered via the Variable Input (VI) command. A "0" indicates that the last data obtained via the VI command was valid.
16	**Accelerating.** A "1" in this bit position indicates that the servo is currently accelerating.
17	**Position Mode.** A "1" in this bit position indicates that the servo is currently operating in position mode.
18	**Velocity Mode.** A "1" in this bit position indicates that the servo is currently operating in velocity mode.

19	**Torque Mode.** A "1" in this bit position indicates that the servo is currently operating in (voltage) torque mode.
20	**Current Mode.** A "1" in this bit position indicates that the servo is currently operating in (current) torque mode).
21	**Reserved.**
22	**Reserved.**
23	**Reserved.**
24	**Limit Mode Abort.** Used by Limit Mode (LM) to determine what action to take upon the occurrence of a limit switch event.
25	**Limit Mode Stop.** Used by Limit Mode (LM) to determine what action to take upon the occurrence of a limit switch event.
26	**Limit- Tripped.** A "1" in this bit position indicates that a Limit- input event has occurred and has been acted upon as set by the Limit Mode (LM) command.
27	**Limit- Enabled.** A "1" in this bit position indicates that the Limit- input has been enabled for action via the Limit Mode (LM) command.
28	**Limit- Active.** A "1" in this bit position indicates that the Limit- input is currently active.
29	**Limit+ Tripped.** A "1" in this bit position indicates that a Limit+ input event has occurred and has been acted upon as set by the Limit Mode (LM) command.
30	**Limit+ Enabled.** A "1" in this bit position indicates that the Limit+ input has been enabled for action via the Limit Mode (LM) command.
31	**Limit+ Active.** A "1" in this bit position indicates that the Limit+ input is currently active.

Command: TT -- Tell Target Position --

This command reports the current target position of the servo. This is the absolute position to which the servo was last commanded to move. It may have been specified directly with the Move to Position (MP) or Move Absolute (MA) commands or indirectly with the Move Relative (MR) command. If the servo axis is in Velocity Mode (VM), then the target position will track the current optimal position (or that which is reported by the Tell Optimal (TO) command).

Related Commands: TO, TP

Command: TV -- Tell Current Velocity --

This command reports the trajectory generator instantaneous commanded velocity for the servo. The value reported has the same units as the Set Velocity (SV) command, see the description of that command for further details. When in Torque Mode (QM) and not under control of the trajectory generator, the value reported indicates the actual velocity of the servo.

Related Commands: SV, SS

Command: VE -- Tell Version --

This command reports the firmware revision level. This revision level exists as a code in the internal RAM memory (see Register Commands). This code should be interpreted as the first byte being the major revision number and second byte being the minor revision number. This allows user programs to determine the firmware revision for compatibility purposes.

Related Commands: Register Commands

4.3 Motion Commands

Motion Commands are those commands that involve or cause the actual movement of a servo. Any position mode (PM) based motion will be carried out using a trapezoidal velocity profile with the maximum velocity determined by the Set Velocity (SV) command. The acceleration and deceleration are the same and both are determined by the Set Acceleration (SA) command.

During the execution of a Position Mode based motion, **the target position (MA or MR) and the maximum velocity (SV) may be changed at any time, however, changes to the acceleration (SA) will be ignored.** If a velocity mode (VM) based motion is under way, the maximum velocity (SV) and the acceleration (SA) may be changed at any time. During this type of motion, the target position will continuously be set equal to the current optimal position. If a torque mode (QM) based motion is under way, the torque setting (SQ) may be changed at any time. During this type of motion, the target and optimal positions will continuously be set equal to the current real position. In either PM, VM or QM, a Stop (ST) or Abort (AB) command will cause motion for the specified servo axis to cease. If a servo axis is in Position Mode (PM) and is commanded into Velocity Mode (VM), the motion will continue (only with no target destination) at the current maximum velocity set by the Set Velocity (SV) command. If a servo axis is in PM or VM and commanded into Torque Mode (QM), the servo output will be reduced to zero, and the unit will be placed in QM. If a servo axis is in VM and commanded into PM, the servo will be stopped at the current acceleration rate and placed into the PM mode. If a servo axis is in QM and commanded into VM, it will attempt to instantly assume the last known maximum velocity and remain in VM. If a servo axis is in QM and commanded into PM, the output will be reduced to zero, and the servo will begin station keeping in PM.

Command: AB -- Abort Motion --

This command causes an emergency stop. The servo stops abruptly but leaves the servo control loop enabled. The target position is changed to be equal to the present position. The command is used for stopping an undesired motion.

Related Commands: ST

Command: CI -- Capture on Index --

This command causes the position counter for the servo to be stored in the internal variable HREG upon receipt of an index pulse. For more information on HREG, see the Register Commands and the listing of the internal variables.

Related Commands: FI, WI

Command: DHn -- Define Home --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command causes the current position of the servo to be defined as 'n'. From then on, all positions used and reported for the servo axis will be relative to that physical position.

Related Commands: FE, FI

Command: DIn -- Set Direction --

Argument: n = 0 for positive, n = 1 for negative
Default: 0

This command sets the move direction for the servo when in velocity mode. Issuing this command with an argument of "0" will cause the servo to move in a positive direction while an argument of "1" will cause it to move in a negative direction.

Related Commands: SA, SV, VM

Command: FEn -- Find Edge of Coarse Home Input --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command is used to initialize the servo at a given position. It will remain in effect until the Home input goes active. At that time, the current position of the servo will be defined as 'n'. This command will neither start nor stop any servo motion. It is up to the user to initiate servo motion before issuing the command and to stop any motion after the command is completed.

Related Commands: WE

Command: FIn -- Find Edge of Index --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command is used to initialize the servo at a given position. It will remain in effect until the Index input goes active. At that time, the current position of the servo will be defined as 'n'. Like the Find Edge (FE) command, this command will not start or stop any servo motions; that is up to the user. Since an index pulse may occur at numerous points of the servo's travel (once per revolution in rotary encoders), a typical application will require a home signal to establish a coarse position reference before the index pulse can be used to fine tune that reference.

Related Commands: CI, WI

Command: GH -- Go Home --

This command causes the servo move to absolute position 0. This is equivalent to a Move Absolute (MA) command when the destination is 0. This command will initiate motion; therefore, a Go (GO) command is not required.

Related Commands: MA, GO

Command: GO -- Go (start motion) --

This command causes the servo to begin motion. When in the Velocity Mode (VM), the axis will accelerate to a constant velocity. When in the Position Mode (PM) the axis will begin to seek the specified target position. The servo must be in the "on" state for motion to occur.

Related Commands: MA, MR, PM, VM

Command: MAn -- Move Target Absolute --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command sets the target position of the servo to absolute position 'n'. The absolute position is relative to the point of initialization (home or 0). As the Move Absolute (MA) command will not initiate a motion, a Go (GO) command must be used. The servo's target position will be adjusted whether the servo is on or off.

Related Commands: GO, MR, PM

Command: MF -- Motor Off --

This command is used to place the servo in the "off" state. The servo's output will go to the null level. This command can be used to prevent unwanted motion or to allow manual positioning of the unit. When the servo is turned off, the target and the optimal positions will follow the real position.

Related Commands: MN

Command: MN -- Motor On --

This command is used to place the servo in the "on" state. When the servo is turned on, its target position is set to its current position so that the servo is not inclined to move. When this command is issued, it will disable the Home and Index interrupt sources and will reset the Error, Fault, Breakpoint reached, Looking for Edge, Looking for Index and Limit Tripped bits in the status word for axis 'a'.

