Shutdown and Recovery

This Technical Document describes the SilverLode Controller/Drivers shutdown and recovery tasks. Other Technical Documents referenced below can be found on our website.

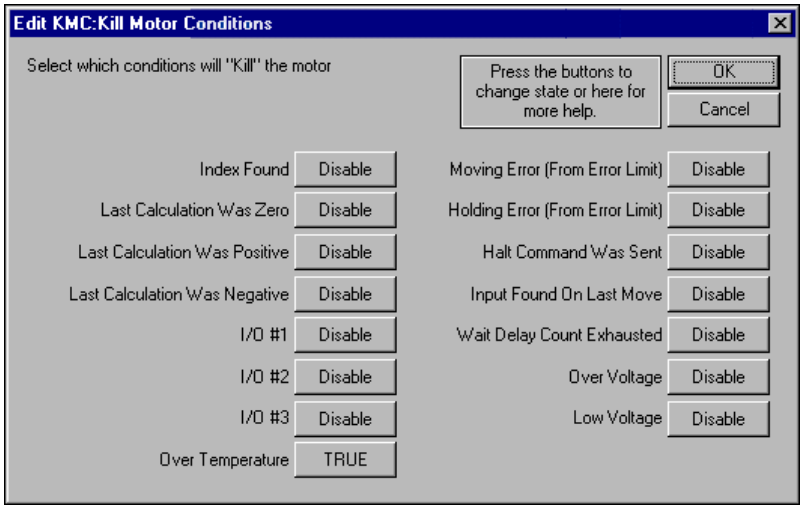
Automatic Shutdown

Every servo cycle (120 microseconds), the SilverLode servo performs an error check based on the settings issued in the Kill Motor Conditions (KMC) (SilverDust Rev 06 or KMX command). If any of these kill conditions are met, the program specified by the Kill Motor Recovery (KMR) command is immediately loaded. In addition, an over voltage condition will always trigger a kill condition if the driver is enabled. The program specified by KMR can then perform any operation, a shutdown, recovery, or other technique.

The default initialization files have a KMC that selects a default recovery program which repeatedly cycles the Status LED (green) located on the back of the servo after a KMC setting is found TRUE. Strategies on how to modify the recovery program based on application requirements are covered in this technical document. The Low Voltage Trip (LVT) command operates similar to the KMC command except that it is triggered by one condition (low voltage), and it loads the program specified by the Power Low Recovery (PLR) command. The default initialization files are set up to run a Power Low Recovery program that disables the servo driver whenever the LVT setting is tripped (voltage drops below specified value). This distinction between interrupts adds the functionality of choosing a recovery process based on whether the servo has found error conditions relating to normal machine operation (KMC settings), or if power has been cut off to the servo (LVT settings). Such conditions most often are used for emergency stops, halting on mechanism jams, hard stop homing routines, program interrupts, and power down scenarios.

Servo Error Conditions (Kill Motor Conditions)

Fifteen different conditions are available to shutdown the servo. Each of these conditions can be independently enabled. When enabled, the state (TRUE or FALSE) that will trigger the kill condition is also set. The conditions are derived from the Internal Status Word (ISW) that is used by the servo in a variety of commands such as the JUMP command. The over voltage trigger is automatically enabled as a kill condition if the motor driver is enabled. The over voltage condition will automatically disable the motor driver, thus the recovery sequence may proceed, such as flashing the LED to indicate the problem.



Kill Motor Conditions Edit Window

Index Found

The internal encoder within the servo can trigger a kill condition since this condition resides within the ISW, and the ISW is the basis for KMC. This condition is available but is not necessarily useful as a shutdown condition. If a an external encoder has been setup, its index signal will also trip this condition

Last Calculation Was Zero, Positive or Negative

When a calculation takes place (Using the CLC command) the result of the calculation will be one of the above. This allows shutdown to occur based on the result of a calculation. If for example, a calculated position parameter should never be negative, setting the “Last Calculation Was Negative” to TRUE will kill the servo when this situation is encountered.

I/O #1, #2 and #3 - (Used for E-Stop conditions)

Three of the seven digital Inputs are available as kill conditions. They are probably most useful when developing an emergency stop (E-Stop) signal. When using the “controlled shutdown” method these inputs can be used to redirect program operation.

