SMC-40 (v1.10)
Software Guide

Also supporting:

- **SMO-40 Analog Input IC**
- **SME-40 Encoder Feedback IC**
- **IBC-400 Stand Alone Motor Controller**
- ***m*STEP-407 Motor Controller - Driver**

Introduction

This manual is intended to provide information on the software features and programming capabilities of the SMC-40 microprocessor and peripheral devices, SMO-40 and SME-40. These devices are available as integrated circuits (IC's) or embedded in higher-level AMS designs. Product versions include:

Master Processors

SMC-40 Master processor supplied as TSSOP28 or PLCC28 IC
IBC-400 Step motor controller using a variation of the SMC-40 processor
***m*STEP-407** Step motor control / driver using another variation of the SMC-40 processor

Peripheral Devices

SMO-40 Analog input IC (also available with IBC-400A and *m*STEP-407-A)
SME-40 Encoder feedback IC (also available with IBC-400E and *m*STEP-407-E)

Variations in command and signal characteristics (to accommodate individual hardware and peripheral options for these products) are indicated in this manual, i.e., the SMC-40 used in the *m*STEP-407 is tailored to accommodate the integral microstep driver for double step output and different default values.

Please refer to the individual product "Hardware Manuals" for specifications and operating procedures.

Technical Update
The AMS "EASI" program (available on our website: www.stepcontrol.com), is a DOS program that will work under Windows 2000 (and older) but may not function properly with Windows XP.

If used in a Windows compatible computer, EASI only functions in the "dumb terminal" mode. Hyper-terminal (a utility supplied with all Windows versions) is recommended for elementary, single axis programming.

For more complex, multi-axis designs, the low cost SIN-11 serial adapter simplifies applications programming and provides reliable software communication protocol with all operating systems. Other advantages in using the SIN-11 include:

- Converts RS-232 to RS-422 signals, allowing one to thirty two axis of communication over long distances.
- Permits multi-axis, Party Line (mini drop network) operation.
- Allows use of USB serial adapters.

IMPORTANT – For models with the "analog" option, all communication is locked out during motion caused by "analog joystick" control. In this instance, the '$'(busy) character is returned. The operator must stop joystick activity (zero speed permits communication). Alternatively, one of the stop functions (ESC, @, or Soft Stop input port) will stop the motion.

NV Memory Overview

The main processor (SMC-40) contains 512 bytes of non-volatile (NV) memory (EEPROM) and 512 bytes of "shadow" RAM. An additional 2048 bytes may be present as external EEPROM.

The NV memory is rated to retain data for 10 years. As with all EEPROMS, the number of times it may be re-programmed is limited. Each time a cell is written a small number of electrons are trapped in the dielectric. After many write cycles (500,000), the dielectric becomes less effective and the cell cannot retain its charge.

Another characteristic of EEPROM is the slow access time. Read time (each byte) is approximately 1mS, write time is 5mS per byte. During reset, the entire 512 bytes of NV memory is copied to the shadow RAM. Fast shadow RAM is then used for all operations except the Store and Clear commands. During normal use, the program and parameters are read/written to the RAM where longevity is not an issue.

When parameters are modified or a user program sequence is entered below address 512, the RAM is written to, not the EEprom. Data will remain in the RAM until reset or modified. The "S" command must be used to store to the EEprom (NV memory write).

Commands that actually write to the EEprom are:

Command	Function
"\"	Writes byte directly to any address.
"C2"	Writes hex FF to RAM program addresses 0 to 199 and 256-511 then stores 512-byte image (includes parameters) to the NV memory.
"S0"	Store parameters to locations 200 to 254 .
"S1"	Store entire 512-byte image (includes parameters) to the NV memory.
\255	Force complete re-initialize of all parameters - execute hardware reset afterwards

To extend the life of the EEPROM in your device it is necessary to be aware of which commands perform writes to the EEPROM, and eliminate those which are not needed.

Note: Use the SAVE command sparingly. The parameters are set so quickly, even in SERIAL mode, that you should let the host download them.

Changing parameters should NOT be done by writing directly to EEPROM because the controller will not recognize the change and may over-write them. Use the commands available to set parameters. Reading on the other hand is non-taxing on the EEPROM.

NV Memory Programming

The above examples were samples of immediate commands. The following is a sample sequence used to store a sequence in the volatile memory (RAM).

	Enter	Remark
	P0<CR>	Place in Program mode. Insert instructions at location 00.
Address		
0	O0<CR>	Set Origin to zero.
1	R10000<CR>	Move 10,000 steps in the "+" direction, relative to Origin.
6	W 0<CR>	Wait until complete.
9	R -10000<CR>	Move 10,000 steps in the "-" direction, relative to Origin.
14	W00<CR>	Wait until complete.
17	J1 3<CR>	Jump to address 1, 4 times.
21	R500<CR>	Move 500 steps in the "+" direction, relative to Origin.
26	P0<CR>	End Program.

Now list the stored program

Q<CR>	Query command.

Verify the Program
The controller will respond with:

```
0   O
1   R    10000.00
6   W    0
9   R    -10000.00
14  W    0
17  J    1     3
21  R    500.00
26
```

Execute the Program

Enter	Remark
G0 1<CR>	Programs start executing at location zero. If the Trace option is on, it will display each instruction prior to execution.

Note: This same program can also be triggered by pulsing the "Go" input and the program can be terminated at any time by hitting the ESCape key.

Edit Program
Example: It is desired to change instruction number 21 from 500 steps to 5,000 steps:

Enter	Remark
P21<CR>	Edit instruction 21.
R5000	Move 5,000 steps in the "+" direction, relative to Origin.
"ESCape"	Terminates Edit mode.

Note: Caution should be exercised when making Program Edits in dumb terminal mode due to variations in command byte length that may effect subsequent command address locations and possible corruption of non-volatile memory storage.

Save the Program
The program may now be preserved by issuing the "S1" command. This command takes several seconds to execute. Try to avoid over usage of the Store commands, as they may affect NV life. Writing to RAM has no such limitation.

Program Command Overview

Immediate (Instant) Commands

These are single character commands that are executed immediately. They are not storable in program memory.

Command	Function	Comment
ESC	Abort	In any mode
^C	Software Reset	In any mode
^N	Assign Name	Single mode only
^P	Enter Party Mode	Single mode only

Set Parameter Commands

These commands are used to set speeds, modes and values used in a system. Each application will require customization of the parameters. As parameters are entered, they are stored in a temporary RAM location. Once the desired values are set, the programmer may enter the "S 0" command to save them to NV storage. Thereafter, each time the controller is reset or powered up, the new parameters will take effect. "C2" will trigger s full clear, erasing all storage, set all parameters to default and store them.

Production systems implementing host control should use the parameter commands to "down-load" settings as part of controller initialization. This facilitates field replacements with only a simple naming procedure required.

As shipped, default values are stored in the NV memory. The "C1" command will reset the controller to the default parameters.

All of the following commands are storable as new user defaults:

Command	Function	Range
(D) Divide step rates	Scale step speeds	0-255
(E) Settle delay	Delay before current setback	0-255
(I) Initial SPS	Start speed in SPS	56- 65,535
(K) Ramp slope	Acceleration, deceleration	0-255,0-255
(V) Slew speed	SPS in index command	56- 65,535
(l) Options (lower case L)	Square wave, invert Lim, dir, enable	Varies with model

Motion Related Commands

The physical direction is dictated by several factors. The convention here assumes that the "+" direction is clockwise rotation as viewed from the motor output shaft end. In some designs it may be desirable to reverse the "+" direction, this is accomplished by reversing the polarity of one motor winding.

Command	Function	Range
+	Index n steps in+ direction	0 to 16,777,215
-	Index n steps in- direction	0 to16, 777,215
@	Decelerate and stop	
F	Find home	56 to 50,000, 0-1
M	Move constantly	56 to 50,000
O	Set origin	±8,388,607
R	Go to absolute Position	±8,388,607
T (nv)	Trip point	
W0	Wait until motion complete	Special case of W
i	Special trip restart	
k	Special trip initiate	

Utility Commands
These commands function to allow read back status or data or set outputs.

Command	Function	Range
A	I/O read/write	0- 128
X	Examine	0-4
Z	Read Position	0-1
[	Read NV Memory	0-2560
]	Read Limits	
\	Write NV Memory	0-2560, 0-255
^	Read moving Status	
N	Read Ramp & Speeds	
u	Emit character	u13 = <CR>

Program Functions
These commands provide the ability to create, store and run functions.

Command	Function	Range
C	Erase memory	0-3
G	Go to / branch	0-2560
g	Branch	0-1
P	Programming on/off	0 to end of memory
Q	List program	0-2560
S	Store to memory	0-2
J, j	Jump to address n times	0-2560
L	Loop on port (input)	

There are three types of memory in the controller.

1. Registers
These temporary storage locations are used for handling parameters and variables. During power-up, they are initialized from those values stored in NV memory. When a parameter command such as "V 1000" is executed the data is processed and stored in the registers.

2. RAM
512 bytes of RAM are available with the basic SMC-40. During power up, the (slower) 512 NV memory bytes are copied into the (fast) RAM where they can be executed at high speed. This is called shadow RAM executed.

3. NV Memory
The SMC-40 contains a 2560 bytes non-volatile (EE) memory. Storage includes user programs and parameter storage (55 bytes). NV memory retains data in excess of 10 years with and without applied power. It is altered by certain commands, as specified in the NV memory section. Excessive writes (500,000+) can wear out the memory.

About Speeds
The controller and software in this product are designed for high-speed performance. The speed (steps per second) is computed using integer math formulas. The actual steps-per-second will be close to that specified, however at higher step rates there are only a finite number of values possible. This is called granularity. As an example, we can chart 15000 SPS with the Velocity (V) command.

V Command SPS	Read SPS (X Command)
V14950	14951
V14960	15012
V15000	15012
V15020	15074
V15070	15074
V15080	15137
V15130	15137
V15140	15200

The affects of this property can be reduced by "dividing" the rate ("D"), or increasing microstep resolution, then compensating by increasing the step rate appropriately (I, V). This technique usually adds the benefit of smoother operation.

SMC-40 Program Commands (In ASCII Character Code Order)

Command	Function		Type	NV Bytes
	Mnemonic	Data 1 (Range)	Data 2 (Range)	Result

Where:

Command:	Keystroke
Function:	Functional description of command
Type:	Immediate = Direct execution Program = Executable in stored program Global = All controller present Default = Initial parameter setting Hardware = Auxiliary I/O
NV Bytes:	Storage requirements in program
Mnemonic:	Single character prefix used in multi-controller protocol; (prefixed by controller "name" assignment in party line mode)
Data 1:	Affected parameters
(Range):	Valid numerical range of parameter(s)
Data 2:	Same as Data 1 (as required)
Result:	Information returned as a result of command execution or examination

Note: A comma separates two parameters.

Command	Function	Type	NV Bytes
^C	Reset Controller	None	N/A
Mnemonic	Data 1	Data 2	Result
(Name) ^C	None	None	None

^C (Reset)

Response to the single character "Ctrl C" (3 decimal, 03 hex) is immediate.

Resets controller to power-up condition, waiting for start sequence. It is analogous to "Ctrl-Alt-Delete" -reboot the computer. All outputs are set high, defaults are reloaded from NV memory, and position is set to zero. This command may not be used within the non-volatile program memory. This command does not modify the NV memory values. If an instruction is stored in memory location 192, it will automatically start execution- see the "G" command.

Command	Function	Type	NV Bytes
^N	Name Controller	None	N/A
Mnemonic	Data 1	Data 2	Result
^N	None	None	None

^N (Name Axis)

Response to the single character "Ctrl N" (14 decimal, 0E hex) is immediate.

This command must be executed with one SMC-40 controller (or any other AMS controller) attached, otherwise a bus conflict will occur and all axes may receive the same name. On entry of a "Ctrl N," the SMC-40 responds with "Name?" Type the desired name and the controller will respond "Save (Y)?" Type "y" (lower case only). "OK" is echoed back and the character is stored in NV memory. The SMC-40 is reset and must be "signed on" using the space character.

Command	Function	Type	NV Bytes
^P	Restart in Party Line Mode	None	N/A

Mnemonic	Data 1	Data 2	Result
^P	None	None	None

^P (Party Line Mode)

In response to the single character "Ctrl P" (16 decimal, 10 HEX) the controller(s) immediately enter the party line mode. One, or up to 32 controllers, may be connected in a "mini drop" network.