Related Commands: LM, LN, MF

Command: MRn -- Move Target Relative --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command generates a relative target position of 'n' counts for the servo. Since the Move Relative (MR) command will not initiate a motion, a Go (GO) command must be used. The servo's target position will be adjusted whether the servo is on or off.

Related Commands: GO, MA, PM

Command: PM -- Enter Position Mode --

Default: Position mode

This command causes the servo to enter the position mode of operation. In this mode, the servo can be commanded to move to a specific target position. The moves will be executed using a trapezoidal velocity profile based upon parameters set by the Set Velocity (SV) and the Set Acceleration (SA) commands.

Related Commands: EG, QM, VM

Command: QMn -- Enter Torque Mode --

Argument: 0 for voltage mode, 1 for current mode
Default: Position mode

This command places the servo axis in the Torque Mode of operation. For mode 0 (or QM0), the output PWM duty cycle (or analog output for appropriate models) can be manually controlled by the user program. Once QM0 has been entered, the Set Torque (SQ) command can be used to set or change the servo output (see the SQ command). Bear in mind, that if the output was being limited by an SQ command before entering torque mode (while in PM or VM), it will remain limited while in QM and cannot be changed until the unit is in PM or VM.

For mode 1 (or QM1), the S100 will use one of its A/D channels to monitor servo output current and attempt to maintain that output current at a steady level as commanded by the user program (see the SQ and SC commands). In that the A/D converter has 10-bit resolution, this will result in a value from 0 to 1023 which represents the instantaneous output current with 0 being approximately 0 Amps and 1023 being approximately 5 Amps with intermediate values being somewhat linear.

In the case of a motor (for example), when an output command of 512 is set (or about 2.5 amps), the PWM output begins to ramp up until the desired output current is reached. If the motor is not loaded enough to require 2.5 amps then the PWM will ramp up to 100% putting the maximum supply voltage to the motor. If at this point the motor is suddenly loaded, the PWM will ramp down to a level sufficient to supply 2.5 amps to the motor.

Mode QM1 will not function with the D/A module option.

Related Commands: EG, PM, VM

Command: SEn -- Set Maximum Following Error --

Argument: 0 <= n <= 16,383
Default: 16,383

This command is used to set the maximum allowable following error (or the difference between the actual position and the optimal position) for the servo. If the following error exceeds the programmed value, the servo will be turned off (except in the case of the related interrupt being enabled) and the Error bit in the status word for axis 'a' will be set. This bit will remain set until the servo is turned back on with the Motor On (MN) command. If the maximum argument of 16,383 is given, then this function is disabled.

Related Commands: DB, TF

Command: ST -- Stop Motion --

This command is used to terminate motion. This command differs from the Abort (AB) command in that the servo will decelerate at its preset rate instead of stopping abruptly. When in Torque Mode, the outputs will be set to null.

Related Commands: AB, WS

Command: VM -- Enter Velocity Mode --

Default: Position mode

This command places the servo in the Velocity Mode of operation. In this mode, the servo can be commanded to move in either direction to a maximum velocity. The servo will move in that direction until commanded to stop. When using Velocity Mode, the user must specify the direction for the servo to move using the Direction (DI) command. After specifying the desired maximum velocity and the desired direction and placing the servo in velocity mode, the servo can be started by issuing the Go (GO) command. While the servo is moving in Velocity Mode, the user can change the velocity by issuing new Direction (DI) and/or Set Velocity (SV) commands. The acceleration rate at which the servo's velocity will change, is determined by the Set Acceleration (SA) command. The acceleration can also be changed at any time.

Related Commands: EG, QM, PM

4.4 Register Commands

The S100 uses part of its nonvolatile RAM (NVRAM) to create a 512 by 32 bit general purpose register space with Register "0" being referred to as the Accumulator.

The registers have many uses including storing data and parameters, performing mathematical operations and controlling command execution. The registers can be manipulated by several commands and can also be used to replace the argument in most commands. For example, if register "6" contains the value "-12000" and the following command is used...

```
MA@6,GO
```

it would use the contents of register "6" as the argument thus giving the same result as if the following command was issued...

```
MA-12000
```

Note that the use of the "@" character is what caused the command to assume a register argument (or indirect argument) instead of a direct argument. If the value following the "@" is not in the range of the 512 registers (0 to 511), an error will be reported. If the value contained by the register of the indirect argument is out of range, an error will also be reported.

As stated earlier, because the registers are within the NVRAM, they are non-volatile and can be used as such. For example, if a register is incremented once every user program cycle, it can be used as an ongoing cycle counter for maintenance purposes, production accounting and etc.

4.4.1 Internal Variables

In certain applications, the user may find it necessary to use data pertinent to the internal operation of the S100. This may be accomplished via the use of the Read Byte (RB), the Read Word (RW) and the Read Long (RL) commands which copy the S100's internal RAM memory to the accumulator and the Write Byte (WB), the Write Word (WW) and the Write Long (WL) commands which copy the accumulator to the S100's internal RAM memory. The listings below tell the location of and describe the internal variables that may be of use to the user.

WARNING: Randomly modifying these variables or other internal RAM not listed will most likely affect the operation of the S100 resulting in unpredictable behavior or possible damage.

Axis Variable Descriptions.

Status	Status word (TS command).
PV	Peak velocity (SV command).
MPV	Negative peak velocity (SV command).
V	Temporal velocity (TV command).
Desp	Desired position (TT command).
Carp	Temporal desired position (TO command).
Ack	Acceleration constant (SA command).
Curp	Current real position (TP command).
HREG	Holding register (CI,IP,IR commands).
IPPOS	Interrupt on position reference.
QI	Integral of current mode error (QM1 command).
PGAIN	Proportional term of PID filter (SG command).
IGAIN	Integral term of PID filter (SI command).
DGAIN	Derivative term of PID filter (SD command).
IL	Integral limit of PID filter (IL command).
CGAIN	Current mode gain (QM1 command).
FVGAIN	Velocity feed-forward of PID filter (FV command).
BIAS	Output offset (OO command).
THRO	Current servo output (TQ,SQ command).
QCMD	Current command (QM1,SQ command).
TLMTPL	Maximum positive servo output (SQ command).
FAGAIN	Acceleration feed-forward of PID filter (FA command).
PERR	Last calculated servo position error (TF command).
OERR	Old servo position error.
MAXERR	Maximum servo position error (SE command).
I	Servo error integral.
DERIV	Servo error derivative.
IMON	Servo output current monitor.
INTRVL	PID derivative sample interval (FR command).
SMPCNT	PID derivative sample counter.
IINTRVL	PID integral sample interval (RI command).
ISMPCNT	PID integral sample counter.
AXIS	Axis number lookup index.
ATYPE	Axis output type.
PHASE	Phase, sense and polarity information (PH command).
DBAND	Position error dead-band (DB command).
DERR	Position error with dead-band.
FCNT	Fault system counter.
FCMP	Fault system comparator.
TLMTMI	Maximum negative servo output (SQ command).

System Variable Descriptions

RecRate	Data recorder sample rate.
RecAddr	Data recorder sample address source.
RecSize	Data recorder sample data size.
AIN0	Analog Input 0. (After TA0,GA0 command).
AIN1	Analog Input 1. (After TA1,GA1 command).
AIN2	Analog Input 2. (After TA2,GA2 command).
AIN3	Analog Input 3. (After TA3,GA3 command).
AIN4	Analog Input 4. (After TA4,GA4 command).
AIN5	Analog Input 5. (After TA5,GA5 command).
AIN6	Analog Input 6. (After TA6,GA6 command).
AIN7	Analog Input 7. (After TA7,GA7 command).
AIN8	Analog Input 8. (After TA8,GA8 command).
AIN9	Analog Input 9. (After TA9,GA9 command).
VERSION	Version number for program use. (VE command).
LST_ERR	Last error code (TE command).
LTIMER	1 Hz LED timer.
ADSEMA	A/D Interrupt system semaphores flags.
SYSSTAT	System status word.
IPEND0	User interrupt pending level 0 -15 (EV,DV command).
IPEND1	User interrupt pending level 16 - 31 (EV,DV command).
SCLOCK	Servo loop counter.
RCLOCK	1 mS Real time clock/counter.
USRSEMA	Boolean flags for user semaphores(BC,BS,CB,SB commands).
IO_DELAY	User digital input delay (ID command).