Over Temperature – (Always Used)

Over-Temperature must always be enabled so that the servo can halt operation if it gets too hot. The over-temperature is actually a combination of the servo driver thermal sensor (not settable) and the internal electronics temperature sensor (Settable using the Maximum Temperature Trip (MTT) command). Either of the two can cause this condition. In addition, the 34-frame driver enable line triggers the same OverTemp flag when voltage drops below roughly +10VDC on the enable line. Note that on the 34 frame servos, the OverTemp error signifies two different faults.

Motion and Holding Errors – (Most Commonly used Condition)

In any servo system, the most common need for servo shutdown is when moving or holding position error exceeds a set limit. Using the Error Limits (ERL) command, the amount of acceptable error can be set for both the Moving Error and the Holding Error. When the error exceeds the limit, these flags will be triggered. Typically, the Moving Error is set to a larger value because of the dynamic conditions encountered during a move. One or both of these conditions can be used for shutdown.

Halt Command was sent – (Used to re-initialize the servo after a Halt)
The Halt (HLT) command, when sent from a host system, will cause the servo to shutdown. This leaves the servo in a disabled state with no torque applied to shaft. By enabling this condition, the servo could process an error recovery routine or re-initialize itself without host intervention.

Sensor Found on Last Move – (Used for "Hard Limit" shutdown)
All motion commands can be stopped using a digital input or other bit state conditions. When an input stops a motion command, this condition is set. If the input was an end-of-travel sensor, the servo could also shutdown in order to halt motion.

Wait Delay Count Exhausted – (Used for a "Watch Dog" timer)
The Delay counter can be pre-set with a time value that will count down automatically. When the counter reaches zero, this condition is set. Setting the Delay counter and enabling this condition prior to an operation that should complete in a given time allows a *Timeout* function. If the operation does not complete before this condition is disabled (disabled by issuing another KMC command), a shutdown will occur. A host could also be setup to update the watchdog, such that if communications was lost a shutdown would occur.

Over Voltage – (Used for handling Over-Voltage conditions)
Over-Voltage will always cause the servo to shutdown regardless of the KMC command settings. However, if some type of error recovery is desired, setting this condition will allow the servo to continue operation after the Over-Voltage. This is useful when the over-voltage is due to heavy deceleration and control of the load must be regained. (SilverDust Rev 06 the over voltage is only automatically enabled if the motor driver is enabled. This allows for a recovery routine that can report the problem without continually retriggering.)

Under Voltage – (Don't use this, use the "Power Low Recovery" instead)
Don't use this unless a servo shutdown is what you really want to do. The servo has a unique feature in that a low voltage condition can be used to trigger a special recovery routine that can be use to exit operation before power is totally lost. Use the Power Low Recovery (PLR) command to set up this operation. The under-voltage threshold can be set using the Low Voltage Trip (LVT) command. This option is provided for backward compatibility reasons.

If any one or more of the preceding conditions are enabled and found to be TRUE, the servo will execute a process defined by the recovery program. The recovery program can do *nothing* (disable the servo and wait, as the default recovery process does) or execute a program at a specified location in the non-volatile memory.

Extended Servo Error Conditions (Kill Motor Extended)

Available on the SilverDust (Rev 06) the Kill Motor Extended (KMX) command allows any of the I/O to be used as a kill motor trigger, as well as an encoder analog fault, encoder re-sync (lost counts), motor temperature (IP65 units with SilverDust IG or IGB controllers), and other more specific conditions. The ISW Over-temperature bit includes several different causes; these have been independently broken out in the IS2 word to simplify troubleshooting and tuning of kill motor recovery routines. This command extends the Kill Motor Conditions (KMC) function. (The KMC command overrides the two extra banks of parameters in the KMX

command, setting them all to zero. The KMX command sets these banks according to the provided parameters. The Clear Status Word command clears the latched bits in both the ISW and the IS2 status words.)

Extended IO (SilverDust IGB)

All 16 bits of the extended I/O – 101 through 116 – are available as trigger conditions for the Kill Motor Extended command. Each bit may be independently enabled, with a separate parameter for each bit to select the active state.

Internal Status Word 2 (IS2)

Any of the bits in IS2 may be used to trigger a Kill Motor Extended command

I/0 4,5,6, and 7

In addition to I/0 1,2, and 3 in the ISW, I/0 4,5,6 and 7 may be used to trigger a Kill Motor Recovery.

