SilverMax®

Command Reference

Revision 4.01
January 2003

QuickSilver Controls, Inc.
580 E. Arrow Hwy, Suite E., San Dimas, CA 91773
888-660-3801 fax: 909-447-7410
www.QuickSilverControls.com

Notice of Court Order Dated 01-26-04

27 January 2004

In December, Animatics submitted a motion requesting the Court to declare the S-Series infringes based on new disassembly techniques and that the production units were not like the prototypes. Although we disagree with the Court's finding and are submitting an emergency appeal with the Federal Court (Appeal's Court), the Court has found the S-Series infringes and has barred the sale, manufacture and use of the product. Please see the 1-26-04 Order on the Legal Page of our website entitled "Order Barring Sale and Manufacturing of S-Series Product-Line Units 01-26-04". Anticipating our emergency appeal, the Court is allowing us to sell the S-Series until February 13th.

The Court has ordered us to notify all distributors and customers who have purchased the S-Series units of this latest order. The Court has ruled that the product-line units do not conform to the prototypes in violation of the prototype–approval order and the permanent injunction. The Court also ordered us to warn the distributors and customers that any further use or sale by them is at their own peril.

Last year we appealed the Court's original December 2002 infringement decision. The ruling of this main appeal is due any week now. We still expect the Appeal's Court to rule that we do not infringe, which would put an end to this case. In addition to appealing the 1-26-04 order, our emergency appeal will request that the main appeal be expedited and that the 1-26-04 Order be stayed pending the outcome of the main appeal.

As stated, we disagree with the 1-26-04 Order and in particular with its assertion that we "secretly" changed the design. For the last year Animatics has had the right to obtain servos from us or our customers to insure they were being assembled correctly and their attorneys always had the right to review our manufacturing procedures. We did nothing in secret and the above measures insured nothing could be done in secret. We documented our procedure knowing full well that it could be seen by the Court. We understood that the Court approved the production units as long as they were as hard to disassemble as the prototypes. We understood this to mean that the Court was relying on us to make good engineering/manufacturing decisions that results in a unit that was as hard to disassemble as the prototype. We believe their new technique is unreasonable with respect to their patent. It requires hitting the electronics with a chisel and the discharge of toxic fumes (See the Court's notice "Notice Re Future Demonstrations 1-14-04").

We are still fighting this aggressively as well as investigating design solutions. We will keep you up to date.

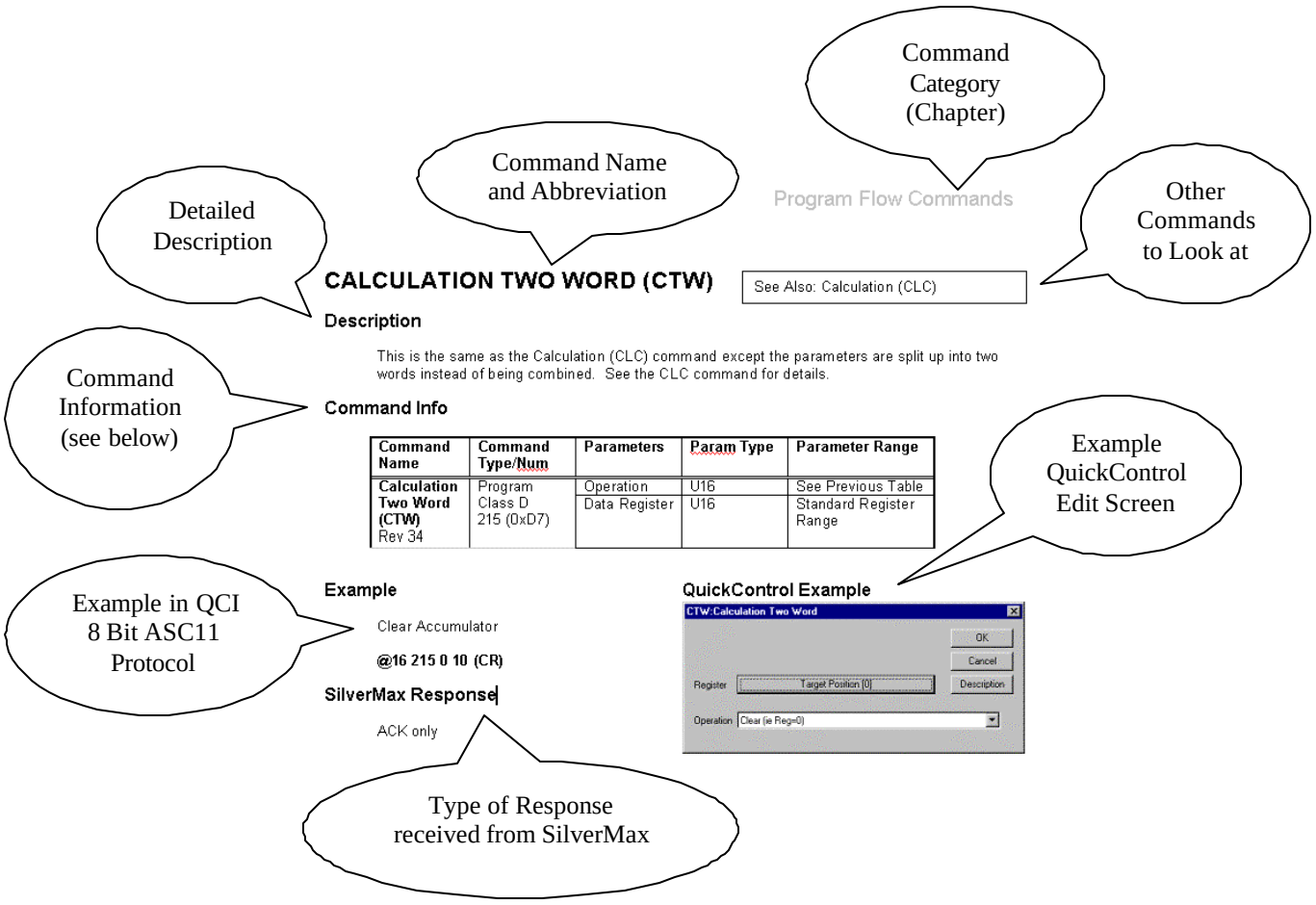
Table of Contents

HOW TO USE THIS MANUAL

The Command Reference contains a detailed description of every SilverMax command. It should be used as a reference not as a tutorial. For general information on SilverMax and QuickControl, please refer to the SilverMax User Manual.

The manual is broken up into several chapters with each chapter detailing a category of commands. For example, there are chapters for Initialization, Mode and Motion commands. Within these chapters, each command is described in one or more pages.

Page Anatomy



Command Information:

Command Name:
- Name of command and its three-letter acronym.
- First SilverMax firmware revision this command appeared. Blank implies the command was available on the first "E" series servos (rev 21 and older).

Command Type/Number:
- Command Type (Program or Immediate). See below for details.
- Command Class (A through F). See below for details.
- Command numbers range from 0 to 255.
- Commands with numbers less than 64 are Host level Immediate Mode only commands (See Command Types below for more details).
- Command numbers 64 or greater are commands that can be contained in a program.
- Commands with numbers 64 or greater will generate a “Busy” “NAK” code if sent to the motor while it is executing a command or a program.
- Commands numbers are given in decimal and hexadecimal format. In the above example the command number is 215 (0xD7). 215 decimal and D7 hex.

Parameters:
- List of parameters for this command
- Parameters must always be included in the command even if the value is “0”.

Parameter Type:
- S32 indicates a signed 32 bit parameter which can range in value from -2147483648 to +2147483647.
- U32 indicates an unsigned 32-bit parameter, which can range in value from 0 to 4294967295.
- S16 indicates a signed 16-bit parameter, which can range in value from -32768 to +32767.
- U16 indicates an unsigned 16-bit parameter, which can range in value from 0 to 65535.

Parameter Range:
- Typical parameter range

Command Types

The SilverMax command structure is divided into two major classifications: Immediate Mode Commands and Program Mode Commands. The Immediate Mode Commands may only be executed via the serial link, while Program Mode Commands may be executed via the serial link or from the nonvolatile memory. Program Mode Commands are temporarily stored in the Program Buffer prior to execution. Before executing a Program, the Program Buffer is filled with the given Program from either the serial communications or the nonvolatile memory.

Immediate Mode

Immediate Mode Commands typically give an immediate result or return data when executed. Most of these commands can be executed at any time even during SilverMax operation. Some Immediate Mode Commands cannot be executed simultaneous to Program Buffer operations. These commands and the conditions for execution are noted in the command description. If command execution is attempted when not appropriate, SilverMax will produce a "NAK Device Busy" response.

Immediate Mode commands do not use the Program Buffer. They are executed as soon as they are received. (Exception- Stop and change velocity immediate. Overwrite the buffer and take over motion and command processing.)

Immediate Mode commands can only be used via the serial communications interface, they cannot be used within a Program that is downloaded to SilverMax for Program execution. A “Host” controller may use Immediate Mode Commands to set up, control, or determine status of a SilverMax.

Program Mode

Program Mode Commands can be executed either from the serial communications interface or from nonvolatile memory. Program Mode Commands, as the name implies, can be part of a Program. When these commands are sent, they are first loaded into the Program Buffer, and then executed. This requires that the buffer not be in use at the time the command is sent. For example, they cannot be executed while the Load Program or Store Program commands are active. If a Program Mode Command is sent while the motor is active, a "NAK Device Busy" response is returned.

Program Mode Commands can also be downloaded to the Program Buffer without being executed. Once a Program has been assembled, it can either be executed immediately or it can be written to the non-volatile Memory. Programs can also be loaded from the Nonvolatile Memory and executed. (In fact the SilverMax factory initialization is a Program which starts at location “0” in the non-volatile Memory and is automatically executed at the application of power.)

Command Classifications

The SilverMax command set has been broken into the following classifications. Each class of command has a set of rules that define how or when a command can be used.
NOTE: “executed” for this section means to “Send a command real-time from a Host controller to SilverMax using the serial communications interface”

Class “A” Commands:

These are serial communications interface only. They may not be contained within a Program and their execution does not incidentally affect the Program Buffer contents. They may be executed at any time.

Class “B” Commands:

These are serial communications interface only. They may not be contained within a Program, but their execution affects the Program Buffer. They may be executed only while the motor is idle (No Motion or Program is running).

Multitasking allows these commands to be executed at any time providing the conditions are appropriate. For example: the “Start Download” command should not be executed when a Program is running or when already in Download mode.

Class “C” Commands:

These are serial communications interface only. They may not be contained within a Program, but their execution affects the Program Buffer. They may be executed only while the motor is idle (No Motion or Program is running). The Program Buffer must also be loaded prior to execution.

Multitasking **–** Allows these commands to be executed when a Motion is running but not when a Program is running.

Class “D” Commands:

These commands can be executed from the serial communications interface or as part of a Program. Their execution from the “Host” affects the Program Buffer. They may only be executed when the motor is idle. They are then stored to the buffer when in download (Program Download) mode. All of these commands have a command code of 64 (hex 0x40) or higher.

Multitasking – Allows these commands can be executed when a Motion is running but not when a Program is running. Most commands will execute immediately while the “Motion” or “Profile Move” commands will be buffered until the current Motion is complete.

Class “E” Commands:

These commands are executed as part of a Program. They may be executed from the serial communications interface but should only be used within a Program or the motor operation may not be what is expected. They rely on what has been previously loaded to the Program buffer for operation. They can only be executed when the motor is idle. They will be stored to the buffer when in download (Program Download) mode. All of these commands have a command code of 64 (hex 0x40) or higher.

Multitasking **–** Allows these commands to be executed while a Motion is running, but care must be taken to avoid unexpected results.

Class “F” Commands

These are serial communications interface only. They may not be contained within a Program, but their execution affects the Program Buffer. They may be executed while the motor is running or idle.

STATUS COMMANDS

Status commands are used to retrieve information from SilverMax. These commands can all be used while SilverMax is executing a motion or a Program.

Clear Poll (CPL)

Description:

This is a complement to the Poll (POL) command. This command is used to clear the Polling Status Word (PSW) bits (See Polling Status Word (PSW) in User Manual for bit definitions). When a status bit is set (“1”) it will remain set until a Clear Poll (CPL) command is sent with the same bit set in its Clear Status Word parameter.

For example, if a POL command gets back a Polling Status Word(PSW) of “0x2000”, bit 13 set (Program completed), of the PSW is set. To reset bit 13, the Clear Status Word parameter must be set to “0x2000”. This will cause bit 13 to be re-set (“0”). All other bits in the PSW will be left unchanged if the corresponding clear bit is not set. New occurrences since the last poll will NOT be cleared (the PSW is double buffered). That is, the information must be read before it is cleared.

See Polling and SilverMax Status Words in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Clear Poll (CPL)	Immediate Class A 1 (0x1) 2 words	Clear Status Word	U16	0 to 65535

Example:

Clear only Bit #13 set in the Polling Status Word
Decimal 8192 = 0x2000 in Hexadecimal

@16 1 8192 (CR)

Clear all the bits set in the Polling Status Word

@16 1 65535 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

ACK only

Poll (POL)

Description:

This command is used to determine the condition of a SilverMax unit. A Poll command can be executed at any time, including while SilverMax is in motion. Executing this command will cause the addressed SilverMax unit to return either an ACK (if no bits of the status are set), or the Polling Status Word (PSW). The PSW contains information about the current state of the SilverMax (See Polling Status Word (PSW) Section in the SilverMax User Manual). The Poll command can be used when checking to see if a motion has completed. This is useful when a system must wait for a SilverMax to complete its operation before performing the next operation. It may be used to verify that the last motion completed without any motion or position errors when the ERROR LIMIT command has been used to set up motion error conditions. The Polling Status Word bits are “Set” when the particular condition takes place. The bits are “re-set” using a Clear Poll command (See Clear Poll command above). Note: Additional conditions that occur after a Poll will show up in the following Poll even if those bits have been cleared in an intervening Clear Poll command. (i.e. they cannot be cleared until they have been read - the data is double buffered.)

See Polling and SilverMax Status Words in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Poll (POL)	Immediate Class A 0 (0x0) *No command will also trigger the poll routine.	NONE	NONE	NONE

Example:

Poll without command number
@16 (CR)

Poll with command number
@16 0 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

ACK only or Pulling Status Word

Response Example:

Response with status
10 0000 2000 (CR)
Response without status
*10 (CR)

Read I/O States (RIO)

Description:

The I/O State Word (IOS) is available for reading back the states of miscellaneous I/O conditions. This word is dynamic and may change every servo cycle (120 usec.).

See I/O State Word (IOS) in User Manual for bit definitions.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Read I/O States (RIO)	Immediate Class A 21 (0x15) 1 word	NONE	NONE	NONE

Example:

Read back the I/O State Word

@16 21 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

I/O State Code

Response Example:

Indicates lines #4, 5, 6, & 7 are
"High" and lines #1, 2 & 3 are also "High"

10 0015 F0F0 (CR)

Read Internal Status Word (RIS)

Description:

The Internal Status Word (ISW) is used in SilverMax to keep track of different conditions that are present in the motor. The Internal Status Word (ISW) can be cleared using the Clear Internal Status (CIS) command.

See Internal Status Word (ISW) in User Manual for bit definitions.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Read Internal Status Word (RIS)	Immediate Class A 20 (0x14) 1 word	NONE	NONE	NONE

Example:

Read back the Internal Status Word

@16 20 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

Internal Status Word (ISW)

Response Example:

Indicates Input #1, 2, 3 “High”, Last Calculation was Zero and Index Sensor was found.

10 0014 00F3 (CR)

Read Program Buffer (RPB)

Description:

Reads the Data that is currently contained in the Program Buffer. The specified number of Words is read from the Program Buffer starting with the given address. Up to 8 words can be read at one time. This limit is due to the size restriction of the Serial Communications Buffer. To read the entire contents of the Program Buffer multiple reads are required. For details on SilverMax memory management, see the User Manual section Programming SilverMax.

NOTE: When reading command codes from Program Buffer the MSB (Most Significant Bit) will be stripped off. For example, if an MRV command is read from the Program Buffer, it will be read as a 0x07 instead of a 0x87.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Read Program Buffer (RPB)	Immediate Class A 6 (0x6) 3 words	Length (in words)	S16	1 to 8
		Address	S16	0 to 199

Example:

Read the first 7 words from Program Buffer

@16 6 7 0 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

Requested number of words read from Program Buffer.

Response Example:

10 0006 0007 0000 9C40 0002 7524 2000 0058 (CR)

Revision (RVN)

Description:

This command returns the revision date firmware, and the buffer sizes. The code revision date and buffer sizes of a SilverMax can be read back so that future upgrades can be dealt with through a software interface. This enables dynamic changes in SilverMax usage when different versions are used.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Revision (RVN)	Immediate Class A 5 (0x05) 1 word	NONE	NONE	NONE

Example:

Read the revision code from SilverMax

@16 5 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

Revision Code (8 Bytes)

Response Example:

SilverMax Revision code

10 0005 1116 1998 0108 0A34 (CR)

The following Revision information is available

Data Type	Data Format	Example Shown above
A. Month	1 Byte	“11” = November
B. Day	1 Byte	“16” = 16th day
C. Year	2 Bytes	“1998” = The year 1998
D. Options Number	2 Bytes	“0108” = Code rev 108
E. Serial Communications Buffer Size	1 Byte	“0A” = 10 Words
F. Program Buffer Size	1 Byte	“34” = 52 Words

OVERRIDE COMMANDS

Halt (HLT)

Description:

This command immediately shuts down any motion in progress (hard stop), disables the single step mode, and then causes the motor to load and run the Kill Motor Recovery program. (see Kill Motor Recovery (KMR) command for details.)

This command stops the execution of all commands, programs and motions. When executed, it will stop any command or program in process. Unless the Kill Motor Recovery Program has been designated and the Kill Enable Driver (KED) has been enabled, the motor driver will be disabled. This shorts the drive to the motor and allows the motor shaft to spin freely.

Bit #10 of the Internal Status Word (ISW) is "set" to indicate that a Halt command was sent. This is useful for determining the cause of the motor shut down when using an internal Kill Motor Recovery Program.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Halt (HLT)	Immediate Class A 2 (0x2) 1 word	NONE	NONE	NONE

Example:

Halt any Command, Program or Motion in process

@16 2 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

ACK only

Restart (RST)

Description:

The Restart command is provided to cause the servo to do a “Soft” reset of the processor and logic circuits. This causes the processor to jump to memory address zero as if the power were just cycled on. If this command does not function, then the SilverMax servo will require a full power down and up to reset the processor. All configurations and settings are returned to power-up defaults. All registers are cleared but nonvolatile memory is not affected.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Stop (STP)	Immediate Class A 4 (0x4) 1 word	NONE	NONE	NONE

Example:

Reset the SilverMax processor.
This is done immediately.
@16 4 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

There is no response due to the resetting of the processor

Stop (STP)

Description:

The Stop command exits the executing program or motion and goes into Hold. If a motion is running, the Deceleration parameter sets the deceleration as follows: If “0”, it uses the executing command’s acceleration value for deceleration. If this value is positive, it uses the given deceleration value. If the deceleration parameter is negative, the servo does an immediate stop (directly to Hold). The Target value is set to the present position. If the servo is not executing a motion, any Program Mode command executing is terminated and the servo returns to idle.

When the Stop command is sent the Program Buffer is over-written (similar to a Clear Buffer). The Program Buffer must be loaded again (Load Program or Load and Run Program) for Program execution.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Stop (STP)	Immediate Class A 3 (0x3) 3 words	Deceleration	S32	**-1** = Stop Immediate or **0** = Stop using previous Acceleration or 1 to 536,870,911

Example:

Stop the SilverMax using the previous command Acceleration parameter

@16 3 0 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

ACK only

INITIALIZATION COMMANDS

ACK Delay (ADL)

Description:

The ACK Delay sets a time delay for SilverMax to wait before sending an Acknowledgement ACK) or DATA after a command has been received. In some cases, the host computer or PLC may not have enough time to set up for reception after having transmitted a command. The ACK Delay configures SilverMax to wait a predetermined length of time to allow for these delays. In the case where an RS-485 network in used, there is often a delay when going from transmit to receive for the RS-485 driver. Setting the delay makes the SilverMax to wait for the line to be free before attempting a transmission.

When the Serial interface is set to “RS-232” a value of “0” causes SilverMax to run in standard RS-232 mode (the Tx line is always driven). With a number of “1” or greater, SilverMax will run in RS-232 multi-drop mode (the Tx line is tri-stated when not transmitting).

The delay parameter is a count that equates to the servo cycle tick. One cycle tick is 120 microseconds. The largest count that can be set is 21845 - this will give a delay of 2.6214 seconds.

When sending this command the delay takes effect before the Acknowledge is sent.

QuickControl will automatically set this parameter at download if the box "Automatically Set ADL depending on the current SIF" is checked (default for QuickControl's initialization wizard). At download, QuickControl asks the SilverMax whether is in RS-232 or RS-485 (SIF) and, with baud rate, sets ADL accordingly.

Recommended settings:
RS-232;
ADL = 0
RS-485 and RS-232 Multi-Drop:

ADL (ms)	Baud Rate
2.8	115200
3.2	57600
4.2	28800
5.2	19200
8.1	9600

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Ack Delay (ADL) Rev 23	Program Class D 173 (0xAD) 2 words	Delay Count 1 tick= 120 uSec	U16	0 to 21845

Example:

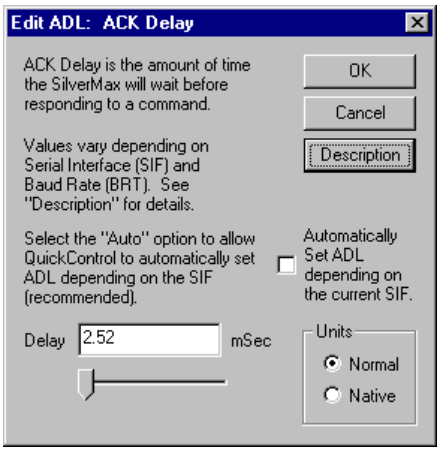
Delay ACK for 2.52 milliseconds
21 x 120 uSec= 2.52 ms

@16 173 21 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Anti-Hunt Constants (AHC)

See Also: **Torque Limits (TQL), Anti-Hunt Delay (AHD)**

Description:

Anti-Hunt Constants sets the thresholds used to determine if the position is sufficiently close to the target to allow the motor to go into and to stay in Anti-Hunt mode. The first parameter is the maximum error (in counts) allowed in the Anti-Hunt mode before the unit will revert to normal closed loop operation. The second parameter is the maximum error allowed to enter the Anti-Hunt mode.

Setting the second parameter to a negative number will cause a slightly different operation when going from no Anti-Hunt into Anti-Hunt (Closed => Open). Normally SilverMax will not go into Anti-Hunt until the error is within the limit and the current torque (current) is less than the Open Loop Holding torque (current). When the error parameter is negative, the torque is not checked.

If the Torque Limits (TQZ) Open Loop Holding and Open Loop Moving parameters have been set to zero, then the parameters in this command set the limits of a conventional dead-band.

QuickControl will automatically set this parameter at download if the box "Use Default for SilverMax" is checked (default for QuickControl's initialization wizard). The defaults are:

Encoder Counts/Rev (CPR)	AHC Parameters Open to Close/Close to Open	Max Recommended
4000	10/4	30
8000	20/8	60
16000	40/16	120

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Anti-Hunt Constants (AHC)	Program Class D 150 (0x96) 3 words	Open to Closed	S16	0 to 140
		Closed to Open	S16	-140 to 140

Example:

Go into Anti-Hunt when within “4” counts of target. Go out of Anti-Hunt when “10” counts away

@16 150 10 4 (CR)

SilverMax Response:

ACK only

QuickControl Example:

See Also: **Anti-Hunt Constants (AHC)**
Anti-Hunt Mode (AHM)

Anti-Hunt Delay (AHD)

Description:

After the conditions are met for Anti-Hunt as specified by the Anti-Hunt Constants (AHC) command, this Anti-Hunt Delay (AHD) specifies the amount of delay before going into Anti-Hunt. This is useful for allowing a system time to "settle" prior to going into Anti-Hunt. See Anti-Hunt Constants (AHC) for more details.

Settling time is a system parameter, which must be analyzed under real working conditions. Using the SilverMax Tuning tool in QuickControl allows viewing of motion profiles for analyzing settling times.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Anti-Hunt Delay (AHD)	Program Class D 230 (0xE6) 2 words	Delay Count in Ticks 1 Tick = 120usec.	U16	0 to 65535 Default= =1250 ticks (150ms)

Example:

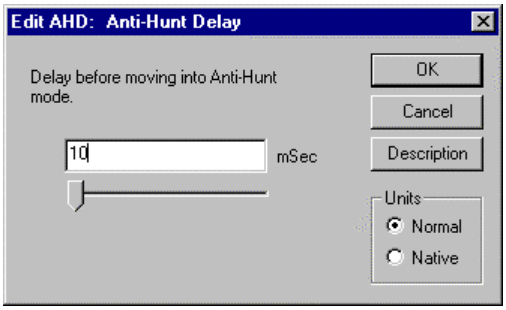
Allow Anti-Hunt 10 milliseconds after a motion is completed.

@16 230 83 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Anti-Hunt Mode (AHM)

See Also: **Anti-Hunt Constants (AHC)**
Anti-Hunt Delay (AHD)

Description:

The default mode of Anti-Hunt automatically switches from open loop to closed loop as soon as a motion begins, and then remains in closed loop for Anti-Hunt Delay time counts after the position error is less than the Closed to Open parameter. Anti-Hunt Mode with Mode=1 bypasses the in motion check, allowing the servo to remain in open loop, even while moving, as long as the error is sufficiently low. A value of Mode=0 switches the Anti-Hunt function back to its default mode of operation.

With Mode=1, some Anti-Hunt Delay (AHD) is useful to keep from switching between moving and stopped while moving at low speeds.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Anti-Hunt Mode (AHM) Rev 29	Program Class D Code (Hex): 219 (0xDB) 2 words	Mode	S16	0 or 1 0 = only when stopped (Default) 1 = moving or stopped

Example:

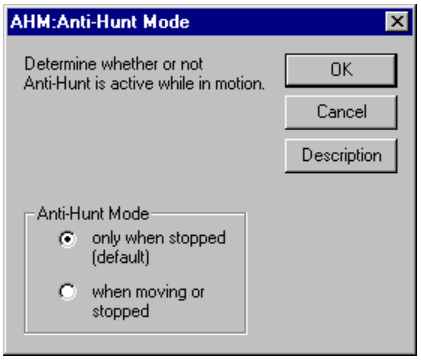
Allow Anti-Hunt Mode only while stopped.

@16 219 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Baud Rate (BRT)

Description:

If a Baud Rate different than the default rate is required, such as with a PLC limited to 9600 baud, the Baud Rate command can be used to change the baud rate to a range of other values. The baud rate is limited to a maximum value of 230400 bits per second. 57600 is the default baud at power up. Other baud rates can be set using values from the table below.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Baud Rate (BRT)	Program Class D 174 (0xAE) 2 words	Speed	U16	3 = 300 (baud) 12 = 1200 24 = 2400 48 = 4800 96 = 9600 192 = 19200 288 = 28800 384 = 38400 576 = 57600 (Default) 1000 =100000 1152 =115200 2304 =230400

Example:

Set the baud rate for 57.6K.

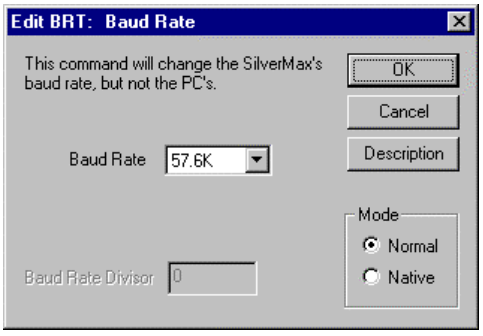
@16 174 576 (CR)

SilverMax Response:

ACK only

NOTE: The baud rate changes immediately, so, the Acknowledge may not be intelligible.

QuickControl Example:



Control Loop Mode (CLM)

Description:

This command sets the control loop of the servo to operate around either Position (default at power up) or Velocity.