AMS - SIN-11 Serial Adapter
If you are using a SIN-11 serial adapter from AMS, the "&" command must be used to initiate party line operation. On receipt of the "&" command, the SIN-11 automatically transmits the ^P character within the party line startup routine. See the serial communication section in your product specific Hardware Manual for more information.

Command	Function	Type	NV Bytes
ESC	Terminate Operation	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
(Name) Esc Char	None	None	Echo #

ESC (Global Abort)

Response to the single character "ESC" (27 decimal, 1B hex) is immediate.

Terminate any active operation and cause the controller to revert to the idle state waiting for a new command. Output ports are NOT affected. Stepping and position counter update will cease immediately without deceleration. The lack of deceleration can cause mechanical overshoot. The controller will echo a "#" character. This command may not be used within the non-volatile program memory.

Command	Function	Type	NV Bytes
+	Index in Plus Direction	Immediate, Program	4

Mnemonic	Data 1	Data 2	Result
(Name) + (n)	Steps (0-16,777,215)	None	None

+ (Index in Plus Direction)

Step in the positive direction for the specified step count. The motor will ramp up, slew, and then ramp down per the previously set parameters. The range is 0 to 16,777,215. The position counter will overflow at 8,388,607.

The motion sequence is:

1. Wait until any previous motion is finished.
2. Energize the motor winding as required.
3. Start stepping at the rate of the initial velocity (I).
4. Accelerate using a profile defined by the fixed table that approximates straight-line acceleration and a slope set by the "K" command.
5. Accelerate until the slew speed, as specified by the "V" command, is attained.
6. Motion continues at the slew speed, until the deceleration point is reached.
7. Decelerate (determined by the second "K" value) to a stop completing the index.
8. If another index is not commanded for the settling period, power down the motor (if auto power down is enabled).

The SMC-40 is factory set with the following program example:

P 0	Enter program mode.
+ 1001	Move 1001 steps in the plus direction.
W 100	Wait 100 milliseconds.
- 1000	Move 1000 steps in the minus direction.
W 100	Wait 100 milliseconds.
Z 0	Display step position.
G 0 0	Go to location 0 and run stored program.
P	Exit program mode.

Note: Prior to execution, ensure that the analog "joystick" function is disabled. Reference "m" command for more information.

Command	Function	Type	NV Bytes
-	Index in Minus Direction	Immediate, Program	4

Mnemonic	Data 1	Data 2	Result
(Name) – (n)	Steps (0-16,777,215)	None	None

- (Index in Minus Direction)

Same as "+" command, only in the opposite direction.

Command	Function	Type	NV Bytes
@	Soft Stop	Immediate, Program	1

Mnemonic	Data 1	Data 2	Result
(Name) @	None	None	None

@ (Soft Stop)

If moving, decelerate immediately to a stop using ramp parameters. If running a program, when this command is entered, the program will terminate after deceleration. The soft stop may be embedded in a program without causing termination.

An example of this command within a program in conjunction with the Loop on Port command as explained later is:

P 0	Enter program mode.
M 2000	Move at a constant step rate of 2000 SPS.
L0 0	Loop to memory address location 0 until port 1 is low.
@	Decelerate and stop program execution.
P	Exit program mode.

Command	Function	Type	NV Bytes
A	Read/Write to Ports	Immediate, Program	2, 2

Mnemonic	Data 1	Data 2	Result
(Name) A (n)	0-128	None	Port Data

A (Port Read/Write)

Note: The scope of this command is dependant on the product hardware design, ranging between three to six I/O ports.

The SMC-40 (chip) has six general-purpose input/output ports. These ports have internal weak pull-up (to 3.6 volts) resistors. An open drain transistor is available when specified as an output. The number (between 0 and 63), specified with the "A" command, is inverted and written to ports 1-6, i.e., "A0" will place high levels (>3.0 volts) on all ports, defining them as all off.

Defining an output requires writing a low level to the desired port bit. "A8" will turn on port 4 (others will be turned off if they were on. The value represents a binary value, i.e., 8, 16, 24 ("24" represents port 4 and 5).

At reset, all ports are set to the off condition (>3 volts).

The following example program shows how to turn on an output port. Some uses for this could be illuminating an LED to signal a sequence is complete, or to operate a valve.

```
P 0     Enter program mode.
A 8     Turn on port 4.
W 60    Wait 60milliseconds.
A 0     Turn off all ports.
P0      Exit program mode.
```

Different products are designed with varying numbers of available ports. The chosen input and output buffers will define whether a given port is to be used as an input or an output.

If you are designing at a chip level, the input buffer must be open drain/collector in order to prevent output conflicts that would damage the SMC-40. Your design may define all six ports as 6 inputs or 6 outputs or some combination, remembering that some functions use ports 1 through 4 as inputs.

Product	Inputs	Outputs
SMC-40	6*	6*
IBC-400	3	1
IBC-400E	3	3 (2 with encoder option)
*m*STEP-407	3	3

* Mutually exclusive

Command	Function	Type	NV Bytes
C	Clear and Restore NV Memory	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
(Name) C (n)	0-3	None	Version

C (Clear and Restore NV Memory)

This command is used to erase and format non-volatile memory. A specified program memory range will be formatted with 0FF Hex data.

The command is controlled using data 1 as follows:

1. C 0- (non-destructive) Restore last stored defaults to registers (the parameters that were last saved). This also includes analog jog and encoder parameters. This non-destructive command does not write to the NV memory.
2. C 1- Initialize the registers with factory defaults (non destructive). The "S 0" command must be used to save to NV.
3. C 2- This destructive command erases and formats the basic (511 bytes) program memory and deletes all programs from storage (takes about 3 seconds). The "Name" is preserved.
4. C 3 - Erase and format all option card memory (takes 10 or more seconds depending on memory size).

Note: Complete re-initialization is accomplished by writing a zero to NV location 255 using the "\" command. This will force a full erase and reset of defaults during a subsequent reset.

"\255" Set factory defaults
^C Reset

Command	Function	Type	NV Bytes
D	Divide Speeds (default= 4)	Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) D (n)	Divider (1-255)	None	None

D (Divide Speeds)

This user default is savable in NV memory using the "S" command.

All speeds during ramping and slewing are divided by the specified number (n). The pre-scale number may range between 1 and 255. Speeds as low as three revolutions per day may be obtained.

As "n" is increased, other parameters (internal speeds) must be increased to maintain a specified output RPM speed. The specified SPS must be doubled to recover the motor shaft speed. D should not be changed while moving at speeds that require ramping.

This command is generally implemented during the initial customer default parameter assignment. However, it may be implemented and changed within a program. Following, is an example:

P 0 Enter program mode.
D 10 Change the divider to 10.
P Exit program mode.

The "X" command displays the affects of this parameter. The analog joystick option (V1.06 and above) also uses this function.

Command	Function	Type		NV Bytes
E	Settle Time (default=100)			2
	Mnemonic	Data 1	Data 2	Result
	(Name) E (n)	5-255 (time *.01 sec)	None	None

E (Delay to Hold Time)

The SMC-40 moving (MVG) signal is an output. The output polarity may be inverted via the "l" (lower case L) command (savable in NV). Because the meaning of low and high can be changed, the term "active" is used here.

MVG will be active between the start of any motion and go inactive some time (specified with this command) after motion stops. There are two possible uses for this signal:

1. As a moving indicator signal where setting E 0 will remove the delay after stepping. An external controller may poll this signal. More than one axis may be collector (drain) 'or'ed together if the active level is defined as a low voltage out.
2. As a current setback or hold control in situations where power reduction is desired.

The E command specifies the delay between motion stop and automatic disable. Enable and disable of the driver is controlled by the "Moving" output signal. Timer settings are in 10mS increments. Thus, the maximum delay is 2.5 seconds. The default is 1.0 second.

On the last step pulse the count down begins. If another step occurs before the timeout (0-2.5 seconds), the timer is reset.

A value of 255 will prevent shut-off. Setting the delay to zero permits remote controllers to poll moving status immediately. This user default is saved in NV memory using the "S" command.

***m*STEP-407 only**
This signal can be used to control shutdown. If jumper "PG" (refer to Hardware Manual, "Top View" drawing of *m*STEP-407 for location) is removed, then shutdown will be disabled. However, the automatic current setback (configured on driver) will provide a low (33%) standby idle current.

Command	Function	Type		NV Bytes
F	Find Home	Immediate, Program		3
	Mnemonic	Data 1	Data 2	Result
	(Name) F (n, d)	SPS (56-65,000)	Direction (0,1)	None

F (Find Home)

The special Home algorithm is intended to eliminate mechanical hysteresis typically found in many switches, encoders and is generally present in the form of system mechanical backlash.

The SMC-40 implements an intelligent homing algorithm whereby home is always approached from the same direction based on the initial logic state of the Home switch and the value (0 or 1) assigned to the "d" direction byte.

Normally Open Home Switch
The Find Home step velocity, using a normally open Home switch (actuation from logic high to low) is programmable over the entire slew velocity available, from 56-65,000 SPS. Once the Home switch is encountered, the system inertia typically overshoots the exact switch transition point so that the controller changes the direction signal and shifts the step speed down to the (I) initial parameter velocity. This direction reversal and speed reduction continues until the exact Home switch actuation point is reached and the Homing function is complete.

Normally Closed Home Switch
The Find Home step velocity, using a normally closed Home switch (actuation from logic low to high) will always be the (I) initial velocity parameter setting. Once the Home switch is actuated all motion ceases and the Homing function is complete. The following table illustrates the possible combinations of switch motion:

Home Switch	"d" Parameter	Direction of Motion
Normally Open (High to Low)	0	Negative
Normally Closed (Low to High)	0	Positive
Normally Open (High to Low)	1	Positive
Normally Closed (Low to High)	1	Negative

This command may be implemented within a program. Following, is an example:

```
P 0        Enter program mode.
F 1000 1   Find the home switch in the "1" direction at a step rate of 1000 SPS.
P          Exit program mode.
```

Note: Prior to execution, ensure that the analog "joystick" function is disabled. Reference "m" command for more information.

Command	Function	Type	NV Bytes
G	Execute Program	Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) G (a, t)	0-2560	T (0-1)	None

G (Go)

The Go command is used to execute a user programmed sequence starting at location "a." Most programs will start at "0", however, you may wish to start at another address. The address MUST begin at a stored instruction address, i.e., "go to" data produces unpredictable results.

If "t" is a one, the TRACE mode is turned on. A display of the current step being executed is produced while the program is running. The list format is the same as that of the "Q" command. The TRACE mode will be in effect until the program execution terminates or until an embedded "Go" without the trace attribute is encountered.

The address range is 0 to 2560. Address locations between 200 through 255 are reserved for parameter storage and may not be used in programs. The controller is factory set with the following program example:

```
P 0        Enter program mode.
+ 1001     Move 1001 steps in the plus direction.
W 100      Wait 100 milliseconds.
- 1000     Move 1000 steps in the minus direction.
W 100      Wait 100 milliseconds.
Z 0        Display step position.
G 0 0      Go to location 0 and run stored program.
P          Exit program mode.
```

During power up the SMC-40 inspects memory location 192. If a value other then 0FF hex is found, then the commands stored there are executed as a power up sequence. The power-up feature allows the designer to place initialization sequences here, such as a homing routine. Only 8 bytes are available, so it may be necessary to use the "G" or "g" command here.

Caution should be used when loading programs to jump (using "G") over this area to avoid unexpected results at power-up. A reset resulting from the ^C command will also trigger the power-up commands when used. A running program can be aborted (usually) with the ESC command.

Command	Function	Type	NV Bytes
I	Initial Velocity (default 2000)	Default, Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) I (n)	SPS (0, 56 to 65,535)	None	None

I (Initial Velocity)

This user default is savable in NV memory using the "S" command.

This parameter sets the initial velocity in steps per second. This is the first speed used at the beginning of acceleration. It must be slow enough that the motor can start without loosing steps (stalling).

As with all velocity parameters, the initial velocity is divided by the divide factor (D). Using the examine (X) command displays updated velocities. The initial velocity applies to:

1. All index commands (+, -, R).
2. First execute in constant velocity.
3. Decelerate to 0 in constant velocity.
4. Final phase in home command if home speed is above initial velocity.