SYSSTAT Variable Bit Definitions

Bit	Purpose
0	**Axis 0 Enabled.** This bit is set to “1” when axis 0 is enabled.
1	**Reserved.**
2	**Reserved.**
3	**Reserved.**
4	**Reserved.**
5	**Pause.** This bit is set to “1” when the space bar is used to pause a user program.
6	**SXOff.** This bit is set to “1” when the S100 receives an XOFF code.
7	**HexMod.** This bit is set to “1” by the HM command and set to “0” by the DM command.
8	**Echo.** This bit is set to “1” by the EN command and set to “0” by the EF command.
9	**HandSh.** This bit is set to “1” by the HN command and set to “0” by the HF command.
10	**Reserved.**
11	**Reserved.**
12	**Reserved.**
13	**Reserved.**
14	**Fail.** This bit is set to “1” by the FN command and set to “0” by the FF command.
15	**BadInp.** This bit is set to “1” by the VI command to indicate an error in user input.

Axis Variable Locations

Variable Name	Type	Axis #1 Address
Status	LONG	01C0H/0448
PV	LONG	01C6H/0454
MPV	LONG	01CAH/0458
V	LONG	01CEH/0462
Desp	LONG	01E0H/0480
Carp	LONG	01E6H/0486
Ack	LONG	01EAH/0490
Curp	LONG	01EEH/0494
HREG	LONG	01F8H/0504
IPPOS	LONG	01FCH/0508
QI	LONG	0200H/0512
PGAIN	WORD	0204H/0516
IGAIN	WORD	0206H/0518
DGAIN	WORD	0208H/0520
IL	WORD	020AH/0522
CGAIN	WORD	020CH/0524
FVGAIN	WORD	020EH/0526
BIAS	WORD	0210H/0528
THRO	WORD	0212H/0530
QCMD	WORD	0214H/0532

Variable Name	Type	Axis #1 Address
TLMTPL	WORD	0216H/0534
FAGAIN	WORD	0218H/0536
PERR	WORD	021AH/0538
OERR	WORD	021CH/0540
MAXERR	WORD	021EH/0542
I	WORD	0220H/0544
DERIV	WORD	0222H/0546
IMON	WORD	0224H/0548
INTRVL	BYTE	0226H/0550
SMPCNT	BYTE	0227H/0551
IINTRVL	BYTE	0228H/0552
ISMPCNT	BYTE	0229H/0553
AXIS	WORD	022AH/0554
ATYPE	BYTE	022CH/0556
PHASE	BYTE	022EH/0558
DBAND	WORD	0230H/0560
DERR	WORD	0232H/0562
FCNT	WORD	0242H/0578
FCMP	WORD	0244H/0580
TLMTMI	WORD	0246H/0582

System Variable Locations

Variable Name	Type	Address
RecRate	WORD	01A6H/422
RecAddr	WORD	01A8H/424
RecSize	WORD	01AAH/426
AIN0	WORD	0600H/1536
AIN1	WORD	0602H/1538
AIN2	WORD	0604H/1540
AIN3	WORD	0606H/1542
AIN4	WORD	0608H/1544
AIN5	WORD	060AH/1546
AIN6	WORD	060CH/1548
AIN7	WORD	060EH/1550
AIN8	WORD	0610H/1552
AIN9	WORD	0612H/1554
VERSION	WORD	0616H/1558
LST_ERR	BYTE	0619H/1561
LTIMER	WORD	061AH/1562
ADSEMA	WORD	061CH/1564
SYSSTAT	WORD	0712H/1810
IPEND0	WORD	0718H/1816
IPEND1	WORD	071AH/1818
SCLOCK	LONG	0722H/1826
RCLOCK	LONG	0726H/1830
USRSEMA	BOOL	072CH/1836
IO_DELAY	BYTE	073EH/1854

Command: AAn -- Add 'n' to Accumulator (Signed) --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command adds the value 'n' to the accumulator.

Command: AC -- Accumulator Complement --

This command causes a (bit-wise) 1's complement of the accumulator.

Command: ADn -- Accumulator Divide by 'n' (Signed) --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command performs a 64 bit by 32 bit signed division with a 64 bit quotient and a 32 bit remainder. **The low order 32 bits of the numerator must be in the accumulator and the high order 32 bits must be in Register 1.** The divisor is specified by 'n'. The lower 32 bits of the quotient will be stored in the accumulator and the upper 32 bits will be stored in Register 1. The remainder will be in Register 2.

Related Commands: AM

Command: AEn -- Exclusive-Or Accumulator with 'n' --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command causes the result of an exclusive-OR of the accumulator with 'n' to reside in the accumulator.

Command: ALn -- Load Accumulator with 'n' (Signed) --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command causes the accumulator to be loaded with the value 'n'.

Command: AMn -- Multiply Accumulator by 'n' (Signed) --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command does a signed multiply of the 32 bit contents of the accumulator and 'n', **leaving the lower 32 bits of the 64 bit product in the accumulator and the upper 32 bits in Register 1.**

Related Commands: AD

Command: ANn -- And Accumulator with 'n' --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command causes the result of the accumulator ANDed with 'n' to reside in the accumulator.

Command: AOn -- Or Accumulator with 'n' --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command causes the result of the accumulator ORed with 'n' to reside in the accumulator.

Command: ARn -- Copy Accumulator to Register 'n' --

Argument: 0 <= n <= 511

This command causes the value in the accumulator to be copied to Register 'n'.

Related Commands: RA

Command: ASn -- Subtract 'n' from Accumulator (Signed)

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command causes the value 'n' to be subtracted from the accumulator.

Command: RAn -- Copy Register 'n' to Accumulator --

Argument: 0 <= n <= 511

This command causes the value in Register 'n' to be copied to the accumulator.

Related Commands: AR

Command: RBn -- Read Byte (8 bits) at RAM location 'n'

Argument: 0 <= n <= 2047

This command reads the 8 bit value at internal RAM location 'n' and copies it to the low 8-bits of the accumulator while clearing the upper 24 bits.

Related Commands: RW, RL, WB, WL, WW

Command: RLn -- Read Long (32 bits) at RAM location 'n' --

Argument: 0 <= n <= 2046

This command reads the 32 bit value at internal RAM location 'n' and copies it to the accumulator. The value specified by 'n' must be evenly divisible by 2.

Related Commands: RB, RW, WB, WL, WW

Command: RWn -- Read Word (16 bits) at RAM location 'n' --

Argument: 0 <= n <= 2046

This command reads the 16 bit value at internal RAM location 'n' and copies it to the low 16-bits of the accumulator while clearing the upper 16 bits. The value specified by 'n' must be evenly divisible by 2.

Related Commands: RB, RL, WB, WL, WW

Command: SLn -- Shift Accumulator Left 'n' bits --

Argument: 0 <= n <= 31

This command causes the accumulator to be shifted left 'n' bits while filling the low order bits with zero.

Related Commands: SR

Command: SRn -- Shift Accumulator Right 'n' bits --

Argument: 0 <= n <= 31

This command causes the accumulator to be shifted right 'n' bits while filling the high order bits with zero.