In velocity mode, the servo is closed around the servo velocity rather than servo position. The Proportional gain term is disabled (zeroed out), and the Integrator acts on the difference in velocities between the Target velocity and the Measured Velocity. The anti-windup on the integrator is configured to smoothly recover from a motion stoppage without over-running the desired velocity.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Control Loop Mode (CLM) Rev 31	Program Class D 166 (0xA6) 2 Words	Mode	U16	0 = Position mode (Default) 1 = Velocity mode

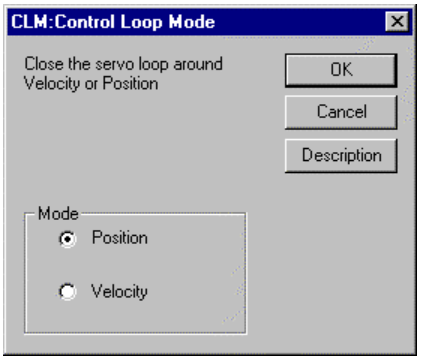
Example:

@16 166 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Control Constants (CTC)

Description:

This command sets the various servo loop gain control constants. These are used in tuning the servo.

QuickControl stores a default set of parameters for each SilverMax type (i.e. 23-3, 23H-1, ...). If Use Default For SilverMax is checked, QuickControl will use the default parameters.

See Tuning SilverMax in the SilverMax User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Control Constants (CTC)	Program Class D 148 (0x94) 8 words	Kv1: Velocity 1 Feedback Gain	U16	0 to 32767
		Kv2: Velocity 2 Feedback Gain	U16	0 to 32767
		Kvff: Velocity Feedfoward Gain	U16	0 to 32767
		Ka: Acceleration Feedback Gain	U16	0 to 32767
		Kaff: Acceleration Feedfoward Gain	U16	0 to 32767
		Kp: Proportional Gain	U16	0 to 32767
		Ki: Integrator Gain	U16	0 to 32767

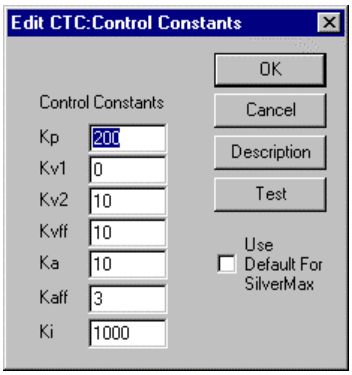
Example:

@16 148 0 10 10 10 10 200 1000 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Disable Done Bit (DDB)

See Also: **Enable Done High (EDH)**
Enable Done Low (EDL)

Description:

Disables the “Done” bit (I/O #1) on the servo. The “Done” bit indicates when the servo is running or idle (See Enable Done Bit for more details.) By default, the “Done” bit is disabled.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Disable Done Bit (DDB)	Program Class D 171 (0xAB) 1 word	NONE	NONE	NONE

Example:

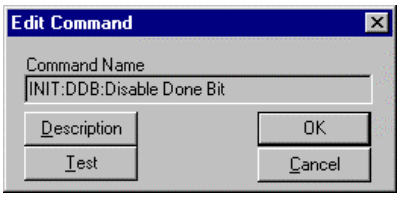
Disable usage of the “Done” bit

@16 171 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Digital Input Filter (DIF)

Description:

Sets up a filter time constant for any of the seven digital inputs. A "0" in the I/O line parameter causes all of the input filter constants to be changed at the same time. Selecting 1,2, 3, 4, 5, 6 or 7 for the I/O line changes only the selected line.

The Filter constant is in "Ticks" (120 usec / tick). Setting the filter constant affects how long a digital state must be held for the SilverMax to “see” the given state. The filter does not require that the input be exclusively in the new state for the entire period, but just that it is in that state sufficiently long for the counter to expire.

For example, with the filter set to 8 Ticks (approximately 1 mS), and transitioning from low to high: 5 high states, followed by 2 low states (such as switch bounce / noise) require another 8-5+2 = 5 ticks of high before a high would be reported. 8 consecutive high levels are not required. This minimizes the effects of noise/contact bounce on the system.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Digital Input Filter (DIF) Rev 22	Program Class D Code (Hex): 252 (0xFC) 3 Words	I/O Line #	U16	0 = All Lines 1 to 7
		Filter Constant	U16	0 to 32767 Default: 83 ticks (10ms)

Example:

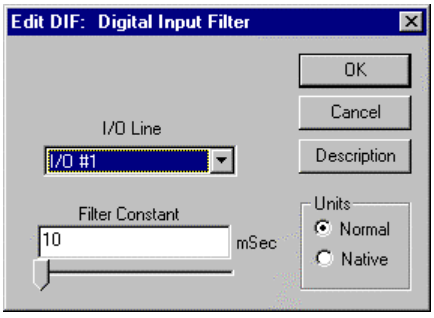
Filter Input #1 so that it must be either low or high for as least 10 milliseconds before the low or high state is accepted.

@16 252 1 83 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Direction (DIR)

Description:

Establishes at initialization the direction in which the servo will turn given a motion in a positive direction. Normally the SilverMax will turn Clockwise (when viewed from the shaft end of servo) when a positive distance or velocity number is used. A negative number will cause the servo to turn counter clockwise. Using the Direction command, this default operation can be reversed.

WARNING: DIR can only be used when SilverMax is being initialized and before the Go Closed Loop (GCL) command is issued. If DIR is used after GCL is can cause the servo to become unstable. Typically this command is only edited within the SilverMax Initialization Wizard while editing the initialization file "SilverMax Factory Default Initialization.qcp".

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Direction (DIR)	Program Class D 184 (0xB8) 2 words	Mode	S16	0 = Normal (CW) (Default) 1 = Reverse (CLW)

Example:

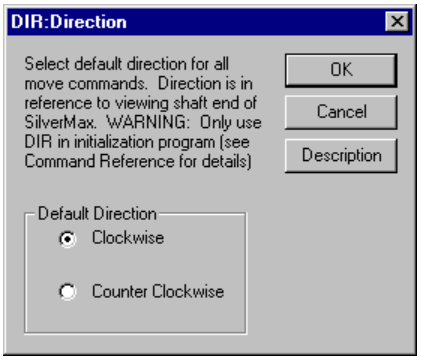
Clockwise

@16 184 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Dual Loop Control (DLC)

See Also: **Select External Encoder (SEE), Single Loop Control (SLC)**

Description:

Configures SilverMax to run in a Dual Loop control mode. In Dual Loop Control, the SilverMax servos its position based on an External Encoder signal. SilverMax commutation, velocity and acceleration feedback information is derived from the internal encoder. Moving and holding error limits also use the external encoder for the Kill Motor Conditions.

The Anti-Hunt mode uses the Position Error derived from the external encoder to establish when to move in and out of Anti-Hunt mode.

When position control of the driven device is needed, this command along with an external encoder connected to the device, will enable direct position control of the device. When using a linear slider, a linear encoder can be used for the external encoder signal.

When entering dual loop control SilverMax sets the current “Target” to the “Current position” (external position) to prevent a sudden motion.

SilverMax must be in Closed Loop Mode for this command to take effect.

By default, SilverMax starts up in Single Loop Control mode. The Dual Loop Control command can be placed within a User Program to place the servo properly into Dual Loop Control. Use of a single loop move prior to entering dual loop move may be used to verify that the external encoder is connected and operational.

The external encoder settings should be initialized prior to using this command or unpredictable results may occur. Use the Select External Encoder (SEE) command to set up the external encoder usage.

NOTE: The Control Constants (CTC) typically need to be configured differently for single loop operation than for dual loop operation. The Velocity and Acceleration parameters for motions become related to external encoder counts rather than internal encoder units. The feedforward acceleration and velocity terms are relative to full speed in external encoder units while the feedback terms are relative to the internal encoder units, thus the feedback terms may need to be different from the feedfoward terms in order to minimize following error. If the external encoder has three times the resolution of the internal encoder, then the feedback terms need to be three times as large as the feedforward terms to balance their gains.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Dual Loop Control (DLC)	Program Class D 243 (0xF3) 1 word	NONE	NONE	NONE

Example:

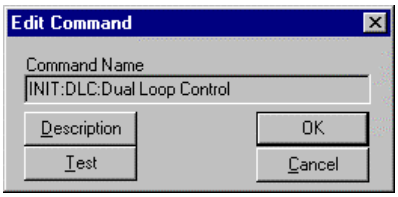
Configure SilverMax for Dual Loop Control

@16 243 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Disable Motor Driver (DMD)

See Also: **Enable Motor Driver (EMD)**

Description:

Disables the motor driver and shorts the windings together. The SilverMax will be unable to move when attempting any motion command. This is a software disable that can be overcome by the Enable Motor Driver (EMD) command, or by setting the Motor Constants (MCT).

Command Info:

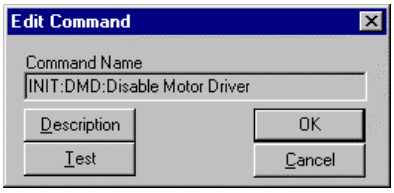
Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Disable Motor Driver (DMD)	Program Class D 228 (0xE4) 1 word	NONE	NONE	NONE

Example:

Disable the Motor Driver electronics

@16 228 (CR)

QuickControl Example:



SilverMax Response:

ACK only

Disable Multitasking (DMT)

See Also: **Enable Multitasking (EMT)**

Description:

Disables SilverMax Multitasking operation. See Enable Multitasking for more information on multitasking operation.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Disable Multitasking (DMT)	Program Class D 226 (0xE2) 1 word	NONE	NONE	NONE

Example:

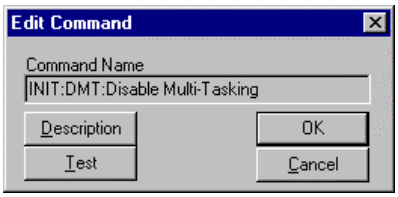
Disable the Multitasking operation

@16 226 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Enable Done High (EDH)

See Also: **Disable Done Bit (DDB)**
Enable Done Low (EDL)

Description:

Enables a “Done” indication on the servo I/O Line #1. The “Done” indicates when the servo is idle and within the Error Limits. When the servo is idle (No pending commands and no active motions) and within the error limits, I/O #1 will be high (“1”), and the Green LED will be lighted, otherwise, I/O #1 will be low (“0”) and the Green LED will be dark.

Note, if multiple commands are in the Program Buffer, all of them must complete (and the error within limits) before the unit is “Done”. Thus, if a PLC were to send a Load and Run command, the entire sequence would have to be completed, and any motions completed, and the error within limits before a “Done” would be reported.

PLCs or Host Controllers can use the “Done” indication for monitoring SilverMax motions to check for completion.

Command Info:

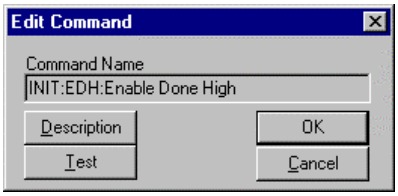
Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Enable Done High (EDH) Rev 22	Program Class D 251 (0xFB) 1 word	NONE	NONE	NONE

Example:

Enable usage of the “Done” indication by setting I/O line #1 High

@16 251 (CR)

QuickControl Example:



SilverMax Response:

ACK only

Enable Done Low (EDL)

See Also: **Disable Done Bit (DDB)**
Enable Done High (EDH)

Description:

Enables a “Done” indication on the servo I/O Line #1. The “Done” indicates when the servo is idle and within the Error Limits. When the servo is idle (No pending commands and no active motions) and within the error limits, I/O #1 will be low (“0”), and the Green LED will be lighted, otherwise, I/O #1 will be high (“1”) and the Green LED will be dark.

Note, if multiple commands are in the Program Buffer, all of them must complete (and the error within limits) before the unit is “Done”. Thus, if a PLC were to send a Load and Run command, the entire sequence would have to be completed, and any motions completed, and the error within limits before a “Done” would be reported.

PLCs or Host Controllers can use the “Done” indication for monitoring SilverMax motions to check for completion.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Enable Done Low (EDL) Rev 22	Program Class D 187 (0xBB) 1 word	NONE	NONE	NONE

Example:

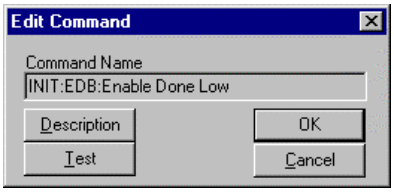
Enable usage of the “Done” indication by setting I/O line #1 Low

@16 187 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Enable Motor Driver (EMD)

See Also: **Disable Motor Driver (DMD)**

Description:

Enables the SilverMax motor driver. The driver is by default enabled, this command is only required if the driver has been disabled using the Disable Motor Driver (DMD) command or disabled by the Kill Motor operation or by an over voltage condition.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Enable Motor Driver (EMD)	Program Class D 227 (0xE3) 1 word	NONE	NONE	NONE

Example:

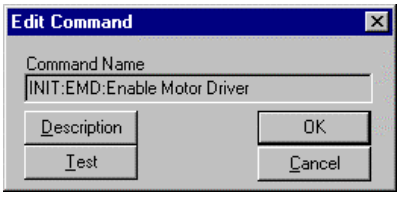
Enable the SilverMax motor driver

@16 227 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Enable Multitasking (EMT)

See Also: **Disable Mutitasking (DMT)**

Description:

Enables SilverMax “Multitasking” operation, which allows motion while executing a program. By default, SilverMax does not continue internal program execution when performing a motion command or while executing in a Velocity Mode, Step and Direction Mode or Input Mode. Enable Multitasking causes SilverMax to continue program execution after a motion command or mode has been started.

Multitasking is useful for conditions where the SilverMax is acting as a Master controller and must initiate other tasks while performing motions. An example of this is performing I/O operations for controlling other machine functions. SilverMax will be able to “Set” or “Clear” the I/O lines at any time. Many other things can be done including modifying motion profiles on-the-fly when executing a Profile Move command or the Register Step and Direction command.

For more details, refer to multi-tasking in SilverMax User's Manual.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Enable Multitasking (EMT)	Program Class D 225 (0xE1) 1 word	NONE	NONE	NONE

Example:

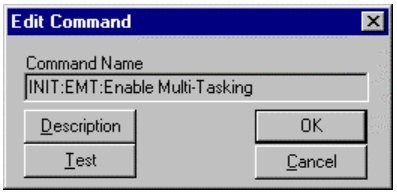
Enable Multitasking operation

@16 225 (CR)

SilverMax Response:

ACK only

QuickControl Example:



See Also: **Kill Motor Conditions (KMC)**

Error Limits (ERL)

Description:

Error Limits sets the Moving Error limits, the Holding Error limits, and the Delay to Hold time following a motion before the servo uses the holding torque limit setting and the holding error limits. At the completion of a motion (the Trajectory Generator has completed the move calculation), the Delay to Holding counter will begin counting down. When the count reaches zero (one count per 120 microseconds), the servo changes to Hold Mode. The Moving Error limit is always checked, while the Holding Error limit is only checked while in the hold mode.

The Delay to Holding also controls when the SilverMax will switch from the moving torque limits to the holding torque limits.

Error conditions are set in both the Internal Status Word and in the Polling Status Word if the error limits are exceeded. See Status Words in User Manual for details.

A special “Drag” mode may be implemented by setting the error limits to negative values. The absolute value of the limit is used to generate the status conditions, while the target is not allowed to get farther than the respective error limit from the servo position. This can be used to set up a “clutch” mode where the output shaft can be dragged around to a desired position and the system will then maintain that position. This is useful for some mechanical alignment procedures, as well as for implementing drag clutch operations.

The defaults for SilverMax and QuickControl's Initialization Wizard are different as described below.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Error Limits (ERL)	Program Class D 151 (0x97) 4 words	Moving Limit	S16	-32768 to 32767 SilverMax Default = 0 QuickControl Default = 500
		Holding Limit	S16	-32768 to 32767 SilverMax Default = 0 QuickControl Default = 200
		Delay to Holding (1 tick = 120usec.)	S16	0 to 65535 SilverMax Default = 100 QuickControl Default = 120

Example:

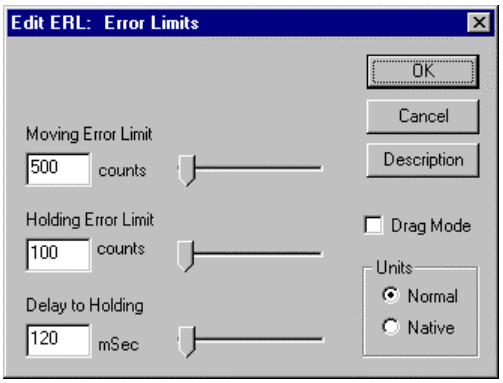
Allow 500 counts of error while moving and 100 counts of error when holding position. Allow 120 milliseconds before going into Hold mode with its tighter error limit.

@16 151 500 100 1000 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Filter Constants (FLC)

Description:

Filter Constants selects the cutoff frequency for the velocity and acceleration filters.

See Tuning SilverMax in the SilverMax User Manual for more details on using this command.

See Scaling in the SilverMax User Manual for details on converting filter values Hz to native SilverMax units.

QuickControl stores a default set of parameters for each SilverMax type (i.e. 23-3, 23H-1, ...). If "Use Default For SilverMax" is checked, QuickControl will use the default parameters both now and at download time.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Filter Constants (FLC)	Program Class D 169 (0xA9) 4 words	Fv1: Velocity 1 Feedback Filter	S16	4096 to 32767
		Fv2: Velocity 2 Feedback Filter	S16	4096 to 32767
		Fa: Acceleration Feedback Filter	S16	4096 to 32767

Example:

Set filters to roll of at 469, 413 and 117 Hz.

$23000 = 32768\ e^{-(469)2\pi(120uS)}$

$24000 = 32768\ e^{-(413)2\pi(120uS)}$

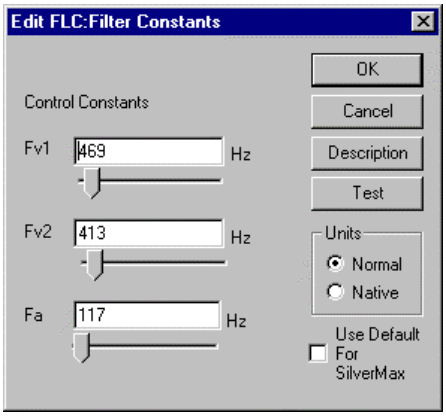
$30000 = 32768\ e^{-(117)2\pi(120uS)}$

@16 169 23000 24000 30000 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Gravity Offset Constants (GOC)

Description:

Establishes a value that compensates for the effects of gravity on the load that the servo is driving. This servo control parameter is designed to neutralize the effect of gravity on mechanisms that operate in other than horizontal orientation. It enables the servo control to operate consistently in both directions of servo rotation by creating a Torque offset that counters the torque required to hold the load in position.

The offset value is in torque units the same as the Torque Limits (TQC) command. If, for example, it requires ¼ of the servo's torque capability to hold the load in position, the value may be set to approximately ¼ of the 100% torque value. (If the full torque value were 20000, the ¼ value would be 5000)

Depending on the direction of the Torque applied to the servo shaft, the value can be set to a negative or positive value.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Gravity Offset Constant (GOC)	Program Class D 237 (0xED) 2 words	Gravity Offset	S16	-32767 to 32767 Default: 0

Example:

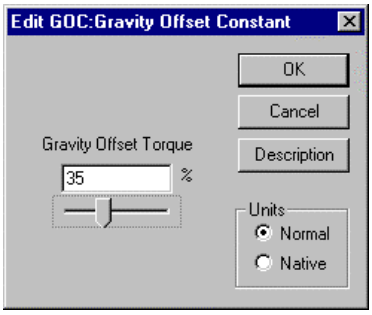
Set the Gravity Offset to 35% Torque for a S-23-3 servo

@16 237 7000 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Identity (IDT)

Description:

The Identity command is used to set the Unit ID and Group ID addresses to which the SilverMax will respond. The SilverMax will accept and respond to any command addressed with the Unit ID. The SilverMax will accept commands sent to either the Group ID or to the Global ID (255), but no response will be sent as multiple units cannot respond at the same instant. No two units should have the same Unit IO when connected on the same network. Multiple units may share a common Group IO when they are on the same network. Do not set Unit ID and Group ID to the same value.

Identities need to be in the range of 1 to 254.

Upon issuing this command, the SilverMax will respond with the new identifier.

See Communication Protocols in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Identity (IDT)	Program 155 (0x9B) 2 words	Group/Unit ID Group ID = Upper Byte Unit ID = Lower Byte	U16	257 to 65278 Default: Unit ID=16 Group ID=20

Example:

To Calculate number: Multiply the Group Identity times 256, then add the Unit Identity

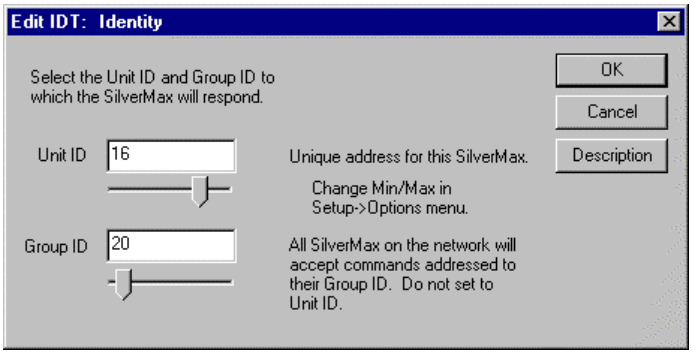
Group = 20, Unit = 16

Identity = (20 * 256) + 16 = 5136

Group Identity of 20, Unit Identity of 16;

@16 155 5136 (CR)

QuickControl Example:



SilverMax Response:

ACK only

Kill Disable Driver (KDD)

See Also: **Kill Enable Driver (KED)**

Description:

Disables the motor driver, shorting across the windings (passive braking) when a Kill Motor Condition is met. If the SilverMax is moving, it will stop immediately in a rapid manner. The motor will be unable to move until re-enabled using the Enable Motor Driver command. This is the default setting for the servo.

See Shutdown and Recovery in User Manual for details.

Command Info:

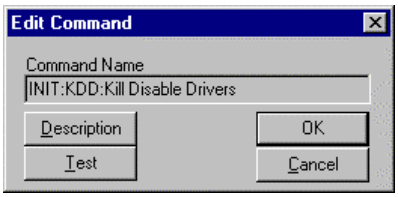
Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Kill Disable Driver (KDD)	Program Class D 183 (0XB7) 1 word	NONE	NONE	NONE

Example:

Disable the Motor Driver electronics when Kill Motor Conditions are met

@16 183 (CR)

QuickControl Example:



SilverMax Response:

ACK only

Kill Enable Driver (KED)

See Also: **Kill Disable Driver (KDD), Enable Multi-Tasking (EMT)**

Description:

Causes SilverMax to leave the motor drivers enabled when a Kill Motor Condition is met. Normally the motor driver is disabled with a Kill Motor Condition, this command can be used to leave the driver enabled if continuing operation is required.

In order for this command to function, SilverMax must be set up for Multitasking operation. Without Multitasking, the driver will be disabled when a Kill Motor Condition occurs.

See Shutdown and Recovery in User Manual for details.

Command Info:

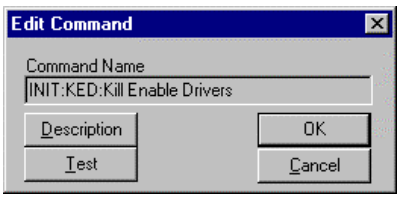
Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Kill Enable Driver (KED)	Program Class D 182 (0xB6) 1 word	NONE	NONE	NONE

Example:

Leave the SilverMax motor driver enabled

@16 182 (CR)

QuickControl Example:



SilverMax Response:

ACK only

Kill Motor Conditions (KMC)

See Also: **Kill Motor Recovery (KMR)**

Description:

The Kill Motor Conditions allows the user to select what conditions will allow a controlled shutdown of the SilverMax unit. The Condition Enable word selects which bits in the Internal Status Word (ISW) will be evaluated (See Internal Status Word (ISW) in User Manual for bit definitions). Conditions are enabled by setting a "1" in the desired bit position of the Condition Enable binary word.

The Condition State word allows the user to specify the state of the selected conditions that will cause the servo to do a controlled shutdown. Note: Over-voltage is always enabled whenever the driver is enabled to protect the drivers from over voltage. An over-voltage condition will always disable the drivers regardless of the of Kill Enable Drivers state.

See Shutdown and Recovery in User Manual for details.

Default has only Over Temperature enabled.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Kill Motor Conditions (KMC)	Program Class D 167 (0xA7) 3 Words	Condition Enable	U16	0 to 65535
		Condition State	U16	0 to 65535

Example:

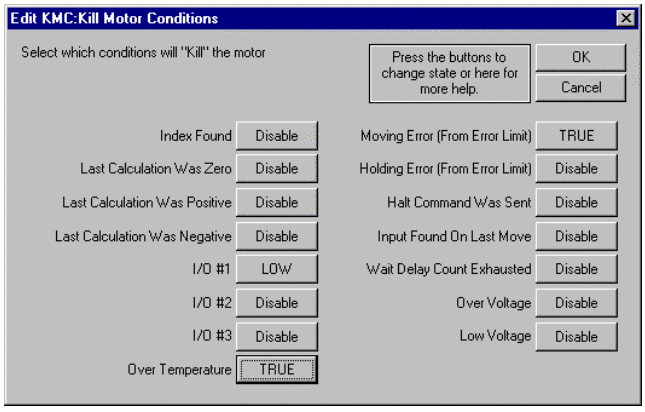
Shut down servo if any of the following conditions are met:
I/O#1 LOW (bit 4)
Over Temp (bit 7)
Moving Error (bit 8)

NOTE: Over Temp TRUE = 0.

Enable = $2^4 + 2^7 + 2^8 == 400$
State = $2^4*0 + 2^7*0 + 2^8*1=256$

@16 167 400 256 (CR)

QuickControl Example:



SilverMax Response:

ACK only

See Also: **Kill Motor Conditions (KMC)**

Kill Motor Recovery (KMR)

Description:

The Kill Motor Recovery sets up options for recovery from a motor shut down. The Kill Motor Conditions (KMC) establishes conditions that will cause the motor to shut down. Using the Kill Motor Recovery the motor can perform a standard or user defined process for re-initializing the motor. User Programs can be executed that have been previously stored in the motor nonvolatile memory. (See Kill Motor Conditions for more detail). See Shutdown and Recovery in User Manual for details.

Three Options available:
"0" – This default state indicates that no recovery program has been designated. The SilverMax drops out of any motion or program that is currently executing and goes into an idle state. The drivers are disabled. At this point SilverMax will sit with no current to the motor, waiting for host intervention using the Serial Interface.

"-1" – Normal operation: -1 is a special parameter value indicating to run the initialization program from non-volatile memory location "0"

"####" – Normal operation: The routine located at #### is loaded and executed.

See Kill Motor Mask for actions taken before the loading of the Kill Motor Recovery routine.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Kill Motor Recovery (KMR)	Program Class D 181 (0xB5) 2 words	Process	S16	0 = Do Nothing -1 = Load and Run Program @ NV Mem adr 0. #### = Load and Run Program at indicated NV Mem adr.

Example:

After motor shutdown load and run "Fault Recovery" program which is stored at 542.

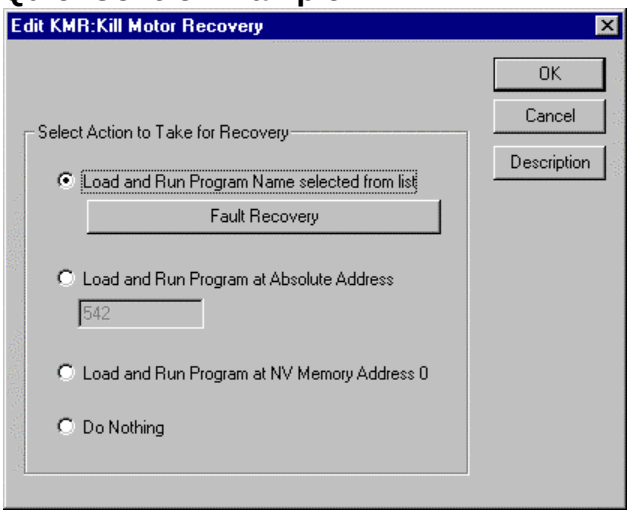
NOTE: In QuickControl, the user only needs to specify the program name. The address is calculated automatically.