This command is generally implemented during the initial customer default parameter assignment; however, it may be used and changed within a program.

Following, is an example:

```
P 0      Enter program mode.
I 100    Change the initial velocity to 100 SPS.
P 0      Exit program mode.
```

Special case (Data 1=0)
The setting is displayed. This is especially helpful in party line mode where the "X" command is not usable.

Command	Function	Type	NV Bytes
J	Jump to Address	Program	4

Mnemonic	Data 1	Data 2	Result
(Name) J (a, n)	Address (0-2560)	N + 1 Times 0-255	None

J (Jump to Address a, n+1 times)

This loop command allows repetition of a sequence up to 255 times. The address specified MUST be a valid instruction address, and is usable only within a program. This instruction may NOT be nested (use the lower case "j" to nest a second loop).

This command may be implemented within a program. Following, is an example:

P 0	Enter program mode.
+ 1000	Move in the plus direction 1000 steps.
J 0 3	Go to and run command at location 0, 4 times.
R 0	Return to zero.
j 0 2	Repeat total 9 times (nesting).
P 0	Exit program mode.

Command	Function	Type	NV Bytes
K	Ramp Slopes (Defaults 5/3)	Default, Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) K (n1, n2)	Accel (0-255)	Decel (0-255)	None

K (Ramp Slope)

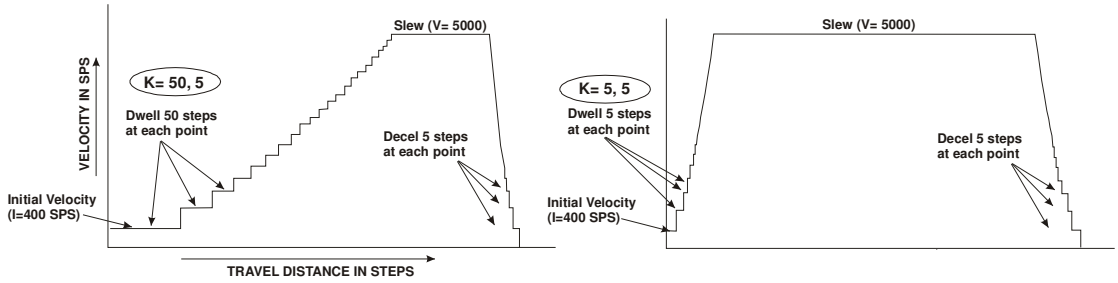
These user defaults are saved in NV memory using the "S" command.

Specify the ramp acceleration and deceleration time. The "K" command is used to adjust the ramp slope during the motor acceleration or deceleration. An internal lookup table defines the profile or shape of the acceleration/deceleration curve. Depending on the values of initial and slew velocities, a number of discrete velocities (risers) are used to define the acceleration or deceleration of the motor armature rotation.

The "K" value determines how many steps are made at each step rate point (riser) on the acceleration curve during ramping. Higher "K" values will increase the dwell time at each discrete point on the acceleration ramp. Lower values of "K" will increase the acceleration rate. A value of 0 will eliminate any ramping.

In practical applications, it is typically easier to decelerate a system, rather than accelerate a system. The separate decelerate parameter feature is a valuable time saver when compared to controllers with fixed acceleration/deceleration times.

The following two examples are of ramped indexes, each 2000 steps with I=400, V=5000, but different "K" values; K50 5 and K5 5:



Note: To modify the ramp slope it is always necessary to enter two (2) data values (from 0 to 255), corresponding to the desired slope for motor acceleration vs. deceleration. The value of "K" can be proportionally changed if the Divide Speed (D command) is increased.

The K command can be issued:

1. As part of a setup.
2. In an application program.
3. As User defined defaults at reset.

This command is generally implemented during the initial customer default parameter assignment. However, it may be implemented and changed within a program.

Following, is an example:

```
P 0        Enter the program mode.
K 100 50   Change the acceleration ramp to 100 and the deceleration ramp to 50.
P          Exit program mode.
```

The analog (joystick) option (V1.06 and above) also uses these settings. See the "X" command to display the values.

Command	Function	Type	NV Bytes
L	Loop on Port	Program	4

Mnemonic	Data 1	Data 2	Result
(Name) L (a, c)	0-2560 (or 511)	Condition (0-9)	None

L (Loop on Port)

Loop on Port will test the specified input port for the required condition (c). If the port is NOT at the required level then the program will jump to the specified address. If the address is to a previous instruction then the program will loop until it becomes the specified level. The program will then continue to the next step. The PLC mode inverts the input high/low definitions.

Input ports are available as follows:

Port	Low	High
1	0	1
2	2	3
3	4	5

The SMC-40 can view all ports as inputs and outputs, restricted by contention with external hardware. Any "output" port can be modified, then subsequently used in conjunction with the L, g, or A128 (read) command.

The SMC-40 has an additional feature of implementing a "wait till" function. This command executes rapidly in program areas below 512 (RAM) while loop tests the condition every 2-3ms when placed in higher addresses. When the condition is met, program execution continues.

This feature is helpful in situations where the condition may be of short duration. This command is usable only in NV memory program execution. Following is an example of this command:

```
P 0        Enter program mode.
L0 4       Stay at location 0 until port 3 is low then go to next command in program.
+ 1000     Move 1000 steps in the plus direction.
P          Exit program mode.
```

Command	Function	Type	NV Bytes
M	Move at Constant Velocity	Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) M	SPS (±56-60,000)	None	None

M (Move at a Constant Velocity)

The "+" or "-" sign determines direction during the move at constant velocity function. The motor will ramp up, or down to a constant velocity. Motion will continue at the given speed until a new velocity is entered. The specified slew speed is in steps per second. Ramp parameters may be modified prior to each velocity command, allowing different ramp slopes. The direction is specified by the sign preceding the velocity. The SMC-40 has the capability of decelerating from full speed in one direction, then accelerating to full speed in the opposite direction with this single command.

Motion may be terminated by:

1. The "M 0" command
2. Soft stop command or interrupt
3. Abort (ESC) interrupt (without deceleration)
4. LimA or LimB becomes active

The default initial velocity is used at the first invocation of the command. The following commands modify effective speeds:

5. (D) Divide
6. (K) Ramp factor

An example of this command within a program, in conjunction with the Loop on Port and Soft Stop commands, is as follows:

```
P 0       Enter program mode.
M 2000    Move at a constant step rate of 2000 SPS.
L0 0      Loop to memory address location 0 until port 1 is low.
@         Decelerate and stop program execution.
P         Exit program mode.
```

Note: Prior to execution, ensure that the analog "joystick" function is disabled. (Reference "m" command for more information).

Command	Function	Type	NV Bytes
N	Read Information	Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) N	0-2	None	None

N (Read Information)

This command returns information pertaining to indexing variables. In terminal (single) mode a complete set of information is listed:

Isps= (sps) Iptr= (pointer) Initial (start) set by "I"
>sps= (sps)>ptr= (pointer) Live (on the fly) values
Isps= (sps) Iptr= (pointer) Slew (run speed) Set by "V" or "M"

This can be useful, particularly in party line mode where the X command is not available to display parameters. As with all party line protocol, only one number will be returned per command.

Command "N 0" (Isps) returns initial velocity (in steps per second) as used in an index, or constant velocity command. It corresponds to the value specified by the "I" command.

Command "N 1" (>sps) returns the current ("live") velocity (in steps per second) when moving. For instance, the speed during the "M" command may be read back.

Command "N 2" (Vsps) returns the slew velocity (in steps per second) as used in an index (+, -, or R commands) only. It corresponds to the value specified by the "V" command.

Command "N3" (Iptr) Returns the starting ramp table pointer. It varies according to the value specified in the "I" command.

Command "N4" (>ptr) Returns the ("live") ramp table pointer. It tracks the acceleration and deceleration at the start and end of an index. If a sudden halt, such as a limit or abort (ESC) occurs, the live values will be undefined, generally at their high-speed values. These are initialized at the beginning of the motion function.

Command "N5" (Vptr) Returns the slew (or running) table pointer. It varies according to the value specified in the "V" command.

One can calculate the number of acceleration "Risers" in ramp, then multiply by the "K" values to determine the total ramp distance. When used with the ramp table, it can be used to predict acceleration (and deceleration) times and distances.

For the current defaults:

Risers = N5-N3 = 32-5 =27
Ramp length (Accel) = risers+1 (one at Initial) * Kup = 28*5=140 steps
Ramp length (Decel) = risers+1 (one at Initial) * Kdn = 28*3=84 steps

Command	Function	Type	NV Bytes
O	Set Origin	Immediate, Program	4

Mnemonic	Data 1	Data 2	Result
(Name) O	Position (±8,388,607)	None	None

O (Set Origin) **See the 'o' (lower case O) for units fitted with encoder feedback**

This command sets the internal 24-bit position counter to the specified value. Zero position for the RELATIVE mode is "0." Signed numbers are used. Hardware reset clears to "0000." The position counter is incremented or decremented for all motion commands. During any index, the position counter is used only for trip value comparison. This counter may be changed without affecting the distance of travel in process.

This command may be implemented within a program. It is very useful when used in conjunction with the Find Home and Relative Positioning commands. Following, is an example:

```
P 0       Enter program mode.
F 1000 1  Find the Home switch in the "1" direction at a step rate of 1000 SPS.
W0        Wait
O         Set origin and counter to 0.
R 1000    Move to position 1000 relative to 0.
P         Exit program mode.
```

Command	Function	Type	NV Bytes
P	Program Mode On/Off	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
(Name) P (a)	Address (0-2560)	None	None, #

P (Program Mode)

The P command is always used in pairs. The first "P" initiates the program mode at the specified address. Once in this mode all commands and data are directed into the NV memory for future execution. Entering the second "P" command will terminate the PROGRAM mode, and then insert an end of program marker (0FFh) in the stored program. The controller will then return to the COMMAND mode.

The program mode may also be terminated with the ESCape character, causing immediate return to the COMMAND mode without inserting the end of program marker. This is useful for editing sections of the program, without requiring that all commands be re-entered.

More than one program may exist at different addresses. These commands can than be executed via the "G (address)" command. There are special address ranges that are assigned to various functions:

Address	Function
0-191	Fast "shadow" RAM
192-194	Power-up routines ('G' command)
200-255	DO NOT USE
256-511	Fast "shadow" RAM
256-511	"g 40000" command
512-2560	Standard NV memory

When programming locations below 512, the commands/data are directed into RAM, where they can be executed. To store this program, you must issue the "S 2" command when you wish to copy into the NV memory.

If programs are located at address 512 and above commands are written directly to the NV memory. During programming, if passing through locations 192 or 199, the SMC-40 will automatically skip to 256 to avoid overwriting data areas.

Note, three operations could contend for location 256. Remember the law "no two objects can occupy the same space" may apply, but then, a crafty programmer may find a way to utilize this "feature."

Some Rules

1. RAM programming – When programming in RAM areas (addresses 0 to 511) the commands enter the RAM where they can be modified or executed during debug or quick tests. You must use the "S 1 " command to save this block into the NV memory where it will be downloaded during a reset and become "permanent" (until another command overwrites it).
2. Option memory programming – when programming to locations 512 and above, the command/data are written directly to the NV memory and special Store commands are not required.

Warning: The "\" (write NV byte) command can write directly to <u>ANY</u> NV location, including program, data, or parameter storage areas.

Command	Function	Type	NV Bytes
Q	List Program	Immediate	3

Mnemonic	Data 1	Data 2	Result
(Name) Q (a)	Address (0-2560)	None	Listing

Q (List Program) (Note: Use in dumb terminal, single line mode).

List program stored in non-volatile memory using the format:

Address Instruction Value 1 Value 2

The values will be displayed only if applicable to the particular instruction type. Twenty instructions are displayed at a time. Use the <CR> key to list up to 20 more commands without pause. ESC quits and any other key single steps the listing.

Command	Function	Type	NV Bytes
R	Index Relative to Origin	Immediate, Program	4

Mnemonic	Data 1	Data 2	Result
(Name) R (n)	Position (±8,388,607)	None	None

R (Index Relative to Origin)

Move, with ramping, relative to the "0" origin. The target position has a range of ±8,388,607 steps from the '0' origin.