Extended IO Fault

This fault is set when units with extended IO, such as the SilverDust IGB, are unable to communicate with the isolated I/O. Every 120uS the new I/O states are written to and read from the isolated I/O section, along with a check byte. If the check byte is not correct, it indicates that the hardware communications path is not working, and the Extended IO Fault is set. This bit is latched, and is cleared by the Clear Internal Status (CIS) command. A common source of error would be the lack of 12-24v power to the isolated I/O section (this section must have its own power connections.)

Encoder Fault

The encoder receiver includes analog comparator circuitry to determine if the signals meet RS-485 specification, including sufficient differential amplitude, excessive common-mode, opens or shorts,.

This bit may be ignored on the SilverDust MG if the encoder has been jumpered to use single ended signals, as these may falsely trigger. The SilverDust MG may not reliably be able to detect open signals (trade off to allow use with single ended encoders.)

Driver Temperature Fault (SD11)

Note that this bit is Active HIGH. It is set if the driver internal temperature sensor indicates over temperature.

Hardware Driver Disabled (SD11)

This bit is set if the Driver Enable on the SilverDust IG/IGB is not enabled (High). This signal may be used to delay power-up initialization until the driver has been enabled.

Motor Temperature Fault

This bit is only valid for IP65 I-Grade motors. Note that this bit is Active HIGH. It is set if the internal temperature sensor indicates over temperature.

Factory Block Driver Disabled (SD11)
The Driver Disable Factory bit also sets active (low) the Over Temp/Driver Disabled bit in the ISW register. This bit is set by routines in the Factory Block (which runs at power-up a/reset before the initialization code starts up. In the SilverDust IG and IGB, this code determines if the unit has been configured to use Motor Memory, and if so, it reads in the parameters, making sure that the motor type matches, and that the checksum on the motor memory block is correct, as well as certain other motor memory parameters. If errors exist, the driver is disabled by a special command within the factory block, and the error code is passed to the initialization routine via Register 41. This allows the initialization code to set the communications parameters, and to flash the error code. This allows the cause of the problem to be identified. The driver may be enabled by correcting the problem and powering up again, or by re-initializing the device. There is no User command to re-enable the driver from a Factory Block fault.

Encoder Re-phased
This bit is only set if the encoder index align has been enabled. It is set if sufficient encoder pulses have been lost so as to require resetting of the encoder to motor phasing information. This bit is cleared with the Clear Internal Status (CIS) command.

Motion Limit Fault (SD11)
HIGH indicates motion is being limited. See bit description in Internal Status Word 2 for more details. This bit is cleared with the Clear Internal Status (CIS) command.

Count Down Timer Active (SD08)
The Count Down Timer (register 245), is not zero, and therefore is counting down. If the user has a process to refresh this timer, then this counter may serve as a “watchdog” counter if a low for this bit is used to trigger a shutdown. This counter may also just be used to terminate a motion, or for other general purpose operations.

Kill Process
The kill process can be carried out two different ways, depending on the initialization settings. The servo default kill process ends any motion and/or program that is currently executing and disables the motor drivers, which cuts off torque to the shaft. This causes the servo to come an uncontrolled stop. This uncontrolled kill provides no deceleration control using only friction to slow the servo to a stop. The servo executes the actions specified by the KMR program.

The second shutdown method leaves the servo operating and calls a specified recovery program (with the exception of over voltage operation, which always disables the driver). This allows the servo to do a controlled shutdown if required. When a kill occurs, the servo leaves its motor drivers enabled allowing controlled deceleration (granted there are no obstructions and inertia has been taken into account). In addition, with Multi-Tasking enabled the servo finishes any motion that was in progress at the time of the kill. This motion will continue to completion unless it is overridden by a command in the KMR program.

Shutdown Commands
The details of the following commands are available in the Command Reference.

Kill Motor Conditions (KMC)
The KMC command allows the user to select what conditions will allow a controlled shutdown of the unit. The Condition Enable Word selects which bits in the Internal Status Word will be evaluated. Conditions are enabled by setting a "1" in the desired bit position of the Condition Enable 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.

Kill Motor Extended (KMX)
This is an extended version of the KMC command, with three Enable Words and Three Condition words.

Kill Motor Recovery (KMR)
The KMR command sets up options for recovery from a shut down. The KMC command establishes conditions that will cause the servo to shut down. Using the KMR command, the servo can perform a standard or user defined process for re-initializing the servo. User programs can be executed that have been previously stored in the non-volatile memory.