@16 181 542 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Low Voltage Processor Trip (LVP)

See Also: **Power Low Recovery (PLR), Low Voltage Trip (LVT)**

Description:

This command is only usable with SilverMax units that provide separate power supply inputs for the processor and for the driver sections, such as the HC series. This command allows the monitoring of the processor power supply for low voltages in the same way that a Low Voltage Trip (LVT) command monitors the driver (or, for single supply motors, the main power supply).

This command sets the input voltage that will trigger a Low Voltage status (Bit #14 in the Internal Status Word(ISW)) and subsequently the Power Low Recovery (PLR) routine (if configured). When a Low Voltage Processor Trip occurs the low voltage trip values, both driver and processor; are overwritten to zero to prevent multiple triggering.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Low Voltage Processor Trip (LVP) Rev 29	Program Class D 131 (0x83) 2 Words	Voltage	U16	0 = Don't Check 10 to 48 Default: 0

Example:

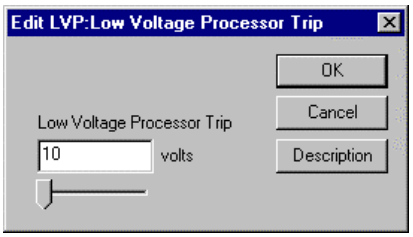
Set LVP to 10 volts

@16 131 10 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Low Voltage Trip (LVT)

Description:

This command sets the input voltage (or driver Input voltage for units that have dual input power supplies) that will trigger a Low Voltage status (Bit #14 in the Internal Status Word(ISW)) and subsequently the Power Low Recovery (PLR) routine (if configured). When a Low Voltage Trip occurs the low voltage trip values associated with the Low Voltage Trip and Low Voltage Processor Trip commands are overwritten to zero to prevent multiple triggering.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Low Voltage Trip (LVT)	Program Class D 212 (0xD4) 2 Words	Voltage	U16	0 = Don't Check 10 to 48 Default: 10V

Example:

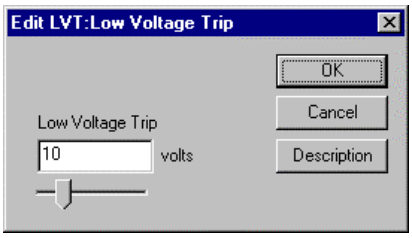
Set Motor to shut down at 10 volts

@16 212 10 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Motor Constants (MCT)

Description:

This command initializes the driver stage to produce appropriate drive signals to the motor dependant on both the motor type and the supply voltage. These constants are factory supplied for the selected motor at the requested power supply voltage. Normally these are set using QuickControl's SilverMax Initialization Wizard. Executing this command also causes the motor driver to be “Enabled”. This will over-ride a disabled driver condition whether it was commanded by a Disable Motor Driver (DMD) or Kill Motor Condition (KMC).

The Edit MCT dialog box gives the user the following 3 options:

Auto: QuickControl will read the servo's voltage at download time and set the parameters accordingly. This is the recommended default setting.

Manual: The user selects the voltage. This option is useful when the voltage in the field is different than the voltage at time of download.

Native: An advanced mode that should only be used at the direction of QuickSilver Controls.

The MCT and PAC commands are tightly coupled. When either of these commands is edited in a QuickControl QCP file, the other command is automatically updated.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Motor Constants (MCT)	Program Class D 168 (0xA8) 9 Words	MC1	S16	0 to 32767
		MC2	S16	0 to 32767
		MC3	S16	0 to 32767
		MC4	S16	0 to 32767
		MC5	S16	0 to 32767
		MC6	S16	0 to 32767
		MC7	S16	0 to 32767
		MC8	S16	0 to 32767

Example:

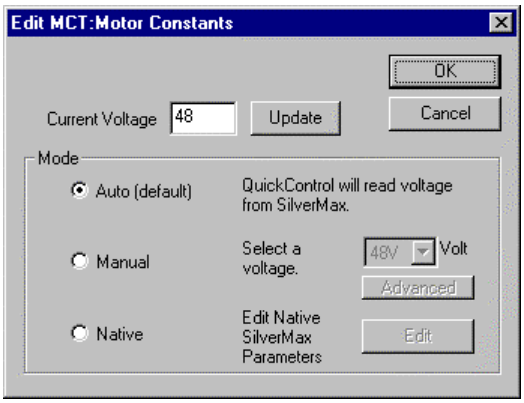
Set up a 23-5 for 24 volt operation

@16 168 1631 14843 31816 2057 1758 2329 32767 8213 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Maximum Temperature Trip (MTT)

Description:

Sets the temperature at which SilverMax will shut down the servo. This is used to prevent internal over-heating of the servo electronics. The value is entered in degrees Celsius integer units. (Example “70” for 70 degrees Celsius). The maximum temperature error condition is OR-ed with the motor driver over temperature condition. Either active will cause an Over Temperature status condition in the Internal Status Word. The temperature can be read using the ANALOG READ INPUT command.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Maximum Temperature Trip (MTT)	Program Class D 214 (0xD6) 2 words	Temperature (°C)	U16	0 = Don’t Check 1 to 80 Default: 0

Example:

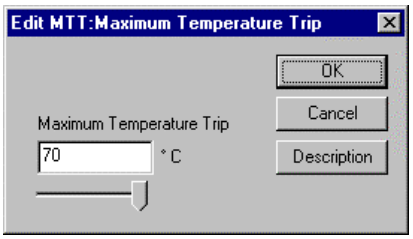
Set Servo to give an error at 70 degrees C

@16 214 70 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Open Loop Phase (OLP)

Description:

The Open Loop Phase is used to set initial motor phase prior to doing motor/encoder alignment. It is primarily used as an element in the algorithmic motor to encoder alignment routine. For positive values, this represents the micro-step position of the motor. Negative values are used to select half step positions – i.e. –2 sets the motor at 1 full step away from a value of 0. Please see the SilverMax Initialization section in the SilverMax User Manual for a more detailed description of the initialization process. This command is normally used only in the initialization procedure.

The use of negative values makes the operation independent of encoder resolution.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Open Loop Phase (OLP)	Program Class D 152 (0x98) 2 words	Phase Angle Count	S16	-7 to 79

Example:

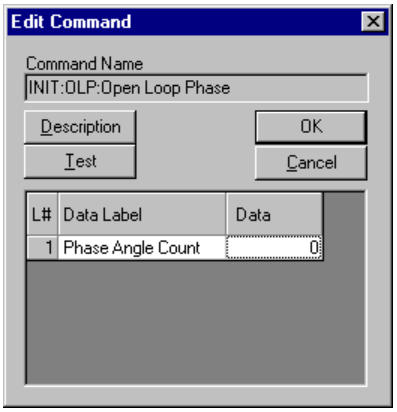
Set the open loop phase to “0”

@16 152 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Over Voltage Trip (OVT)

Description:

Sets the voltage at which SilverMax will cause a motor shutdown. This command is mainly used to prevent over-voltage from the power regenerated during deceleration. The voltage value is entered in integer units (example: “48” for 48 volts). If an over-voltage condition is detected, a motor shutdown is executed that shorts the motor phase windings to reduce regenerated power flowing into the power supply input which boosts the supply voltage.

NOTE: The Kill Enable Driver (KED) command does not allow the motor driver to stay enabled when an Over Voltage Trip occurs. This condition always disables the motor driver.

The motor driver is disabled when this condition occurs and must be re-enabled using the Enable Motor Driver (EMD) command or by re-writing the Motor Constants (MCT).

The factory default is set at 52 volts. A power supply voltage that exceeds 52 volts may cause the motor to shutdown at power up. Unregulated power supplies with excessive voltage ripple can cause an over voltage trip, even though an average reading meter may report the voltage as within specification. The over voltage trip may also activate when doing rapid decelerations with large inertias, or using the SilverMax as a clutch without using a Clamp Module between the SilverMax and the power supply.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Over Voltage Trip (OVT)	Program Class D 213 (0xD5) 2 words	Voltage	U16	1 to 53 (52 = Default)

Example:

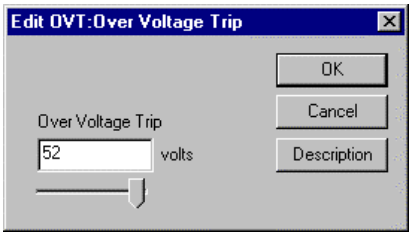
Shut down the Motor if the input voltage exceeds 52 volts

@16 212 52 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Phase Advance Constants (PAC)

Description:

Sets the motor phase advance constants. These are motor type and power supply voltage dependent to optimize motor torque at high speed. Factory set for optimal performance.

PAC uses the same QuickControl dialog box as MCT. For more details, see MCT.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Phase Advance Constants (PAC)	Program Class D Code (Hex): 172 (0xAC)	PAC1	S16	0 to 60
		PAC2	S16	0 to 2400
		PAC3	S16	0 to 200

Example:

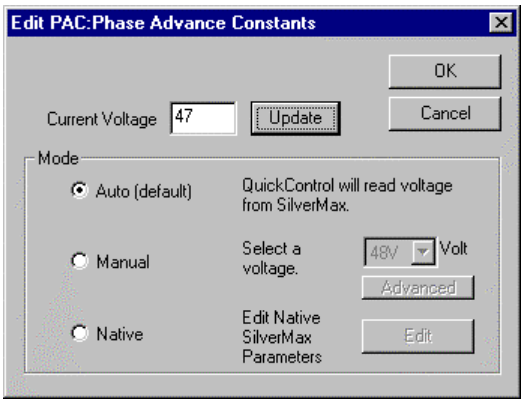
Phase advance for a 23-5 SilverMax

@16 172 5 160 37 (CR)

SilverMax Response:

ACK only

QuickControl Example:



See Also: **Low Voltage Trip (LVT)**

Power Low Recovery (PLR)

Description:

This command designates which routine from nonvolatile memory to run if the power supply voltage drops below that specified by see Low Voltage Trip. A value of zero indicates to shut down the motion and then to do nothing. See Shutdown and Recovery in User Manual for details.

Four Options are available:
Load and Run Program (QuickControl Only) - Press this button to select which program you want to load and ran for the PLR.

Load and Run Absolute Address - Enter the non-volatile memory address of the program you want to load and run for the PLR.

Load and Run Program at NV Memory Address 0 - Load and run the program stored at 0. By default this is the initialization program.

Do Nothing – This default state indicates that no recovery program has been designated. The SilverMax drops out of any motion or program that is currently executing and goes into an idle state. The drivers are disabled. At this point SilverMax will sit with no current to the motor, waiting for host intervention using the Serial Interface.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Power Low Recovery (PLR)	Program Class D 208 (0xD0) 2 Words	Process	S16	0 = Do Nothing -1 = Load and Run Program @ NV Mem adr 0. #### = LRP @ NV Mem adr.

Example:

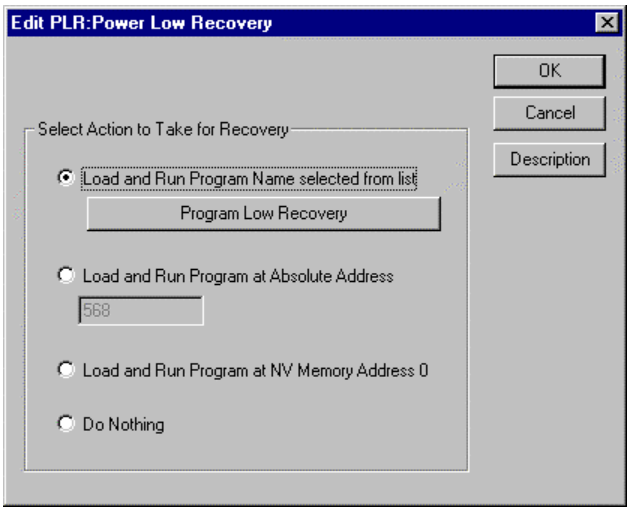
If power low condition exists load and run "Program Low Recovery" program which is stored at 568.

@16 208 568 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Protocol (PRO)

Description:

Allows the user to select the desired communications protocol. Two different protocols are available: the 8-Bit ASCII and the 9-Bit Binary. See SilverMax User Manual for more information on Protocols.

When PRO is sent while the servo is in download mode (see Start Download (SDL) command), the servo will respond with an ACK. If this command is sent in Immediate Mode, the response will be in the new protocol.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Protocol (PRO)	Program Class D 185 (0xB9) 2 words	Mode	S16	0 = 9-Bit 1 = 8-Bit (Default)

Example:

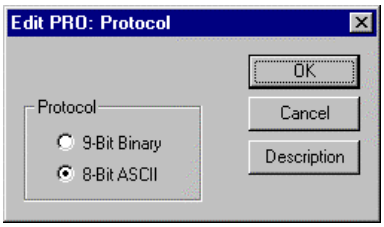
Select the 8-Bit ASCII Protocol

@16 185 1 (CR)

SilverMax Response:

See above.

QuickControl Example:



S-Curve Factor (SCF)

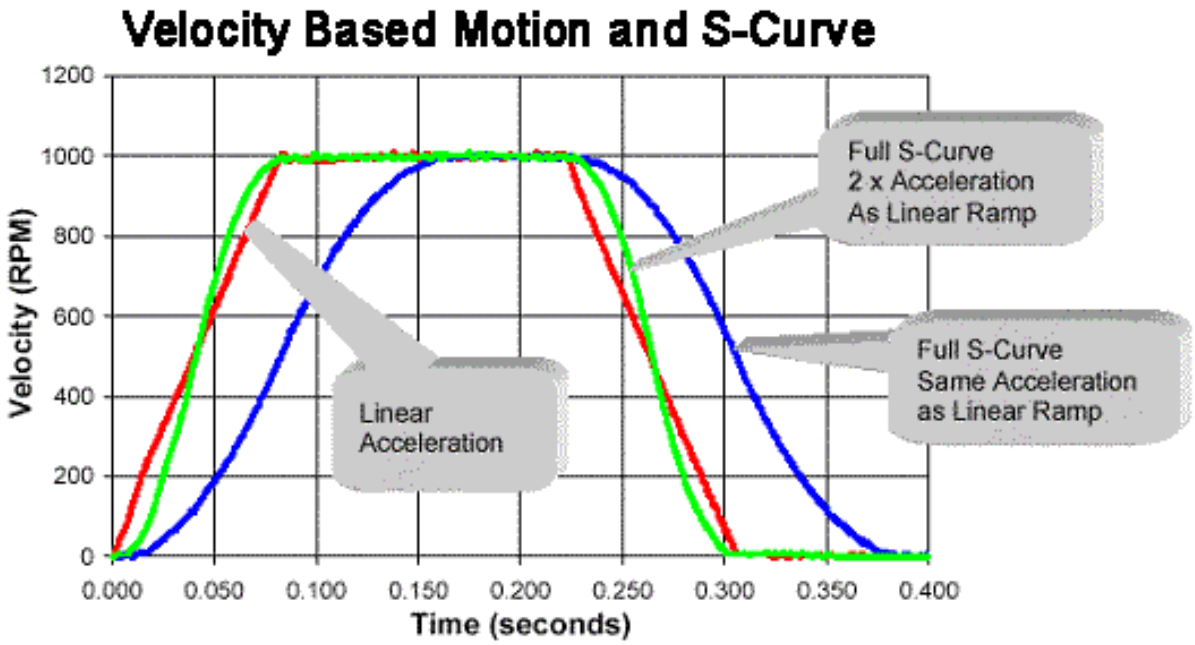
Description:

Using this command, the shape of motion profile acceleration can be set from linear to full s-curve. This command can be set at any time (except for during a motion) allowing each motion profile to be tailored for the best shape. SCF only affects the following basic motion commands and their register based deviations.

Move Relative, Time Based (MRT)
Move Relative, Velocity Based (MRV)
Move Absolute, Time Based (MAT)
Move Absolute, Velocity Based (MAV)

SCF is not available in the Step & Direction (i.e. SSD) , Profiled Move (i.e. PMC) , Input Mode (i.e. PIM) or the Velocity modes (i.e. VMP).

A full s-curve will minimize the rate of change of acceleration (or jerk) for a trapezoidal motion. In a full s-curve time based move, the actual acceleration used is double that of a pure trapezoidal (zero s-curve) motion. In a velocity based move, the time to complete the move increases proportionally to the amount of s-curve specified. In order to have the full s-curve move complete in the same time as the trapezoidal move, the acceleration parameter must be doubled. The following chart shows a velocity based move with zero s-curve (trapezoidal), full s-curve, and full s-curve with the specified acceleration doubled.



Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
S-Curve Factor (SCF)	Program Class D 195 (0xC3) 2 words	Factor	S16	0 = Trapezoidal 1 to 32766 = s-curve 32767 = Full s-curve Default: 0

Example:

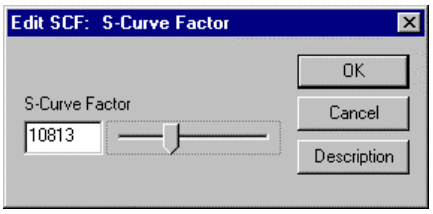
Use some S-Curve on next Motion Profile

@16 195 10813 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Select External Encoder (SEE)

Description:

Selects the desired input format for an external encoder or step/direction input. External encoders can be used by the “Step and Direction” commands, as well as the Dual Loop mode. If an external encoder is not being used, the inputs are ignored. The count since cleared or powered up is available in the register 200 - “External Encoder”. A sensing of the designated index source causes the external encoder counter contents to be copied to register 201 - external encoder index. SEE does not tri-state the selected inputs. If an I/O is already set LOW or HIGH it will remain that way after the SEE command.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Select External Encoder (SEE)	Program Class D 192 (0xC3) 4 words	Index Source	S16	0= I/O #6 1= I/O #3
		Index State	S16	0 = Falling 1= Rising
		Encoder Style	U16	0 = A/B Quad on I/O 4 & 5 1 = Step Up/Dn on I/O 4 & 5 2 = Step & Dir on I/O 4 & 5 3 = Step & Dir on I/O 2 & 3

Example:

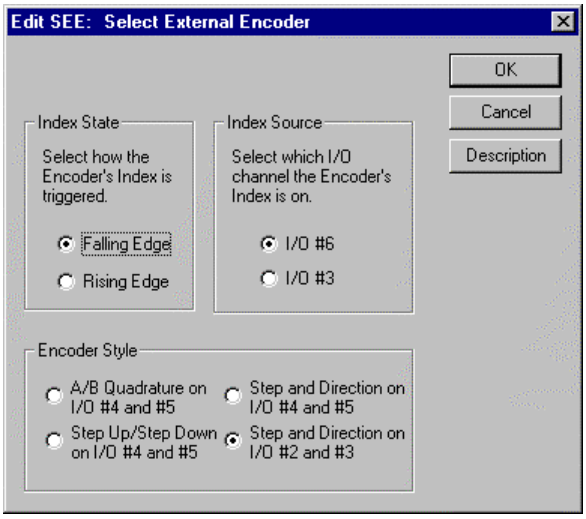
Set up the External encoder inputs for Falling edge, Index on input #6 and Step & Dir on #2 & #3.

@16 192 0 0 3 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Select Encoder Filter (SEF)

Description:

Selects the desired digital filter for the external encoder signals. The default is a 150 nS. The other option is an 800nS filter. The increased filter time may help applications using the external encoder or step/direction inputs in a noisy environment. The filter is applied to each external encoder interface line (all 3 I/O usable at same time). This filter only affects the external encoder count, not any I/O that may also be looking at these same lines.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Select Encoder Filter (SEF) Rev 29	Program Class D 130 (0x82) 2 words	Filter Enable	S16	0 = 150nS (Default) 1= 800nS

Example:

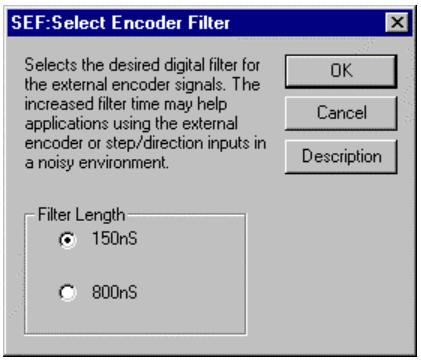
Set up the External Encoder filter for 800nS:

@16 130 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Serial Interface (SIF)

Description:

Allows the user to select between RS-232 and RS-485 serial communications hardware interface. This command is usually used at power up as part of the initialization program. Care should be taken when using this command, as communications may be lost if the Host Controller is not compatible with the new hardware setting.

QuickControl will automatically set this parameter at download if the box "Set to SIF currently being used by SilverMax" is checked. At download, QuickControl asks the SilverMax whether is in RS-232 or RS-485 and then sets the SIF command accordingly.

If this command is sent in Immediate Mode, the response will be in the new interface.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Serial Interface (SIF)	Program Class D 186 (0xBA) 2 words	Mode	S16	0 = RS-232 (Default)

Example:

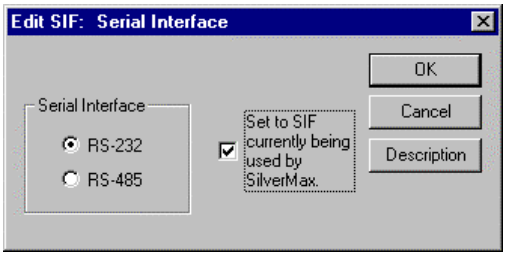
Set up SilverMax to use RS-232 for the serial interface

@16 186 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



See Also: **Select External Encoder (SEE), Dual Loop Control (DLC)**

Single Loop Control (SLC)

Description:

Configures SilverMax to run in the standard “Single Loop” control mode. All SilverMax encoder information for commutation, position, velocity and acceleration control is derived from the Internal Encoder.

If a motion is running, the servo Trajectory Generator must be shut down prior to executing this command or a sequence error will result.

When entering Single loop control SilverMax sets the current “Target” to the “Current position” (Internal Position from the Internal Encoder).

By default, SilverMax starts up in Single Loop Control mode.

See the Dual Control Loop (DLC) command for cases where external encoder position control is required. Switching between Single Loop and Dual Loop modes usually requires changing the control loop tuning.

Command Info:

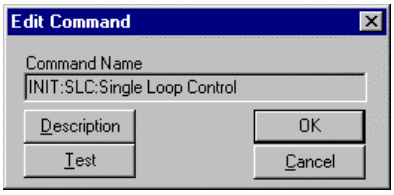
Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Single Loop Control (SLC)	Program Class D 244 (0xF4) 1 word	NONE	NONE	NONE

Example:

Configure SilverMax for Single Loop Control

@16 244 (CR)

QuickControl Example:



SilverMax Response:

ACK only

Soft Stop Limits (SSL)

See Also: **Write Register, Program Type (WRP) & Write Register, Immediate Type (WRI)**

Description:

Sets position limits for “End of Travel” control. Two registers are used to store the end limits. End of travel positions must be stored in the specified registers using a Write Register command (i.e. WRP or WRI). The register selection sets aside two registers in succession. Any motion affecting the Target is limited so as to keep the target more than the first register value and less than the second register value. If the target is beyond a limit, only motion in the direction toward that limit is allowed. The motion exceeding a given limit is hard stopped at the point that the limit is encountered – no ramping occurs. Internally, the motion calculations continue, but their effect is not directed to the Target value.

The registers can be set with limit values ranging from -2,147,483,647 to +2,147,483,647. This is the full positional range of the SilverMax. The limits consider the position as “Linear” rather than “Cyclic”. If the position attempts to wrap-around (going past the full range values), the Soft Stop Limits will prevent this movement.

The first register is used for the lower limit, which is checked when the direction of a motion is negative. The second register is used for the upper limit, which is checked when the direction is positive.

If the Lower Limit is set more positive than the Upper Limit, this will create a Dead Zone. If the servo’s position is in the Dead Zone, it will not be able to move. No error checking is done on the Data Register values to prevent this condition.

If the limits are set so that the Target is outside of the permitted range, only motions toward the permitted range are effective. Drag mode may allow the Target to be forced outside the permitted range if manually moved. Again, only motions that move toward or within the permitted range will have effect.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Soft Stop Limits (SSL)	Program Class D 221 (0xDD) 2 words	Data Register (First of two)	S16	0 = Not Used 10 to 39

Example:

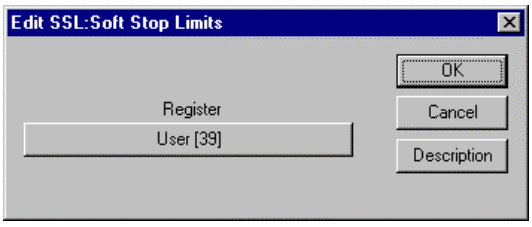
SilverMax uses Data Registers 39 & 40 for end a travel position limits

@16 221 39 (CR)

SilverMax Response:

ACK only

QuickControl Example:



See Also: **Error Limits (ERL) & Anti-Hunt Constants (AHC)**

Torque Limits (TQL)

Description:

This command sets the torque limits for the different operating modes of the SilverMax unit. The unit may be in either Open Loop or Closed Loop mode, and in either moving or holding position. The four parameters supplied set the limits on the output torque for all four combinations: Closed Loop Holding, Closed Loop Moving, Open Loop Holding, and Open Loop Moving.

SilverMax operates in moving mode whenever the target velocity or motion (i.e. SSD) is non-zero. The unit will continue to use the moving torque limits until Delay to Holding time ticks (120uS each) (see Error Limits command) after the last non-zero target velocity. This allows a higher moving torque limit to persist up to seven seconds after the last motion (or motion step of the Step and Direction command). Following this period (which power up defaults to zero), the holding torque levels are used.

The Open Loop torque (holding or moving) is used by the Anti-Hunt mode for the torque level that will be used to prevent hunting.

Typically, servos have a 100% torque level corresponding to 20,000 (see the table below). In QuickControl, the TQL command automatically scales percentage values to numeric values for all servo types. Peak torque above 100% is permitted, but at a reduced duty cycle and duration.

The setting for the Closed Loop parameters establishes the maximum torque limit within the control loop, but the servo current (and thus heat) will only be what is needed to generate the required torque. The Open Loop parameters, on the other hand, will cause the respective servo current (and heat) regardless of load. For most operations, the open loop parameters should be set to 50% or less to minimize servo heating; these are only used if the Anti-Hunt is enabled.

To determine the actual maximum torque that may be applied to the load, refer to the torque curve charts in found in the QuickSilver Controls Product Catalog. The 100% torque applied to the load depends on the speed of operation. 100% torque at stall may be 100 oz-in while the 100% torque at 2000 RPM may be 50 oz-in (data from 23H-3 Torque Curve).

When using TQL in QuickControl's Normal units, the conversion to native SilverMax torque units is done automatically for the user at time of download.

NOTE: The actual continuous torque levels are application dependent based many variables (i.e. motor type, speed, voltage, ambient temperature, heat sinking, air motion....).

See Torque Control in SilverMax User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Torque Limits (TQL)	Program Class D 149 (0x95) 5 words	Closed Loop Holding	U16	0 to 32767
		Closed Loop Moving	U16	0 to 32767
		Open Loop Holding	U16	0 to 32767
		Open Loop Moving	U16	0 to 32767

Torque parameter range changes depending on motor type. Use the following table to determine the range for your SilverMax™.

SilverMax™ P/N	100% Torque	Max Torque
Any 17 or 23	20000	30000
34N-1	20000	22500
34H-1	16383	24575
34H-2	16383	16851
34H-3	14745	14745
34H-4	10619	15929
Any 34HC	20000	32767

Example:

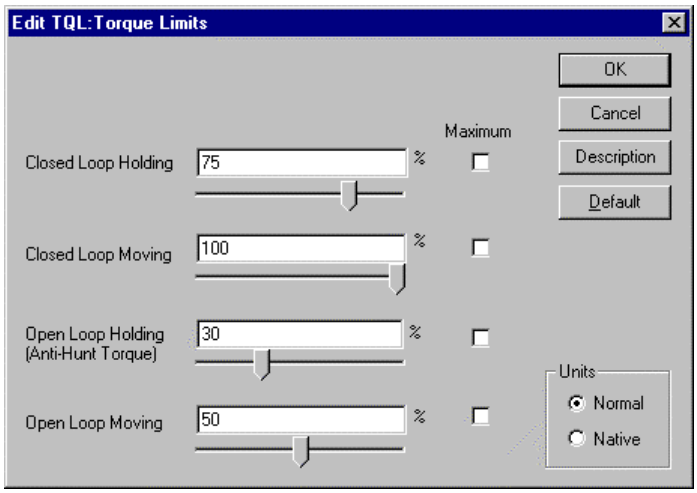
Set torque to:
Closed Loop Holding 75%
Closed Loop Moving 100%
Closed Loop Holding 30%
Closed Loop Moving 50%

@16 149 15000 20000 6000 10000(CR)

SilverMax Response:

ACK only

QuickControl Example:



Torque Ramp Up (TRU)

Description:

Ramps up the torque limit values by the increment given up to the final value. This is used mainly during servo initialization. Only Ramps up Open Loop Torque Limits. This command slowly brings up the Open Loop motor current to avoid a harsh or sudden movement during servo power up. This is done just prior to the algorithmic alignment of the motor rotor to the encoder. The increment sets how much current will be added each servo cycle (120usec).