The motion sequence is:

1. Wait until any previous motion is finished,
2. Read the current position then calculate the distance to the new target position,
3. Energize the motor winding,
4. Start stepping at the rate of the initial velocity (I),
5. Accelerate using a profile defined by the fixed table that approximates straight-line acceleration and a slope set by the "K" command,
6. The acceleration continues until the slew speed as specified by the "V" command is attained,
7. Motion continues at the slew speed, until the deceleration point is reached,
8. Decelerate (determined by the second "K" value) to a stop completing the index,
9. If another index is not commanded for the settling period, power down the motor (if auto power down is enabled).

This command may be implemented within a program. It is very useful when used in conjunction with the Origin command. Following, is an example:

```
P 0     Enter program mode.
O       Set origin and counter to 0.
R 1000  Move to position 1000 relative to 0.
P       Exit program mode.
```

Note: Prior to execution, ensure that the analog "joystick" and encoder functions are disabled. Reference "m" and "e" commands for more information.

Command	Function	Type	NV Bytes
S	Save Parameters to NV Memory	Immediate	2

Mnemonic	Data 1	Data 2	Result
(Name) S	Type (0,1)	None	None

S (Save)

The SMC-40 uses fast RAM and registers when executing commands. Both parameters and programs are entered into this working memory (below address 512). You must save the data to NV memory if you want to keep it (like saving to disk). Once saved, the data is recalled during power up sequence.

S 0
The parameter area (locations 200 to 255) are saved in the NV memory and will be recalled as defaults during subsequent power-on resets. All of these parameters are saved as a block from the working registers in the SMC-40. Frequent use of this command should be avoided, as memory longevity may be affected.

S 1
This command copies the two program areas from RAM (0 to 199 and 256 through 511) into NV shadow memory for preservation.

Command	Function	Type	NV Bytes
T	Set and Enable Trip Point	Default, Program	5

Mnemonic	Data 1	Data 2	Result
(Name) T (n)	Position (±8,388,607)	Address (0-255)	None

T (Trip Point)

During motion operations, the position counter is continuously updated. If the trip point function is enabled, the position is continuously compared to the programmed trip position. When equality is detected, a trip event will be triggered. If a program is running, a call or "Go Sub" will be made to the specified address between 1 and 255.

Programs located at the specified address can perform almost any function, including turning on/off ports and setting new trip points. A trip point cannot be "reentered" i.e., when executing a trip subroutine and a new trip is set as part of the routine, the new trip cannot be triggered until the end of the first trip routine. Routines located below address 512 will execute faster because of the "Shadow RAM" feature. Trip service routines should not contain index, wait or time consuming instructions.

Disable
To turn off the trip function, use 0 (zero) as the address parameter. The trip is not currently usable in the encoder mode.

The following is an example (all commands are followed by a <CR>):

1. Write program to location 0 (zero):

P0		Enter program mode at address 0.
0	A8	Turn port 4 on.
2	+2000	Rotate motor 2000 steps in the plus direction.
6	P0	Exit program mode.

2. Write program to location 100:

P100		Enter program mode at address 100.
100	A129	Read port states.
102	A0	Turn port 4 off.
104	P0	Exit program mode.

3. Set trip point:
In "dumb terminal" mode, enter T1000 100. This tells the controller to run the program located at address 100 when the step position is 1000.

4. Run program:
Enter the "G" command. Port 4 will turn on and the motor will start moving. When the motor position is at 1000, the program will vector to address 100 and run that sequence. The number 8, signifying port 4, will appear on the screen.

Command	Function	Type	NV Bytes
V	Final (Slew) Velocity (default= 10,000)	Default, Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) V (n)	SPS (56- 65,535)	None	None

V (Set Slew Speed)

This is the maximum speed to be used after acceleration from the initial velocity. The maximum speed will be limited by the motor capability and/or power driver circuitry.

The final output velocity is divided by the value of "D." This value is independent of constant velocity (M) or home (F) speeds and is used when indexing absolute or relative (+, -, R commands).

This command is generally implemented during the initial customer default parameter assignment. However, it may be implemented and changed within a program. Following is an example:

P 0	Enter program mode.
V 10000	Change the slew velocity to 10000 SPS.
P	Exit program mode.

Special case (data 1=0):
The setting is displayed. This is especially helpful in party line mode where the "X" command is not usable.

This user default is savable in NV memory using the "S" command.

Command	Function	Type	NV Bytes
W	Special case "wait while moving"	Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) W 0	10 ms. (0-65,535)	None	None

W (Wait)

Wait until motion is complete. Note; this only applies to "indexed" motion commands (i.e.: W0 does not apply to "M+4000").

Using this command with zero time can provide a method of delaying subsequent command execution until an index is completed.
The following example program makes a move, waits for motion to complete, then turns on an output port. Some uses for this could be illuminating a LED, signaling a sequence is complete or operating a valve.

P 0	Enter program mode.
+ 1000	Index 1000 steps in the plus direction.
W 0	Wait for index to finish.
A 8	Turn on port 4.
W 500	Wait 5 seconds.
A 0	Turn off port.
P 0	Exit program mode.

Command	Function	Type	NV Bytes
X	Examine Settings	Immediate	2

Mnemonic	Data 1	Data 2	Result
X	0-4	None	Display Setting

X (Examine Parameters)

X0 Examine index parameters
The Examine command produces different responses, depending on the mode of operation and data 1. When NOT in the multi-controller mode (non-daisy chain or party line) the display is as follows:

K= 5/3, I= 2001/4, V= 10014/4, E= 100, N=A, Encoder= OFF

Where:
- K= Ramp up/ramp down
- I= Initial velocity / **D**ivider
- V= Slew velocity/ **D**ivider
- E= Settle time (delay before moving signal off)
- N= Controller name

In the multi-controller (party line) mode the data is returned in the following format:
- mm[LF]
- mm= model (26)

X1 Examine option settings
Displays any options that are enabled
Options include:
- (1) Invert limit inputs = "**+InvLim"**
- (2) Auto wait for completion = "**+AutoW"**
- (4) Square wave out (divides sps by 2) = "**+Square"**
- (8) Invert moving output signal = "**+InvMvg"**
- (16) Invert direction output signal = "**+InvDir"**

The options are set as a byte with the l (lower case L) command.

X2 Examine analog parameters
If this option is not present, a "not installed" message is displayed.

The Examine "2" command produces a display of the parameters for the analog jog system as follows:
3(dz= dz), 4(dir), 5(I)=ii, 6(V)=vv, 7(hys)=hh, 8(Mult)=xx

Where:
dz = Dead zone
Dir = (0/1) Set direction used for unidirectional jog
ii= Start speed times xx
vv= Top speed limit times xx
hh= Hysteresis
xx= Multiplier

Refer also to the SMO-40 section.

X 3: Examine encoder settings
If this option is not present, a "not installed" message is displayed.

e=0, h=1600, d=12, r=4, t=30, s=5, v=50, w=0, x=0

Where:
e = Encoder Line count (0 = Off)
h= Driver Full/half or microsteps per revolution
d = Dead Zone (position maintenance)
r = Stall Retries allowed (4)
t = Stall Tolerance (allowed error 30%)
s = Sample interval (every 5 full steps)
v = Hunt speed during position maintenance
w = Timeout for hunt attempt (seconds)
x = Distance limit for hunt attempt (steps)

X4 Mtop
Displays program NV memory size in bytes.

512 or 2559 (with standard external memory)

Command	Function	Type	NV Bytes
Z	Read and Display Current Position	Immediate	2

Mnemonic	Data 1	Data 2	Result
(Name) Z	Readout Mode (0-1)	None	Position

Z (Read Position)

During a motor move command, the value will change depending on the direction of travel. The counter is programmable by the "O" command.

The SMC-40 has the option of continuous readout via the serial interface. This is useful only in single (dumb terminal) mode. The "Z 1" command enables this operation. Any change in position causes the position data to be sent to the serial output. The readout is terminated by a Z0 command. Z1 is not allowed during party line operation. The readout mode will be defaulted as "On" if a SAVE command is issued. This mode is only practical using single controller protocol.

The controller is factory set with the following program example:

P 0	Enter program mode.
+ 1001	Move 1001 steps in the plus direction.
W 100	Wait 100 milliseconds.
- 1000	Move 1000 steps in the minus direction.
W 100	Wait 100 milliseconds.
Z 0	Display step position.
G 0 0	Go to location 0 and run stored program.
P	Exit program mode.

Command	Function	Type	NV Bytes
[	Read NV Memory	Immediate	3

Mnemonic	Data 1	Data 2	Result
(Name) [	Address (0-2560)	Sequential Bytes (0-255)	Displayed Values

[(Read NV Memory)

The user may display any byte of the NV memory. The address specifies the desired location to access. At addresses 0 to 511 the NV memory is <u>always</u> Read (not the RAM).

The data contained at the specified location is output as a decimal value.

Example:

[0 20

The result from this command would be 20 sequential bytes starting at location 0 and finishing at location 19.

Command	Function	Type	NV Bytes
\	Write to NV Memory	Immediate	N/A
	Mnemonic	Data 1	Data 2	Result
	(Name) \ (a, d)	Address (0-2560)	Data (0-255)	None

\ *(Write to NV Memory)*

This command allows the programmer to modify any location in the memory. The command being changed must be done so in decimal format. Special step sequences may be entered, and all initialization constants may be changed. (Reference "Memory Map" in the Addendum section for more details).

The life expectancy of the NV memory may be affected by this command. Addresses 0-511 write directly to both NV memory AND the shadow RAM. This is a very powerful command and care must be taken not to overwrite other needed sections of the nonvolatile program.

This command complements the Read NV Memory ([ ) command.

Command	Function	Type	NV Bytes
]	Read Limits, Hardware	Immediate, Program	2
	Mnemonic	Data 1	Data 2	Result
	(Name)]	0-1	None	Status

] *(Read Limits, Hardware)*

This command allows the user to examine the status of the various switch inputs. The result will contain the real time state of the limit switch inputs in binary values as follows:

Decimal value	128	64	32	16	8	4	2	1
Bit position	7	6	5	4	3	2	1	0
SMC-40	Lb	La	Hm	NA	NA	NA	NA	NA

Where:
LA = Limit "A" switch
LB = Limit "B" switch
Hm= Home switch (32 = low input)

Example:
] 0 192

The result (192) from this command, in decimal format, indicates limits A and B are on.

Command	Function	Type	NV Bytes
^	Read Moving Status	Immediate, Program	1
	Mnemonic	Data 1	Data 2	Result
	(Name) ^	None	None	Status

^ *(Read Moving Status)*

The host may use this command to determine the current moving status that exists within the SMC-40. A non-zero value indicates moving.

The result from this command would be a decimal number. Any number other than 0 indicates the controller is moving. This command is not valid if the analog joystick option is being used.

Command	Function	Type	NV Bytes
g	Branch on ports 1-3	Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) g (n, t)	0	T (0-1)	None

g (lower case G: Special Branch)

The controller will read the input ports 1 through 3, then branch to an address based on the state of input ports 1 through 3. There are eight possible memory addresses that this command may "go to" starting at NV address 256 with spacing of 16 bytes.

P1	P2	P3	Address
0	0	0	256
1	0	0	272
0	1	0	288
1	1	0	304
0	0	1	320
1	0	1	336
0	1	1	352
1	1	1	368

To prevent any confusion to the controller, each address should have a program associated with it even if it is simply "g 0 0" to go back into the "branch to mode." This instruction is analogous to "on PORT go to." The "T" flag is useful for debugging (see the G command).

Command	Function	Type	NV Bytes
i	Restart Special i Trip	Default, Program	5

Mnemonic	Data 1	Data 2	Result
(Name) I (n)	Next Trip Position ±8,388,607	Port (0-63)*	None

i (lower case I; Start & Restart Special Trip) **-Preliminary**

See lower case "k" command. *Actual values are determined by the hardware configuration.
The "special i" command performs several functions

1. Set the output port(s).
2. Sets the "trip" position. A zero will stop special trip.
3. Initializes the tripped program address to 100 or140.
4. Resets the 24 bit position (Z) counter to zero. The encoder (if any) position is not changed.

When the "i" command is part of the triggered function, the above sequence will execute and the position counter is reset to zero.

Note that resetting the position counter does not affect an "in process" index, however the position counter readings will be modified.

The tripped program address is determined by the direction of travel. If travel is in the + direction, the address will be 100. If travel is in the - direction, the address will be 140.

On execution the command "i 180 0"

1. "0" clears the outputs.
2. 180 sets the "+" trip point and sets program pointer to 100.