Kill Enable Driver (KED)
KED causes the servo to leave the servo drivers enabled when a KMC setting is met. Normally the motor driver is disabled when a KMC setting is met. The KED command is used to leave the driver enabled if continuing operation is required. In order for this command to function, the servo must be set up for Multi-Tasking operation (See Enable Multi-Tasking (EMT)). Without Multi-Tasking, the driver will be disabled when a KMC setting is tripped, whether or not this command has been issued. This command is very useful when a controlled shutdown of the servo is needed. For example, if there is a need to slowly ramp down the speed of the servo, a Velocity Mode, Program Type (VMP) command can be used in the recovery program to decelerate to zero velocity with a given deceleration. Note: the driver is always disabled if an over voltage condition is detected.

Kill Disable Driver (KDD)
Configures the servo to disable the motor driver and shorts the windings together when a KMC setting is met. If the servo is moving, it will stop immediately in an uncontrolled manner. The servo will be unable to move until re-enabled using the Enable Motor Driver (EMD) command. This is the default setting for the servo. This is the default state of the servo.

Other Relevant Commands

Error Limits (ERL)
Error Limits set the Moving Error limits and Holding Error limits used by the KMC command. Kill conditions triggered by error limits set bits contained in both the Internal Status Word and in the Polling Status Word. The Polling Status Word bits can alert a host controller to the condition if the servo is being polled. If Drag Mode is enabled, then the position error will never exceed the limits, and these conditions are not useful as Kill Conditions. See Command Reference for more information on the Error Limits Command

Enable Multi-Tasking (EMT)
Enables Multi-Tasking operation, which allows motion to occur while executing a program in the background. Multi-Tasking is required when using the Kill Enable Drivers Command (KED). See the Command Reference for details on this command.

Enable Motor Driver (EMD)
Enables the servo driver. The driver is enabled by default, so this command is only required if the driver has been disabled by either the Disable Motor Driver (DMD) command or a kill condition. This command does not re-enable the driver if it has been disabled by the Factory Block.

Recovering or Restarting After a Kill Motor Condition

Simple Shutdown Routine

This program selects KMC settings, but selects no recovery program. Note that if the servo shuts down, a host would need to poll the servo in order to determine that shutdown occurred. This situation requires constant outside monitoring and intervention if immediate post fault action is needed. For most applications, the next method would be more applicable. In addition, this program can be used to shutdown the servo indefinitely until an operator resets manually.

Line# Oper	Label	Command
1:REM		This program allows the motor shutdown routine to be tested. Host intervention must be used to bring the motor back up and running. Move back and forth until Motion Error occurs and the servo shuts down on this Kill Motor Condition.
2:REM		Set torque values low. The values are set low to allow the shaft or flywheel to be manually stopped by the hand.
3:TQL		Torque Limits: Closed Loop Holding = 15 % Closed Loop Moving = 9 % Open Loop Holding = 15 % Open Loop Moving = 9 %
4:REM		Set up the "Error Limits" that establish the amount of error which will cause the motor shutdown.
5:ERL		Error Limits: Moving Limit = 200 counts Holding Limit = 100 counts Delay to Holding = 100 ticks
6:REM		Set the conditions on which the servo should automatically shut down.
7:KMC		Kill Motor Conditions: If Over Temperature or Moving Error
8:REM		A simple move "back-and-forth" loop. The servo will continue its motion until the Kill Motor Conditions are met.
9:MRT	FOREVER	Move 2 revs @ ramp time=500.04 mSec total time=2000.04 mSec
10:MRT		Move -2 revs @ ramp time=500.04 mSec total time=2000.04 mSec
11:JMP		Jump to "FOREVER"

To return the servo to operation, a host must send either of the following command string(s):

RESTART (RST)
The servo acts as though it was just powered on by resetting itself and jumping to the default program in NV memory. This will clear all registers, including the current position

ASCII example:
@16 4 <CR>

- OR -

To maintain the current position, and other internal data:
TARGET TO POSITION (TTP)

Clear the accumulated error; sets the target to coincide with the current measured position..
CLEAR INTERNAL STATUS (CIS)
Clear any latched bits in the ISW or IS2 that triggered kill.
POLL (POL)
Read the error bits; these must be read before they may be cleared.
CLEAR POLL (CPL)
Clear the moving error in the PSW as well.
KILL MOTOR CONDITIONS (KMC)
Reset the servo shutdown conditions.