Related Commands: SL

Command: TRn -- Tell Contents of Register 'n' --

Argument: 0 <= n <= 511

This command reports the value contained by Register 'n'.

Command: WBn -- Write Byte (8 bits) to RAM location 'n' --

Argument: 0 <= n <= 2047

This command copies the low 8-bits of the accumulator to the internal RAM location 'n'.

Related Commands: RB, RL, RW, WL, WW

Command: WLn -- Write Long (32 bits) to RAM location 'n' --

Argument: 0 <= n <= 2046

This command copies all 32 bits of the accumulator to the internal RAM location 'n'. The value of 'n' must be evenly divisible by two.

Related Commands: RB, RL, RW, WB, WW

Command: WWn -- Write Word (16 bits) to RAM location 'n' --

Argument: 0 <= n <= 2046

This command copies the low 16-bits of the accumulator to the internal RAM location 'n'. The value of 'n' must be evenly divisible by two.

Related Commands: RB, RL, RW, WB, WL

4.5 Sequence Commands

The S100 includes commands that provide for conditional sequence command execution based on the register data, I/O state and etc. These Sequence Commands are illustrated by the following general forms:

- If the condition is true, command execution will continue normally. If the condition is false, the next two commands in the command line or the macro will be skipped.
- If the condition is true, command execution will continue normally. If the condition is false, the rest of the command line or the macro will be skipped.
- If the condition is true, command execution will continue normally. If the condition is false, command execution will be suspended until the condition becomes true.

Command: DFn -- Do if I/O Channel is "Off" --

Argument: 0 <= n <= 63

This command will cause command execution to continue if I/O channel 'n' is "Off"; otherwise, the rest of the command line or the macro will be skipped.

Related Commands: CF, CN, DN

Command: DNn -- Do if I/O Channel is "On" --

Argument: 0 <= n <= 63

This command will cause command execution to continue if I/O channel 'n' is "On"; otherwise, the rest of the command line or the macro will be skipped.

Related Commands: CF, CN, DF

Command: EP -- End Program --

This command will cause program execution to cease and return the ">" prompt.

Related Commands: RC

Command: IBn -- If Accumulator is Below 'n' (Signed) -

Argument: -2,147,483,647 <= n <= 2,147,483,647

If the contents of the accumulator is less than (signed comparison) the value 'n', command execution will continue; otherwise, the next two commands in the command line or the macro will be skipped.

Command: ICn -- If Bit 'n' of Accumulator is Clear (0) --

Argument: 0 <= n <= 31

If bit 'n' of the accumulator is clear (equals 0), command execution will continue; otherwise, the next two commands in the command line or the macro will be skipped.

Command: IEn -- If Accumulator Equal to 'n' --

Argument: -2,147,483,647 <= n <= 2,147,483,647

If the accumulator is equal to the value 'n', command execution will continue; otherwise, the next two commands in the command line or the macro will be skipped.

Related Commands: IU

Command: IFn -- If I/O Channel is "Off" --

Argument: 0 <= n <= 63

If the I/O channel specified by 'n' is in the "off" state, command execution will continue; otherwise, the next two commands in the command line or the macro will be skipped.

Related Commands: CF, CN, IN

Command: IGn -- If Accumulator is ">" 'n' (Signed) --

Argument: -2,147,483,647 <= n <= 2,147,483,647

If the contents of the accumulator is greater than (signed comparison) the value 'n', command execution will continue; otherwise, the next two commands in the command line or the macro will be skipped.

Command: INn -- If I/O Channel is "On" --

Argument: 0 <= n <= 63

If the I/O channel specified by 'n' is in the "on" state, command execution will continue; otherwise, the next two commands in the command line or the macro will be skipped.

Related Commands: CF, CN, IF

Command: IPn -- Interrupt on Absolute Position 'n' --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command is used to determine when the servo has achieved the specified real position 'n' referenced by the home position (or zero). When that position has been reached, the "breakpoint reached" flag in the status register will be set. This command can be issued before or after the servo has been commanded to move.

Related Commands: IR, TB

Command: IRn -- Interrupt on Relative Position 'n' --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command is used to determine when the servo has achieved the specified real position 'n' referenced by the current position when this command is executed. When that position has been reached, the "breakpoint reached" flag in the status register will be set. This command can be issued before or after the servo has been commanded to move.

Related Commands: IP, TB

Command: ISn -- If Bit 'n' of Accumulator is Set (1) --

Argument: 0 <= n <= 31

If bit 'n' of the accumulator is set (equals 1), command execution will continue; otherwise, the next two commands in the command line the or the macro will be skipped.

Related Commands: IC

Command: IUn -- If Accumulator Unequal to 'n' --

Argument: -2,147,483,647 <= n <= 2,147,483,647

If the accumulator is unequal to the value 'n', command execution will continue; otherwise, the next two commands in the command line or the macro will be skipped.

Related Commands: IE

Command: RPn -- Repeat --

Argument: 0 <= n <= 65,535

This command causes the command line to repeat 'n' more times. If 'n' is not specified or is "0", the command line is repeated indefinitely.

Command: WAn -- Wait 'n' milliseconds --

Argument: 0 <= n <= 65,535

This command causes a wait period of 'n' milliseconds before going on to the next command.

Command: WEn -- Wait for Edge --

Argument: 0 or 1

This command waits until the Coarse Home Input Active bit in the status word is at the logic state specified by 'n' before continuing command execution. If 'n' is not specified or is 0, it will wait for the Home input to go inactive. If 'n' is 1, it will wait for the Home input to go active.

Note: Please note that this command works differently as compared to that of the IMCSA controller.

Related Commands: FE

Command: WFn -- Wait for I/O Channel "Off" --

Argument: 0 <= n <= 63
Default: "Off"

This command waits until I/O channel 'n' is in the "off" state before continuing command execution.

Related Commands: CF, CN, WN

Command: WI -- Wait for Index --

This command will wait for the occurrence of an Index input signal before continuing to the next command. If a Find Index (FI) or Capture Index (CI) command has not been issued prior to this command, no waiting will occur.

Related Commands: CI, FI

Command: WNn -- Wait for I/O Channel "On" --

Argument: 0 <= n <= 63
Default: "Off"

This command waits until I/O channel 'n' is in the "on" state before continuing command execution.

Related Commands: CF, CN, WF

Command: WPn -- Wait for Absolute Position 'n' --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command is used to determine when the servo has achieved the specified real position 'n' referenced by the home position (or zero). Until that position has been reached, command execution will be suspended. This command can be issued before or after the servo has been commanded to move; however, if the servo is not moving, then command execution may be suspended indefinitely.

Related Commands: WR

Command: WRn -- Wait for Relative Position 'n' --

Argument: -2,147,483,647 <= n <= 2,147,483,647

This command is used to determine when the servo has achieved the specified real position 'n' referenced by the current position when this command is executed. Until that position has been reached, command execution will be suspended. This command can be issued before or after the servo has been commanded to move; however, if the servo is not moving, then command execution might be suspended indefinitely.

Related Commands: WP

Command: WSn -- Wait for Stop --

Argument: 0 <= n <= 65535

This command will wait until the servo trajectory has stopped moving for 'n' milliseconds before continuing to the next command.

Related Commands: ST

4.6 Learned Position Storage (LPS) Commands

The S100 uses part of its nonvolatile RAM (NVRAM) to create a 256 item table for storing positions.

The LPS table is accessed by three commands: Learn Position (LP), Learn Target (LT) and Move to Position (MP). The purpose for the LPS table is to allow the user to store pre-determined positions for later use (such as in contouring) as the NVRAM will retain data even when powered down.