On execution the command "i -180 0"

1. "0" clears the outputs.
2. -180 sets the "-" trip point and sets program pointer to 140.

Command	Function	Type	NV Bytes
j	Jump to Address	Program	4

Mnemonic	Data 1	Data 2	Result
(Name) J (a, y)	Address (0-2560)	y + 1 Times (0-255)	None

j (lower case J; Jump to Address)

This loop command is identical to the upper case "J" command, but uses a different counter, permitting two-dimensional array indexing, as in the following example:

```
P 0          Enter program mode.
0   + 1000   Move in the plus direction 1000 steps.
4   W 0      Wait until move is complete.
7   -100     Move in the minus direction 100 steps.
11  W 0      Wait until the move is complete.
14  j 7 9    Repeat –100 steps 10 times.
18  J 0 3    Go to and run command at location 0, 4 times.
P            Exit program mode.
```

Command	Function	Type	NV Bytes
k	Special k Next Trip Point, Port Output	Default, Program	5

Mnemonic	Data 1	Data 2	Result
(Name) k (n)	Next Trip Position ±8,388,607	Port (0-63)*	None

k (lower case K; Trip Output Value) **-preliminary**

*Actual values are determined by the available output port configuration.

The latency described in use of the "T" (trip) command can be avoided via use of the "i" and "k" (both lower case) commands. Both of these commands implement a trip mode similar to the T command, but there actions are performed in real time.

The best way to illustrate the power of these commands is with an example such as the design of a scanner.

1. A detector is attached to a slide.
2. An accurate sample strobe needs to be generated every 200 steps in either direction.
3. The slide is indexed 6000 steps in the + (positive) direction.
4. The slide is indexed 6000 steps in the - (negative) direction.

Starter code:
In this example, a starter sequence starts at zero. It may be any available address, but must not conflict with the "triggered" code sequence located at 100 or 140.

```
P 0          Start programming mode.
0   i 190 0  Set outputs off, trip position=190, trip code =100, zero position.
5   + 6000   Index 6000 steps in +direction.
9   W 0      Force wait until index complete.
```

Now do the reverse:

```
12  i –200 0 Set outputs off, trip position=-200, trip code =140, zero position.
17  - 6000   Index 6000 steps in -direction.
21  W 0      Force wait until index complete.
24  P 0      End program.
```

The "real time" code starts here. The first segment starts at trip address 100. The output port will go low (zero volts) at step position 190 and high at step 200. At step 200, the position is reset to zero and trip position is set at 190 again. Thus during motion, pulses will be generated every 200 steps. The pulse width is dependant on the step speed (SPS).

```
P 100          Program for (+) moves.
100   k 200 8  Turn on port 4 and set next trip to step 200.
               The trip address pointer is advanced to 105.
               The step position remains at 190.
105   i 190 0  Turn off port 4 and  set next trip to step 190
               The trip address pointer is reset to 100
               The step position is reset to zero
      P 0      End segment
```

This segment is used when motion is in the negative direction.
We have decided that the pulse width should be very short – about 10 microseconds
The port command ("A") is permitted (no others) to change the outputs with the i or k commands

```
P 140
140   i -200 8  Turn on port 4 and set next trip to step 200.
                The trip address pointer is reset to 140.
                The step position is reset to zero.
                Check for contiguous "A" command.
145   A 0       Turn off port(s). Pulse width will be about 10 uS.
147   P 0       End segment
      S 1       Save programs to NV
      G 0       Execute program.
```

Notes:

1. The physical motor travel will be 6000 steps, even though the position counter has been reset 30 times.
2. The cycle (pulse output) will repeat 30 times in each direction.
3. The position counter ends up at zero.

Encoder equipped models

The special "i" and "k" commands do NOT function with encoder positioning. Encoder feedback should be disabled ("e 0"). Even though the encoder feedback is off, the encoder counter will track position and may be read externally.

Note: Failure to store the program in shadow memory will result in loss of all commands. Once they are stored, they will automatically reload with every reset.

Command	Function	Type	NV Bytes
l	Hardware Options (default =0)	Default, Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) l (a, d)	Option Flags		None

l *(lower case L; Option Flags)*

This command configures several options, primarily relating to input/output operating modes and defining external hardware. Several options invert the sense of input signals.

Flags and Numbers
Several commands use "on-off" flags to enable or disable some feature. The data supplied is in decimal ranging between 0 and 255. The corresponding binary bits are called flags. There are 8 flags, each equaling binary values of 1, 2, 4, 8, 16, 32, 64, 128.

Option Table

Flag	Bit	Mode	Function
1	0	Invert Limits	Both inputs must be held low to allow a move.
2	1	Auto Wait	Wait until index is done before executing next command.
4	2	Square	
8	3	Invert MVG out	
16	4	Invert Direction	
32	5	TS mode	
64	6		
128	7	Read back	Read back status.

An "S" (save) command must be used to preserve the settings. Default with CLEAR =all zero.

Two or more flags can be specified for multiple options, i.e., flag 1 + flag 4= 5. So, command "l 5" will invert limits and use square wave. When reading flags, the host software must convert decimal to binary to evaluate the results.

Limit Polarity (Flag 1)
The input levels on the travel limit sensors are inverted, allowing source type sensors such as hall-effect devices to be used. This command cannot swap the limit directions. When this bit is set, motor travel in either direction is inhibited unless the appropriate limit inputs are forced low. The PLC mode will invert the logic levels (sourcing input necessary to trigger a limit).

Example:
- l 1 Invert the limit software.
- l 5 Invert limits and square wave step.

Auto Wait (Flag 2)
Assume the SMC-40 is idle. When index commands are issued, motion will start immediately and another command can execute-while the index is running. In the auto-wait mode, the index must complete before any new command is processed.

Example:
P 0
+1000
A8
P 0

With Auto Wait off, port 4 will turn on immediately (in a few steps). With Auto Wait on, port 4 will turn on after the 1000 steps.

Example:
P 0
+1000
-1000
P 0

Here the W0 has little affect, because the +1000 index must complete before the –1000 index commences. However, a third command would have to wait until the "-1000" index completes.

Mode (Flag 4) - Square
The SMC-40 normally generates 10us low going step pulse output. This mode has the effect of adding a flip-flop to the output, producing a 50% duty cycle square wave. Like a flip-flop, the step rate will be divided by 2. This feature is useful when interfaced to drivers requiring longer step pulses.

Invert MVG output (Flag 8)
Some drivers require a high rather than low signal to activate an "enable" or setback input.

Invert Direction output (Flag 16)
The direction output signal is inverted. a plus command will cause rotation in the reverse direction, however the position counter will still count up. This is Useful as a quick way to simulate reversing motor wires.

T State Mode (Flag 32)
Higher values result in slower SPS. This allows all speed inputs to be specified in time rather than steps-per-second (SPS). When the SPS is used, integer math is used to determine "time-per-step", this is the actual value used when generating step rates. Using the T-state mode allows a more precise speed specification.

The formula is:
2763000 / SPS, where D (Divider)= 1

Thus, the value to obtain 10,000 SPS is 2763 and the value to obtain 5,000 SPS is 5526. The value 65535 yields the lowest basic speed of 42.16 SPS without using a divider.

Unused Mode (Flag 32) and Mode (Flag 64)
Reserved for future use.

Read Back
The command "l 128" will read and display the option byte in decimal.

Command	Function	Type	NV Bytes
u	Print a character	Immediate, Program	3
	Mnemonic	Data 1	Data 2	Result
	Prn	0-126		

u (lower case U; Print Character)

Echo any ASCII character including control, number, punctuation, A thru Z and a thru z. It is useful for debug operations.

Example; print a <CR>:

```
0    A 129    * Read port
2    u 13     * Print carriage return
4    G 0      *Loop
```

Using zero generates CR and LF.

SMO-40 Analog Control Overview

Bi-Directional Operation

The 8-bit analog to digital converter includes a voltage range of 0 to 5 volts. Assuming a joystick or potentiometer is attached and is centered, the wiper voltage should be 2.5 volts. The voltage can be confirmed using the "9" command ("9 1" allows continuous readout).

If you enter the "m 4" command, the bi-directional mode will be enabled. There will be a dead-band around 2.5 volts, preventing unwanted drift to cause motion. Motion will start in the "plus" direction when the wiper voltage exceeds dead-band. Motion will be in the "minus" direction when the voltage goes below the dead-band. If the measured voltage is near zero the single direction mode is activated.

The input voltage must be above the "dead-band" for motion to begin. The motor direction is controlled by the mode command. Integral ramping prevents motor "stalls" that could be caused by abrupt input changes.

The analog input is digitized to one of 256 values, with 128 corresponding to 2.5 volts. In bi-directional mode, this corresponds to 127 points in clockwise direction or 127 points in counter clockwise direction and 2.5 volts at zero or stopped condition.

Details:

Defaults for this example: (all parameters are savable in NV memory using the "S" command).

Parameter	Command	Default	Formula	Value	Units
Divide	D	4			
Initial	I	2,000	I/D	500	Steps per second
Slope	K	5,3			
Dead Zone	3 (dzo)	15	dzo*20	300	Millivolts (mV)
Start Speed	5 (ssp)	8	ssp*mul/D	8*100/4=200	Steps per second
Top Speed	6 (tsp)	240	tsp*mul/D	240*100/4=6,000	Steps per second
Hysteresis	7 (hys)	5	hys*20	100	Millivolts (mV)
Multiplier	8 (mul)	100			

Assume input voltage is 2.50 (corrected via auto calibrate function or trim input "zero center").

1. When in the dead band (2.5 $\pm$ 0.3 volts) motion is zero.
2. At start voltage (2.8 volts) step speed will be +200 SPS.
3. At start voltage (-2.8 volts) step speed will be -200 SPS.
4. Speed increases with increasing voltage, capped at 6,000 SPS.
5. Hysteresis: voltage must decrease by 100mV (hys) before speed changes.

Uni-directional Operation

Uni-directional operation always rotates in the same direction starting at zero volts input. The procedure is similar to the bi-directional mode, except the input must be at (or near) zero volts for zero speed.

Analog Control Commands

Commands are received and executed by the master processor (SMC-40) like any other command. A high-speed data link communicates with the option co-processor (SMO-40). The SMO-40 performs the following functions:

1. Analog to digital conversion / measurement of input "position" voltage
2. Generation of step rates proportional to input voltage
4. Programmable dead band prevents motion in a stop zone
5. Programmable start speed, 56 SPS minimum
6. Programmable top speed limit 65,000 SPS maximum
7. Programmable multiplier
8. Automatic range and zero algorithms

When the analog (joystick) mode is selected, some special parameters govern speed and functions, while acceleration/deceleration slope (K), and divider (D) are common to all index functions.

Command	Function	Note	Default
"m"	Mode	Enable modes	
"1"	Calibrate	Auto calibrate	
"X 2	Examine	Display analog parameters	
"3"	Dead Zone	Prevent drift*	15
"4"	Direction	Uni-directional mode*	2 (+Dir)
"5"	Start speed	Slowest speed*	8
"6"	Top Speed	Fastest speed*	240
"7"	Hysteresis	Prevent speed dither*	5
"8"	Multiplier	Scale range*	100
"9"	Read A to D	Display input voltage	
<	Jam A to D		
D	Divide	Common to all*	
K	Accel/Decel	Common to all*	

*These user defaults are saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
m	Motion mode (default= 0)	Default, Immediate, Program	2
	Mnemonic	Data 1 / Data 2	Result
	(Name) m (n)	Option	None

m (lower case M; Motion Mode)

This command applies to units that include options such as analog joystick or external step/direction, shuttle encoder or encoder feedback.

Analog Joystick Operation
The analog joystick is first initialized and enabled using the "1" (number 1) command. Once initialized, it may be disabled using the "m0" command (idle) or motion commands ("+", "-", "R", "F", "M") and subsequently re-enabled with an "m4" command.

IMPORTANT – All communication is locked out during motion caused by "Analog Joystick" control. In this instance, the '$'(busy) character is returned. The operator must stop joystick activity (zero speed permits communication). Alternatively, one of the stop functions (ESC,@, or Soft Stop input port) will stop the motion and idle the jog mode.