The ramp up time is calculated by taking the final value divided by the increment times 120usec.

Example:
20000/5 = 4000
4000 * 0.00012 = 480 milliseconds

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Torque Ramp Up (TRU)	Program Class D 222 (0xDE) 3 words	Final Torque	S16	0 to 32767
		Increment per 120usec	S16	1 to 32767

Example:

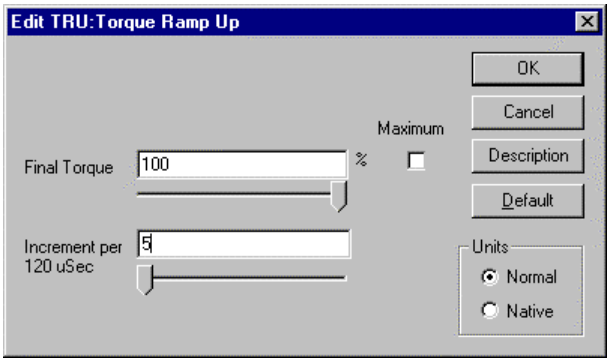
Set open loop current to 20000 (100%) in 4000 servo cycles (480 milliseconds)

@16 222 20000 5(CR)

SilverMax Response:

ACK only

QuickControl Example:



MODE COMMANDS

Mode Commands put SilverMax into required and special modes of operation. Some modes give SilverMax the ability to input analog or Step and Direction signals for motion control.

Go Closed Loop (GCL)

Description:

Puts SilverMax into closed loop operation. This is typically only done one time during initialization. This command is used to put SilverMax into closed loop mode if the unit has been placed into open loop mode. This command sets the phase relationship between the servo rotor and the encoder for closed loop operation. (See SilverMax Initialization in the SilverMax User Manual for more information.)

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Go Closed Loop (GCL)	Program Class D 142 (0x8E) 1 word	NONE	NONE	NONE

Example:

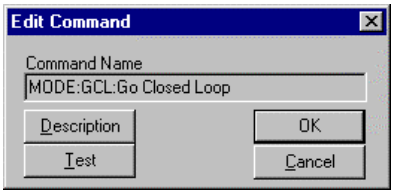
Put SilverMax into closed loop mode

@16 142 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Go Open Loop (GOL)

Description:

Puts SilverMax into open loop operation. This is the default servo power up mode. This command is used during servo initialization to aid in aligning the servo rotor to the encoder (See SilverMax Initialization in the SilverMax User Manual for more information.)

The command can also be used to force the servo into open loop mode. This is not recommended for normal operation, as the servo performance is severely degraded.

If the servo is in Dual Loop Control (DLC) operation when this command is encountered, it is forced back into Single Loop Control.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Go Open Loop (GOL)	Program Class D 143 (0x8F) 1 word	NONE	NONE	NONE

Example:

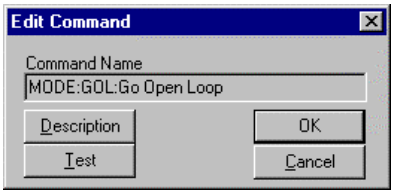
Put SilverMax into open loop mode

@16 143 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Position Input Mode (PIM)

Description:

Puts the SilverMax into a position control mode. Uses the contents of registers #12 -18 for position control processing. The Position Input Mode may be exited via I/O by configuring the I/O Exit State and enable. As with other motions, the Position Input Mode may also be exited via the Stop command, the Change Velocity Program, and the Profiled Move Exit command.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

Position Input Mode uses a set of Data Registers and processing to allow sophisticated manipulation of the input data. This allows the input signal to be calibrated to give the desired position control (see Input Mode Usage in SilverMax User Manual for more details on using this mode).

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Position Input Mode (PIM)	Program Class D Code (Hex): 216 (0xD8)	Filter Constant	S16	0 to 32767
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

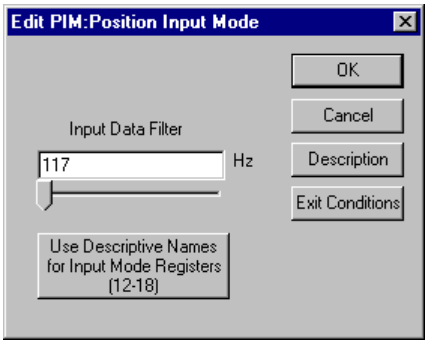
Position Input mode using a 117 Hz filter.
$30000 = 32768\ e^{-(117)2\pi(120uS)}$

@16 216 30000 0 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



See Also: **Scaled Step & Direction (SSD)**

Registered Step & Direction (RSD)

Description:

This command works the same as the Scaled Step and Direction (SSD) command below except that the “scaling” value is found in a User Data Register specified in the parameter.

The scaling value should be stored in the register prior to executing this command, but it may be modified at any time after the Step and Direction mode is initiated. All commands able to modify register contents can be used to set the scaling value. This includes the Calculation command (CLC), which can be used to dynamically adjust the scaling value when in Multitasking operation.

See High-Speed Signals in User Manual and for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Registered Step & Direction (RSD)	Program Class D 223 (0xDF) 2 words	Data Register	U16	10 to 40

Example:

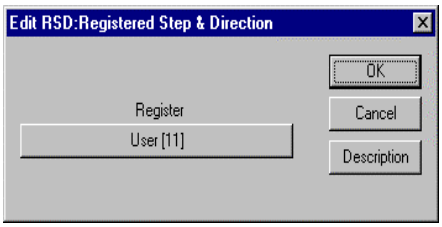
Put SilverMax into a Set & Direction mode using User Data Register #11 for the scaling value

@16 223 11 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Scaled Step & Direction (SSD)

See Also: **Select External Encoder (SEE)**

Description:

The Scaled Step and Direction (SSD) command causes the system to exit the hold mode of operation and follow a counter as a target position. The target position and target velocity are controlled by the step and direction input.

The Select External Encoder (SEE) command must be issued prior to SSD to configure the external input (encoder). See Select External Encoder for details.

Scale Factor: The maximum positive value of 32767 is approximately 2.88° per step clock.

1 to 1 Ration dependent on encoder Counts/Rev (CPR):

Encoder Counts/Rev (CPR)	1 to 1 Scale Factor
4000	1024
8000	512
16000	256

A negative value for the scaling factor reverses the state of the Direction Input (this is so that no external logic is required to invert the Direction input).

The SEE command causes the external inputs (encoder) to be counted and stored in 200. The Step and Direction inputs drive a counter, which is sampled every 120uS. The counts detected are scaled and summed to any remaining fractional count left from the prior period, with the whole count being applied to the Target Position value. The fractional remainder is saved for the following period. Counts in excess of the maximum (+31, -32) counts per sample period are accumulated for use in the following sample period to handle sample period to sample period variations. The command velocity should not exceed 4000RPM to prevent count loss.

For SilverMax Revision 29 and higher the counter is also double low pass filtered to generate an estimate of the Target Velocity for the Velocity Feedforward term. The filter time constants used are the same as is used for the Velocity #1 and Velociy#2 Filters in the feedback loop. All of the filter time constants in the system are set by the Filter Constants command. The system switches to Moving Torque and Error Limits checking upon the first Step pulse being received, and remains with these settings until the Hold Delay time has passed since the last Step pulse has been received and since the Target Velocity filter has been allowed to decay to zero. Once these time limits have elapsed, the system reverts to Holding Torque limits and Holding Error limits.

Some Anti-Hunt Delay (AHD) may be needed to avoid switching between moving and holding modes at low step rates.

As with other moves, the Scaled Step and Direction move may be exited via the Stop command, or by the Change Velocity Program, and the Profiled Move Exit command if multitasking is enabled.

See High-Speed Signals in User Manual and for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Scaled Step & Direction (SSD)	Program Class D 180 (0xB4) 2 words	Scale Factor	S16	-32767to 32767

Example:

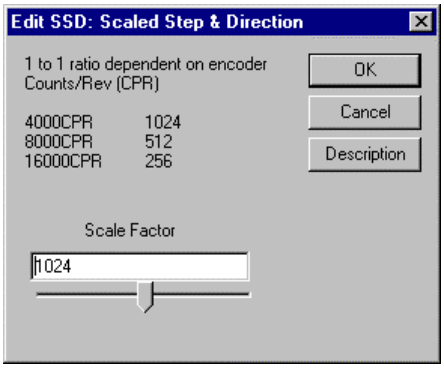
Put SilverMax into a Step & Direction mode with a 1:1 Scale Factor assuming a 4000 CPR encoder.

@16 180 1024 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Torque Input Mode (TIM)

Description:

Puts the SilverMax into a torque control mode. Uses the contents of data registers #12 -18 for torque control processing while the servo is moving.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

Torque Input Mode uses a set of Data Registers and processing to allow sophisticated manipulation of the input data. This allows the input signal to be "calibrated" to give the desired torque control. (See Input Mode Usage in SilverMax User Manual for more details on using this mode).

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Torque Input Mode (TIM)	Program Class D 218 (0xDA) 4 words	Filter constant	S16	0 to 32767
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

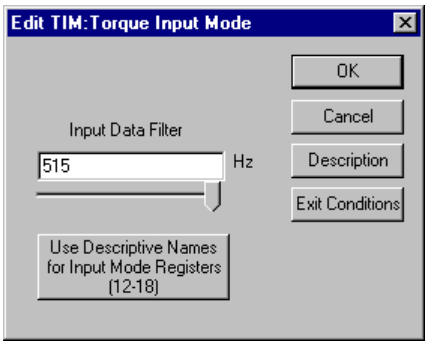
Torque Input mode using a 515 Hz filter. Exit if Input #1 is high ("1")
$30000 = 32768\ e^{-(117)2\pi(120uS)}$

@16 216 2222 –1 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Velocity Input Mode (VIM)

Description:

Puts the SilverMax into a Velocity control mode. Uses the contents of registers #12 -18 for velocity control processing.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

Velocity Input Mode uses a set of Data Registers and processing to allow sophisticated manipulation of the input data. This allows the input signal to be calibrated to give the desired velocity control. (See Input Mode Usage in SilverMax User Manual for more details on using this mode)

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Velocity Input Mode (VIM)	Program Class D 217 (0xD9) 4 words	Filter Constant	S16	0 to 32767
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

Velocity Input Mode using a 117 Hz filter.

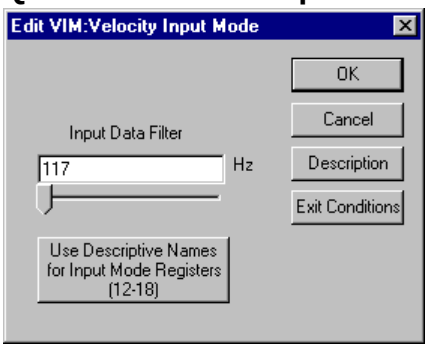
$30000 = 32768\ e^{-(117)2\pi(120uS)}$

@16 217 30000 –1 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Velocity Mode, Immediate Mode (VMI)

Description:

Accelerates the servo from the present velocity to the indicated velocity using the given acceleration. If the servo has an active move operation in progress, that motion is taken over from its current velocity, and servo ramps to the new velocity at the given acceleration rate. Any program operating is stopped and the contents of the command buffer are modified. This command is used when the velocity mode needs to be controlled from a Host controller. This command can only be used through the serial interface. See the Velocity Mode, Program Mode (VMP) command for embedding this type of command in a program.

NOTE: If the acceleration is negative, any accumulated position error is removed and the absolute value of the acceleration is then used.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Scaling in User Manual for more details on SilverMax acceleration and velocity units.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Velocity Mode, Immediate Mode (VMI)	Immediate Class A 15 (0xF) 7 words	Acceleration	S32	-1 to -1,073,741,823 or 1 to 1,073,741,823
		Velocity	S32	-2,147,483,647 to +2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

Put SilverMax into velocity mode running at 200 RPM.

@16 15 200000 107374200 0 0(CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

ACK only

Velocity Mode, Program Mode (VMP)

Description:

Accelerates the servo to the indicated velocity using the given acceleration. This command may be run from within a program. When this command is executed in a program, the motion will continue until the velocity reaches zero. Issuing the command with a non-zero velocity and stop on I/O enabled will allow the servo to run at velocity until the selected stop configuration is met; the velocity then ramps down to zero and the motion ends . This command can also be used through the serial interface, however a Nak Busy will be reported when a Program or a motion command is executing. See the Velocity Mode**,** Immediate Mode (VMI) command above for velocity mode using the serial interface. If multitasking is enabled, this command will take over any executing motion with out the completion of that motion, and may be used to shutdown a motion if the new velocity is zero.

NOTE: If the acceleration is negative, the accumulated position error is removed and the absolute value of the acceleration is used.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Scaling in User Manual for more details on SilverMax acceleration and velocity units.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Velocity Mode, Program Type (VMP)	Program Class D 159 (0x9F) 7 words	Acceleration	S32	-1 to -1,073,741,823 or 1 to 1,073,741,823
		Velocity	S32	-2,147,483,647 to +2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

Put SilverMax into velocity mode running at (See Scaling):

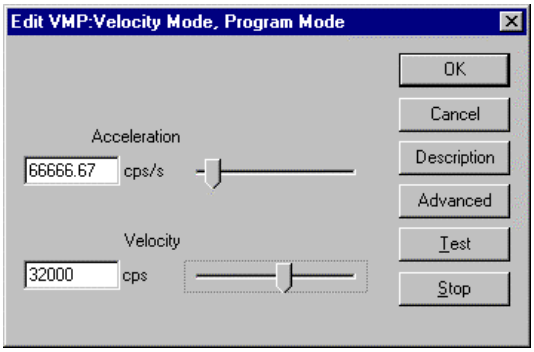
Vel = 32000 cps
Acc = 666666.67

@16 159 64425 257698038 0 0(CR)

SilverMax Response:

ACK only

QuickControl Example:



MOTION & PROFILE MOVE COMMANDS

Motion & Profile Move commands make up the set of commands that use the SilverMax Trajectory Generator to perform simple or complex motions.

Extended Register Move Absolute, Time Based (XAT)

Description:

The Extended Register Move Absolute performs an absolute position move using move parameters contained in the indicated User Data Registers. This command works like the basic Move Absolute, Time Based (MAT) command in all other ways.

The move parameters are retrieved from the User Data Registers in the following order:
If Starting Data Register = N:

N = Position
N + 1 = Acceleration Time
N + 2 = Total Time

The move parameters must be written into the User Data Registers prior to executing the move command. The range of the move parameters stored in the User Data Registers must be the same as those used in the basic Move Absolute, Time Based (MAT) command. The Write Register commands can be used to write the move parameters into the Data Registers. The Write Register commands do not type check the parameters when written, therefore errors may occur at time of execution. Changing the move parameters during an actual move will have no affect on the move in progress.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Extended Register Move Absolute, Time Based (XAT)	Program Class D 236 (0xEC) 4 words	Starting Data Register	S16	Standard Register Range
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

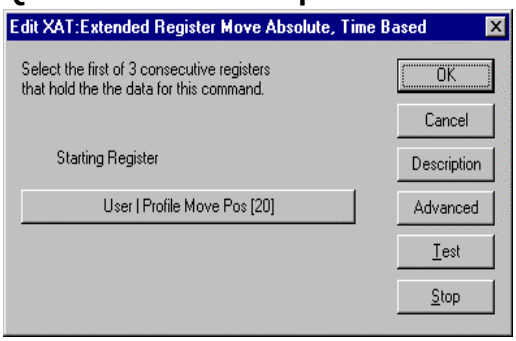
Move SilverMax using parameters from User Data Registers #20-22.

@16 236 20 0 0(CR)

SilverMax Response:

ACK only

QuickControl Example:



Extended Register Move Absolute, Velocity Based (XAV)

Description:

The Extended Register Move Absolute performs an absolute position move using move parameters contained in the indicated User Data Registers. This command works like the basic Move Absolute, Velocity Based (MAV) command in all other ways.

The move parameters are retrieved from the User Data Registers in the following order:
If Starting Data Register = N:

N = Position
N + 1 = Acceleration
N + 2 = Velocity

The move parameters must be written into the User Data Registers prior to executing the move command. The range of the move parameters stored in the User Data Registers must be the same as those used in the basic Move Absolute, Velocity Based (MAV) command. The Write Register commands can be used to write the move parameters into the Data Registers. The Write Register commands do not type check the parameters when written, therefore errors may occur at time of execution. Changing the move parameters during an actual move will have no affect on the move in progress.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Extended Register Move Absolute, Velocity Based (XAV)	Program Class D 234 (0xEA) 4 words	Starting Data Register	S16	Standard Register Range
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

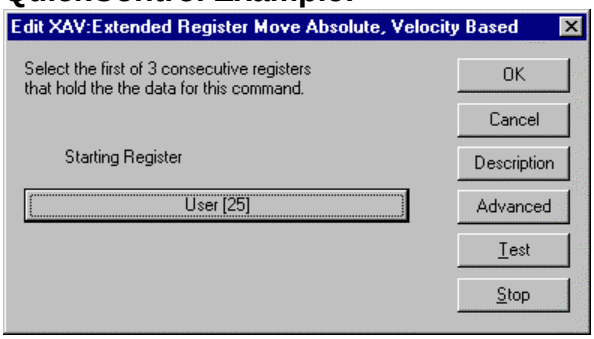
Move SilverMax using parameters from User Data Registers #25-27.

@16 234 25 0 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Extended Register Move Relative, Time Based (XRT)

Description:

The Extended Register Move Relative performs a relative distance move using move parameters contained in the indicated User Data Registers. This command works like the basic Move Relative, Time Based (MRT) command in all other ways.

The move parameters are retrieved from the User Data Registers in the following order:
If Starting Data Register = N:
N = Distance
N + 1 = Acceleration Time
N + 2 = Total Time
The move parameters must be written into the User Data Registers prior to executing the move command. The range of the move parameters stored in the User Data Registers must be the same as those used in the basic Move Relative, Time Based (MRT) command. The Write Register commands can be used to write the move parameters into the Data Registers. The Write Register commands do not type check the parameters when written, therefore errors may occur at time of execution. Changing the move parameters during an actual move will have no affect on the move in progress.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Extended Register Move Relative, Time Based (XRT)	Program Class D 235 (0xEB) 4 words	Starting Data Register	S16	Standard Register Range
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

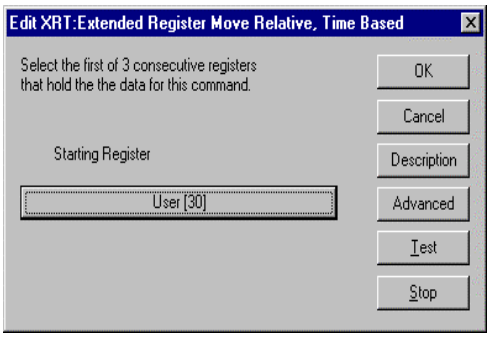
Move SilverMax using parameters from User Data Registers #30-32.

@16 235 30 0 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Extended Register Move Relative, Velocity Based (XRV)

Description:

The Extended Register Move Relative performs a relative distance move using move parameters contained in the indicated User Data Registers. This command works like the Move Relative, Velocity Based (MRV) command in all other ways.

The move parameters are retrieved from the User Data Registers in the following order:
If Starting Data Register = N:
N = Distance
N + 1 = Acceleration
N + 2 = Velocity
The move parameters must be written into the User Data Registers prior to executing the move command. The range of the move parameters stored in the User Data Registers must be the same as those used in the basic Move Relative, Velocity Based (MRV) command. The Write Register commands can be used to write the move parameters into the Data Registers. The Write Register commands do not type check the parameters when written, therefore errors may occur at time of execution. Changing the move parameters during an actual move will have no affect on the move in progress.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Extended Register Move Relative, Velocity Based (XRV)	Program Class D 233 (0xE9) 4 words	Starting Data Register	S16	Standard Register Range
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

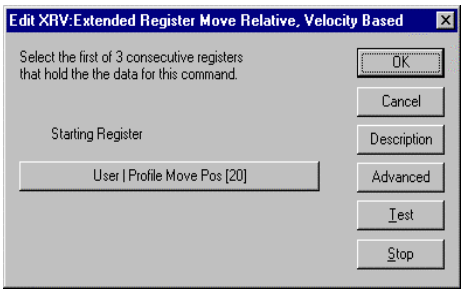
Move SilverMax using parameters
from User Data Registers #20-22.

@16 233 20 0 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Hard Stop Move (HSM)

See Also: **Velocity Mode, Program Type (VMP)**

Description:

This command provides a way, while in multitasking operation, to execute a hard stop of any move or mode from within a program. A hard stop immediately halts the trajectory generator (motion commands) or stops the current mode, in either case the motor will come to an abrupt stop. In many situations, this may cause the motor to overshoot the stop position and oscillate until settled. More controlled stops can be accomplished by using the Velocity Mode which allows a user selectable deceleration to “0” velocity (stopped). The Profile Move Exit (PMX) command may similarly be used to halt an existing motion with a controlled deceleration.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Hard Stop Move (HSM)	Program Class D 229 (0xE5) 1 word	NONE	NONE	NONE

Example:

Stop the SilverMax immediately.

@16 229 (CR)

SilverMax Response:

ACK only

QuickControl Example:

Interpolated Move Start (IMS)

See Also: **Profiled Move (PMV), Interpolated Move Queue Clear (IMQ), Interpolated Move Write (IMW)**

Description:

This command provides a means generating an arbitrary motion from either nonvolatile memory, or from a host via the serial interface. Before issuing this command, Register #17 should be written to point to the source of the profile data, while Register #19 should contain a deceleration value to use if the data stream were to become interrupted. This command takes succeeding sets of 4 data values, and copies them, within a single normal control cycle, into the associated registers used by the profiled move command. It then waits the designated number of cycles (120 microseconds each) before loading the next set of data. At the same time, a Profiled Move operation is running in the background, using the given data. Each set of data represents a timed slice of the total motion, consisting of a constant acceleration period ramping to the new velocity, followed by a constant velocity period until the next set of data is loaded. If the given destination is reachable in the time slice given the other parameters, the target position will come to rest there until the next set of data is loaded that requires the motion to begin again. Complex moves involving multiple axis may be generated that may run from either internal Non volatile memory, or from the Serial interface.

These two distinct modes of operation are selected by the value of the data in Register #17. A non-zero value indicates the address of the first of the four registers that will hold the data. Once this command is executed, the contents of the first of these registers will be copied to Register #18 to be used as a time countdown. The continuing operation of this command will decrement Register #18 each cycle. No external modification of Register #18 should be made while this mode is active.

The second register in this bank of four contains the target position for this time segment. If the segment is intended to end with the velocity non-zero, the value should be the approximate actual position + or - 1/2 full scale. This indicates the desired direction of travel. If the final velocity of this segment is zero, then the position should be the desired stopping position. This value is automatically copied to Register 20 for use by the profiled move operation. Note that the velocity for the last segment should not be set to zero, but left as the initial value for the segment.

The third register in this bank of four contains the acceleration or deceleration magnitude (positive values only) for this segment of the move. It is copied to Register # 21 and #23 for use by the profiled move operation.

The fourth register in this bank of four contains the speed (absolute value of the velocity) for this segment of the move. It is automatically copied to Register #22 for use by the profiled move operation.

After the four values have been copied, The upper word of Register #17 is set to “1” to indicate the data has been transferred and is now stale. If updating from internal memory, a multitasking program should be looping until this register has been modified. Next it needs to either point to the next set of data (Register 34, for example) if the data representing the next segment has already been loaded, or it needs to load the data representing the next segment of the move into the same set of Registers, then re-write Register #17 with the starting Register address. If the current movement segment ends while the upper half of Register #17 is non-zero, then the deceleration data in Register #19 is used to decelerate the motion to a stop. An error flag is set in bit 12 of the Status word to indicate a timeout in the data stream. The motion is stopped, but the program continues in operation. The final motion segment of a move is denoted by setting the segment time counter, the data in the first word of the four data words to zero. The final four words of segment information are copied into the respective profiled move operation registers, and the move continues as a profiled move.

This move may also be driven from the Serial Interface. The first method is to initialize the registers, as was explained above, but using the serial interface to write the registers and monitor Register # 17 to determine when the next data needs to be supplied. The preferred method is to use the Interpolated Move Queue Clear and Interpolated Move Write commands. In this mode, the Interpolated Move Clear , a program mode command, is used to clear our any existing data from the a four deep by four long word software FIFO specifically provided for this use. Next , Register #17 is set to zero (0) to indicate data will be drawn from the circular queue. Register #19 should be initialized with an appropriate deceleration value to use to stop the motion in the case of loss of communications. The Interpolated Start Move command may then be issued. The motion will not start until the first data set has been written to the software FIFO (circular queue). The queue may be kept full by the host via the Serial Interface. The buffering makes it easier to keep one or multiple axis fed with data. It also eliminates an extra register read to determine when data is required. (See Interpolated Move Write Queue (IMW) command for details.) The Interpolated move continues until either the Segment Time read is a zero, which terminates as detailed above, or until the queue is found empty when data is required, which uses the deceleration data in Register # 19 to bring the motion to an end, setting error bits as described above.

See Interpolated Motion Control in the User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Interpolated Move Start (IMS) Rev 29	Program Class D 253 (0xFD) 1 word	NONE	NONE	NONE

Register usage:

Register #17 Points to Register containing Segment Data, or “0” for Queued operation.
Register #18 Used internally to hold segment time countdown
Register #19 Holds data loss deceleration value
Registers # 20 to 24, as defined in the Profiled Move command (PMV**)**

Note: Register #17 is modified following each non-queued data transfer at the start of each segment.See notes above.

Example:

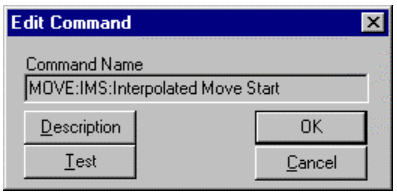
Start Interpolated Move.

@16 253 (CR)

SilverMax Response:

ACK only

QuickContol Example:



Interpolated Move Queue Clear (IMQ)

SeeAlso: **Profiled Move (PMV), Interpolated Move Start (IMS), Interpolated Move Write (IMW)**

Description:

This Command clears any data that may have been left in the Interpolated Move Queue. This queue is a software FIFO (First in first out) buffer capable of holding data for up to four interpolated motion segments, the data for each segment consisting of four long words (32 bits each) of data. If the data is able to fit within the queue, it is accepted and the communication is acknowledged. If the queue is full, the request is answered with a NAK – Full response. This just indicates that the host is successfully keeping the queue filled. The same data should be sent again until it is positively Acknowledged.

See Interpolated Move in User Manual for details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Interpolated Move Queue Clear (IMQ) Rev 29	Program Class D 254 (0xFE) 1 word	NONE	NONE	NONE

Example:

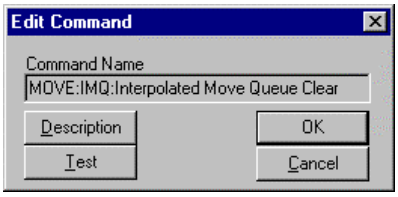
Stop the SilverMax immediately.

@16 254 (CR)

SilverMax Response:

ACK only

QuickContol Example:



Interpolated Move Write Queue (IMW)

See Also: **Profiled Move (PMV), Interpolated Move Start (IMS), Interpolated Move Clear (IMC)**

Description:

This Command writes data to the Interpolated Move Queue through the Serial Interface. This queue is a software FIFO (First in first out) buffer capable of holding data for up to four interpolated motion segments, the data for each segment consisting of four long words (32 bits each) of data. If the data is able to fit within the queue, it is accepted and the communication is acknowledged. If the queue is full, the request is answered with a NAK – Full response. This NAK is to be expected: it just indicates that the host is successfully keeping the queue filled. The same data should be sent again until it is positively Acknowledged.