ENABLE MOTOR DRIVERS (EMD)

Re-enable the servo drivers so the servo can begin motion.

ASCII example:
@16 146 <CR>
@16 167 256 256 <CR>
@16 227 <CR>

Kill Motor Recovery from Uncontrolled Shutdown

Handling the Shutdown

The servo shutdown may occur while in motion. If this is true there may be a period of time required to allow the servo to come to a complete stop.

Fault Recovery Program

The simplest way to deal with this is to wait using the Delay (DLY) command. Set a delay sufficient to allow the system to come to a halt. This depends on the system implementation; the needed time delay will vary greatly. Once the servo has stopped the error processing can begin. Systems with large inertias may want to monitor the measured position, and keep restarting the delay as long as motion has been detected since the start of the previous delay.

Processing the Error Condition

Depending on the shutdown condition enabled, there may be a number of different things that must be done to clear the error, continue operation, or simply alert the system or operator of an error. Alerting the system or operator may involve toggling an output. The program example shows a Moving Error triggered recovery that will allow the servo to re-home and continue with operation. The first operation is to correct the existing error; do this using a Target to Position (TTP) command. This removes any Position Error that may exist. Then, Clear Internal Status (CIS) and reissue the KMC and KMR commands to reset the KMC and KMR operation.

Line# Oper	Label	Command
1:REM		This program performs an "uncontrolled" shutdown of the servo. The servo immediately shorts the windings on the motor upon a motor shutdown condition. Then the recovery program re-enables the drives and returns the servo to readiness.
2:REM		Define the "Kill Motor Recovery" process that will be used. In this case, the servo calls a "Recovery" Program to deal with the error.
3:KMR		Kill Motor Recovery: Program = "Fault Recovery"
4:REM		Set the conditions on which the servo should automatically shut down.
5:KMC		Kill Motor Conditions: If "I/O #3" is LOW
6:REM		A simple, move "back-and-forth" loop. The servo will continue its motion until the Kill Motor Conditions are met.
7:MRT	FOREVER	Move 2 revs @ ramp time=500.04 mSec total time=2000.04 mSec
8:MRT		Move -2 revs @ ramp time=500.04 mSec total time=2000.04 mSec
9:JMP		Jump to "FOREVER"

Uncontrolled Shutdown Example

Restoring operation

After a shutdown the system can be restarted automatically by adding a few extra steps to the Recovery Program. If a homing routine is to follow, the Torque Limits could be set to a lower value required for homing. Next, the servo drivers must be re-enabled using the EMD command. The servo is now ready for operation. From here, a homing or other corrective action program can be executed putting the system back into service.

Kill Motor Recovery from Controlled Shutdown

Initializing for Controlled Shutdown

Controlled shutdown requires adding a few extra elements. The most important is to add an EMT command. Multi-Tasking will allow the servo to continue a program operation while processing the shutdown condition. Also, issue the KED command. This command over-rides the default setting that disables the motor drivers on shutdown, allowing the servo to complete any programmed motion while processing the shutdown condition.

NOTE: be sure to set the wanted KMR recovery routine before setting the KMC/KMX to enable detection of an error, or, if an error already exists, the routine will just stop, as the associated recovery routine has not been specified at the time such error is enabled for detection!

Line# Oper	Label	Command
1:REM		A fault has occurred. By default the servo disables the Motor Driver. So now, the motor must be returned to operating status.
2:REM		Give the motor a little time to stop. This can be increased if needed.
3:DLY		Delay for 500 mSec
4:REM		Now that the motor is stopped, set the "Target Position" to the "Actual Position". This effectively zeroes the Position error.
5:TTP		Target to Position
6:REM		Reset Internal Status Word in order to clear any latched bits.
7:CIS		Clear Internal Status
8:REM		Re-set the "Kill Motor Conditions". This is required before motion is resumed again.
9:KMC		Kill Motor Conditions: If "I/O #3" is LOW
10:REM		Now, re-enable the motor driver. Form here on the motor will have power to the rotor
11:EMD		Enable Motor Driver
12:REM		Just a wait before calling the next program
13:DLY		Delay for 500 mSec
14:REM		Run a system recovery program from here. This following program could be a "Homing" routine that would recover from the error.
15:LRP		Load And Run Program: Program = "Home"

Recovery Program

Handling the Shutdown

In the controlled shutdown scenario, the servo does not stop the motion in process. Instead, the recovery program is called allowing the servo to stop the motion (or not) in a controlled manner. In the example shown (Fault Recovery Program) a Velocity Mode command is issued with a velocity of zero. This will bring the servo to a controlled stop.