The LPS table overlaps registers 256 - 512 in the general purpose register space so that LPS entry 0 is the same as register 256. This allows for the calculation of predefined positions that can be accessed via the LPS commands.

Command: LPn -- Learn Current Position --

Argument: 0 <= n <= 255

This command causes the current real position of the servo (that position reported by the Tell Position (TP) command) to be stored in location 'n' of the Learned Position Storage table. This command is useful for "teaching" positions to the S100.

Related Commands: LT, MP

Command: LTn -- Learn Target Position --

Argument: 0 <= n <= 255

This command causes the current ideal target position of the servo (that position reported by the Tell Target (TT) command) to be stored in the location 'n' of the Learned Position Storage table. This command is useful for setting up a table of target positions via a downloading procedure.

Related Commands: LP, MP

Command: MPn -- Move to Index Table Position 'n' --

Argument: 0 <= n <= 255

This command causes the position contained by the Learned Storage Position table location 'n' to become the new target position. This command will not initiate a motion; therefore, a Go (GO) command may be required.

Related Commands: LP, LT

4.7 Macro Commands

Command instructions can be entered directly and executed immediately, but the S100 also has the capability of using commands to form other commands called "macros". These macros are stored in the nonvolatile RAM (NVRAM) and can be executed automatically. Macros are created by stringing together one or more commands (with arguments), separated by commas, and indicating where they are to be stored. It should be noted that macros can use indirect as well as direct arguments. The S100 allows for the creation of 256 macros consisting of a total of nearly 890 command instructions. Macro calls via the Macro Call (MC) command or the interrupt system, may be nested up to 25 calls deep.

An example of a macro might be...

```
>MD5,SV1000000,SA10000,MA25000,GO,WS100<CR>
```

Once this command line is entered, it will become the definition for macro 5. When macro 5 is used via the Macro Call (MC), Macro Sequence (MS) or Macro Jump (MJ) commands, the commands contained within the macro will be executed automatically.

There are a few necessary restrictions when creating macros. When using the Macro Define (MD) command, it must be placed first in the command line and it may be used only once. This also implies that it cannot be used as part of a macro. The Reset Macro (RM) command is similar in that it too cannot be used as part of a macro.

Another feature of the S100 is the ability to automatically execute macro "0" on power-up or after a Reset (RT) command. If macro "0" is not defined, the user will receive the ">" prompt, and the S100 will wait for manual input. If macro “0” is defined, the S100 will automatically generate a “MS0” command thereby executing macro “0”.

Command: DVn -- Disable Interrupt Vector 'n' --

Argument: 0 <= n <= 31

This command disables interrupt vector 'n'. This prevents any possibility of an interrupt being caused by that source.

Related Commands: EV, LV

Command: EVn -- Enable Interrupt Vector 'n' --

Argument: 0 <= n <= 31

This command enables interrupt vector 'n'. This allows the possibility of an interrupt being caused by that source.

Related Commands: DV, LV

Command: JPn -- Jump to Command, Absolute --

Argument: 0 <= n <= 31

This command causes execution of a macro to jump to the absolute command specified by 'n'. Commands are numbered sequentially starting from 0. If 'n' is specified whereas the command would attempt to jump past the end of the macro, command execution will resume as if the macro had ended.

Related Commands: JR

Command: JRn -- Jump to Command, Relative --

Argument: 0 <= n <= 31

This command causes execution of a macro to jump to the command specified by 'n' relative to the current command location. If 'n' is specified as zero, then this command will jump to itself. If 'n' is specified such that the command would attempt to jump past the end of the macro, command execution will resume as if the macro had ended. If 'n' is specified such that the command would attempt to jump to a point before the beginning of the macro, an error will be reported.

Related Commands: JP

Command: LVn -- Load Interrupt Vector 'n' --

Argument: 0 <= n <= 31

This command loads interrupt vector table entry 'n' with the contents of the low 8-bits of the accumulator. The following program fragment will cause execution of macro "10" upon the activation of digital input 0 (while a macro program is running).

```
MD1,AL10    ; Load accumulator with number for macro "10".
MD2,LV0     ; Load that number to interrupt vector 0.
MD3,EV0     ; Enable interrupt vector 0.

MD10,MG"Input 0 was just activated.",RC
```

When input 0 is activated, the line "Input 0 was just activated." will be displayed.

Related Commands: DV, EV

Command: MCn -- Macro Call --

Argument: 0 <= n <= 255

This command allows a previously defined macro specified by 'n' to be called like a subroutine. When this command is used, the current macro being executed is pushed to the macro stack and execution of macro 'n' begins. If macro 'n' has not been defined, then an error will be reported. After execution of the defined macro, command execution will continue immediately after the Macro Call (MC) command. The Macro Call (MC) command can be used any place in a macro. Macro calls may be nested up to 25 times; however, it is NOT advisable for a macro to call itself.

Related Commands: RC, UM

Command: MDn -- Macro Definition --

Argument: 0 <= n <= 255

This command is used to define a new macro. Any duplication of numbers will simply result in the loss of the previously defined macro using that number and the loss of the memory that it used. The Macro Define (MD) command must be the first command in the command line or an error will be reported.

Related Commands: RM, TM

Command: MJn -- Macro Jump --

Argument: 0 <= n <= 255

This command may be used to "Jump" to a previously defined macro command. Once the S100 begins executing the new macro, it has no record of how it got there. This means that any commands that appear after the Macro Jump (MJ) command will not be executed. If there is no macro defined by the number 'n', an error will be reported. Once the end of the macro is encountered, macro execution stops (See the Macro Sequence (MS) command). The Macro Jump command can be used any place in a command string or macro command. It is also acceptable for a macro to jump to itself.

Related Commands: MC, MS

Command: MSn -- Begin Macro Sequence --

Argument: 0 <= n <= 255

This command will cause macros to be executed sequentially beginning with macro 'n' until an undefined macro or an End Program (EP) command is encountered. This command can be used anywhere in a command string or macro command. If the Macro Jump (MJ) or Macro Call (MC) commands are encountered, macro execution will still continue to execute sequentially.

Related Commands: MC, MJ

Command: RC -- Return from Macro Call --

When executed, this command will cause immediate return to the calling macro (assuming there was one).

Related Commands: MC, interrupts

Command: RMn -- Reset Macro(s) --

Argument: 0 <= n < 255

This command is used to delete one or more macros. If an argument is used, the macro specified by 'n' will be deleted and the memory it used will be lost. If no argument is used, all macros will be deleted, and all macro memory will be recovered. A Reset Macro command (RM) should be used before entering or downloading a new set of macro commands.

Related Commands: MD

Command: TMn -- Tell Macros --

Argument: -2 <= n <= 255

The Tell Macro (TM) command will display the commands which make up any macros that have been defined. If 'n' >= 0 and 'n' <= 255, then the macro specified by 'n' be displayed. If 'n' = -1, then all the macros will be displayed preceded by the individual macro number. If 'n' =-2, then all macros will be displayed and will be preceded by the letters "MD" and the individual macro number. This is useful when downloading programs from the S100 to a computer in that you need not re-enter the Macro Define (MD) command and number at the beginning of each macro.

Related Commands: MD

Command: UMn -- Un-push Macro(s) --

Argument: 0 or 1

This command is used for controlling the macro subroutine call and return stack. If this command is used with no argument or an argument of "0" then one macro will be removed from the macro stack. If there is no macro to be removed then a MACRO_STACK_UNDERFLOW error will be reported. If the argument is "1" then the macro stack will be completely reset. The UM command is used in the event that a macro is paused due to a program interrupt and it is not desirable to return to that macro, or under some circumstances, a program may need to completely reset itself.