The four long words of data associated with each Interpolated Move segment are:
Time ticks:Indicating the number to 120 microseconds time slices the segment is to last. A "0" indicates it is the last segment of the move.
Position. Unless the segment is intended to come to a halt at a given location, this should be full scale positive if the final velocity of the segment is positive, or full scale negative if the final velocity of the segment is to be negative.
Acceleration: Indicates the acceleration or deceleration to be used in reaching the requested velocity or position.
Velocity: Indicating the desired ending velocity magnitude (speed) of the segment, or maximum velocity to use if coming to a stop within this segment. Will only be reached if it is consistent with the starting velocity, the acceleration, and the segment time.

See "Scaling" in User Manual for details on scaling the parameters to engineering units.
See Interpolated Move in User Manual for details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Interpolated Move Write Queue (IMW) Rev 29	Immediate Class A 25 (0x19) 9 words	Time	S32	0 to +2,147,483,647
		Position	S32	-2,147,483,648 to +2,147,483,647
		Acceleration	S32	1 to 2,147,483,647
		Velocity	S32	0 to +2,147,483,647

Example:

Move to position -456 at acceleration 7890 at velocity 1234 and wait for 123 ticks before getting the next command.

@16 25 123 –456 7890 1234 (CR)

.

SilverMax Response:

ACK
NAK –FULL

Response Example:

* 10 (CR)

Negative Acknowledge (NAK) , Command 25 (0x19), Reason = Queue Full ! 10 0019 0006 (CR)

The NAK indicates that the data was rejected. This should commonly happen if the Host is keeping ahead of the SilverMax unit. It means that the queue has been kept full, and to send the data again. Periodic polling should also be done to see that the move has not improperly ended due to the Host falling behind the consumption rate of the SilverMax, and data not being available when needed.

QuickContol Example:

Immediate (Host) Mode Command Only

Move Absolute, Time Based (MAT)

Description:

Move Absolute initiates a move to an absolute position. Absolute positions are based on the incremental encoder resident in SilverMax. The move profile uses time as the constraint for the Acceleration period and for the total move.

Position is in encoder counts. For a SilverMax with an encoder that provides 4000 counts per revolution, one revolution of the motor equals 4000 counts.

Time based moves make the desired move within the times given. The acceleration and velocity are calculated to accommodate the time requirements. If the acceleration or velocity values needed to make the move exceed the maximum permissible limit, or the ramp times exceed 65534 ticks (120uS each) the move will not be executed and an error code will be set in the Polling Status Word. The Polling Status Word bit #14

The Total Time is a time value in ticks. Ticks are based on update rate of the servo cycle, which is 120 microseconds long. The Acceleration Time is also represented by ticks. This time value defines the time needed to accelerate up to the calculated velocity. The Total Time and distance must be consistent with the maximum velocity constraints of the motor to prevent the generation of an error. Acceleration Time can be no greater the ½ the Total Time value and not greater than 7.864 seconds in time (65534 ticks).

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Move Absolute, Time Based (MAT)	Program Class D 176 (0xB0) 9 words	Position	S32	-2,147,483,648 to +2,147,483,647
		Acceleration Time	U32	0 to 65534
		Total Time	U32	2 to 2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

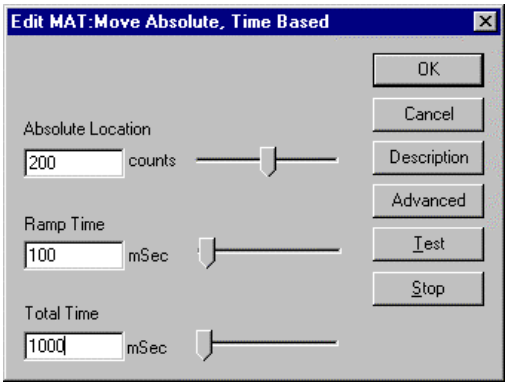
Move SilverMax to position 200 in 1.0 seconds with a 0.1 second acceleration.

@16 176 200 83 8333 0 0(CR)

SilverMax Response:

ACK only

QuickControl Example:



Move Absolute, Velocity Based (MAV)

Description:

Move Absolute initiates a move to an absolute position. Absolute positions are based on the incremental encoder resident in SilverMax. The move profile uses acceleration and velocity as the constraints.

Velocity based moves use an acceleration and velocity parameter to accomplish the motion profile. If the Velocity value needed to make the move exceeds the maximum permissible limit, the move will not be executed and an error code will be set in the Polling Status Word. The Polling Status Word bit #14 Foreground Command Error will be set. Foreground command errors are generated when a command cannot be executed due to parameters that are out of range. If the Acceleration value exceeds the maximum permissible value, the Acceleration is limited to the maximum.

Acceleration parameter can be no greater than ½ the Velocity parameter.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Move Absolute, Velocity Based (MAV)	Program Class D 134 (0x86) 9 words	Position	S32	-2,147,483,648 to +2.147.483.647
		Acceleration	U32	1 to 1,073,741,823
		Velocity	U32	0 to 2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

Move SilverMax to position 0 at 56000 cps (see Scaling).

@16 134 0 96637 450971566 0 0(CR)

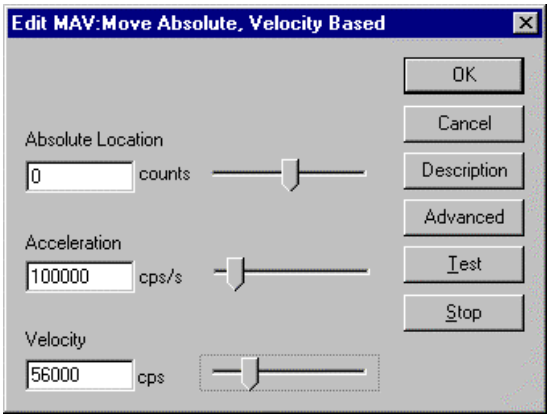
SilverMax Response:

ACK only

Response Example:

* 10 (CR)

QuickControl Example:



Move Relative, Time Based (MRT)

Description:

Move Relative initiates a distance move relative to the current target position. Relative Distances are based on the incremental encoder resident in SilverMax. The move profile uses time as the constraint for the acceleration period and for the total move.

The Distance units are in encoder counts. For a SilverMax with an encoder that provides 4000 counts per revolution, one revolution of the motor is 4000 counts.

Time based moves will attempt to make the desired move within the times given. The acceleration and velocity are calculated to accommodate the time requirements. If the acceleration or velocity values needed to make the move exceed the maximum permissible limit the move will not be executed and an error code will be set in the Polling Status Word. The Polling Status Word bit #14 Foreground Command Error will be set. Foreground command errors are generated when a command cannot be executed due to parameters that are out of range.

The Total Time is a time value in ticks. Ticks are based on the time of a servo cycle which defines each tick as 120 microseconds long. The Acceleration Time is also represented by ticks. This time value defines the time needed to accelerate up to the calculated velocity. The Total Time and Distance must be consistent with the maximum velocity constraints of the motor (4000 RPM) to prevent the generation of an error. Acceleration Time can be no greater than ½ the Total Time value and not greater than 7.864 seconds in time (65534 ticks).

To convert Total Time and Acceleration Time to seconds multiply by 0.00012.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Move Relative, Time Based (MRT)	Program Class D 177 (0xB1) 9 words	Distance	S32	-2,147,483,648 to +2.147.483.647
		Ramp Time	U32	0 to 65534
		Total Time	U32	2 to 2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

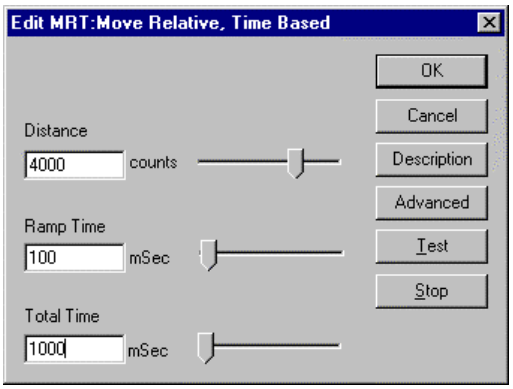
Move SilverMax 4000 counts from its current position. Do the move in 1 second with a 0.1 second acceleration.

@16 177 4000 833 8333 0 0(CR)

SilverMax Response:

ACK only

QuickControl Example:



Move Relative, Velocity Based (MRV)

Description:

Move Relative initiates a distance move relative to the current target position. Relative distances are based on the incremental encoder resident in SilverMax. The move profile uses acceleration and velocity as the constraints.

Relative Distance is an encoder count value that the motor will move from its current position. Providing a positive value will cause the motor to move in a positive count direction. Giving a negative value will cause the motor to move in negative count direction. The polarity of the distance number always determines the direction that the motor will spin. This is not true for the Absolute Position commands, which will move the motor in the required direction to go to the requested Position.

The Distance units are in encoder counts. For a SilverMax with an encoder that provides 4000 counts per revolution, one revolution of the motor is 4000 counts.

Velocity based moves use an acceleration and velocity parameter to accomplish the motion profile. If the Velocity value needed to make the move exceeds the maximum permissible limit, the move will not be executed and an error code will be set in the Polling Status Word. The Polling Status Word bit #14 Foreground Command Error will be set. Foreground command errors are generated when a command cannot be executed due to parameters that are out of range. If the acceleration value exceeds the maximum permissible value, the acceleration is limited to the maximum.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Move Relative, Velocity Based (MAV)	Program Class D 135 (0x87) 9 words	Distance	S32	-2,147,483,648 to +2.147.483.647
		Acceleration	U32	1 to 1,073,741,823
		Velocity	U32	0 to 2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

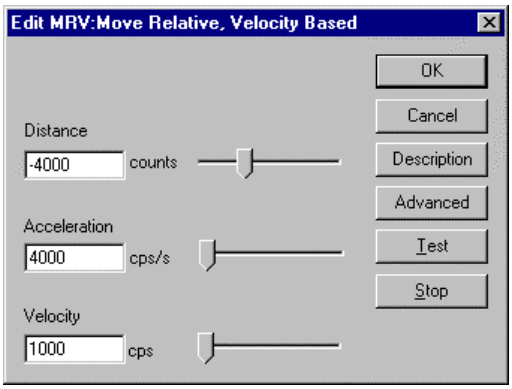
Move SilverMax -4000 counts from its current position at 1000cps.

@16 135 -4000 3865 8053064 0 0(CR)

SilverMax Response:

ACK only

QuickControl Example:



Pre-Calculated Go (PCG)

See Also: **Pre-Calculate Move (PCM)**

Description:

After a Pre-Calculate Move (PCM) command has been successfully executed, the pre-calculated move is in a ready state. The PCG command initiates the move, which will begin immediately (within 120 usec.)

If the pre-calculation was bad (parameters out of range) or a second motion command precedes the PCG, a program error will result and the motor will be shut down.

It is not required for the PCG command to immediately follow the Motion command. Other commands that do not affect the Motion intermediate values can be executed before the PCG. A second motion command or the velocity mode command before the PCG will cause a program error.

Command Info:

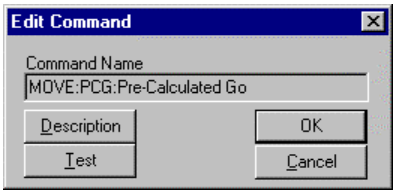
Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Pre-Calculated Go (PCG)	Program Class D 232 (0xE8) 1 word	NONE	NONE	NONE

Example:

Perform the Calculation Task on following Motion command.

@16 232 (CR)

QuickControl Example:



SilverMax Response:

ACK only

Pre-Calculate Move (PCM)

See Also: **Pre-Calculated Go (PCG)**

Description:

The Pre-Calculate Move command causes SilverMax to perform just the calculation task on a motion command. The motion command to be precalculated must immediately follow the Pre-Calculate Move command or no action is taken. When this command has been successfully executed, the pre-calculated move will be in a ready state waiting for the Pre-Calculated Go command.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Pre-Calculated Move (PCM)	Program Class D 231 (0xE7) 1 word	NONE	NONE	NONE

Example:

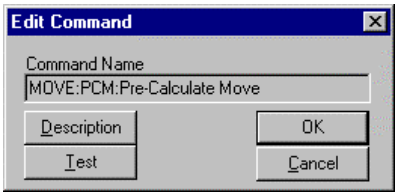
Start the pre-calculated move.

@16 231 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Profile Move Continuous (PMC)

Description:

The Profile Move commands are distinct from the Motion commands in that the move parameters can be modified while the motion is in progress. A change in a move parameter updates the move immediately and can alter the move profile “real-time”.

The Profile Move Continuous puts SilverMax into a move that does not end unless explicitly commanded. During the move, any move parameter can be updated either by a Host controller using the serial interface or by an internal program (Multitasking operation is required).

With this feature, any motion profile shape can be accomplished by changing the appropriate parameter at the desired time. Five parameters are associated with this command. Each of the parameters is dedicated to a specified User Data Register. Modifying the contents of the Data Register modifies the parameter.

See Scaling in User Manual for more details on SilverMax acceleration and velocity units.

The following table shows the list of the parameters and their associated Data Register:

Data Register	Description	Data Range	Comment
20	**Position**	-2,147,483,648 to +2,147,483,647	This is an “Absolute” destination value.
21	**Acceleration**	2 to 1,073,741,823	Sets the acceleration rate that is used when increasing the move speed.
22	**Velocity**	0 to 2,147,483,647	The maximum speed that is allowed during a move
23	**Deceleration**	2 to 1,073,741,823	Sets the deceleration rate that is used when decreasing the move speed.
24	**Offset**	-2,147,483,648 to +2,147,483,647	A distance value to move that is added to the current position when a “Stop Condition” is encountered

Data Registers must be pre-loaded with the move parameters prior to issuing the Profile Move Continuous command.

Profiles Moves begin immediately after executing the command (within 120 usec.). The motor is accelerated using the Acceleration parameter until the maximum Velocity is reached. Deceleration begins when the distance of the move is such that the Absolute Position is achieved at the same time the motor has decelerated to “0” velocity. Depending on the parameters the maximum velocity may never be reached (Triangle Move).

During a Profile Move, SilverMax is constantly recalculating its intermediate move values (every 120 usec.). This is done by taking the given move parameters, the current position and current velocity and adjusting what is required to hit the absolute position. This means that SilverMax can even go from a Velocity Mode into a Profile Move without needing to stop first (Multitasking operation is required). Remember that the move calculations are being done continually. Therefore, the parameters can be changed at any time and affect the motion in process.

The Acceleration and Deceleration parameters should typically be no greater than a ratio of 100:1 of each other (one value is no greater than 100 times the other) for numerical stability. For higher ratios user must verify proper operation.

The Position parameter can act as a Relative Distance value by using the Add To Register command to increase or decrease the Position value. (See Add To Register for more details)

The Offset parameter is used to extend a move by the Offset Distance after a Stop Condition is encountered. In cases where a move needs to continue a prescribed distance past the point where a sensor triggers a stop, this parameter can be used to precisely control that offset distance to be moved. Note that the offset is automatically negative if the direction of motion is negative when the input is found. The Offset parameter allows trailing edge registration operations.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Profile Move in User Manual for details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Profile Move Continuous (PMC)	Program Class D 240 (0xF0) 3 words	Stop Enable	S16/U16	See above
		Stop State	S16/U16	See above

Example:

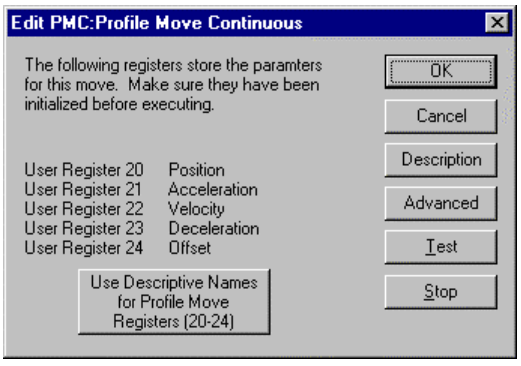
Put SilverMax into a continuous Profile move. Stop if Input #1 is high (“1”).

@16 240 –1 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



See Also: **Profile Move Continuous (PMC)**

Profile Move Override (PMO)

Description:

The Profile Move Override command allows a Profile Move Continuous to end when the Position is achieved. Normally the Move Continuous will not end until explicitly stopped by a Stop Condition or another command. The Override provides a graceful way to end the move so that the entire motion is completed with the motor stopping at the defined position. PMO will also override all other motions, including Step and Direction, if multitasking is enabled.

PMO operates exactly like the Profile Move command except that it does not wait for the previous motion to complete.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Profile Move Override (PMO)	Program Class D 249 (0xF9) 3 words	Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

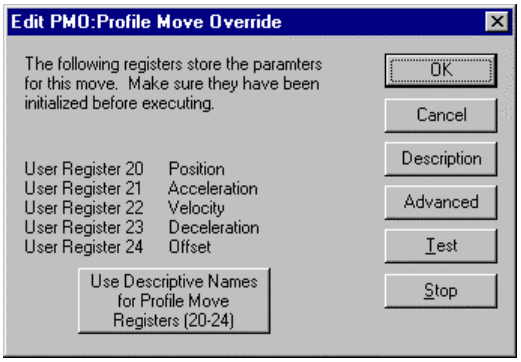
End the current Profile move when at “Position”. Stop if Input #1 is high (“1”).

@16 249 –1 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



See Also: **Profile Move Continuous (PMC)**

Profile Move (PMV)

Description:

The Profile Move command works identical to the Profile Move Continuous except that when the Position is achieved, the move ends and the trajectory generator goes inactive. All of the parameters including the position can be changed while the move is executing. Once the move has ended, changing the parameters will have no effect.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Profile Move (PMV)	Program Class D 241 (0xF1) 3 words	Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

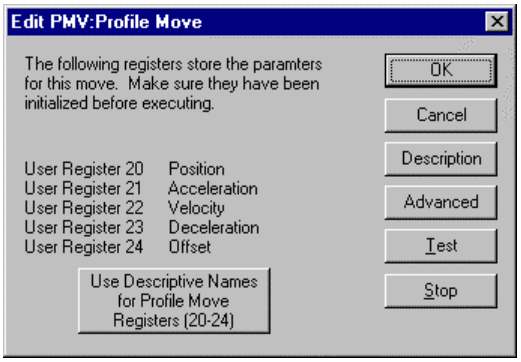
Start a SilverMax Profile move.
Stop if Input #1 is high (“1”).

@16 241 –1 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



See Also: **Profile Move Continuous (PMC)**

Profile Move Exit (PMX)

Description:

Exits the current Profile Move allowing the move to stop using the Deceleration parameter stored in Data Register #23. This command will work to stop any Motion, Profile Move or Mode (as long as register 23 has been initialized). The deceleration begins immediately and the profile destination will not be reached.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Profile Move Exit (PMX)	Program Class D 242 (0xF2) 1 word	NONE	NONE	NONE

Example:

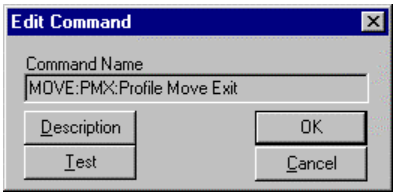
Exit the current move.

@16 242 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Register Move Absolute, Time Based (RAT)

Description:

The Register Move Absolute performs an absolute move using a position value contained in the indicated User Data Register. This command works like the basic Move Absolute, Time Based (MAT) command in all other ways.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Register Move Absolute, Time Based (RAT)	Program Class D 178 (0xB2) 9 words	Data Register	S32	Standard Register Range
		Acceleration Time	U32	0 to 65534
		Total Time	U32	2 to 2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

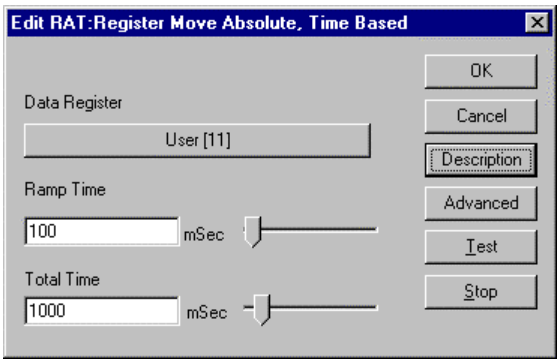
Move SilverMax to position indicated by User Data Register #11 in 1000 mSec with a 100 mSec acceleration (see Scaling in User Manual).

@16 178 11 833 8333 0 0(CR)

SilverMax Response:

ACK only

QuickControl Example:



Register Move Absolute, Velocity Based (RAV)

Description:

The Register Move Absolute performs an absolute move using a position value contained in the indicated User Data Register. This command works like the basic Move Absolute, Velocity Based (MAV) command in all other ways.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Register Move Absolute, Velocity Based (RAV)	Program Class D 160 (0xA0) 9 words	Data Register	S32	Standard Register Range
		Acceleration	U32	1 to 1,073,741,823
		Velocity	U32	0 to 2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

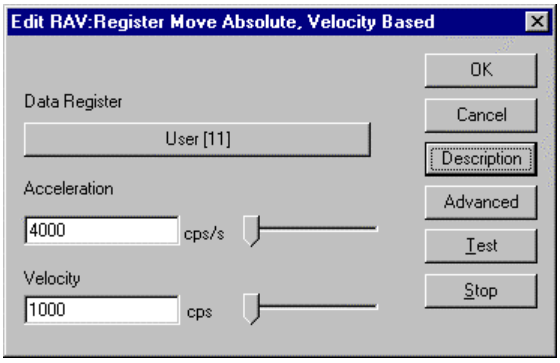
Move SilverMax to position indicated by User Data Register #11 at vel=1000 cps and acc=4000cps/s.

@16 160 11 3865 8053064 0 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Register Move Relative, Time Based (RRT)

Description:

The Register Move Relative performs a relative move using a distance value contained in the indicated User Data Register. This command works like the basic Move Relative, Time Based (MRT) command in all other ways.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Register Move Relative, Time Based (RRT)	Program Class D 179 (0xB3) 9 words	Data Register	S32	Standard Register Range
		Acceleration Time	U32	0 to 65534
		Total Time	U32	2 to 2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

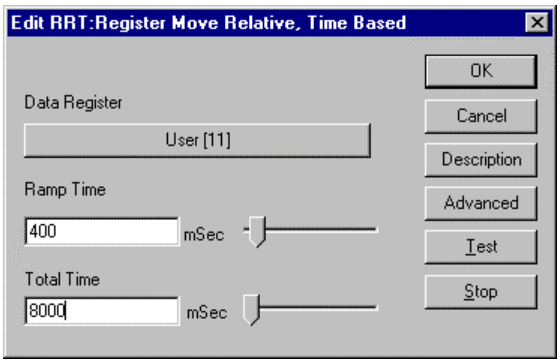
Move SilverMax to position indicated by User Data Register #11. Do the move in 8 seconds with a 0.400 second acceleration.

@16 179 11 3333 66664 0 0(CR)

SilverMax Response:

ACK only

QuickControl Example:



Register Move Relative, Velocity Based (RRV)

Description:

The Register Move Relative performs a relative move using a distance value contained in the indicated User Data Register. This command works like the basic Move Relative, Velocity Based (MRV) command in all other ways.

Stop Enable and Stop State parameters are used to stop the move. Stop Enable values -1 through -7 represent I/Os 1-7 while Stop State values 0 and 1 which whether to stop on specified I/O going LOW or HIGH respectively. Values of 0 in both parameters indicates no stop condition. For advanced uses of these parameters, see Using Inputs to Stop Motions in the SilverMax User Manual.

See Basic Motion in User Manual for more details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Register Move Relative, Velocity Based (RRV)	Program Class D 161 (0xA1) 9 words	Data Register	S32	Standard Register Range
		Acceleration	U32	1 to 1,073,741,823
		Velocity	U32	0 to 2,147,483,647
		Stop Enable	S16/U16	See Above
		Stop State	S16/U16	See Above

Example:

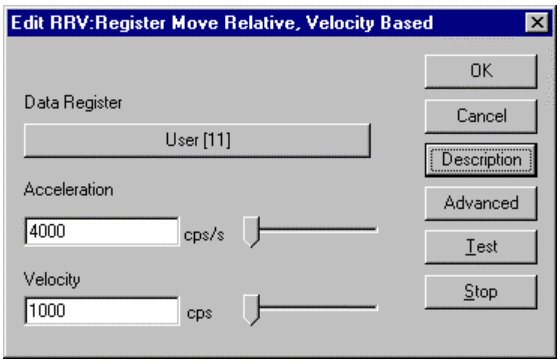
Move SilverMax to position indicated by User Data Register #11 at vel=1000cps and acc=4000cps/s.

@16 161 11 3865 8053064 0 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



PROGRAM FLOW COMMANDS

See Also: **Calculation Two Word (CTW)**

Calculation (CLC)

Description:

The Calculation command provides basic math, logic and other function using Data Registers. The command uses two parameters (combined into a single word), Operation and Data Register, to perform all of its defined operations. Several of the operations have two Operands to perform the calculation. When two Operands are required, Data Register #10 is used as one of the operands while the selected Data Register is used for the second operand. Typically, Data Register #10 Accumulator is used as the destination for a two-operand operation. For single Operand operations, the selected Data Register is used as the source and/or destination.

Data Register #10 is typically used as an accumulator but may also be the Selected Data Register.

The Multiply operations operate on the entire 32 bit word, but only return the 32LSB of the result.

The Divide command takes a 32 signed Dividend and a 16 bit positive divisor (up to 32767), and produces a signed 32 bit quotient. MOD takes the same parameters, but returns the standard (positive) modulo value.

When performing math functions the read only data registers can be used as the selected data register. Data cannot be saved or written to these registers due to their read only nature. User data registers can be used for any purpose as they are designed for both read and write operations. (See Data Register Commands in User Manual for details and definitions of Data Registers.)

Calculations affect the conditions of the Internal Status Word. Depending on the result of an operation one of three different conditions will occur (zero, positive, negative). (See Internal Status Word in the SilverMax User Manual for more details.)

NOTE: There are two related Calculation Commands, Calculation (CLC) and Calculation Two Word (CTW). CLC requires byte combination of the Operation and Register parameters, whereas CTW breaks these into separate parameters. CLC only uses 2 words in the Program Buffer while CTW uses 3 words. As CLC requires the combination of two bytes into a word, it may be too difficult to use in applications programmed without QuickControl (i.e. host programming).

Calculation Code Table

Firmware Rev.	Code	Operation	Description
E-Series	0	**Clear**	Clears out the indicated register
E-Series	1	**Add**	Adds the selected register to register 10 with the result remaining in register 10
E-Series	2	**Subtract**	Subtracts the selected register from register 10 with the result remaining in register 10
E-Series	3	**Copy**	Copies the selected register to register 10
E-Series	4	**Increment**	Increments the selected register
E-Series	5	**Decrement**	Decrements the selected register
E-Series	6	**Absolute Value**	Takes the absolute value of the selected register, saves it back into selected register
E-Series	7	**Subtract Target Position**	Subtracts the selected register from both the Target and Position registers. This allows for doing modulo calculations
E-Series	8	**Save**	Copies Register #10 to the selected Register.
E-Series	9	**Load High Word**	Loads the high word of the selected register (upper 16 bits), with sign extend, into register #10.
E-Series	10	**Load Low Word**	Loads the low word of the selected register (lower 16 bits), with sign extend, into register #10.
E-Series	11	**And**	Bitwise AND the selected register with register #10 with the result in placed in register #10
E-Series	12	**Or**	Bitwise OR the selected register with register #10 with the result in placed in register #10
E-Series	13	**Xor**	Bitwise Exclusive OR the selected register with register #10 with the result placed in register #10
E-Series	14	**Div**	Divide signed 32 bit long word of Register #10 by the positive valued of low word of selected Data Register. 32 bit result is placed Register #10 NOTE: Prior to Rev x33, divide was only 16/16 bit with a 32 bit result.
E-Series	15	**Umult**	Unsigned multiply of register #10 32 bit long word and 32 bit long word of selected register. 32 LSB of result is placed in Register #10. (User must keep terms appropriate such that the result fit in a 32 bit result field. NOTE: Prior to Rev x33, multiply was only 16x16 bit with a 32 bit result.
E-Series	16	**Mult**	Signed multiply of Register #10 32 bit long word of and the signed 32 bit long word of selected register. 32 bit LSB of result is placed in Register #10. User must select values that limit the signed product to fit in 32 bits. NOTE: Prior to Rev x33, multiply was only 16x16 bit with a 32 bit result.
E-Series	17	**Pack**	Replace the upper word of Register #10 with the low word of the selected register
27	18	**Load Indirect**	Loads Register #10 with the contents of the Register addressed by the data within the Selected Register (selected register is a pointer to the data location).
27	19	**Save Indirect**	Saves Register #10 contents to the Register addressed by the data within the Selected Register (selected register is a pointer to the data save location).
27	20	**Save High**	Saves the Low word of Register #10 to the High word of the selected register. Used to write to half of a combined word register.
27	21	**Save Low**	Saves the Low word of Register #10 to the Low word of the selected register. Used to write to half of a combined word register
27	22	**Shift Left**	Performs a 32 bit Left Shift of the selected Register.
27	23	**Shift Right Sign Extended**	Performs a 32 bit sign extended right shift of the selected Register. Implements a signed divide by 2.