Processing the Error Condition

Depending on the shutdown condition enabled, there may be a number of different things that must be done to clear the error and continue operation or simply alert the system or operator of an error. Alerting the system or operator may involve toggling an output. In this example nothing was required other then re-issuing the KMC command.

Continuing Operation

Continuing is less complicated because the servo does not have to be re-initialized. This may be as simple as doing a Load and Run Program of the corrective action program.

Power Loss Management and Recovery

The servo needs a constant voltage source in order to function. Unfortunately, power can be interrupted, so there are built-in functions to help deal with this. At approximately 10.5V, the servo disables the driver circuit. The DSP still function at this voltage; but the servo is no longer able to apply torque to the shaft. Disabling the drivers prevents any erratic movement by the servo at low voltages. If the voltage drops below 7.5V, the control electronics start shutting down. All data stored in the volatile memory (RAM) is lost. Upon the voltage rising above this lower limit, the servo essentially powers back on and goes through its default initialization.

When the voltage drops below approximately 7.5V, the servo loses its position information. The encoder is not an absolute encoder. Without power, the servo cannot monitor its position. If any movement occurs during a powered down period, the servo will need to re-establish its current position, by some method such as re-homing. A back-up power supply provides an alternative, by keeping the internal circuitry active. Note that only the voltage dropping below the threshold will cause this information to be accidentally lost. The drivers being disabled, for any reason, will not clear the position information.

Note that both the SilverNugget N3 and the SilverDust IG and IGB have auxiliary Processor Power inputs. The Processor Power on the SilverDust IG and IGB are not fully isolated as they are on the SilverNugget N3.

Line# Oper	Label	Command
1:REM		A fault has occurred. By default the servo automatically zeros the Torque Limits and Disables the Motor Driver. Give the motor a little time to stop. This can be increased if needed
2:VMP		Velocity Mode: acc=8000 cps/s, vel=0 cps
3:DLY		Delay for 500 mSec
4:REM		Re-set the "Kill Motor Conditions". This is required if after this recovery routine normal operation is restored.
5:KMC		Kill Motor Conditions: If "I/O #3" is LOW or Moving Error
6:REM		Just a wait before calling the next program
7:DLY		Delay for 500 mSec
8:REM		Run a system recovery program from here. This following program could be a "Homing" routine that would recover from the error.
9:LRP		Load And Run Program: Program = "Home"

Line# Oper	Label	Command
1:REM		Move back and forth until Motion Error occurs and runs Kill Motor Recovery. Use the "Kill Enable Drivers" to allow the servo to leave the motor drivers enabled during motor shutdown.
2:KED		Kill Enable Drivers
3:REM		Enable Multi-Tasking operation so that a "Motion" can continue even when the "Kill Recovery" program is called
4:EMT		Enable Multi-Tasking
5:REM		Define the "Kill Motor Recovery" process that will be used. In this case a "Recovery" Program will ba call to deal with the error
6:KMR		Kill Motor Recovery: Program = "Fault Recovery"
7:REM		Enable the "Input 3 LOW" condition for motor shutdown
8:KMC		Kill Motor Conditions: If "I/O #3" is LOW or Over Temperature
9:REM		Move the motor back and forth waiting for a "User" induced Motion error
10:MRT	FOREVER	Move 40000 counts @ ramp time=200 mSec total time=2000 mSec
11:MRT		Move -40000 counts @ ramp time=200 mSec total time=2000 mSec
12:JMP		Jump to "FOREVER"

Power Low Recovery Commands

Low Voltage Trip (LVT)

Used in conjunction with the Power Low Recovery command, this command sets the input voltage that will trigger a Low Voltage status (Bit #14 in the Internal Status Word). When a Low Voltage Trip occurs, the servo will stop immediately, then load and run the program defined by the PLR command. This command allows the user to shut down the servo properly when power is lost. The current position of the servo could be stored to non-volatile memory for pseudo absolute-encoder functionality. The LVT trip setting should by set as high as possible for any given power supply. A higher setting causes the servo to trip sooner giving it more time to write data to the non-volatile memory.