Related Commands: MC, interrupts

4.8 Input / Output (I/O) Commands

The user is able to manipulate the S100's I/O channels via the use of several commands. These include setting or clearing outputs, reading inputs and altering the logic type of both.

Command: BIn -- Bulk Input from I/O Port 'n' --

Argument: 0 <= n <= 7

This command reads the value at the 8-bit digital input port and copies it to the low 8-bits of the accumulator with the lower channel being the lowest bit in the accumulator. The unused bits of the accumulator will be set to 0. The state of the inputs will be determined by the Channel High (CH) and Channel Low (CL) commands and the inputs will have been debounced if the Input Debounce (ID) command is in effect.

Related Commands: BO

Command: BOn -- Bulk Output to I/O Port 'n' --

Argument: 0 <= n <= 7

This command copies the low 8-bits of the accumulator to the digital output port. The state of the outputs will be determined by the Channel High (CH) and Channel Low (CL) commands. Unused bits in the accumulator are ignored.

Related Commands: BI

Command: CFn -- Turn I/O Channel 'n' "Off" --

Argument: 0 <= n <= 63
Default: "OFF"

This command will cause I/O channel 'n' to assume the "OFF" state. The actual output state will depend on whether the channel is set active HIGH (CH command) or active LOW (CL command).

Related Commands: CH, CL, CN

Command: CHn -- Make I/O Channel 'n' Active High --

Argument: 0 <= n <= 63
Default: Active HIGH

This command causes I/O channel 'n' to assume an active HIGH mode.

Related Commands: CL

Command: CLn -- Make I/O Channel 'n' Active Low --

Argument: 0 <= n <= 63
Default: Active HIGH

This command causes I/O channel 'n' to assume an active LOW mode.

Related Commands: CH

Command: CNn -- Turn I/O Channel 'n' "On" --

Argument: 0 <= n <= 63
Default: "OFF"

This command will cause I/O channel 'n' to assume the "ON" state. The actual output state will depend on whether the channel is set active HIGH or active LOW.

Related Commands: CF, CH, CL

Command: IDn -- Input Debounce 'n' milliseconds --

Argument: 0 <= n <= 7
Default: 0

This command determines the length of debounce time, if any, that is applied to the digital inputs. The digital inputs are sampled once every millisecond. What this means is that an input must remain in a given state for 'n' samples before it is considered valid. If the argument 'n' is "0", then no input debouncing in performed. This debouncing applies to the following commands: BI, DF, DN, IF, IN, WF, WN.

4.9 Future Expansion Interface

For future expansion, the S100 provides a proprietary high speed serial data bus. Such uses might include various external devices such as thumb-wheel switches, LED or LCD displays, switch panels and additional number or types of input and output. This provides custom expansion flexibility in certain qualified OEM applications.

4.10 Serial Communications and Miscellaneous Commands

These commands control operation of the serial communication's interface and cover the balance of the S100 functions not fitting in the other categories.

Command: BK -- Break --

This command will cause the rest of the command line or macro to be skipped. This command is used along with the Sequence Commands for conditional command execution.

Command: BRn -- Set Baud Rate --

Argument: 300, 600, 1200, 2400, 4800, 9600 or 19,200
Default: **9600**

This command allows the user to change the baud rate at which the serial communication's interface operates. Once this command has been issued with a valid argument, the communication's interface will then immediately operate at the specified baud rate. This baud rate will remain in place, even after power cycling the unit, until it is changed again. **Please note that the default baud rate is 9600.**

Command: CDn -- Capture Data --

Argument: 0 <= n <= 16383

This command allows for the capture and recording of data from an internal variable with the user being able to later download and plot this data for analysis. **The number of samples that can be recorded depends on the amount of macro memory available at the time that the Capture Storage (CS) command is issued.**

The argument determines the number of samples to record. If an argument is used that is greater than the allocated storage space, then the argument will be truncated to fit that space. All data sizes are extended to 32 bits before they are stored.

Before using the CD command, a few things must be taken care of. The first of these is to define an area to store the data. The Capture Storage (CS) command is used for this in that it allocates a storage area in the macro memory to where the data can be recorded.

The next thing to do is to specify the location and size of the data to be recorded. The internal 16-bit variable “RecAddr” (see Register commands) is used to specify the location (or address) of the variable to be recorded. The internal 16-bit variable “RecSize” is used to specify the data size. If bit 0 of “RecSize” is equal to “1” then 8 bit data samples are recorded. If bit 1 of “RecSize” is equal to “1” then 16 bit data samples are recorded. If bits 0 and 1 of “RecSize” are both “0” then 32 data samples are recorded.

The last thing that needs to be specified is the sample rate at which the data is captured. This is set by writing the appropriate value to the 16-bit variable “RecRate”. The data sample rate will be whatever the servo sample rate is (see SS command) multiplied by “RecRate”.

After using the CD command, the Dump Data (DD) command can be used to display the data that was captured.

Example:

To make 1000 records of the actual position you would issue the following commands:

```
CS1000            ; Reserve space for 1000 samples.
;
;
;
AL4,WW422         ; Servo rate = 1mS so Data rate = 4mS.
AL494,WW424       ; Set address of Curp.
AL0,WW426         ; Set for 32-bit variable size.
```

The CS command should be used only once. After that, the CD and DD commands can be used repeatedly.

Related Commands: CS, DD

Command: CSn -- Capture Storage --

Argument: 0 <= n <= 16383

This command is used to allocate storage space for the data recorder commands. The argument determines the maximum number of samples that can be recorded. **The number of samples that can be recorded depends on the amount of macro memory available at the time that the Capture Storage (CS) command is issued.** If an argument is used that is greater than the macro memory available, then an error will be reported. Because information pertaining to this command is stored in non-volatile memory, once this command is issued, the storage space will remain, even after power-cycling. To recover the macro memory used by this command, use the Reset Macros (RM) command.

Related Commands: CD, DD

Command: DDn -- Dump Data --

Argument: 0 <= n <= 16383

This command is used to dump data that has been previously recorded by the Capture Data (CD) command to the display. The argument determines the number of recorded samples to display. If an argument is used that is greater than the allocated storage space, then the argument will be truncated to fit that space.

Related Commands: CD, CS

Command: DM -- Decimal Mode --

Default: Decimal Mode

This command causes all numerical input and output to be interpreted as decimal or base 10. Numbers will be output with a leading "-" if they are less than zero.

Related Commands: HM

Command: EF -- Echo Off --

Default: Echo On

This command suppresses the echoing of characters received by the serial communication's interface. It is normally used in the half-duplex mode of operation in serial communications.

Related Commands: EN

Command: EN -- Echo On --

Default: Echo On

This command causes characters received from the serial communication's interface to be echoed as they are received. It is normally used in the full-duplex mode of operation in serial communications.

Related Commands: EF

Command: GAn -- Get A/D Channel

Argument: 0 <= n <= 9

This command will cause a conversion on A/D channel 'n' and will store the result in the bottom 10 bits of the accumulator. Unused bits will be set to 0.

Note: Only inputs 0, 7, 8 and 9 are used.

Related Commands: TA

Command: HF -- Hardware Handshaking Off --

Default: Hardware handshake off

This command does not function and is retained for backward compatibility reasons.

Command: HM -- Hexadecimal Mode --

Default: Decimal Mode

This command causes all numerical input and output to be interpreted as hexadecimal or base 16. Numbers will be output as 2, 4 or 8 digits with leading 0's if they are positive and leading F's if they are negative.

Related Commands: DM

Command: HN -- Hardware Handshaking On --

Default: Hardware handshake off

This command does not function and is retained for backward compatibility purposes.

Command: MG[[""][:n][:N]] -- Display Message --

Argument: 0 <= n <= 511

This command allows for the display of an optional text string and / or an optional register variable with the additional option of inhibiting the carriage return / linefeed (CRLF) at the end.