27	24	**Shift Right**	Performs a 32 bit right shift of the selected Register. Implements an unsigned divide by 2.
33	25	**Modulo**	Performs a modulo (remainder) calculation using the signed 32 bits of Register 10, with the positive (only) divisor being the lower word of the selected Data Register. Note the remainder will always be positive, following the standard Modulo format.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Calculation (CLC)	Program Class D 165(A5) 2 words	Operation = Upper Byte Data Register = Lower Byte	U16	Operation: See Previous Table Data Register: Standard Register Range

Example:

Decrement Accumulator

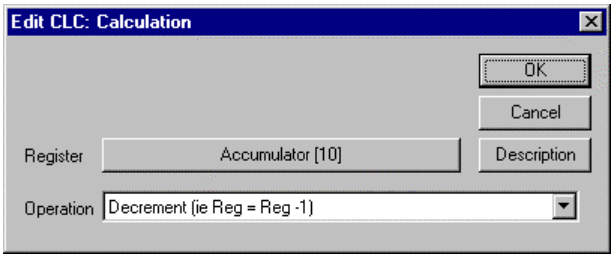
Operation/Register = 5 * 2^8 + 10 = 1290

@16 165 1290 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Calculation Two Word (CTW)

See also: **Calculation (CLC)**

Description

This is the same as the Calculation (CLC) command except the parameters are split up into two words instead of being combined. See the CLC command for details.

Command Info

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Calculation Two Word (CTW) Rev 34	Program Class D 215 (0xD7) 3 Words	Operation	U16	See Previous Table
		Data Register	U16	Standard Register Range

Example

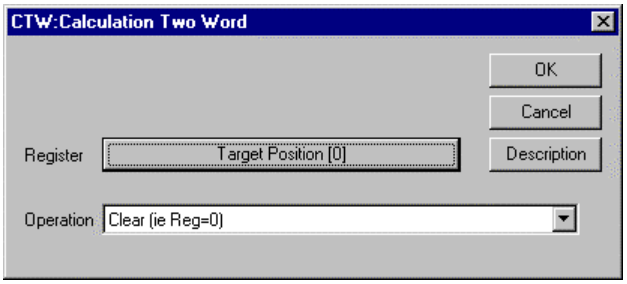
Clear Accumulator

@16 215 0 10 (CR)

SilverMax Response

ACK only

QuickControl Example:



Clear Program (CLP)

Description:

The Clear Program prepares the SilverMax unit for downloading into the Program Buffer. First, the Program Buffer is cleared. Then the buffer pointer is set to the beginning of the buffer. This command is used prior to a Start Download command. It sets up the buffer to properly receive a program.

This command may also be used to end the program download initiated by a Start Download (SDL) command.

See SilverMax Memory Model in User Manual for details on downloading programs

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Clear Program (CLP)	Immediate Class B 8 (0x8) 1 word	NONE	NONE	NONE

Example:

Clear the SilverMax Program Buffer.
@16 8 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

AKC only

Delay (DLY)

See also: **Delay In Ticks (DLT)**

Description:

This command is the same as the Delay In Ticks (DLT) command. It has the same command number and parameters. The only difference is how QuickControl scales them. To SilverMax they are exactly the same. QuickControl takes the parameter entered and multiplies it by 8.3333 to convert milliseconds to servo ticks (120uSec/tick).

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Same as Delay In Ticks (DLT)

Example:

Cause SilverMax to delay program execution by 1.2 seconds.

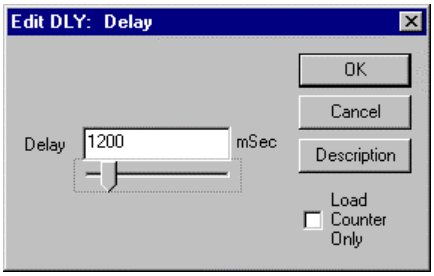
1200mSec * 8.3333 = 10000 ticks

@16 140 10000 (CR)

SilverMax Response:

ACK only

QuickControl Example:



See also: **Delay (DLY)**

Delay In Ticks(DLT)

Description:

The Delay In Ticks command sets a Delay Counter (Register 5) with the supplied parameter. The counter is decremented every servo cycle (120 microseconds). If the Tick Count is positive, the given value is used for the counter and a WAIT DELAY command is automatically executed. If the value is negative, the absolute value of the parameter is loaded into the counter and the execution continues on to the next command in the Program Buffer. The Delay Counter (Register 5) is Read/Write for code version REV321 and higher.

The Delay Counter, if not being used for program delays, may be used in a "watch dog" configuration. The counter would be kept refreshed by writing to Register 5 through the serial interface. The Delay Counter Bit in the Kill Motor Mask would be enabled. As long as the counter is not allowed to reach zero, the Delay Counter Bit in the Internal Status Word stays low. If the host were to shutdown or lose communications, the counter would reach zero, causing a Kill Motor operation, running a recovery routine if one were enabled.

A Tick Count equals 120 microseconds in time. To convert to seconds multiply the Tick Count by 0.00012. A one second delay (rounded off) is 8333 Tick Counts .

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

NOTE: DLT has the same command number as Delay (DLY). To SilverMax there is no difference between these two commands. Only QuickControl distinguishes between DLT and DLY with DLT have units of ticks and DLY having units of milliseconds.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Delay (DLT)	Program Class D 140 (0x8C) 3 words	Delay Count 1 tick=120 uSec.	S32	-2,147,483,647 to 2,147,483,647

Example:

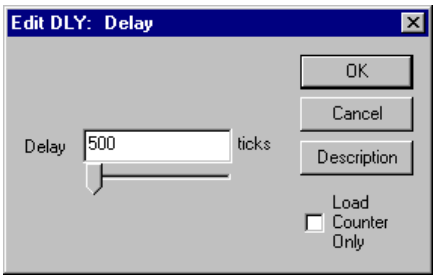
Cause SilverMax to delay program execution by 500 ticks.

@16 140 500 (CR)

SilverMax Response:

ACK only

QuickControl Example:



End Program (END)

Description:

SilverMax programs typically end when the last line of the program is completed. If the program needs to end based on a Conditional Jump, the End Program command can be inserted in the program at the desired point. When this command is executed, the currently running program will stop executing and the motor will be placed in a Host Mode. Programs can also be stopped by issuing any of the Override Commands.

NOTE: An END command is automatically inserted at the end of programs loaded in the Program Buffer.

This command does nothing if sent to the motor when it is in Host Mode.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
End Program (END)	Program Class D 128 (0x80) 1 word	NONE	NONE	NONE

Example:

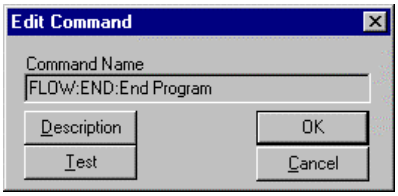
As part of a program, end program execution.

@16 128 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Jump (JMP)

Description:

The Jump command allows looping and other conditional branching inside a program based on the condition of the Internal Status Word (ISW) (see User Manual for bit definitions). The Condition Enable word selects which bits in the Internal Status Word will be evaluated. A “1” in the bit position enables looking at that bit. The Condition State word allows the user to specify the state (High “1” or Low “0”) of the selected bits that will cause the jump to occur. Setting the Condition Enable word to “zero” forces an unconditional jump to the specified Program Buffer location. A match on any of the selected bits caused the jump to occur.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Jump (JMP)	Program Class E 162 (0xA2) 4 words	Condition Enable	U16	0 to 32767
		Condition State	U16	0 to 32767
		Program Buffer Address	U16	0 to 199

Example:

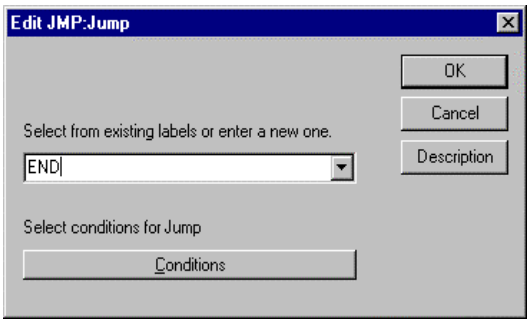
Jump to Program Buffer location 0 last calculation was zero. See Internal Status Word (ISW) in User Manual for bit definitions.

@16 162 2 2 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Jump On Input (JOI)

Description:

The Jump On Input command allows looping and other conditional branching inside a program based on the condition of an External Digital Input or an Internal Input. This command is actually the same number as the Jump command (JMP), however, by using a negative number for the first parameter the usage of the command changes (see Internal Status Word (ISW) Enable Code definition in User Manual for details). The Jump Enable Code selects which input will be evaluated. The Input State allows the user to specify the state (High "1" or Low "0") of the selected input that will cause the jump to occur. Setting both parameters to "zero" forces an unconditional jump to the specified Program Buffer location.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Jump On Input (JOI)	Program Class E 162 (0xA2) 4 words	Enable Code	S16	-1 to -14
		Input State	S16	0 = "Low" 1 = "High"
		Program Buffer Address	U16	0 to 199

Example:

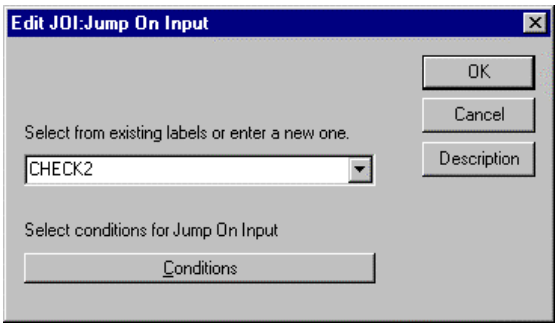
Jump to Program Buffer location 10 if digital input #5 is High "1".

@16 162 -5 1 10 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Jump On AND I/O State (JAN)

Description:

The Jump on AND I/O State command allows looping and other conditional branching inside a program based on the condition of the I/O State Word (IOS) (see User Manual for bit definitions). The IOS Condition Enable selects which inputs will be used in the AND-ed evaluation. The IOS Condition State allows the user to specify the states (High “1” or Low “0”) of the selected inputs that will cause a TRUE condition for each of the inputs. Setting both parameters to “zero” forces an unconditional jump to the specified Program Buffer location.

The condition TRUE evaluation is done by AND-ing the enabled inputs together. If all the enabled inputs are TRUE a jump will occur.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Jump On Inputs, And-Ed (JAN) Rev 22	Program Class E 250 (0xEE) 4 words	U16	IOS Condition Enable	0 to 65535
		U16	IOS Condition State	0 to 65535
		U16	Program Buffer Address	0 to 199

Example:

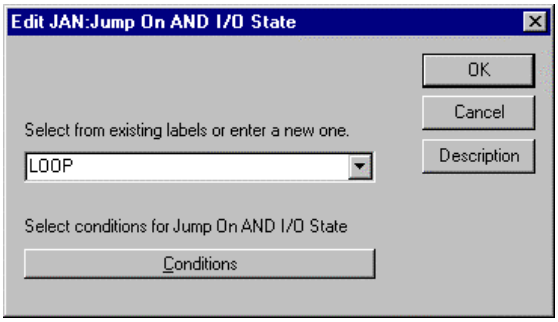
Jump to Program Buffer location 10 if digital inputs #4, #5, #6 and #7 are High (“1”). See I/O State Word (IOS) in User Manual for bit definitions.

@16 250 61440 61440 10 (CR)

SilverMax Response:

ACK only

Quick Control Example:



Jump On NAND I/O State (JNA)

Description:

The Jump On NAND I/O State command allows looping and other conditional branching inside a program based on the condition of the I/O State Word (IOS) (see User Manual for bit definitions). The IOS Condition Enable selects which inputs will be used in the NAND-ed (Negative AND-ed) evaluation. The IOS Condition State allows the user to specify the states (High "1" or Low "0") of the selected inputs that will cause a TRUE condition for each of the inputs. Setting both parameters to "zero" forces an unconditional jump to the specified Program Buffer location.

The condition TRUE evaluation is done by NAND-ing the enabled inputs together. If all the enabled inputs are TRUE a jump will NOT occur. This means that a jump will always occur when any of the conditions are FALSE.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Jump On Inputs, And-Ed (JNA)	Program Class E 238 (0xEE) 4 words	IOS Condition Enable	U16	0 to 65535
		IOS Condition State	U16	0 to 65535
		Program Buffer Address	U16	0 to 199

Example:

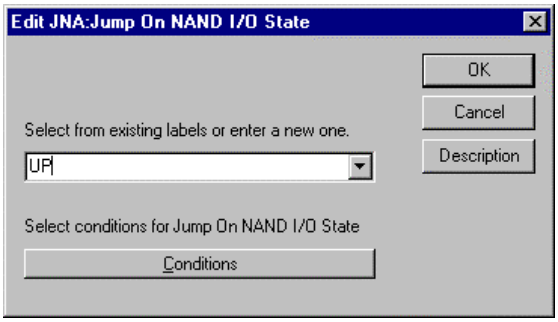
Don't jump to Program Buffer location 10 if digital inputs #4, #5, #6 and #7 are High ("1"). See I/O State Word (IOS) in User Manual for bit definitions.

@16 238 61440 61440 10 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Jump On OR I/O State (JOR)

Description:

The Jump On Inputs, OR-ed command allows looping and other conditional branching inside a program based on the condition of the I/O State Word (IOS) (see User Manual for bit definitions). The IOS Condition Enable selects which inputs will be used in the OR-ed evaluation. The IOS Condition State allows the user to specify the states (High “1” or Low “0”) of the selected inputs that will cause a TRUE condition for each of the inputs. Setting both parameters to “zero” forces an unconditional jump to the specified Program Buffer location.

The condition TRUE evaluation is done by OR-ing the enabled inputs together. If any of the enabled inputs are TRUE a jump will occur.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Jump On Inputs, Or-Ed (JOR)	Program Class E 239 (0xEF) 4 words	IOS Condition Enable	U16	0 to 65535
		IOS Condition State	U16	0 to 65535
		Program Buffer Address	U16	0 to 199

Example:

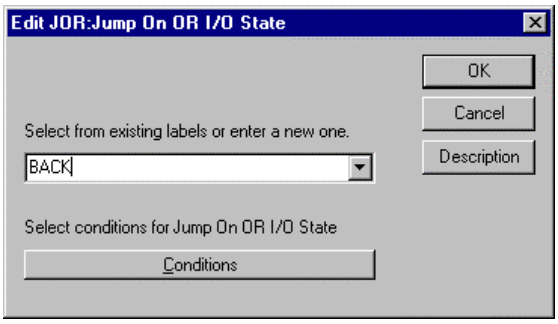
Jump to Program Buffer location 0 if input #1 or #2 or #3 is High (“1”). See I/O State Word (IOS) in User Manual for bit definitions.

@16 162 112 112 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Jump On Register Greater Or Equal (JGE)

Description:

The Jump On Register Greater or Equal command allows looping and other conditional branching inside a program based on the comparison of the contents of the given register with the value of the compare parameter. Note: Internally JRE, JGE, JNE, JLT all share the same Command number, with the difference indicated in the high byte of the Operation/Register parameter (see JRE command for details). For this command, the Operation/Register parameter is equal to the register number + 256.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Jump On Register Greater or Equal (JGE) Rev 29	Program Class E 137 (0x89) 5 words	Operation (High Byte) Register (Low Byte)	U16	Operation: =1 Register: Standard Register Range
		Value	S32	-2,147,483,648 to +2,147,483,647
		Program Buffer Address	U16	0 to 199

Example:

Jump to Program Buffer location 10 if Register # 32 is greater or equal to "1200"

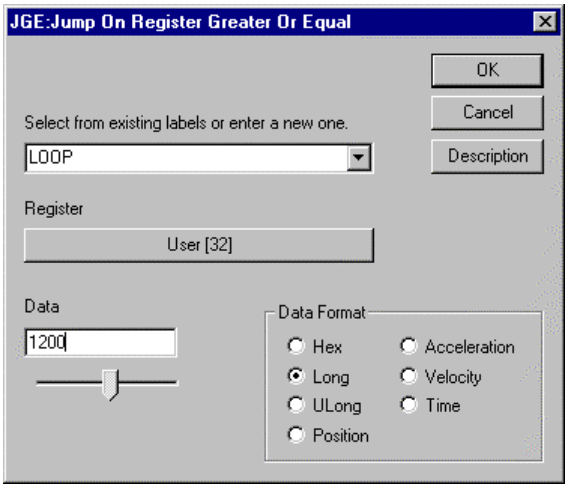
Operation/Register=256 + 32=288

@16 137 288 1200 10 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Jump On Register Less Than (JLT)

Description:

The Jump On Register Less Than command allows looping and other conditional branching inside a program based on the comparison of the contents of the given register with the value of the compare parameter. Note: Internally JRE, JGE, JNE, JLT all share the same Command Code, with the difference indicated in the high byte of the Operation/Register parameter (see JRE command for details). For this command, the Operation/Register parameter is equal to the register number + 512.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Jump On Register Less Than (JLT) Rev 29	Program Class E 137 (0x89) 5 words	Operation (High Byte) Register (Low Byte)	U16	Operation: =2 Register: Standard Register Range
		Value	S32	-2,147,483,648 to +2,147,483,647
		Program Buffer Address	U16	0 to 199

Example:

Jump to Program Buffer location 10 if Register # 32 is less than to “1200”.

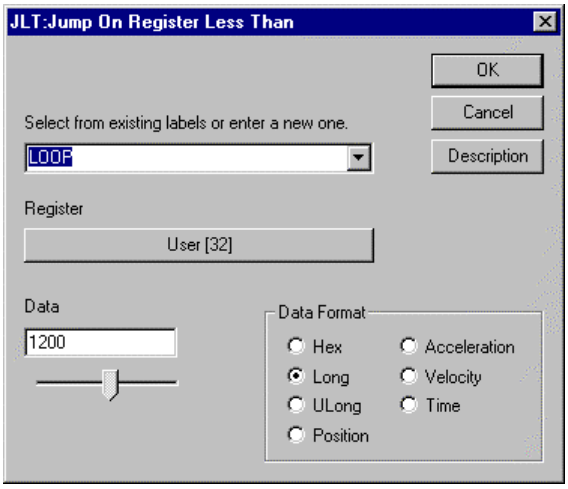
Operation/Register = 512 + 32 = 544.

@16 137 544 1200 10 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Jump On Register Not Equal (JNE)

Description:

The Jump On Register Not Equal command allows looping and other conditional branching inside a program based on the comparison of the contents of the given register with the value of the compare parameter. Note: Internally JRE, JGE, JNE, JLT all share the same Command Code, with the difference indicated in the high byte of the Operation/Register parameter (see JRE command for details). For this command, the Operation/Register parameter is equal to the register number + 768.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Jump On Register Not Equal (JNE) Rev 29	Program Class E 137 (0x89) 5 words	Operation (High Byte) Register (Low Byte)	U16	Operation: =3 Register: Standard Register Range
		Value	S32	-2,147,483,648 to +2,147,483,647
		Program Buffer Address	U16	0 to 199

Example:

Jump to Program Buffer location 10 if Register # 32 is not equal to "1200"

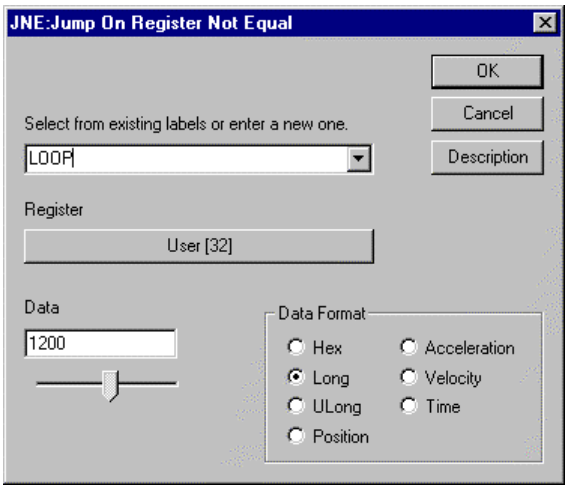
Operation/Register = 768+32= 800.

@16 137 800 1200 10 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Jump On Register Equal (JRE)

Description:

The Jump On Register Equal command allows looping and other conditional branching inside a program based on the comparison of the contents of the given register with the value of the compare parameter. Note: Internally JRE, JGE, JNE, JLT all share the same Command Code, with the difference indicated in the high byte of the Operation/Register Parameter.

The Operation is automatically handled for you by QuickControl. The following is provided for those not using QuickControl.

Operation	Equivalent Command	Operaion/Register Param Value
0	JRE	Register #
1	JGE	Register # + 256
2	JLT	Register # + 512
3	JNE	Register # + 768

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Jump On Register Equal (JRE) Rev 29	Program Class E 137 (0x89) 4 words	Operation/Register Operation (High Byte) Register (Low Byte)	U16	Operation: = 0 Register: Standard Register Range
		Value	S32	-2,147,483,648 to 2,147,483,647
		Program Buffer Address	U16	0 to 199

Example:

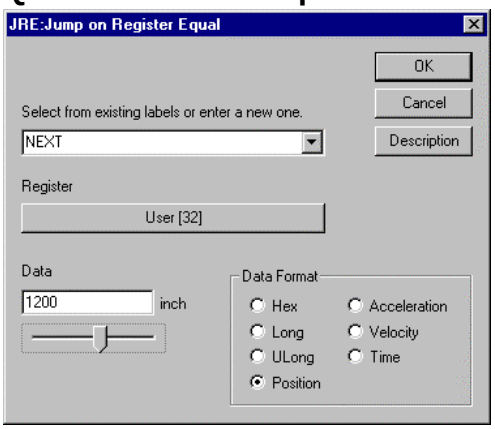
Jump to Program Buffer location 10 if Register # 32 is equal to "1200"

@16 137 32 1200 10 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Load Program (LPR)

Description:

The Load Program transfers a program from the nonvolatile memory to the Program Buffer. The number of words to be transferred is read from the location given in the NV Memory Address parameter. This count is automatically stored in the first word, along with a checksum, when the program is written into nonvolatile memory.

The content in the first NV Memory Address of the program is the length in words of the program size and the checksum of the program. The first command is read from the address following the Length & Checksum word, with subsequent words transferred up to the size indicated in the Length.

During the load process, the data is used to calculate a checksum value. When the load is complete, the calculated checksum is compared to the stored checksum. If the checksums do not agree Bit #14 in the Polling Status Word is set ("1") to indicate a program load failure.

This command only transfers the program into the Program Buffer; it does not cause execution to begin. Once loaded into the Program Buffer a Run Program command must be issued to begin program execution. The program will remain in the buffer until removed by the Clear Buffer command or over loaded by another Load Program command.

See SilverMax Memory Model in User Manual for details on downloading programs.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Load Program (LPR)	Immediate Class B 14 (0x0E) 3 words	NV Memory Address	U16	Valid NV Memory
		Word Count The Count is typically set to "0"	U16	**0** = use count stored at first address location. **1 to 199** = read the literal word count.

Example:

Load the program stored at NV Memory Address #110.

@16 14 110 0 (CR)

QuickControl Example:

Immediate (Host) Mode Command Only

SilverMax Response:

ACK only

Load And Run Program (LRP)

Description:

The Load and Run Program transfers a program from nonvolatile memory to the Program Buffer and executes it. This command combines the function of the Load Program and the Run Program together in one command. The Load and Run Program is designed to be placed inside of programs to enable branching off to other programs stored in NV memory.

The content in the first NV Memory Address of the program is the length in words of the program size and the checksum of the program. The first command is read from the address following the Length & Checksum word, with subsequent words transferred up to the size indicated in the length.

During the load process, the data is used to calculate a checksum value. When the load is complete, the calculated checksum is compared to the stored checksum. If the checksums do not agree, Bit #14 in the Polling Status Word is set ("1") to indicate a program load failure. (This may occur if data and/or programs overlap their usage in nonvolatile memory.)

After a load is complete and no errors were encountered, a Run Program will be initiated starting the program and dropping SilverMax into the Program Mode. Programs that contain errors will shut down the motor and exit execution when an error is encountered. Bit #12 (Program errors) of the Polling Status Word will be set indicating program execution error.

The program will remain in the buffer until removed by the Clear Buffer command or over loaded by another Load Program command.

See SilverMax Memory Model in User Manual for details on downloading programs.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Load And Run Program (LRP)	Program Class D 156 (0x9C) 2 words	NV Memory Address	S16	NV Memory Range

Example:

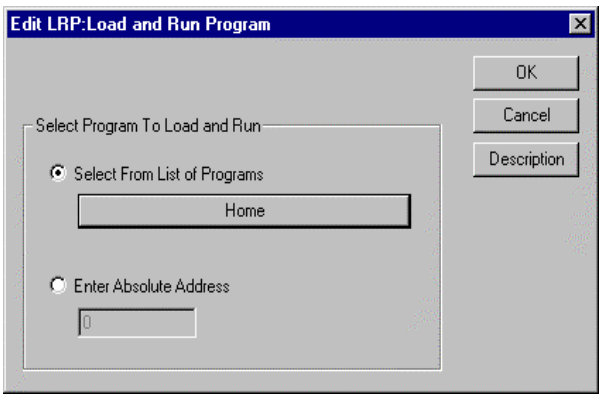
Load and Run the Program stored at NV Memory Address #150.

@16 156 150 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Program Call (PCL)

Description:

If the conditions are met, PCL jumps to the specified Program Buffer location (program label in QuickControl) and continues executing commands until a Program Return (i.e. PRT or PRI) command is encountered. A Program Return command causes the execution to continue at the command after the PCL.

Only one PCL can be executed at one time (no nested routines). If a second PCL is executed before a Program Return the program will error, Stop execution and Bit #12 in the Polling Status Word will be set. The PCL and Program Return must both be in the Program Buffer (same QuickControl program).

See Jump (JMP) command for details on the Condition State and Condition Enable parameters.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Program Call (PCL)	Program Class D Code (Hex): 201 (0xC9) 4 words	Condition State	U16	0 to 65535
		Condition Enable	U16	0 to 65535
		Program Buffer Location	U16	0 to 199

Example:

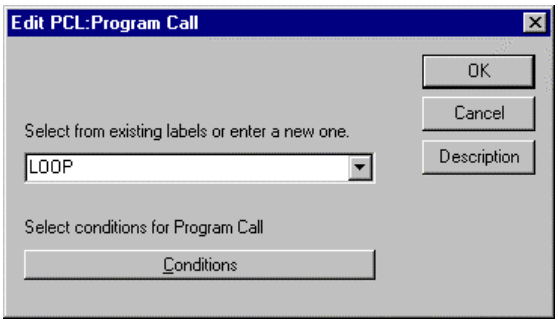
Call Program Buffer location #50 if digital input #1 is High “1”.

@16 201 50 16 16 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Program Call On Input (PCI)

Description:

The Program Call on Input command (PCI) works the same as Program Call (PCL) except the format of the call conditions. See Program Call (PCL) for details.

See Jump on Input (JOI) command for details on using the Enable Code and Enable State parameters.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Program Call On Input (PCI)	Program Class D 201 (0xC9) 4 words	Enable Code	S16	0 or -14
		Enable State	S16	0 to 1
		Program Buffer Address	U16	0 to 199

Example:

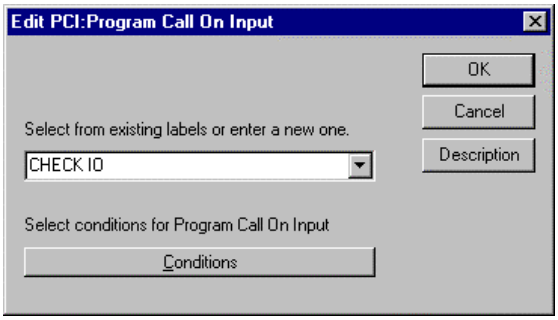
Call Program Buffer location #50 if digital input #2 is High “1”.