Power Low Recovery (PLR)

If the input voltage drops below a set value (see Low Voltage Trip) a user program can be called that can perform a power loss exit routine. This command selects what to do in case of power loss. A Load and Run Program is issued which runs the power loss exit routine from a specified NV Memory address.

Other Related Commands

Disable Motor Driver (DMD)
Disables the motor driver and shorts the windings together (On SilverNugget N3, the windings are not shorted, but rather the driver transistors are disabled.. The servo will be unable to move when attempting any motion command. This is a software disable that can be overcome by the EMD command.

Register Store Non-Volatile (RSN)
Stores data from a Data Register to the selected non-volatile memory address. See the Command Reference for details on this command.

Register Store Multiple (RSM)
Stores data from an array of Data Registers to the selected non-volatile 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 non-volatile memory. Data is also copied from the Data Registers sequentially.

Recovering and Processing Saved Information

- Position
- Direction
- Last Point in Program

In many applications, the servo must retain its absolute position. Once the LVT limit is reached, the Power Low Recovery program is called, which can use the Register Store Non-Volatile (RSN) command to write the current position and other data to the non-volatile memory.

Power Low Recovery and Restoration of Direction and Position

The following QuickControl program is an example of the process and commands needed to recover the last known position and direction of the servo when the supply power was shut off.

Time Needed For Saving to NV Memory

Maximum time to write data to the NV memory ranges from is 10mS per write operation. Each data storage command uses at least two write operations, but additional operations are needed if the data crosses a page boundary (addresses divisible by 32), Therefore, when using the Register Store Non-Volatile (RSN), the maximum time needed to save 5 data registers is (i.e. using RSN 5 times):

5 data registers use (5x2)+2=12 words. This will always fit in 1 or 2 memory pages

(2 pages + 1 extra) x (10 ms/write) = 30 ms

In order to minimize save time, the Register Store Multiple (RSM) command stores multiple registers (up to ten sequential registers) using the same page write cycle. RSM uses the same identification and checksum words for all registers. RSM can store up to 10 registers in 1.5x the time needed for the one RSN command.

Backup Power Alternatives

If a complete power loss occurs just before the final resting point and the servo doesn't quite have time to save all pertinent information, and then additional capacitance placed across the Power Bus may be used to extend the time. When the driver is disabled (by a command or a supply voltage below 10.5), the servo only draws power for the control circuitry. Based on the energy equation for capacitance and the time needed, the extra capacitance can be determined as follows:

$$C = \frac{4 * W * T}{V1^2 - V2^2}$$

W=2.5W (SilverNugget), 1.7W (SilverDust)
V1=Voltage setting of Low Voltage Trip
V2=7.5V
T=Total time needed to save data

This value gives a good estimate of the size of the extra capacitor needed. However, this estimate is based on the assumption that the servo is the only device drawing power from the capacitor. An even larger capacitor is needed if there are any other devices powered from the same line. Alternatively, if the nominal power supply is higher, 48v for example, the Low Voltage Trigger (LVT) could be selected to activate at a higher voltage than the 10.5v setting, for example, 36v. This would provide significant extra shutdown time.

A back-up power supply allows the servo to continuously log actual position even in the event of total power loss. Given the 10.5 to 7.5 volt range (between the point of driver disable and loss of circuit operation), a nominal 9V battery provides an ideal backup supply. When implemented with the QuickSilver voltage clamp, a simple drop-in solution is created. A customer-supplied charger maintains the charge on the battery, which immediately provides power when the main power is lost. Internal diodes in the voltage clamp are biased to ensure that the main power doesn't drive the 9V battery. The SilverDust IG and IGB provide a separate Processor Power input pin which may be used to supply power to the processor when the driver power is not present. (The processor power and the driver power are diode "Or'ed" to power the processor.) The Processor Power is not limited to 7v, but may be as high as 48v.

Using the Kill Motor Process for Program Flow

The issue of resuming program execution from the line where the KMC/KMX setting was tripped may be desirable in some applications. If the program structure is simple enough, the recovery program can be written to jump back to a predetermined location in the main program when completed with the recovery process (or whatever task it is designed for). The recovery program will have to WRP a specific value to a user register and LRP the main program. The first line of the main program will recognize that the system has recovered from the recovery program based on the register set by the WRP command, using the Jump on Register Equals (JRE), or a similar command.