The text string must be enclosed by quotes """ and may be up to 127 characters long. Additional parameters must be separated by a colon ":" .

The following are all the valid examples. Note that the "N" option inhibits a CRLF. Parameters must be given in specific order.

```
>MG                          ; Display CRLF only.

>MG"THE BLACK BEAR IS "      ; Display "THE BLACK BEAR IS " followed by
                             ; CRLF.

>MG0                         ; Display contents of register 0 followed
                             ; by CRLF.

>MGN                         ; This command displays nothing followed
                             ; by nothing.

>MG"THE BLACK BEAR IS ":0    ; Display "THE BLACK BEAR IS " followed by
                             ; the contents of register 0 followed by
                             ; CRLF.

>MG"THE BLACK BEAR IS ":N    ; Display "THE BLACK BEAR IS " followed by
                             ; nothing.

>MG0:N                       ; Display the contents of register 0
                             ; followed by nothing.

>MG"THE BLACK BEAR IS ":0:N  ; Display "THE BLACK BEAR IS " followed by
                             ; the contents of register 0 followed by
                             ; nothing.
```

Command: NO -- No Operation --

This command does nothing. It can be used to cause short delays in command line executions or to fill out commands (see Sequence Commands).

Command: RT -- Reset --

This command performs a complete restart of the S100, including restoration of all default conditions, such as acceleration and velocity, and leaves all servo axis' in the "off" state.

Command: VI[[""][:n][:N]] -- Variable Input --

Argument: 0 <= n <= 255

This command allows for the display of an optional text string, an optional carriage return / linefeed (CRLF) and the entry of integer numeric operator input to a register variable.

The text string must be enclosed by quotes """ and may be up to 127 characters long. Additional parameters must be separated by a colon ":". Entered numbers will be interpreted (as decimal or hexadecimal) as determined by the current mode set by the DM or HM commands. If the value entered by the operator is in error (indeterminate), then the Bad Input bit (bit 15 of the variable SYSSTAT) is set, otherwise it is cleared. If no operator input is entered (only a carriage return is received), then no register will be altered. If no argument is given and operator input is received,
then register 0 will be the default recipient of the input value.

The following are all the valid examples. Note that the "N" option causes a CRLF. Parameters must be given in specific order.

```
>VI                      ; Wait for operator input (if any) to register 0.

>VI"ENTER NUMBER: "      ; Display "ENTER NUMBER: " and wait for
                         ; operator input (if any) to register 0.

>VI5                     ; Wait for operator input (if any) to
                         ; register 5.

>VIN                     ; Display a CRLF and wait for operator input
                         ; (if any) to register 0.

>VI"ENTER NUMBER: ":5    ; Display "ENTER NUMBER: " and wait for
                         ; operator input (if any) to register 5.

>VI"ENTER NUMBER: ":N    ; Display "ENTER NUMBER: " followed by a
                         ; CRLF and wait for operator input (if any)
                         ; to register 0.

>VI5:N                   ; Display a CRLF and wait for operator input
                         ; (if any) to register 5.

>VI"ENTER NUMBER: ":5:N  ; Display "ENTER NUMBER: " followed by a CRLF
                         ; and wait for operator input (if any) to
                         ; register 5.
```

Command: XFn -- Set XOFF Code --

Argument: 0 <= n <= 255
Default: 19

This command does not function and is retained for backward compatibility purposes.

Command: XNn -- Set XON Code --

Argument: 0 <= n <= 255
Default: 17

This command does not function and is retained for backward compatibility purposes.

Command: ZF -- Format NVRAM --

Argument: n = 123

This command re-formats and re-initializes the S100's non-volatile static ram (NVRAM). This means that any macros will be deleted and their space recovered, register contents will be set to 0 and the baud rate (among other things) will be set to their default values. **Evidence of the baud rate change will not be immediate but will occur when the unit is either power cycled or the Reset (RT) command is issued.** This command is intended for use by the manufacturer to place the unit in a known state after testing but is sometimes useful as a "when all else fails" means under certain circumstances. This command must include an argument "key code" of "123" in order to function. This is to reduce the possibility of accidental execution.

Command: ZZ -- Dump Memory --

Argument: 0 <= n <= $3FFFF

This command does a display dump of the memory starting at address 'n'. This command is intended for use by the manufacturer for diagnosing purposes.

---

5. Appendix A, S100 Error Code Definitions

1 - ARGUMENT ERROR.

This error indicates that a command argument was not given or was specified out of the permissible numerical range.

2 - INVALID COMMAND.

This error indicates that an invalid or unrecognized command was specified in the command line.

3 - INVALID MACRO COMMAND.

This error indicates that an invalid or unrecognized command was used in defining a macro.

4 - MACRO ARGUMENT ERROR.

This error indicates that a command argument was not given or was specified out of the permissible numerical range in defining a macro.

5 - MACRO NOT DEFINED

This error indicates that execution of an undefined macro was attempted.

6 - MACRO OUT OF RANGE.

The S100 allows for a maximum of 256 macros (numbered 0 to 255). This error indicates that an attempt was made to access a macro out of this boundary.

7 - OUT OF MACRO SPACE.

The S100 allows for a maximum of 256 macros with up to 256 commands per macro and approximately 15800 bytes of macro storage space. Each macro command requires 6 bytes of macro storage memory and each macro requires an overhead of 1 byte. This error indicates that so many macros/macro commands have been defined that there is no remaining memory to define more.

8 - CAN'T DEFINE MACRO IN A MACRO.

This error indicates that an attempt was made to define a macro from another macro that is currently being executed. This is not allowed on the S100 as to prevent loss of program control due to possible "nesting".

9 - CAN'T DEFINE MACRO WHILE SERVO ENABLED.

This error indicates that attempt was made to define a macro while a servo axis the was enabled (i.e.: "MN"). This is not allowed on the S100 as to prevent loss of program and/or servo control due to macro memory space definition contention.

10 - MACRO JUMP ERROR.

This command indicates that an attempt was made to jump ("MJ" command) to a command within a macro that does not exist.

11 - OUT OF MACRO STACK SPACE.

When a macro is called by another macro (via the "MC" command or a macro interrupt), the return macro and macro command number must be saved along with other internal variables. This error indicates that
the memory space set aside for this purpose has been exhausted and no more "calls" may be attempted. The S100 is capable of macro calls nested 25 deep.

12 - MACRO MUST BE FIRST COMMAND.

When defining a macro, the "MD" command must be the first command in the command line. This error indicates that this requirement was not met.

13 - STRING ERROR

When using a MG or VI command, no closing quote was encountered.

14 - MACRO STRING ERROR

When using a MG or VI command in defining a macro, no closing quote was encountered.

15 - SYNTAX ERROR

Indicates the improper usage of the MG or VI commands.

16 - MACRO SYNTAX ERROR

Indicates the improper usage of the MG or VI commands while defining a macro.

17 - AXIS RANGE ERROR

The axis specified is out the possible numerical range.

18 - INTERRUPT MACRO NOT DEFINED

During the course of interrupt processing, at attempt was made to go to an undefined macro.

19 - INTERRUPT MACRO STACK ERROR

Indicates that the macro stack has run out of space during interrupt processing.

20 - MACRO STACK OVERFLOW

The macro stack has run out of space.

21 - MACRO STACK UNDERFLOW

An attempt was made to "pop" a macro off of the macro stack when there was no macro to pop.