"n"	Function	Comment
m0	Idle	Encoder and jogs off
m1		N/A
m2		N/A
m3		Uni-directional
m4		Bi-directional
m5		X step (encoder only)
m6		X shuttle (shuttle control)

Command	Function	Type	NV Bytes
1	Auto Calibrate	Immediate	2

Mnemonic	Data 1	Data 2	Result
Calibrate	0-1	N/A	Display Voltage

1 (Auto Calibrate)

The auto calibrate procedure measures the input voltage and automatically sets the mode and zero as follows:

1. If the voltage is less than 0.5 volts, uni-directional range is selected and the input offset is automatically adjusted for zero volts.
2. If the voltage is between 2 volts and 3 volts (2.5 $\pm$ 0.5V), then bi-directional mode is chosen and the auto-zero will correct the offset to read 2.5 volts.
3. The voltage reading and an "out of range" is displayed if neither criteria is met.
4. The input should be as close as possible to the ideal 2.500 or 0.00 voltage prior to auto calibration. Many joysticks feature trim adjustments.
5. Once calibrated, the m command can be used to enable/disable the jog system, without re-calibration.

This function requires the analog input buffer circuitry (operational amplifiers) described above.

Command	Function	Type	NV Bytes
X2	Examine Settings	Immediate	2

Mnemonic	Data 1	Data 2	Result
X	0-4	None	Display Setting

See "X" command in SMC-40 section.

Command	Function	Type	NV Bytes
3	Dead Zone	Immediate	2

Mnemonic	Data 1	Data 2	Result
DZ	1-255	None	None

3 (Dead Zone)

When at standstill the control voltage can be either zero (uni-directional) or 2.5 volts (bi-directional). This parameter defines the threshold voltage where motion begins. It is used to prevent slight changes from causing unwanted motion or drift.

Default is 15 ($\pm$ 300mV). A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
4	Uni-directional jog direction	Immediate	2

Mnemonic	Data 1	Data 2	Result
Dir	0-2 (Default 0)	N/A	None

4 (Unidirectional Direction)

When the analog "jog" is used in the one direction mode, 0 volts is stopped and increasing (positive) voltage increases speed. This command specifies the speed.

Data	Function	Note
0	Use input signal	Uses direction input port located on encoder
1	+Direction	Fixes direction as positive
2	-Direction	Fixes direction as negative

In order to use the external input, the encoder option (SME-40) is required to provide input connection. While any user convention is permissible, AMS defines positive rotation as clockwise when viewed looking at the motor shaft (facing the mounting flange).

This parameter is storable in NV memory as user default

Command	Function	Type	NV Bytes
5	Start Speed (SPS)	Immediate	2

Mnemonic	Data 1	Data 2	Result
Vmin	1-255 (default 8)	None	None

5 (Start Speed)

This specifies the first speed to start moving at when dead zone is exceeded. This number is multiplied by the "mult" value (specified in the "8" command).

The SPS value (mult*V) has a minimum value of 56. Any value less then 56 will be set to 56. The output step rate will be divided by the value specified in the "D" command. The default value will be (8*100)/4 or 200 SPS.

A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
6	Speed Limit	Immediate	6

Mnemonic	Data 1	Data 2	Result
Vmax	Vmin to 255 (default 240)	None	None

6 (Top Speed)

This sets a maximum speed limit for jogging. Increasing input voltage beyond this does not increase the speed. The formula is (Vmax*mul)/D. The default will be (240*100)/4, corresponding to 24000/4 SPS.

A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
7	Hysteresis	Immediate	2

Mnemonic	Data 1	Data 2	Result
Hysteresis	1-255	None	None

7 (Hysteresis)

This function prevents speed dithering that normally occurs in analog to digital conversion. Factors such as noise or shaking of the speed control can cause the "speed" to switch between lookup table values, causing speed dithering. Hysteresis prevents this by requiring a larger voltage change between increasing and decreasing input voltage.

For instance, normally the step speed will change for every 20 millivolt (mV) increase or decrease of the input. If the hysteresis is set to 10, the following operation takes place:

1. When increasing speed, the speed increases for every 20mV. To decrease speed you have to decrease the voltage by 200mV.
2. Once the speed starts decreasing, further decreases occur for every 20mV reduction. It takes a 200mV increase to start increasing speed once again.

A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
8	Multiplier	Immediate	2

Mnemonic	Data 1	Data 2	Result
Mult	1-255 (default 100)	N/A	None

8 (Multiplier)

This parameter multiplies the speeds by a constant. It is possible to obtain a speed in excess of 65,000 steps per second. The multiplier is further modified by the value of "D." Note, this parameter only applies to the speeds produced by the analog input, while the "D" value is global and applies to indexing and constant velocity functions as well.

The multiplier is useful to:

1. Match microstep resolution.
2. Enhance ramp smoothness.
3. Provide desired motor speed range.

A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
9	Read voltage	Immediate	2

Mnemonic	Data 1	Data 2	Result
Volts	0-1	None	Display Voltage: 0-255 or 0.0-5.0V

9 (Read Voltage)

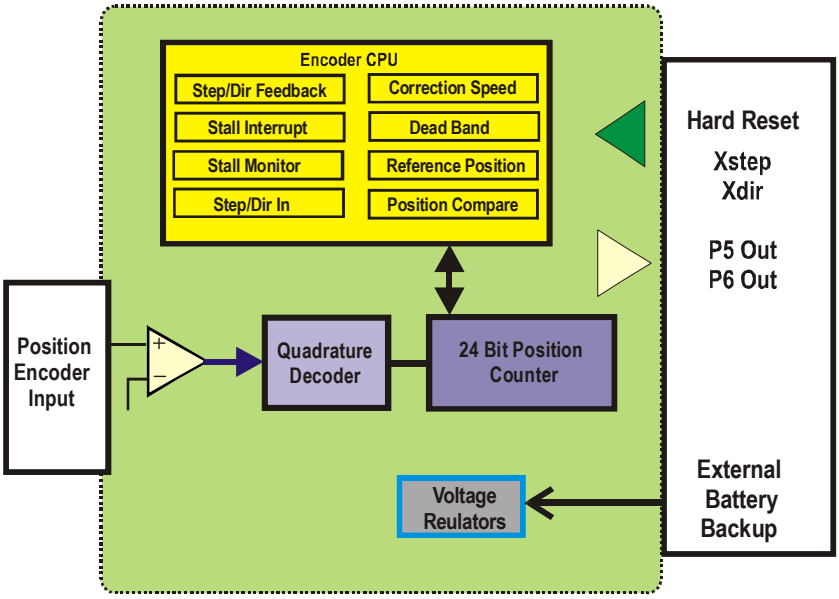
Using this command, the "input" voltage is read. In the party line mode a value between 0 and 255 is returned. In the terminal mode, the 8 bit value is converted to display the 'input" voltage.

The command "9 1" in the terminal mode causes continuous readings to be displayed and is useful for setup and test.

Note: The command "9 1" also disables the joystick function.

SME-40 Encoder Feedback Overview

Encoder feedback will add "servo like" operation to an application. The system is completely digital, without pots to tweak.



ENCODER MODULE

Encoder feedback is composed of the following elements:

1. An incremental encoder with A-B (quadrature outputs). In contrast to earlier AMS designs, most slot (line) count models may be used.
2. An input decoder and counter. The quadrature encoder inputs are filtered and multiplied by an X4 decoder then applied to the 24 bit hardware position counter.
3. An encoder processor- The SME-40 embedded processor reads the position counter, supervising proper operation. It communicates with the master SMC-40 processor via a 2-wire high speed bus.

Encoder feedback uses a supervisory method. The 24 bit binary position is read and actions are executed based on the mode. The first mode is stall detection where encoder motion is monitored while stepping. If the encoder position fails to change, a stall is flagged.

The second function is position maintenance where the SME-40 constantly monitors and maintains the "target" position within a dead band (also known as servo mode).

In the following example, we will assume that the motor is a standard 200 step per revolution motor, the driver is 1/8th micro step and the encoder is 500 lines (slots), resulting in 2,000 encoder counts per revolution.

When the design specifies a "target" position, the position value represents a desired encoder position so the controller must have certain parameters available in order to convert the index distance into motor steps.

A. Encoder counts per revolution= 2000.
B. Micro steps per revolution= 1600 (1/8th micro step = 200*8= 1600).

1. Setup encoder = "e 500" (see description of "e" command)
2. Setup step size = "h1600" (see description of "h" command)

Starting with the position at zero (O 0), a command of "R2000" will compute to 1600 step pulse outputs to the driver and the final encoder position will be 2000.

SME-40 Encoder Feedback Commands

Parameter Summary (Unless specified, all commands are lower case)

Command	Function	Description
X 3	Examine encoder values	Displays these settings
d	Deadband (0-255)	0=Off, 1-255 = "drift" allowed
e	Encoder lines (zero disables)	Pulses per revolution = line count times 4
h	Motor resolution	Number of (micro) steps per revolution
o	Set origin (±8millon)	Force the 24 bit encoder counter
q	Fault functions	Read/reset fault flags
r	Retries (0-255)	On stall detect, a new index is calculated
s	Stall factor	Amount of error allowed before stall detect
t	Test distance	Sample for stall every "n" steps
v	Seek speed	Sets shaft speed for hunt
w	Hunt timeout – sets FAULT	Seconds allowed to complete hunt "cycle"
x	Hunt distance - sets FAULT	Trigger Fault when "n" steps are exceeded
y	Lock at current position	Position maintenance ON

Command	***Function***	***Type***	***NV Bytes***
d	Set Deadband and Enable	Immediate, Program, Default	3

Mnemonic	***Data 1***	***Data 2***	***Result***
(Name) d	0-255 Deadband Size	None	

(d) Set Deadband and Enable

This command specifies the differential position distance, in encoder counts, that the motor shaft is permitted to differ (drift) from the exact position before an automatic correction (hunt) is executed. When a correction is required, the position is re-homed to the desired position. The total deadband value is double the specified distance, thus a value of 10 will maintain the position within ±10 encoder steps.

After completing a move using automatic position correction, further position corrections will be automatic. Full motor power is maintained and the moving output signal is asserted "on." The minimum practical value is affected by encoder resolution, backlash, and hunt step rate/resolutions.

A value of 0 disables this function. A non-zero value activates position maintenance immediately, locking the position to the current encoder position. An abort command (ESC) will shut off the maintenance.

Example:
The encoder position is 1100 and the "current position" (target) register contains 2000.

Note, the current position should be equal (or very close) to the encoder position.

This example assumes that the motor has stalled, without subsequent correction, leaving the two registers out of synchronism. When a "d 15" command is executed, the following actions take place:

1. The encoder position is placed into the "current position" register.
2. The position maintenance mode is turned "on."

The "current position" now equals 1100 and the motor is "locked" into position 1100. The deadband has a tolerance of plus or minus 15 encoder steps from the "current position" register. The encoder position is allowed to wander within this 30-count range. If the encoder position exceeds the specified range, i.e., 1116, an automatic hunt cycle (position correction) is triggered. This will "servo" on the encoder position, adjusting until the encoder is at 1100. Excessive hunt speed (v command) also can cause overshoot – resulting in mechanical oscillations.

Note, the Fault detect function can be used to stop excessive hunting.

If the Hold and Run currents are set to widely different values, a position shift will occur on auto-power down. This may trigger a hunt cycle that will power up the windings. This cycle will continue to repeat, resulting in a periodic shaft oscillation at a low rate. Aside from handling of positions, the "d" and "g" commands perform almost identical functions.

User defaults can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
e	Enable Encoder	Immediate, default	3

Mnemonic	Data 1	Data 2	Result
(Name) e	0-2000 Line Count		

(e) Set Line Count (Enable Encoder)

This command specifies the encoder resolution, in lines, for one revolution of the motor. The controller multiplies this value times 4 for actual encoder position readings. If the encoder is not mounted directly on the motor shaft, the appropriate ratio must be applied, i.e. a 2:1 reduction with a 400 line encoder would require "e 200" as a value.

A 500-line encoder will produce 2000 steps (4*500) per motor revolution. The data is based on a standard 200 step per revolution (1.8°) motor. Non-standard applications must be scaled appropriately.

A value of 0 disables all encoder functions, and subsequent indexing is in motor steps rather than encoder steps. When the encoder is disabled, it is still possible to read the encoder position command via the "z"(or upper case Z) command.

This feature allows verification of microstep and encoder parameters. ***See the Z command for a simple procedure to determine the "e" value.***

Another use is the ability for the host to readout encoder position. Because of the low cost, the SME-40 could be used as an encoder reader only.