@16 201 50 -2 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Program Return (PRT)

Description:

The Program Return command is used as a complement to the Program Call command. Program execution continues at the command immediately following the Program Call. See Program Call (PCL) for details.

If a Program Return is executed without a previous Program Call, the program will error, stop execution and set Bit #12 in the Polling Status Word.

Program Return can be set up to conditionally execute using the Internal Status Word. This works identical to the Program Call and Jump commands.

Placing a “0” in both parameters will cause an unconditional return.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Program Return (PRT)	Program Class D 202 (0xCA) 3 words	Condition Enable	U16	0 to 65535
		Condition State	U16	0 to 65535

Example:

Return from Call if the last I/O #1 is High.

@16 202 16 16 (CR)

SilverMax Response:

ACK only

QuickControl Example:

Program Return On Input (PRI)

Description:

The Program Return command is used as a complement to the Program Call command. Program execution continues at the command immediately following the Program Call. See Program Call (PCL) for details.

If a Program Return on Input is executed without a previous program called, the program will error, Stop execution and set Bit #12 in the Polling Status Word.

Program Return on Input can be set up to conditionally execute using the Digital Inputs. This works identical to the Program Call on Input and Jump on Input commands.

Placing a “0” in both parameters will cause an unconditional return.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Program Return On Input (PRI)	Program Class D 202 (0xCA) 3 words	Enable Code	S16	0 to -14
		Enable State	S16	0 or 1

Example:

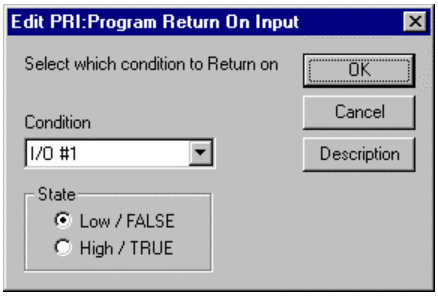
Return from Call if the Input #1 is Low (“0”).

@16 202 -7 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Run Program (RUN)

Description:

Executes the program that has been previously loaded into the Program Buffer. This command will clear the download mode, set the program pointer to “0” and start the program.

The Program Buffer can be filled using the Start Download command from the Host controller (see Start Download below). It can also be filled using the Load Program command that will move a program from the nonvolatile memory into the Program Buffer (see Load Program above).

Any Command or Program remaining in the Program Buffer can be executed over again using this command. A Stop command will stop the program but not clear the buffer. When in Host Mode, Program Mode commands sent to the SilverMax will remain in the buffer until another Program Mode command is sent or a Program is loaded. The Run Program command can be used to repeat the previous Program Mode command.

Sending this command while a Program or Command is executing will give a SilverMax NAK – Busy response.

NOTE: Sending a Program Mode command while in Host mode actually loads that command into the start of the command buffer with an END command inserted behind it and then that (short) program is run.

See SilverMax Memory Model in User Manual for details on downloading and running programs.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Run Program (RUN)	Immediate Class C 10 (0x0A) 1 word	NONE	NONE	NONE

Example:

Run the Program or Command that was previously loaded into the Program Buffer.

@16 10 (CR)

QuickControl Example:

Immediate (Host) Command Only

SilverMax Response:

ACK only

Start Download (SDL)

Description:

This command puts SilverMax into a program download mode**.** Program Mode commands that are sent after a Start Download command are automatically appended to the Program Buffer rather than being executed. Once in the Program Buffer, they can be executed as a program or stored to nonvolatile memory. The program download mode is terminated by a Store Program (SPR), a Run Program (RUN) or a Clear Program (CLP) command.

Immediate Mode commands sent to the SilverMax when in download mode are not appended to the buffer. Depending on the command, it will be immediately executed or it will cause an error.

See SilverMax Memory Model in User Manual for details on downloading programs.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Start Download (SDL)	Immediate Class B 9 (0x9) 1 word	NONE	NONE	NONE

Example:

Put SilverMax into the program download mode.

@16 9 (CR)

QuickControl Example:

Immediate (Host) Command Only

SilverMax Response:

ACK only

Store Program (SPR)

Description:

The Store Program command stores a program into Nonvolatile Memory onboard the SilverMax unit. The currently loaded program will be stored at the address number indicated in the address parameter of the command. A program must be downloaded in the Program Buffer before the Store Program is used. The program download mode is terminated by this command.

The length of the program (in words) and a Checksum are written to the indicated memory address, followed by the program. The length is used by the Load Program (LPR) or Load & Run Program (LRP) command to know the size of the program to load from nonvolatile memory. Because the length is written to the first address location, add 1 word to overall length for keeping track of memory usage. The Checksum is used by the Load Program or Load & Run Program command to determine the data integrity. This prevents corrupted or partially overlapping programs from attempting execution.

This command leaves a background routine running until the programming of the nonvolatile memory has completed. Once completed, Bit #15 in the Polling Status Word is set. Bit #14 is set if the command attempts to write beyond the allowed memory space. Execution time of this command varies depending on the number of words written.

See SilverMax Memory Model in User Manual for details on downloading programs.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Store Program (SPR)	Immediate Class C 13 (0x0D) 2 words	NV Memory Address	U16	Valid NV Memory Range

QuickControl Example:

Immediate (Host) Command Only

Example:

Store the currently loaded program into NV memory at address 1000.

@16 13 1000 (CR)

SilverMax Response:

ACK only

Wait Delay (WDL)

Description:

The Wait Delay command waits until the Delay Counter (Register 5) has decremented all the way to zero. Once it has reached zero, this command is exited and the next command in the Program Buffer is executed.

The Delay Counter is initialized using the Delay (DLY) command with a negative value parameter or by directly writing to register 5. This causes the counter to begin the count down to zero. When the count has expired the Wait Delay exits and allows the program to continue. (See Delay command above for more details.) The Delay counter may also be written with any of the register manipulation commands, either from the Serial Interface or from the program.

This command is useful when a timer needs to be set before a series of other commands are executed with a wait at the end. This allows a program or sub-routine to execute with precise timing.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Wait Delay (WDL)	Immediate Class D 141 (0x8D) 1 word	NONE	NONE	NONE

Example:

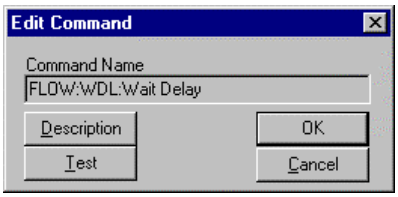
Cause program to wait until Delay Count is expired.

@16 141 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Wait On Bit Edge (WBE)

Description:

During program execution, the Wait on Bit Edge command causes SilverMax to wait until a condition is true. This is a very fast check that is done every servo cycle (120microseconds). Placing this command in a program will cause the program to wait on the current line until the input condition is met. There is no wait limit; therefore, this can put SilverMax into an endless wait state. The I/O bit condition is edge triggered. The input must transition from High to Low for the Falling and Low to High for the Rising condition to be true.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Enable Code	Input Source	Description
1	External	I/O #1
2	"	I/O #2
3	"	I/O #3
4	"	I/O #4
5	"	I/O #5
6	"	I/O #6
7	"	I/O #7
8	"	Current Index Sensor
9	"	Internal Index Sensor
10	"	External Index Sensor
11	Internal	Moving Error Status
12	"	Holding Error Status
13	"	Trajectory Generator
14	"	Delay Counter Active

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Wait on Bit Edge (WBE)	Program Class D 204 (0xCC) 3 words	Enable Code	S16	1 to 14
		Enable State	S16	0 = falling (High to Low) 1 = rising (Low to High)

Example:

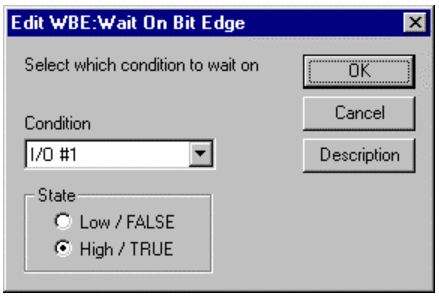
Cause program to wait until I/O #1 goes from Low to High.

@16 204 1 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Wait On Bit State (WBS)

Description:

During program execution, the Wait on Bit State command causes SilverMax to wait until a condition is true. This is a very fast check that done every servo cycle (120microseconds). Placing this command in a program will cause the program to wait on the current line until the input condition is met. There is no wait limit; therefore, this can put SilverMax into an endless wait state.

The I/O bit condition is state triggered, if the condition is true when the command is encountered no waiting will occur.

See Wait On Bit Edge (WBE) for definition of Enable Code.

See Program Flow Control Using Inputs and Data Registers in User Manual for general program flow control information and examples.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Wait On Bit State (WBS)	Program Class D 194 (0xC2) 3 words	Enable Code	S16	1 to 14
		Enable State	S16	0 = “Low” 1 = “High”

Example:

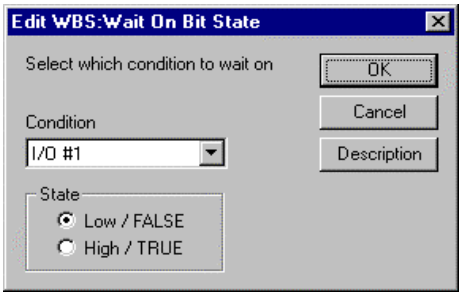
Cause program to wait until I/O #1 is Low.

@16 194 1 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



I/O COMMANDS

Analog Continuous Read (ACR)

Description:

The Analog Continuous Read does continuous read of a selected analog channel into a User Data Register. Readings are taken every servo cycle (120 usec.) and transferred into the selected Data Register. A number of different analog channels are available with this command.

Reading and filtering of all channels into dedicated registers occur continuously in the background (see Data Register Appendix in User Manual for register numbers). This command only copies the current filtered register to the given register.

The internal Analog to Digital Converter (ADC) is a 10-bit version, which yields approximately 0.005 volts per ADC count. SilverMax filters (5 millisecond) the inputs and scales them up to a 15-bit value (32 counts = 0.005 volts), but the resolution remains the same. Note that the maximum reading corresponds to 32*1023 = 32736.

Other types of data can be collected using the Analog Continuous Read. The Input Power Supply voltage and Internal Temperature can both be read into a Data Register. This information can be used for program conditional control or for direct read out to a Host controller.

Selecting Analog Channel “0” disables the Analog Continuous Read.
Only one ACR is active. As soon as another ACR is issued, any previous ACR is cancelled.
To read multiple analog inputs, the Analog Read Input (ARI) Command must be issued repeatedly.

See Using Analog Inputs in the SilverMax User Manual for more information.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Analog Continuous Read (ACR)	Program Class D Code (Hex): 207 (0xCF) 3 words	Analog Channel #	S16	0 = (Disable) 1 = Analog #1 2 = Analog #2 3 = Analog #3 4 = Analog #4 5 = Analog #1 and Analog #2 6 = Analog #3 and Analog #4 7 = V+ (non-calibrated) 8 = Temperature (ADC counts)
		Data Register	S16	Standard Register Range

Example:

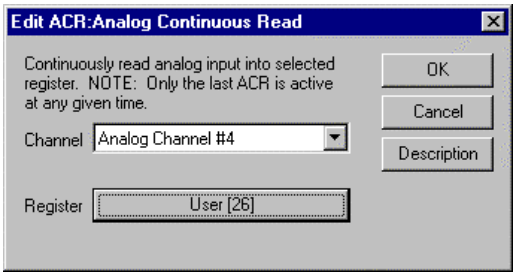
Configure Analog input #4 to do a continuous read to Data Register #26.

@16 207 4 26 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Analog Read Input (ARI)

Description:

The Analog Read Input does a single read of a selected Analog Channel into a User Data Register. A reading is taken only once and transferred into the selected Data Register.

Reading and filtering of all channels into dedicated registers occur continuously in the background (see Data Register Appendix in User Manual for register numbers). This command only copies the current filtered register to the given register.

See Analog Continuous Read (ACR) for details.

See Using Analog Inputs in the SilverMax User Manual for more information

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Analog Read Input (ARI)	Program Class D 193 (0xC1) 3 words	Analog Channel #	S16	1 = Analog #1 2 = Analog #2 3 = Analog #3 4 = Analog #4 5 = Analog #1 and Analog #2 6 = Analog #3 and Analog #4 7 = V+ (non-calibrated) 8 = Temperature (ADC counts) 9 = V+ Scale Factor 10 = Processor V+ 11 = Driver Temperature 12 = Processor V+ Scale Factor
		Data Register	S16	Standard Register Range

Example:

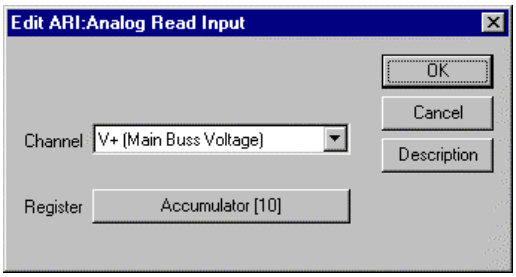
Read into data register #10 the V+ input voltage.

@16 193 7 10 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Configure I/O (CIO)

Description:

Configures the selected digital I/O bit for input or output. When setting as an output the logic level state is also set. Each I/O bit is individually set using this command, the power-up default is all I/O bits are inputs. This prevents I/O conflicts.

Care must be taken when configuring I/O. If the I/O inputs #4, #5, #6 or #7 are being used as analog inputs (#1, #2, #3 or #4), setting these I/O bits to output mode will override analog input.

See Using SilverMax I/O in the SilverMax User Manual for more information on I/O usage and conflicts.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Configure I/O (CIO)	Program Class D 188 (0xBC) 3 words	I/O Line #	S16	1 to 7
		Mode	S16	-1 = Input 0 = Clear (Low) 1 = Set (High)

Example:

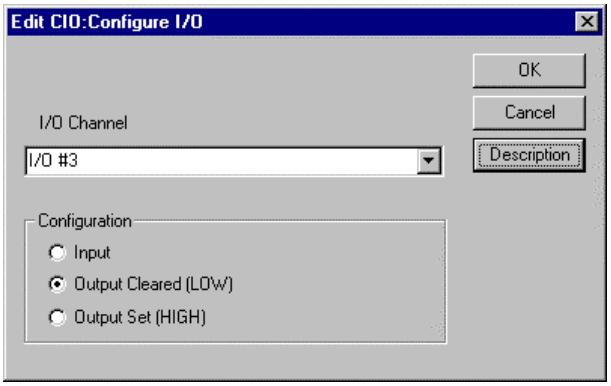
Set I/O bit #3 as output “Low”.

@16 188 3 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Clear Output Bit (COB)

Description:

Clears the selected Digital I/O bit to a logic Low (“0”) condition (Output = 0 volts). If the I/O was configured as an input this will reconfigure the bit as an output and clear it to logic Low (“0”).

See Using SilverMax I/O in the SilverMax User Manual for more information on I/O usage and conflicts.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Clear Output Bit (COB)	Program Class D 206 (0xCE) 2 words	I/O Line #	S16	1 to 7

Example:

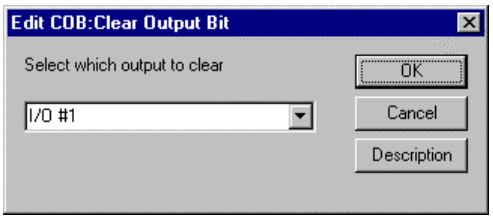
Clear I/O bit #1 to a low (“0”) state.

@16 206 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Disable Encoder Monitor (DEM)

Description:

Turns off the Enable Encode Monitor mode. If the Enable Encode Monitor mode was set this command will take it out of the monitor mode and return the Digital I/O to normal operation. (See Enable Encode Monitor below for more details.)

Command Info:

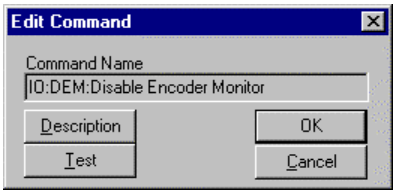
Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Disable Encoder Monitor (DEM)	Program Class D 171 (0xAB) 1 word	NONE	NONE	NONE

Example:

Turn off monitoring of the Internal Encoder.

@16 171 (CR)

QuickControl Example:



SilverMax Response:

ACK only

Enable Encoder Monitor (EEM)

Description:

The Enable Encoder Monitor command is used to output the SilverMax Internal Encoder signals to the Digital I/O. It causes a buffered copy of the raw encoder signals to be output to three digital lines for external viewing. The Encoder A signal is output to I/O line #1, the Encoder B signal to I/O bit line #2 and the Encoder Index signal is output to I/O line #3.

These signals have the same output specifications as the generic digital outputs. I/O lines #1, #2, and #3 are not available in Bit Output mode (either set or clear) while the encoder outputs are enabled. Similarly, the Encoder outputs may not be enables while any of the three I/O lines are in output mode. Either of these conflicts will cause a Sequence error and will terminate the program. See Using SilverMax I/O in the SilverMax User Manual for more information on I/O usage and conflicts.

To exit this mode, use the Disable Encoder Monitor command.

For using the encoder output for controlling or sending signals to other external devices see the modulo commands below. These commands are designed to be more flexible in outputting encoder signals.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Enable Encoder Monitor (EEM)	Program Class D 170 (0xAA) 1 word	NONE	NONE	NONE

Example:

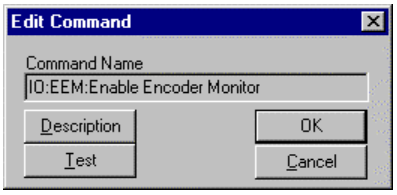
Turn on monitoring of the Internal Encoder.

@16 170 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Modulo Clear (MDC)

Description:

The Modulo Clear takes SilverMax out of modulo output mode and frees up Digital I/O bits #6 & #7 for normal usage.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Modulo Clear (MDC)	Program Class D 190 (0xBE) 1 word	NONE	NONE	NONE

Example:

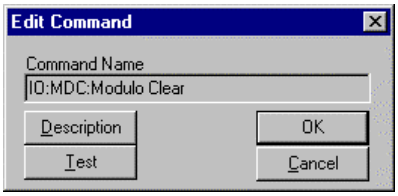
Turn off modulo output.

@16 190 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Modulo Set (MDS)

Description:

The Modulo Set command is used to enable, select the source, divisor, and format of the modulo outputs. These outputs make use Digital I/O bits #6 & #7 (See I/O Configuration Chart), and require that these bits to be configured as Inputs before issuing this command.

Either the internal or an external encoder may be selected as the source for modulo output. This allows the modulo counter to be used with external sources.

The output can be configured in three different ways: 1) A/B quadrature, which is the normal output mode for an encoder, 2) Step up & Step Down, which gives a square wave output on I/O bit #6 when the encoder is counting positive or on I/O bit #7 when the encoder is counting negative, and 3) Step and Direction, which gives a square wave output on I/O bit #6 and a Direction output on I/O bit #7.

The Count sets up a divider that is the Modulo Count. In Step output modes, the square wave rate equals the encoder rate divided by the count.

To exit this mode, use the Modulo Clear (MDC) command. See Using SilverMax I/O in the SilverMax User Manual for more information on modulo usage.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Modulo Set (MDS)	Program Class D 189 (0xBD) 4 words	Count	S16	1 to 32 (External) (1 to 8 on 34HC)
		Encoder Source	S16	0 = Internal 1 = External
		Output Format	U16	0 = A/B Quad 1 = Step Up/Dn 2 = Step & Dir

Example:

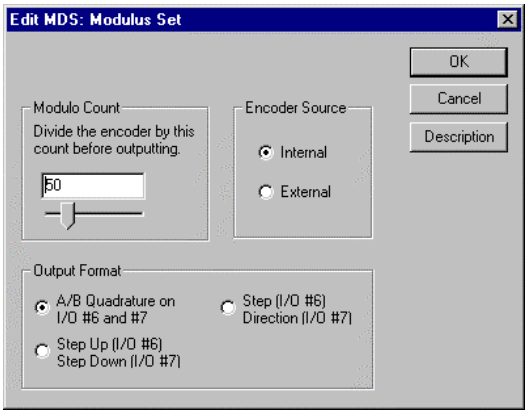
Divide internal encoder counts by 50 and output in A/B Quadrate format.

@16 189 50 0 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Modulo Trigger (MDT)

Description:

The Modulo Trigger allows digital I/O #1 to act as a gating or triggering signal.

Trigger mode #0 disables modulo output until I/O #1 goes from logic Low (“0”) to High (“1”). Mode #1 enables modulo output for continuous operation. Mode #2 will gate the modulo output whenever I/O #1 is high (“1”).

Mode #0 is edge triggered and can be used to as a one shot trigger. Mode #2 is used as a one shot reset for mode #0. Mode #0 is set up by first going into mode #2. If I/O #1 is low, this will disable modulo output, which resets the one shot. Mode #0 can now be set and will wait until I/O #1 goes from Low to High before enabling modulo output.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Modulo Trigger (MDT)	Program Class D 191 (0xBF) 2 words	Trigger Mode	S16	**0** = Disable until I/O #1 is High **1** = Enable **2** = Gate modulo using I/O #1

Example:

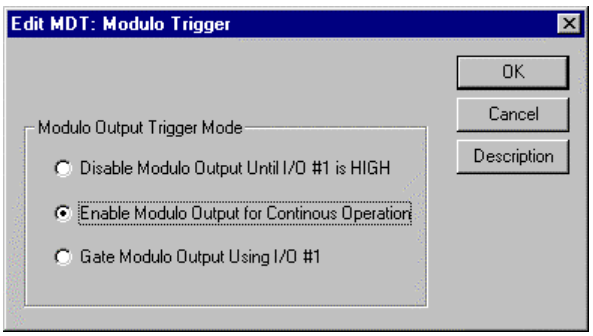
Enable continuous modulo output.

@16 191 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Position Compare (PCP)

Description:

Causes I/O #1 to toggle its state when the motor position is equal to or greater than (absolute value) the “Position” value contained in the first of two User Data Registers. The I/O #1 state will toggle again when the motor position is greater than the “Modulo”(second Data Register) of the “Position” value. On a compare, if I/O #1 is a logic Low (“0”) it will be set to a High (“1”). If I/O #1 is a logic High (“1”) it will be set to a Low (“0”). I/O #1 must be set to “Output” mode using the Set Output Bit, Clear Output Bit or the Configure I/O commands.

The First Data Register = “Position”
The Second Data Register = “Modulo”

The first Data Register contains the “Position” value that starts the compare process and executes the first toggle. The second Data Register contains the “Modulo” value used for creating a continuous toggle of the output. After the first position compare, the “Modulo” value is added to the “Position” value to set up for the next compare. This will continue until the Position Compare is disabled (Setting values to “0”) or the motion completes. If the “Modulo” value is “0” only a single compare takes place.

Position Compare can be done in both positive and negative position moves. The “Modulo” is added as an absolute value to the “Position”.

This command is accomplished using a software compare (updated every 120 usec.) and therefore may have a small delay of 120 microseconds from a compare to the actual I/O #1 change of state.

Command Info:

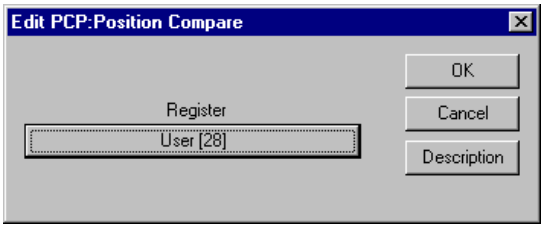
Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Position Compare (PCP)	Program Class D 245 (0xBF) 2 words	Data Register	U16	0 = Disable Usage Standard Register Range

Example:

Enable Position Compare using Data Register #28 for the Position compare value and #29 for the Modulo value.

@16 245 28 (CR)

QuickControl Example:



SilverMax Response:

ACK only

See Also: **COB**

Set Output Bit (SOB)

Description:

Sets the selected Digital I/O bit to a logic High (“1”) condition (Output = 5 volts). If the I/O was configured as an input this will reconfigure the bit as an output and set it to logic High (“1”).

See Using SilverMax I/O in the SilverMax User Manual for more information on I/O usage and conflicts.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Set Output Bit (SOB)	Program Class “D 205 (0xCD) 2 words	I/O Line #	S16	1 to 7

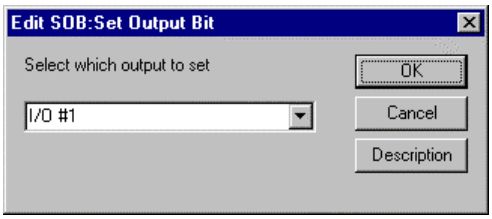
Example:

Set I/O bit #1 to a logic High (“1”) state.
@16 205 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



DATA REGISTER COMMANDS

A number of 32-bit Data Registers have been allocated for special purposes. Data Registers are used as data storage locations that may be used and modified by a Host controller or by SilverMax internal functions. They provide data storage for the distance and position parameters for Register motion profile commands. They can also be used by the Calculation command as data variables for more complex calculations, such as shortest paths for rotary motions. The Input Modes use data registers for Offset and Scaling factors.

See User Manual appendix for Data Register definitions.

Add To Register (ATR)

Description:

The Add to Register command adds the included data into the selected 32 bit Data Register. This command is similar to the Write Register commands except it is designed to add to the existing value instead of overwriting it. The data parameter is “signed” so that a negative value can be added, which works as subtraction for decrementing.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Add To Register (ATR)	Program Class D 248 (0x9A) 4 words	Data Register	U16	Standard Register Range
		Data	S32	-2,147,483,648 to +2,147,483,647

Example:

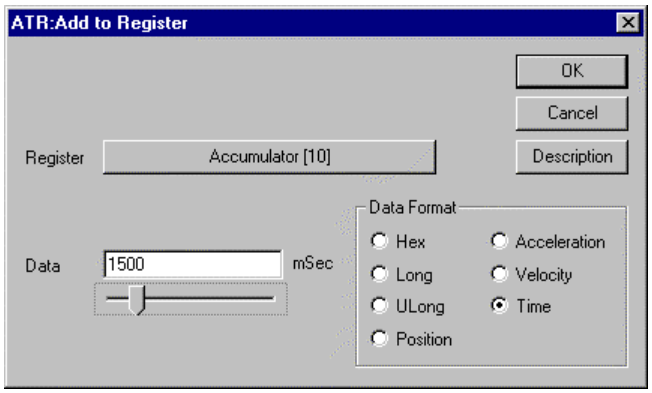
Add the number “1.5 SEC.” to data register #10.

@16 248 10 12500 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Register Load Multiple (RLM)

Description:

Loads an array of data from the selected nonvolatile memory address to an array of data registers. A checksum value is verified to insure good data.

During the load process, the data is used to calculate a checksum value. When the load is complete, the calculated checksum is compared to the stored checksum. If the checksums do not agree bits #14 & #12 in the Polling Status Word are set (“1”) to indicate a register load failure.

See Nonvolatile Memory in User Manual for details on loading and soring data.

INDIRECT ADDRESSING: The Nonvolatile Memory may be indirectly addressed by putting the wanted address into Register # 10, and then using a NV Memory Address of zero. The zero address triggers the indirect addressing mode, as location “0” is reserved as the start of the initialization program. This may be used to recall a long series of numbers from NV memory via a loop operation.