6. Appendix B, Summary of S100 Commands

Parameter Commands

DB -- Set Dead Band
FA -- Feed-forward, Acceleration
FF -- Fail Input Off
FN -- Fail Input On
FR -- Derivative Sampling Frequency
FV -- Feed-forward, Velocity
IL -- Integration Limit
LF -- Limit Off
LM -- Limit Mode
LN -- Limit On
OO -- Output Offset
PH -- Phase
RI -- Sampling Rate of Integral
SA -- Set Acceleration
SC -- Set Current Gain
SD -- Set Derivative Gain
SG -- Set Proportional Gain
SI -- Set Integral Gain
SQ -- Set Torque
SS -- Set Servo Speed
SV -- Set Velocity

Reporting Commands

TA -- Tell A/D Channel
TB -- Tell Breakpoint
TC -- Tell Channel
TD -- Tell Derivative Gain
TE -- Tell Last Command Error
TF -- Tell Following Error
TG -- Tell Position Gain
TI -- Tell Integral Gain
TK -- Tell (K)Constants
TL -- Tell Integration Limit
TM -- Tell Macros
TO -- Tell Optimal Position
TP -- Tell Real Position
TQ -- Tell Torque
TR -- Tell Register
TS -- Tell Status
TT -- Tell Target Position
TV -- Tell Current Velocity
VE -- Tell Version

Motion Commands

AB -- Abort Motion
CI -- Capture Index
DH -- Define Home
DI -- Set Direction
FE -- Find Edge (Home)
FI -- Find Edge (Index)
GH -- Go Home
GO -- Go (start motion)
MA -- Move Absolute
MF -- Motor Off
MN -- Motor On
MR -- Move Relative
PM -- Position Mode
QM -- Torque Mode
SE -- Set Maximum Following Error
ST -- Stop Motion
VM -- Velocity Mode

Sequence Commands

DF -- Do if Channel "Off"
DN -- Do if Channel "On"
EP -- End Program
IB -- If Accumulator Below
IC -- If Accumulator Bit is Clear
IE -- If Accumulator Equal to 'n'
IF -- If Channel "Off"
IG -- If Accumulator is ">" 'n'
IN -- If Channel is "On"
IP -- Interrupt on Absolute Position
IR -- Interrupt on Relative Position
IS -- If Accumulator Bit is Set
IU -- If Accumulator Unequal to 'n'
RP -- Repeat
WA -- Wait
WE -- Wait for Edge (Home or Index)
WF -- Wait for Channel "Off"
WI -- Wait for Index
WN -- Wait for Channel "On"
WP -- Wait for Absolute Position
WR -- Wait for Relative Position
WS -- Wait for Stop

Learned Position Storage Commands

LP -- Learn Current Position
LT -- Learn Target Position
MP -- Move to Position

Macro Commands

DV -- Disable Interrupt Vector
EV -- Enable Interrupt Vector
JP -- Jump Absolute
JR -- Jump Relative
LV -- Load Interrupt Vector
MC -- Macro Call
MD -- Macro Definition
MJ -- Macro Jump
MS -- Macro Sequence
RC -- Return from Call
RM -- Reset Macro(s)
TM -- Tell Macro(s)
UM -- Unpush Macro

Serial Comm. and Misc. Commands

BK -- Break
BR -- Baud Rate
CD -- Capture Data
CS -- Capture Storage
DD -- Dump Data
DM -- Decimal Mode
EF -- Echo Off
EN -- Echo On
GA -- Get A/D Channel
HF -- Hardware Handshaking Off
HM -- Hexadecimal Mode
HN -- Hardware Handshaking On
MG -- Display Message
NO -- No Operation
RT -- Reset
VI -- Variable Input
XF -- Set XOFF Code
XN -- Set XON Code
ZF -- Format NVRAM
ZZ -- Dump Memory

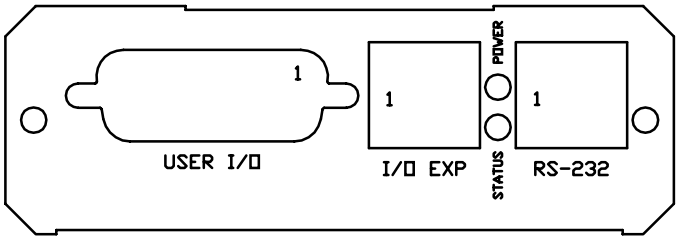
Register Commands

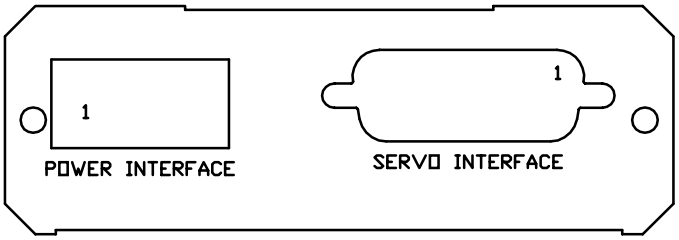
AA -- Accumulator Add
AC -- Accumulator Complement
AD -- Accumulator Divide
AE -- Accumulator Exclusive-Or
AL -- Accumulator Load
AM -- Accumulator Multiply
AN -- Accumulator And
AO -- Accumulator Or
AR -- Copy Accumulator to Reg.
AS -- Accumulator Subtracted
RA -- Copy Register to Accum.
RB -- Read Byte
RL -- Read Long
RW -- Read Word
SL -- Accumulator Shift Left
SR -- Accumulator Shift Right
TR -- Tell Register
WB -- Write Byte
WL -- Write Long
WW -- Write Word

Input / Output (I/O) Commands

BI -- Bulk I/O Input
BO -- Bulk I/O Output
CF -- Turn Channel "Off" --
CH -- Make Channel Active "On"
CL -- Make Channel Active "Off"
CN -- Turn Channel "On"
ID -- Input Debounce Delay

7. Appendix C, S100 Connector Pin Definitions





J1 - Servo Interface : 15-Pin Female D-Sub
Mating Connector: NorComp# ET15P
Digi-Key# 215M-ND

1. Encoder A+
2. Encoder Index+
3. Encoder B+
4. +5 VDC
5. +5 VDC
6. +5 VDC
7. Home Input
8. Limit+ input
9. Encoder A-
10. Encoder Index-
11. Encoder B-
12. Common
13. Common
14. Fault input
15. Limit- Input

J2 - User I/O Interface : 26-Pin H/D Female D-Sub
Mating Connector: NorComp# HDT26P
Digi-Key# T826M-ND

1. Input 6
2. Input 4
3. Input 2
4. Input 0
5. Common
6. Output 6
7. Output 4
8. Output 2
9. Output 0
10. Input 7
11. Input 5
12. Input 3
13. Input 1
14. Common
15. Output 7
16. Output 5
17. Output 3
18. Output 1
19. +5 VDC
20. +5 VDC
21. +5 VDC
22. +5 VDC
23. Common
24. Analog Input 7
25. Analog Input 8
26. Analog Input 9

J3 - I/O Expansion Interface : 6-Pin Modular Jack
Mating Connector: AMP# 5-641337-3
Digi-Key# A9093-ND

1. Receive Data- input
2. Transmit Data- output
3. Receive Data+ input
4. Transmit Data+ output
5. Clock+ output
6. Clock- output

J4 - Power Interface : 4-Pin 5.08mm Centers Phoenix
Mating Connector: OnShore# EDZ95004
Digi-Key# ED1719-ND

1. Main power return
2. Main V+ power input
3. Motor+ output
4. Motor- output

J5 - RS-232 Comm. Interface: 6-Pin Modular Jack
Mating Connector: AMP# 5-641337-3
Digi-Key# A9093-ND

1. Handshake output
2. Handshake input
3. Receive data input
4. Transmit data output
5. Common
6. +5 VDC

8. Index

G

I

L

M

O

P

R

S

T