A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
h	(Micro) Steps per Revolution (SPR)	Immediate, Program, Default	2

Mnemonic	Data 1	Data 2	Result
(Name) h	0-32,000 (Micro) Steps		

(h) Motor Resolution

The "h" command specifies the number of motor steps for one revolution (SPR) of the motor. Microstep size is determined by the driver design and motor design. Commonly used motors have 200 full-steps (1.8 degrees per step) per revolution.

While encoder feedback prefers microstep drive resolution, full (200 SPR) or half (400 SPR) step may be used.

For the *m*STEP-407, use "h 2000" (for 1/10th step). Reference the "Z" command for a simple procedure to determine the "h" value.

A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
o	Force Encoder Position	Immediate, Program	5

Mnemonic	Data 1	Data 2	Result
(Name) n	±8388607 Position		

(o) Force Encoder Position

This command will "jam" a 24-bit number into the encoder counter. It may be useful in a system with power fail and recovery capabilities.

Command	Function	Type	NV Bytes
q	Read and Reset Encoder Fault	Immediate, Program, Default	2

Mnemonic	Data 1	Data 2	Result
(Name) q	0-1 Read/Reset		Flags

(q) Encoder Fault Read and Reset

Command "q0" will read and return the fault flags binary number, weighted as follows:

Binary	Condition
0	Flags are reset – normal
1	Encoder fault detected
2	Encoder retries exhausted
4	Encoder time or distance (hunt) fault
8	Limit switch latch (with or without encoder feedback)

Products with the encoder feedback feature implement fault detection. When a fault event occurs motion attempts cease and flags are set. If the design includes port 6 as an output, this hardware signal may be used by external equipment.

Two types of encoder faults are possible and they are mutually exclusive; meaning only one may be triggered at once.

Using the "q0" command, a possible encoder fault condition may be detected:

Result = 8 or 10 (Limit switch was hit)
This function is available in all models with or without encoder option. The flag is set if a limit switch input stopped motion.

Result = 3 (Retries exhausted) If the dead band is disabled ("d 0") then the retry count, as specified with the "r n" command, decrements to zero (caused by n stalls or obstructed move) this flag will be set.

Result =5 (Hunt fault) This fault condition is specified by the timeout "w(seconds)" or distance "x(steps)." Hunting is a servo function that is triggered for position maintenance. The designer may specify a time limit or distance to allow before quitting the hunt and notifying the host. If the timeout mode is used, then the timer may be between 1 and 255 seconds. If the distance limit is used, up to 65,535 steps of hunting can be allowed.

Correcting the fault:
Using the "w1" will reset the flags only. Executing an index command (R, +, or -) will reset the flags and execute the index. Entering a dead band value > 0 will reset the flags then trigger a hunt.

Command	Function	Type	NV Bytes
r	Set Retries	Immediate, Program, Default	2

Mnemonic	Data 1	Data 2	Result
(Name) r	0-255 Retry Count		

(r) Set Stall Retry Count

This command will automatically re-attempt to execute a new index if, during the course of an index, a stall condition is detected.

The following steps will be performed:

1. Stop motion.
2. Read encoder position.
3. Automatically execute a new index based on the new computation.

Upon exhaustion of the retry count the controller will still attempt to acquire the desired position if the Hunt (Deadband) feature is enabled. Limit inputs or Abort (ESC) commands will terminate retries. If the Deadband function is not enabled and stalls occur then the control will stop.

A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
s	Sample step count	Immediate, Program, Default	3

Mnemonic	Data 1	Data 2	Result
(Name) s	0-255 Full Steps		

(s) Sampling Distance

As the motor is stepped, the stall detect routine is triggered periodically. This command sets a number of full steps (based on a 1.8 degree motor). Assuming a 200-step/rev motor, a value of 5 will sample the position every 9.0 degrees of rotation.

The encoder processor uses this number to calculate the ideal encoder change that should transpire. For instance, a 500-line encoder will advance 50 counts for every five full motor steps.

In the real world, there will be a tolerance in the actual encoder reading. Factors include encoder accuracy, mechanical backlash, motor variations, and numerous other factors. The 't' command is used to relax the required advance-per-sample.

A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
t	Stall tolerance	Immediate, Program, Default	3

Mnemonic	Data 1	Data 2	Result
(Name) s	0-100%		

(t) Stall Tolerance

This feature is used to determine if the motor has stalled, or slipped poles during a move. The encoder counter is read every "s" number of steps (sampling distance) and verifies that it has changed a minimum count in the specified direction. If 't' was set to 100%, it would require a 50-count minimum (as in the above example), change for each sample. False stall detections will occur.

The value is not critical, but should be below 90%. Designs with remotely mounted encoders (not on motor shaft) will have varying amounts of backlash and may require adjustments of the s, t, and d parameters.

Command	Function	Type	NV Bytes
v	Hunt Velocity	Immediate, Program, Default	3

Mnemonic	Data 1	Data 2	Result
(Name) v	0-255		

(v) Hunt Velocity

This command specifies the hunt step rate to be used during deadband repositioning. The motor shaft speed is based on the resolution, (specified by the "h" parameter). The hunt step rate is designed to maintain a constant motor RPM by adjusting the step rate relative to the steps per revolution specified.

Approximate RPM for values of v:

"v" Speed	1/2 SPS (h=400)	1/8 SPS (h=1600)	RPM
5	52	210	7.9
10	105	420	15.8
100	1000	4000	150
200	2000	7560	283

Notes:

1. Values of "v" below 5 and above 200 will not be proportional.
2. There is a $\pm$ 2.5% tolerance on the actual speeds.
3. The hunt does not permit ramping.
4. The hunt speed setting should be high enough for a quick response while hunting. Caution must be exercised to avoid stalling or rough motion that can be caused by system resonances.
5. The position maintenance is operating in a servo mode so excessive speeds or small dead zones can cause overshoot, resulting in mechanical oscillations (see also dead band).
6. A value of zero disables stall detection.

A user default can be saved in NV memory using the "S" command.

Command	Function	Type	NV Bytes
w	Fault timeout	Immediate, Program, Default	2

Mnemonic	Data 1	Data 2	Result
(Name) w	0-255 seconds		

(w) Hunt Timeout Fault

This command specifies how long to permit a hunt (position maintenance) attempt. A value of 0 disables this function. The designer should set the time to a high enough value to prevent erroneous trips. The timeout will begin at the point where the index phase ends. Any Hunt Distance Fault (x) mode will be disabled. When this is enabled, Port 6 (if implemented) will activate to a low voltage output

Command	Function	Type	NV Bytes
x	Fault timeout	Immediate, Program, Default	3

Mnemonic	Data 1	Data 2	Result
(Name) w	0-65535 steps		

(x) Hunt Distance Fault

This command specifies how long to permit hunt (position maintenance) attempts. A value of 0 disables this function. The designer should set the time to a high enough value to prevent erroneous trips. The timeout will begin at the point where the index phase ends. Any Fault Distance mode will be disabled.

Command	Function	Type	NV Bytes
z	Read Encoder Position	Immediate, Program	1

Mnemonic	Data 1	Data 2	Result
(Name) z	0-1 Display Mode		Position ±8388607

(z) Read Encoder Position

This command will read and display the current encoder position. During motor move commands the value will change depending on the direction of travel. The position counter is reset by the "O" command. There is an option of continuous readout via the serial interface. The "z 1" command enables this operation. Any change in position causes the position data to be sent to the serial output. The readout is terminated by a “Z” command only. The Readout mode will be defaulted “On” if a Save command is issued.

Command	Function	Type	NV Bytes
Z	Read Encoder Position	Immediate, Program	1

Mnemonic	Data 1	Data 2	Result
(Name) z	0-1 Display Mode		Position ±8388607

Z (Upper Case Z) Read Step and Encoder Positions

The upper case Z command displays the current step counter position as well as the encoder position (as described above). This is useful to read both position registers and determining the values to be used for the “h” and “e” commands.

Calibration in Application (or, when all else fails)

Where there are several "variables" that may be unknown or not readily available (including step resolution, picket fence encoders, encoder resolution and/or mechanical "gear" ratio) the following technique may be used to "measure" the unknown parameters.

You must be able to determine when the motor shaft has made one revolution (several full revolutions can be used if the results are scaled appropriately).

After power up and sign-on, in single axis terminal mode, perform these steps:

Step	Command	Remark	Note
1	e 0	Turn off encoder	
2	O 0	Reset position	
2	Z 1	Display both positions	Upper case Z
3	+nnn	Determine steps for full revolution of motor	See "h" command
4		Read and note encoder value (must be positive)	+ yyy
5	R 0	Return to zero position	
6		Repeat steps 3 to 5 to verify	

Enter the values:

"h nnn" – micro steps per revolution
"e yyy/4)" - line count is ¼ readout

All numbers must be integers.

Addendum

Useful Party Line Functions

Note that there is no method available to read the working registers directly. If for instance the value of "D" is changed, it will only be located in the volatile RAM or registers of the microprocessor. An "S" command must be issued to update the EE memory as a block. Once updated the "[" command can be used to read out the byte values (0-255). One and two byte values generally conform to standard byte (8 bits) and integer (16 bits) format.

The data (speeds) for some commands are encoded therefore pointers are not meaningful. There are special commands that permit read-back of decoded speeds and pointers.

These commands use the active registers; hence, they are "real time" readings.

Command	Returns	Defining command
N 0	Initial velocity (SPS)	"I" initial velocity
N1*	Current (Live) velocity (SPS)	M,+,-, and analog motions
N2	Slew velocity (SPS)	"V" Slew Velocity
N3	Starting ramp table pointer	"I" Initial velocity
N4*	Current (Live) Pointer	M,+,-, and analog motions
N5	Slew pointer	"V" Slew Velocity

*The "live" parameters can exceed the slew settings.

Memory Maps

The following locations are accessible through the NV memory read/write commands ([, /):

NV Address	SMC-40 Controller Parameter	NV Address	SMO-40 Analog Parameter
0-191	User program	201	Analog mode
192-198	Power up execution	202	Hysteresis (7)
201-211	See SMO-40, Analog parameters	203	Dead zone (3)
212-222	See SME-40, Encoder parameters	204	Multiplier (8)
224	Configuration byte	205	Reserved
225	Options	206	Direction (4)
226	Divide factor (D)	207	Start Speed (5)
227-228	Initial velocity low and high bytes (I)	208	Top Speed (6)
229-230	Pointer value (I)	209	Reserved
231-232	Slew speed (V) low and high bytes	210	Reserved
233-234	Pointer value (V)	211	Reserved
235	Acceleration ramp factor (K)	**NV Address**	**SME-40 Encoder Parameter**
236	Deceleration ramp factor (K)	212-213	Encoder lines (e)
237-239	Trip point low, mid and high bytes	214-215	Motor SPR (h)
240	Port value for trip (“k” data)	216	Tolerance (t)
241	Settle time	217	Dead band (d)
242	Name	218	Sample Dist (s)
243-254	Reserved	219	Hunt Speed (v)
255	NV initialized flags	220	Retries (r)
256-511	User program or data storage	221	Hunt timeout
256-511[1]	Branch area commands	222	Distance limit
512-2560	External NV memory	223	

[1]Committed only when specific command is being used, otherwise used as general-purpose storage. Locations 200 thru 255 are protected from the “Clear” command.

The location and use of these locations are for information only and are subject to change at any time, without notice. It is intended that parameters be modified using appropriate commands. Most of the data contained in these locations are in binary code and should not be changed.

Default Table

The following default values are written to NV memory after the ‘Clear’(C 1) command:

Parameter	*m*STEP-407	All Others
Initial Velocity (I)	2000	2000
Slew Velocity (V)	10000	10000
Divide Factor (D)	1	4
Ramp Slope (K)	5,3	5,3
Trip Point (T)	0	0
Limit Polarity (l)	0	0
Auto Position Readout (Z)	0	0
Settle time	100	100
Options (see “l” command)	0	8

Command Execution Details

This information is intended to familiarize the programmer with the internal operations involved in executing a command.

For each MOTION command there are four cycles; Entry, Execution, Result, and Completion. Other commands have three cycles; Entry, Execution and Result. In the idle state, the controller continually tests for go or command input. The following describes each operation that takes place on receipt of a command.