NOTE: If this command is used in QuickControl's "Normal Mode", many of the complexities go away. See Register File System in SilverMax User Manual for details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Register Load Multiple (RLM)	Program Class D 197 (0xC5) 4 words	Number of Registers	S16	1 to 10
		Starting Data Register	S16	Standard Data Range > 10 0 for indirect addressing
		NV Memory Address	U16	Max Size at NV memory

Example:

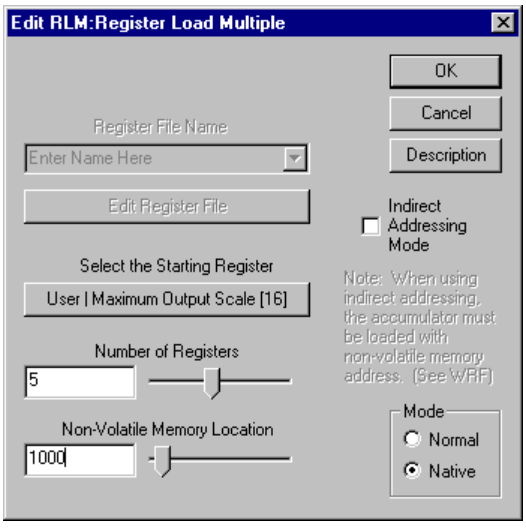
Sequentially Load 5 data registers starting at #16 with the data from NV memory address 1000

@16 197 5 16 1000 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Register Load Nonvolatile (RLN)

Description:

Loads data from the selected Nonvolatile Memory address into the selected Data Register. A Checksum value is verified to insure good data.

See Nonvolatile Memory in User Manual for details on loading and storing data.

The loading process is the same as used by the Register Load Multiple with only one register being loaded. The data selected must be stored using the Register Load Nonvolatile or the Register Load Multiple using “1” for the Number of Registers.

INDIRECT ADDRESSING: The Nonvolatile Memory may be indirectly addressed by putting the wanted address into Register # 10, and then using a NV Memory Address of zero. The zero address triggers the indirect addressing mode, as location “0” is reserved as the start of the initialization program. This may be used to recall a long series of numbers from NV memory via a loop operation.

NOTE: If this command is used in QuickControl's "Normal Mode", many of the complexities go away. See Register File System in SilverMax User Manual for details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Register Load Nonvolatile (RLN)	Program Class D 199 (0xC7) 3 words	Data Register	S16	Standard Reg Range > 10 0 for indirect addressing
		NV Memory Address	U16	Max Size of NV Memory

Example:

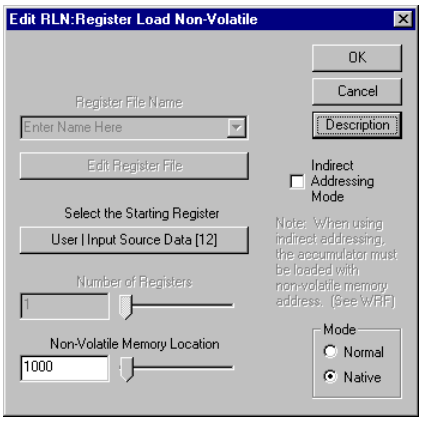
Load data register #12 with data from NV memory address 1000.

@16 199 12 1000 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Read Register (RRG)

Description:

The Read Register command reads back data from a selected 32-Bit Data Register using the Serial Interface. Since it is an Immediate Mode, this command can be used at any time, even during program execution. Any Data Register can be read back using this command including registers #0 through #40 & #200 through #232.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Read Register (RRG)	Immediate Class A 12 (0x0C) 2 words	Data Register	S16	Standard Register Range

Example:

Read back the motor's current position.

@16 12 1 (CR)

QuickControl Example:

Immediate (Host) Command Only

SilverMax Response:

Data Register data

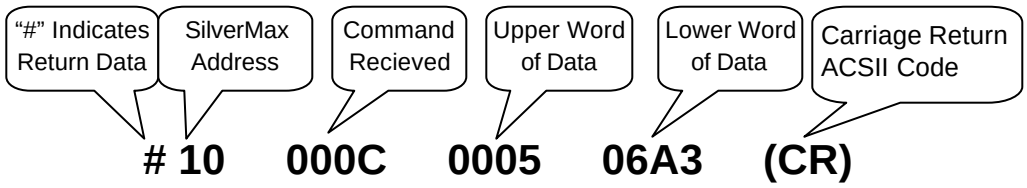
Response Example:

Example: Read Register command that requests the "Current Position" from SilverMax #16 (which is "10" in Hexadecimal); the last 8 digits represent the 32-bits of position data.

The current position = 329,379 in decimal

10 000C 0005 06A3 (CR)

The return data breaks down as follows:



Register Store Multiple (RSM)

Description:

Stores data from an array of Data Registers to the selected NV Memory address. A Checksum value is calculated from the array and stored with the array. Data from the selected Data Registers is stored sequentially to NV Memory. Data is also copied from the Data Registers sequentially.

The Length of the array and a Checksum are combined and written to the first NV Memory Address, followed by a “0” then the array of data. The length is used by the Register Load Multiple command to know the size of the array to load from Nonvolatile Memory. The Checksum is used by the Register Load Multiple command to determine the data integrity. This prevents the loading of bad data that could cause erratic operation. The “0” prevents an array of data from being executed as a program.

See Non-Volatile Memory in User Manual for details on storing data.

INDIRECT ADDRESSING: The Nonvolatile Memory may be indirectly addressed by putting the wanted address into Register # 10, and then using a NV Memory Address of zero. The zero address triggers the indirect addressing mode, as location “0” is reserved as the start of the initialization program. This may be used to recall a long series of numbers from NV memory via a loop operation.

NOTE: If a rapidly changing data register is stored to NV Memory, the data has the possibility of being inaccurate. This is due to the write cycle of the SilverMax DSP. The cycle performs two 16 bit writes from the 32 bit data register to NV Memory. If the “data” in the register changes before the second 16 bit write cycle, then it will be incorrect. A NV Checksum error will also occur when the data is read back via the RLM command. It is advisable to copy the data from the changing register to a user register and then storing the user register to NV Memory.

NOTE: If this command is used in QuickControl's "Normal Mode", many of the complexities go away. See Register File System in SilverMax User Manual for details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Register Store Multiple (RSM)	Program Class D 196 (0xC4) 4 words	Number of	S16	1 to 10
		Starting Data Register	S16	Standard Register Range 0 for indirect addressing
		NV Memory Address	U16	Max Size at NV Memory

Example:

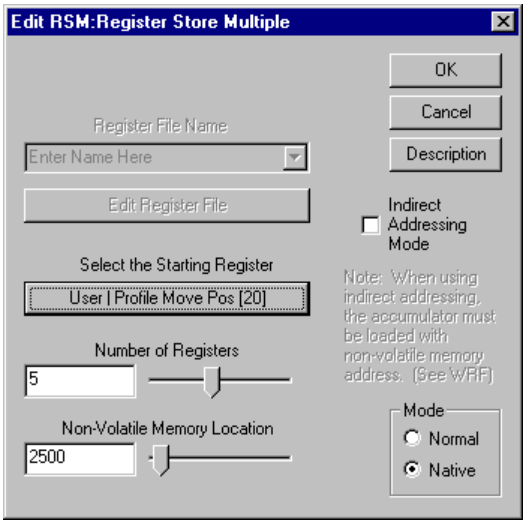
Store 5 data registers starting at #20 to NV memory starting at address 2500

@16 196 5 20 2500 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Register Store Nonvolatile (RSN)

Description:

Stores data from a Data Register to the selected Nonvolatile Memory address. A Checksum value is calculated from the data and stored with the data.

The storing process is the same as used by the Register Store Multiple with only one register being stored. The data selected may be loaded using the Register Load Nonvolatile or the Register Load Multiple using “1” for the Number of Registers.

See command Register Store Multiple (RSM) for more details.

NOTE: If this command is used in QuickControl's "Normal Mode", many of the complexities go away. See Register File System in SilverMax User Manual for details.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Register Store Nonvolatile (RSN)	Program Class D 198 (0xC6) 3 words	Data Register	S16	Standard Register Range 0 for indirect addressing
		NV Memory Address	U16	Max Size at NV Memory

Example:

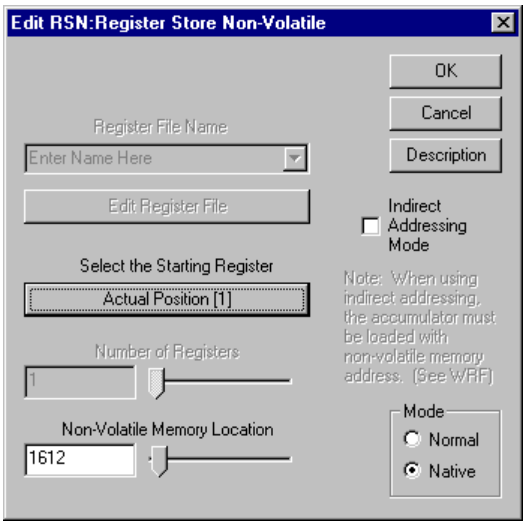
Store data register #1 to NV memory address 1612.

@16 198 1 1612 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Write Cmd Long Word (WCL)

Description:

This command allows program space starting at the selected program buffer location to be overwritten with the 32 bit data in the selected register. This allows for self modification of the command parameters within the program buffer. Any of the command parameters can be dynamically modified within the program. This command specifically is intended to modify 32 bit parameters. Extreme care should be used when writing any self modifying code to prevent unwanted outcomes. The QuickControl tool has support for this command, which simplifies its application, and enforces consistency checks. However, values being transferred are dynamic, based on the contents of the selected register; the range of the data is *not* verified at transfer, so undesired results may be obtained if out of range parameters are assembled into the program buffer, including Sequence Error shutdowns. This command does allow for great flexibility by allowing any of the parameters to be made register based.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Write Cmd Long Word (WCL) Rev 29	Program Class D 138 (0X8A) 3 words	Data Register	S16	Standard Register Range
		Program Buffer Address	S16	Valid NV Memory Range

Example:

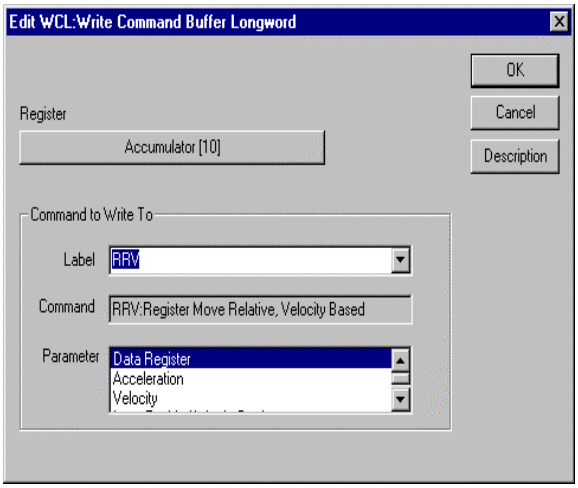
Overwrites the Command Buffer contents at locations 122 and 123 with the contents of Register #10.

@16 138 10 122 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Write Cmd Word (WCW)

Description:

This command allows program space starting at the selected program buffer location to be overwritten with the lower word 16 bit data in the selected register. This allows for self modification of the command parameters within the program buffer. Any of the command parameters can be dynamically modified within the program. This command specifically is intended to modify 16 bit parameters. Extreme care should be used when writing any self modifying code to prevent unwanted outcomes. The QuickControl tool has support for this command, which simplifies its application, and enforces consistency checks. However, values being transferred are dynamic, based on the contents of the selected register; the range of the data is *not* verified at transfer, so undesired results may be obtained if out of range parameters are assembled into the program buffer, including Sequence Error shutdowns. This command does allow for great flexibility by allowing any of the parameters to be made register based.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Write Cmd Long Word (WCL) Rev 29	Program Class D 139 (0X8B) 3 words	Data Register	S16	Standard Register Range
		Program Buffer Location	S16	Valid NV Memory Range

Example:

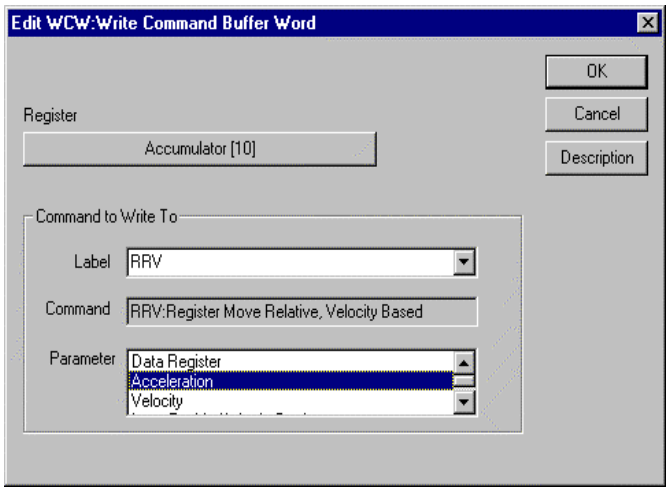
Overwrites the program buffer contents at locations 122 with the contents of the lower word of Register #10.

@16 138 10 122 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Write Register File (WRF)

Description:

WRF has the same command number as Write Register, Program Mode (WRP). WRF allows QuickControl to provide properties of Register Files and Register File Arrays. For details on these data structures, refer to Register File System in SilverMax User Manual.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Write Register File (WRF)	Program Class D 154 (0x9A) 4 words	Data Register	U16	Standard Register Range
		Data	S32/U32	0 to 4,294,967,295 or -2,147,483,648 to +2,147,483,647

Example:

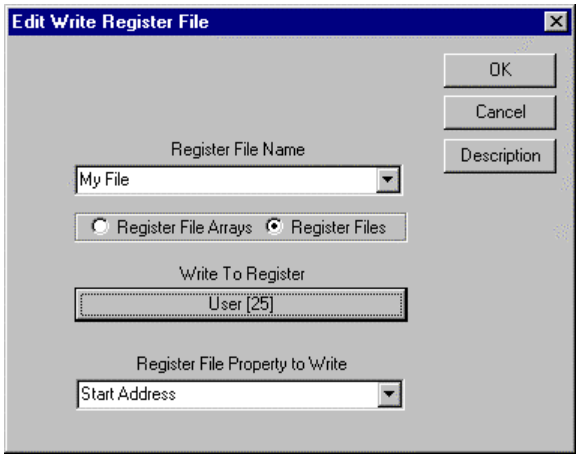
Write the number “1” to data register #10.

@16 154 10 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Write Register, Immediate Mode (WRI)

Description:

This command writes the given data into the selected 32 bit Data Register. Using the Serial Interface this command can be used at any time, even during program execution.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Write Register, Immediate Mode (WRI)	Immediate Class A 11 (0x0B) 4 words	Data Register	U16	Standard Register Range
		Data	S32/U32	0 to 4,294,967,295 or -2,147,483,648 to +2,147,483,647

Example:

Write the number “8000” to data register #12.

@16 11 12 8000 (CR)

QuickControl Example:

Immediate (Host) Command Only

SilverMax Response:

ACK only

Write Register, Program Mode (WRP)

Description:

The Write Register command writes the included data into the selected 32-bit Data Register. This command is similar to Write Register, Immediate Mode except it is designed to be embedded in a program and cannot be use through the serial interface while a command or program is being executed.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Write Register, Program Mode (WRP)	Program Class D 154 (0x9A) 4 words	Data Register	U16	Standard Register Range
		Data	S32/U32	0 to 4,294,967,295 or -2,147,483,648 to +2,147,483,647

Example:

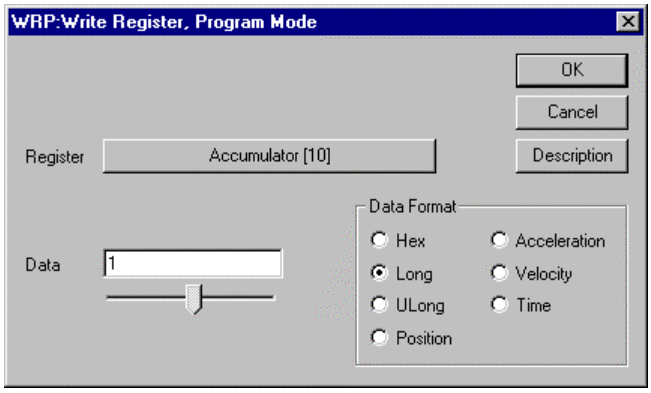
Write the number “1” to data register #10.

@16 154 10 1 (CR)

SilverMax Response:

ACK only

QuickControl Example:



MISC. COMMANDS

Check Internal Status (CKS)

Description:

This command checks the conditions of the Internal Status Word in the same manner as does the Jump command. If the condition enabled is true, bit #6 of the Polling Status is set to “1”. A zero in the Condition Enable parameter unconditionally sets bit #6 of the Polling Status Word.

This command may be used to convey information from a program executing back to the host processor that is polling the SilverMax.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Check Internal Status (CKS)	Program Class D 164 (0xA4) 3 words	Condition Enable	U16	0 to 65535
		Condition State	U16	0 to 65535

Example:

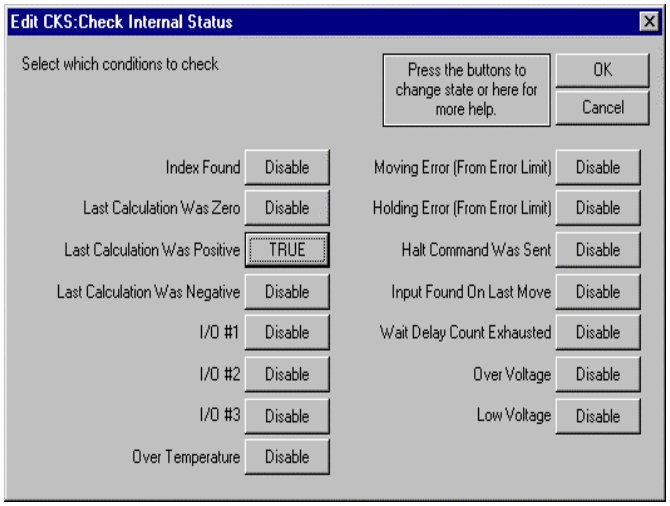
Check for a Last Calculation Was Positive and report to Host using Polling Status Word.

@16 164 4 4 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Clear Internal Status (CIS)

Description:

The Internal Status Word (ISW) is used to indicate different conditions or states in SilverMax (see Internal Status Word (ISW) in User Manual for details). Several of the conditions are "latched" and therefore are persistent even after the condition has changed. The CIS command is used to clear the latched conditions in the ISW.

This command should be used after a Kill Motor condition has occurred before normal operation can be restored.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Clear Internal Status (CIS)	Program Class D 163 (0xA3) 1 word	NONE	NONE	NONE

Example:

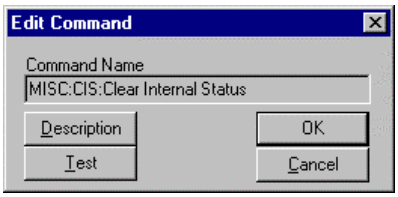
Clear the Internal Status Word.

@16 163 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Clear Max Error (CME)

Description:

The Maximum Error (absolute value of the Position Error) is updated and latched each servo cycle. The value is limited to a single word, saturating at 32767 (0x7FFF) as a maximum value. This command allows the Maximum Error value to be reset to zero so that the Maximum Error for a new motion profile may be determined.

The Maximum Error value is stored in a Dedicated Data Register and may be read using the Read Register command.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Clear Max Error (CME)	Program Class D 147 (0x93) 1 word	NONE	NONE	NONE

Example:

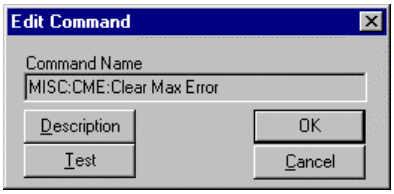
Clear the Maximum Error value.

@16 147 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Target To Position (TTP)

Description:

This command copies the current Position value into the Target register. This is useful for removing errors when an obstruction is encountered without losing track of position. This allows the next motion to move and ramp as expected rather than having to unwind the accumulated error. This is useful for homing against a hard stop where error is intentionally introduced, and for removing error before enabling the motor drivers after they have been disabled.

The Target value is updated by the Trajectory Generator, the Step & Direction mode or one of the Input Modes. The SilverMax servo loop uses the Target value as the input position parameter. If the motor is unable to achieve the Target position windup will occur. This command removes the windup error.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Set Target To Position (TTP)	Program Class D 146 (0x92) 1 word	NONE	NONE	NONE

Example:

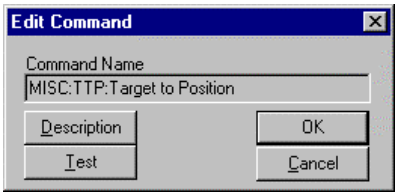
Sets the Target to the current position.

@16 146 (CR)

SilverMax Response:

ACK only

QuickControl Example:



Zero Target (ZTG)

Description:

This command zeros the Target register. Before doing this, the current Position Error (Target – Position) is copied into the Position Register. This is useful for homing routines to denote the current location as “Zero” so that all other locations can be defined as an offset from “Zero”.

This command does not remove any windup, whatever Position Error exists before this command will remain. To zero the Target and clear the Position Error use the Zero Target & Position command.

Command Info:

Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Zero Target (ZTG)	Program Class D 144 (0x90) 1 word	NONE	NONE	NONE

Example:

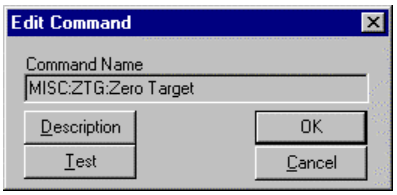
Sets the Target to zero (“0”) does not clear the position error.

@16 144 (CR

SilverMax Response:

ACK only

QuickControl Example:



Zero Target & Position (ZTP)

Description:

This command zeros the Target register and the Position register. This command zeros out both registers and removes any Position Error that may exist. This is useful for homing routines to denote the current location as “Zero” so that all other locations can be defined as an offset from “Zero”.

This command removes any Windup that may exist from a previous motion.

Command Info:

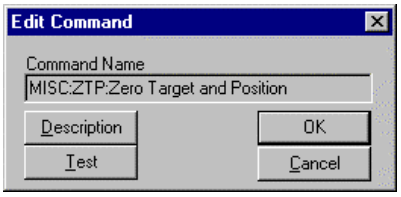
Command Name	Command Type/Num	Parameters	Param Type	Parameter Range
Zero Target & Position (ZTP)	Program Class D 145 (0x91) 1 word	NONE	NONE	NONE

Example:

Sets the Target & Position to zero (“0”).

@16 145 (CR

QuickControl Example:



SilverMax Response:

ACK only

SilverMax Command Set - Numeric/TLA List

Some commands share the same command number. This occurs when a SilverMax command accepts alternate parameters or has multiple uses.

Command Number	Three Letter Acronym (TLA)	Command Name	Reference: Page Number
		Immediate Mode Commands	
0	POL	Poll	11
1	CPL	Clear Poll	10
2	HLT	Halt	17
3	STP	Stop	19
4	RST	Restart	18
5	RVN	Revision	15
6	RPB	Read Program Buffer	14
8	CLP	Clear Program	115
9	SDL	Start Download	136
10	RUN	Run Program	135
11	WRI	Write Register, Immediate Type	165
12	RRG	Read Register	158
13	SPR	Store Program	137
14	LPR	Load Program	129
15	VMI	Velocity Mode, Immediate Type	77
20	RIS	Read Internal Status Word	13
21	RIO	Read I/O States	12
25	IMW	Interpolated Mode Write Queue	91
131	LVP	Low Voltage Processor Threshold	49
134	MAV	Move Absolute, Velocity Based	95
135	MRV	Move Relative, Velocity Based	98
137	JRE	Jump On Register Equal	174
137	JGE	Jump On Register Greater Or Equal	123
137	JLT	Jump On Register Less Than	126
137	JNE	Jump On Register Not Equal	127
138	WCL	Write Command Long Word	162
139	WCW	Write Command Word	163
140	DLY	Delay	117
140	WDL	Wait Delay	174
142	GCL	Go Closed Loop	69
143	GOP	Go Open Loop	70
144	ZTG	Zero Target	172
145	ZTP	Zero Target And Position	27
146	TTP	Set Target To Position	171

Command Number	Three Letter Acronym (TLA)	Command Name	Reference: Page Number
		Program Mode Commands	
147	CME	Clear Max Error	168
148	CTC	Control Constants	28
149	TQL	Torque Limits	65
150	AHC	Anti-Hunt Constants	23
151	ERL	Error Limits	40
152	OLP	Open Loop Phase	53
154	WRP	Write Register, Program Mode	166
154	WRF	Write Register File	166
155	IDT	Identity	44
156	LRP	Load And Run Program	130
159	VMP	Velocity Mode, Program Mode	78
160	RAV	Register Move Absolute, Velocity Based	108
161	RRV	Register Move Relative, Velocity Based	110
162	JMP	Jump	124
162	JOI	Jump On Input	121
163	CIS	Clear Internal Status	169
164	CKS	Check Internal Status	168
165	CLC	Calculation	115
166	CLM	Control Loop Mode	27
167	KMC	Kill Motor Conditions	47
168	MCT	Motor Constants	51
169	FLC	Filter Constants	42
170	EEM	Enable Encoder Monitor	148
171	DDB	Disable Done Bit	29
171	DEM	Disable Encoder Monitor	147
172	PAC	Phase Advance Constants	55
173	ADL	Ack Delay	21
174	BRT	Baud Rate	26
176	MAT	Move Absolute, Time Based	93
177	MRT	Move Relative, Time Based	96
178	RAT	Register Move Absolute, Time Based	107
179	RRT	Register Move Relative, Time Based	109
180	SSD	Scaled Step And Direction	73
181	KMR	Kill Motor Recovery	48
182	KED	Kill Enable Driver	46
183	KDD	Kill Disable Driver	45
184	DIR	Direction	31
185	PRO	Protocol	57
186	SIF	Serial Interface	62
187	EDL	Enable Done Low	37
188	CIO	Configure I/O	145

Command Number	Three Letter Acronym (TLA)	Command Name	Reference: Page Number
		Program Mode Commands	
189	MDS	Modulo Set	150
190	MDC	Modulo Clear	149
191	MDT	Modulo Trigger	151
192	SEE	Select External Encoder	60
193	ARI	Analog Read Input	144
194	WBS	Wait On Bit State	140
195	SCF	S-Curve Factor	58
196	RSM	Register Store Multiple	159
197	RLM	Register Load Multiple	156
198	RSN	Register Store To Nonvolatile	161
199	RLN	Register Load From Nonvolatile	157
201	PCL	Program Call	131
201	PCI	Program Call On Input	132
202	PRT	Program Return	133
202	PRI	Program Return On Input	134
204	WBE	Wait On Bit Edge	138
205	SOB	Set Output Bit	153
206	COB	Clear Output Bit	146
207	ACR	Analog Continuous Read	142
208	PLR	Power Low Recovery	56
211	CAI	Calibrate Analog Input From Nonvolatile	176
212	LVT	Low Voltage Trip	49
213	OVT	Over Voltage Trip	54
214	MTT	Maximum Temperature Trip	52
215	CTP	Calculation Two Parameters	112
216	PIM	Position Input Mode	71
217	VIM	Velocity Input Mode	76
218	TIM	Torque Input Mode	75
219	AHM	Anti-Hunt Mode	25
221	SSL	Soft Stop Limits	64
222	TRU	Torque Ramp Up	67
223	RSD	Registered Step & Direction	72
225	EMT	Enable Multitasking	39
226	DMT	Disable Multitasking	35
227	EMD	Enable Motor Driver	38
228	DMD	Disable Motor Driver	34
229	HSM	Hard Stop Move	81
230	AHD	Anti-Hunt Delay	24
231	PCM	Pre-Calculate Move	101
232	PCG	Pre-Calculated Go	100
233	XRV	Extended Register Move Relative, Velocity Based	84

Command Number	Three Letter Acronym (TLA)	Command Name	Reference: Page Number
		Program Mode Commands	
234	XAV	Extended Register Move Absolute, Velocity Based	82
235	XRT	Extended Register Move Relative, Time Based	83
236	XAT	Extended Register Move Absolute, Time Based	81
237	GOC	Gravity Offset Constant	43
238	JNA	JUMP ON INPUTS, NAND-Ed	123
239	JOR	JUMP ON INPUTS, OR-Ed	124
240	PMC	Profile Move Continuous	102
241	PMV	Profile Move	105
242	PMX	Profile Move Exit	106
243	DLC	Dual Loop Control	32
244	SLC	Single Loop Control	63
245	PCP	Position Compare	152
248	ATR	Add To Register	155
249	PMO	Profile Move Override	104
250	JAN	Jump On Inputs, Anded	120
251	EDH	Enable Done High	36
252	DIF	Digital Input Filter	30
253	IMS	Interpolated Mode Start	87
254	IMQ	INTERPOLATED MODE QUEUE CLEAR	90

Index

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Y

Z