Cycle 1: Entry
A. Serial command and data information is placed in a command line buffer as received. Editing is permitted in SINGLE controller mode. ESCape aborts operation and returns to idle state. A carriage RETURN (Line Feed for party line) terminates the entry cycle and initiates execution.

Cycle 2: Execution
The command is processed. In the case of two consecutive action commands, execution will be delayed until any previous completion cycle has been completed.

Cycle 3: Result
The result cycle outputs any numerical result required by the command, i.e., the position. The result type is signed numerical data, preceded by space padding and followed by a Carriage Return and Line Feed. If the result does NOT produce numeric data then the Carriage Return, Line Feed output indicates execution is complete.

Cycle 4: Completion
The completion phase is required for any Action command cycle.

Action Commands

Action Command	Completion Cycle
GO	Until last instruction is complete
Step Resolution	Until previous action complete
Constant Speed	Until previous ramp is complete
Find Home	Until home is found
Relative Move	Until full index is complete
+Step Index	Until full index is complete
- Step Index	Until full index is complete

During the completion cycle (except for “GO”), any non-action command such as “Read Position” may be executed.

The controller has the capability to “queue up” another action command during the completion cycle resulting from a preceding action command. The execution and result cycle of this “Pending” command is delayed until the completion phase is complete. This interval is called the PENDING PERIOD. During this PENDING PERIOD, the only input accepted is the one character interrupt (abort) command, limit switches, soft stop input and hard stop (ESCape).

External indication of PENDING PERIOD end, execution and result cycle of the pending instruction is the carriage RETURN or Line Feed in the party line mode. The GO command is regarded as a command that has a continuous pending (Instructions Queued) period.

Interrupt Commands
Interrupt commands are single character commands that will interrupt the operation in process as follows:

Abort
Any action command may be terminated using the ESCape character.

Process	Resulting Action
Command line input	Clear input buffer.
Program mode	Exit without inserting "END".
Action command	Terminate all motion (HARD STOP).
Program execution	Terminate execution, Hard Stop.

If more then one process is active then ALL are aborted.
Abort is Global – all controller halt.

Soft Stop "@"
The Soft Stop "@" can be either a command (Immediate mode), or a single character interrupt (Program mode). The Soft Stop operates only when motion resulting from action commands or instructions is taking place.

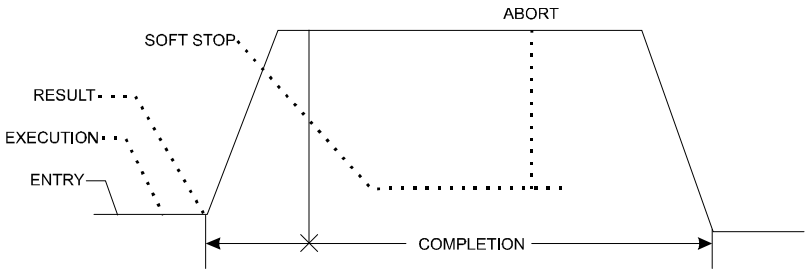
Soft Stop Interrupt
After velocity deceleration, the process is terminated.

Process	Resulting Action
Pending period	Decelerate and cancel pending instruction.
Program execute	Decelerate then terminate execution.

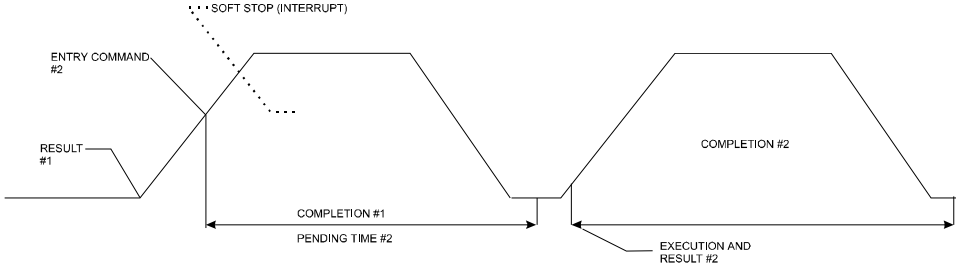
During PENDING PERIODS that are a result of multiple Constant Velocity commands (inter-speed ramping), deceleration will be delayed until the previous ramp-to-speed has been completed.

Homing
Home speed is a special case of the constant velocity command. Homing does NOT employ a deceleration ramp on reaching the home sensor.

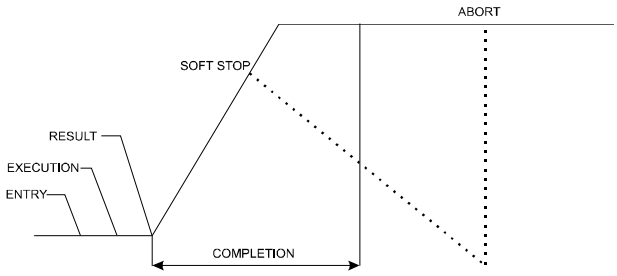
Command Cycle Examples
Index Cycle Resulting From +, -, R Commands



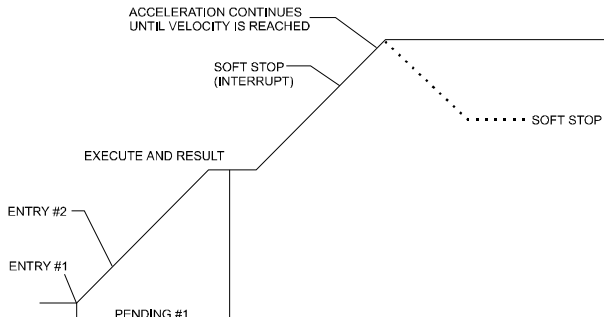
Queued Index Cycle Resulting From +, -, R Commands



Constant Velocity Cycle Resulting From M Command



Constant Velocity Cycle from 2nd M Command



Note: Strange Warning characters may appear (single mode display only) if bad or unknown data is entered.

Example:

Character	Explanation
"<"	A speed lower than the minimum has been entered.
"?"	Unknown or unused command.
"#"	Escape, null line or line overflow.
"E"	NV memory error.

Command Table

(	Unused			Q	List Program	List program	0
)	Unused			R	Absolute Index	Move to absolute position	4
*	Reserved	SIN-10 / SIN-11		S	Store Parameters	Save all	0
+	CW Index (1-8388607)	Step count	4	T	Trip Point	Trigger commands	5
,	Test			U	Reserved		
-	CCW Index (1-8388607)	Step count	4	V	Slew Speed	SPS in index command	3
.	Spare (period)			W	Wait, (delay)	10MS increments	3
/	Repeat	Repeat last command	0	X	Examine Parameters	Show speeds & settings	0
0	Mode	(same as m)		Y	Unused		3
1	Calibrate	AutoCal and Set	2	Z	Display Step Position (0,1)	Readout (continous)	2
2	Unused		2	[	NV Read	Show contents	3
3	Dead Zone (1-255)	Set analog dead band	2	\	NV Write	Write to NV address	0
4	Direction	For uni-dir jog	3	]	Read Limits	Status	2
5	Start Speed (1-255)	Jog initial speed	2	^	Read Moving Status	Non zero = moving	1
6	Top speed (1-255)	Limit highest jog speed	2	'	Copywrite	AMS	
7	Hysterisis (1-255)	Prevent speed dither	2	_	Unused(underscore)	Underscore	
8	Multiplier	Jog speed scale	2	a	Unused		
9	Read A2D	Display joy voltage	2	b	Unused		2
:	Reserved		2	c	Unused		1
;	Get Status		2	d	Encoder Deadband	EFB dead zone	
<	A2D JAM	Reserved	2	e	Encoder Enable	Encoder line count	
=	Unused	Reserved		f	Unused		
>	Unused			g	Branch	Branch on port 1-4	2
?	Reserved			h	Usteps/Rev	Microstep size	3
@	Soft Stop	Decel if moving and stop	1	i	Special Trip Restart		5
A	Port I/O (0-129)	Read/Write user ports	2	j	Jump1 (address, 0-255)	2nd jump counter	4
B	Unused			k	Special Trip Initiate	Position, port	5
C	Erase Memory	Clear NV, reset defaults	0	l	Set Options	Invert lim, etc	2
D	Divide Step Rates	Scale index jog sw,etc	2	m	Mode Set	Analog and encoder	2
E	Settle delay	Current setback delay	2	n	Config	GO options	
F	Find Home	Seek home sensor	3	o	Set encoder position	Origin	5
G	GO (address,1)	Execute user program	3	p	Unused		
			2	q	Fault control	Read/clear encoder faults	
I	Initial SPS (0-65k)	Basic start SPS	3	r	Stall Retries	Encoder feedback	2
J	Jump (address,0-255)	"Go to" n times	4	s	Sample Distance	Sample every 'n' steps	2
K	Ramp Slope	Aceleration, deceleration	3	t	Tolorance (0-100%)	Stall sensitivity	2
L	Loop on Port	Repeat until low/high	4	u	Print Character	u 13= CR	2
M	CV Move	Motion with ramping	3	v	Hunt Speed	Speed for encoder hunting	
N	Show things	SPS, ramp length		w	Hunt timeout (seconds)	Fault after time limit	
O	Set Origin (0 ±8388607)	Set position counter	3	x	Hunt distance (steps)	Fault after "n" steps	
P	Program (address)	Enter/exit programming	0	y	Lock Here (current position)	Servo position hold	
				z	Read Encoder Position		
^C	Software Reset		X	|	Abort command		
^N	Name Axis		X				
^P	Party Line Mode	Enter dumb partyline mode	X	&	SIN-10/11 command	Enter smart party line mode	X
E	Escape character	Abort/Terminate	X				

ASCII Command Table

Ctrl	Char	Dec	Hex	Code		Dec	Hex	Char		Dec	Hex	Char		Dec	Hex	Char
^@		00	00	NUL		32	20			64	40	@		96	60	`
^A	☺	01	01	SOH		33	21	!		65	41	A		97	61	a
^B	☻	02	02	STX		34	22	“		66	42	B		98	62	b
^C	♥	03	03	ETX		35	23	#		67	43	C		99	63	c
^D	♦	04	04	EOT		36	24	$		68	44	D		100	64	d
^E	♣	05	05	ENQ		37	25	%		69	45	E		101	65	e
^F	♠	06	06	ACK		38	26	&		70	46	F		102	66	f
^G	•	07	07	BEL		39	27	‘		71	47	G		103	67	g
^H	◘	08	08	BS		40	28	(		72	48	H		104	68	h
^I	○	09	09	HT		41	29	)		73	49	I		105	69	i
^J	◙	10	0A	LF		42	2A	*		74	4A	J		106	6A	j
^K	♂	11	0B	VT		43	2B	+		75	4B	K		107	6B	k
^L	♀	12	0C	FF		44	2C	,		76	4C	L		108	6C	l
^M	♪	13	0D	CR		45	2D	-		77	4D	M		109	6D	m
^N	♫	14	0E	SO		46	2E	.		78	4E	N		110	6E	n
^O	☼	15	0F	SI		47	2F	/		79	4F	O		111	6F	o
^P	►	16	10	DLE		48	30	0		80	50	P		112	70	p
^Q	◄	17	11	DC1		49	31	1		81	51	Q		113	71	q
^R	↕	18	12	DC2		50	32	2		82	52	R		114	72	r
^S	‼	19	13	DC3		51	33	3		83	53	S		115	73	s
^T	¶	20	14	EC4		52	34	4		84	54	T		116	74	t
^U	§	21	15	NAK		53	35	5		85	55	U		117	75	u
^V	▬	22	16	SYN		54	36	6		86	56	V		118	76	v
^W	↨	23	17	ETB		55	37	7		87	57	W		119	77	w
^X	↑	24	18	CAN		56	38	8		88	58	X		120	78	x
^Y	↓	25	19	EM		57	39	9		89	59	Y		121	79	y
^Z	→	26	1A	SUB		58	3A	:		90	5A	Z		122	7A	z
^[	←	27	1B	ESC		59	3B	;		91	5B	[		123	7B	{
^\	∟	28	1C	FS		60	3C	<		92	5C	\		124	7C	|
^]	↔	29	1D	GS		61	3D	=		93	5D	]		125	7D	}
^^	▲	30	1E	RS		62	3E	>		94	5E	^		126	7E	~
^_	▼	31	1F	US		63	3F	?		95	5F	_		127	7F	