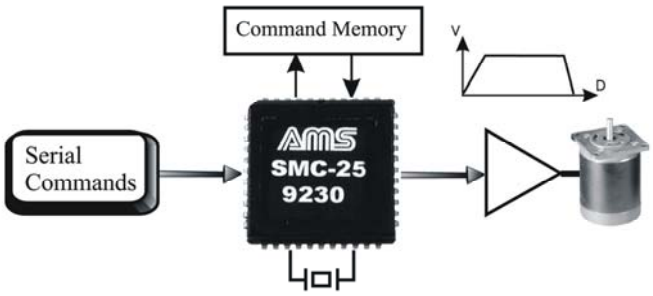
"SMC-25" (V1.09)
STEPPER MOTOR
CONTROL I.C.
MANUAL

SECTION 1: PRODUCT OVERVIEW

The SMC-25 is a microcomputer chip capable of indexing stepping motors. An index sequence consists of:

1. Start at the specified initial velocity,
2. Accelerate to the set slew speed,
3. Decelerate and stop at the specified target position.



It offers built-in phase stepping sequences for Full, Half, and Wave drive modes. A custom user sequence capability provides a means to sequence 8 states with up to four devices.

The SMC-25 has an instruction set of over 30 commands that include loop on port, count delays, and set/clear ports. In addition to six general-purpose ports, jog, limit and home sensor inputs are also available.

Software version 1.09 has enhanced the powerful Trip (T) function and related (i) and (k) commands.

Direct Phase Control Logic Outputs

Four outputs are used to control external phase power drivers. Built-in translator sequences include Full Step, Half Step, and Wave drive. A binary mode produces a 50% duty cycle for remote translators. In addition A custom user table of up to 8 states may be substituted for the built-in choices, making for a very flexible product.

Auxiliary Step Input

This input may be pulsed from an auxiliary step pulse source. Usable in conjunction with the direction I/O port, the SMC-25 acts as a translator with position counter and limit switch functions.

Power-up Program

On reset a special non-volatile (NV) memory address (1600) is tested for instructions to execute. The program can include home sequences or other special operations.

Expanded I/O

Six general purpose input/output ports are available.

Trip Behavior

The trip event allows any value to be sent to the six ports.

Case Sensitive Commands

Commands are case sensitive allowing an expansion of the command set.

High Speed Program Memory

A segment of the program may be stored in these locations when speed critical execution is required. This is useful for:

A. Fast download and executing "macro" sequences
B. Fast execution of more complex trip sequences

Daisy Chain – not recommended

The SMC-25 may be configured to operate in "Daisy Chain" protocol. Several controllers may be connected in series, requiring only a single host computer RS-232 port. Each controller is assigned a unique "Name" as the first action after reset.

Other than a single axis RS-232 application, **Daisy Chain is not the preferred protocol** because of compounding of delays, complex wiring, and system maintenance when more axis are added.

Party Line - preferred mode

The SMC-25 has the ability to be operated in a multiple axis Party Line mode. The architecture is analogous to a mini network. This parallel mode permits full duplex communications with all SMC-25 devices. Listening simultaneously to incoming commands, it is a much faster communication protocol in systems with many axes. Signals to control line driver "output enable" are provided when the requested axis wakes up.

Full Jog Capability

Two directional jog inputs allow jogging in both directions at one of two independent, programmable speeds. A speed select input determines the speed. The readable position counter is continually updated, simplifying teach or alignment applications.

Trip Point

The Trip Point is a programmable position that allows pre-defined operations to be executed when the motor position matches the Trip Point value. A typical application may be to turn on a valve when a desired position is passed.

Bi-directional Ramping

The SMC-25 can ramp either up or down to the specified constant velocities. The ramp slope may be altered prior to changing speeds. Trip Point can be used to trigger velocity changes. The SMC-25 also supports independent deceleration ramping.

Program/Constant Storage

Provision is made to implement up to 2,048 bytes of NV memory. The NV memory allows storage of all speeds, parameters and user programs. The low cost NV memory is connected via a two wire interface. Direct read and write commands allow host use of the memory.

Go Input

A Go input is provided that allows execution of user sequences that have been preloaded into the NV memory. A simple pulse will start the sequence any time. Use of a terminal, host, etc., is not required thus allowing low cost, stand-alone operation.

SECTION 2: HARDWARE

Design Considerations

The SMC-25 requires a minimum of external components for operation. Good design practice with HMOS and CMOS devices, such as the SMC-25, desires buffering or isolation of all input and output signals that have to travel more then a couple of inches, especially off of a circuit board and in noisy environments. Simple local buffering may be achieved through use of logic buffers such as 7404's, 7406's, 7407's or other low cost devices. Noisy environments may dictate use of optical isolation. The low input sink requirements (800ua. max.) of the SMC-25 permits most isolators to directly drive the inputs. Outputs should be buffered to increase L.E.D. drive current.

Required components for all applications
1. Clock (crystal or TTL oscillator)
2. Power on reset
3. Serial interface, RS-232 or RS-422
4. Motor drivers.
5. Buffers for input / output
6. Non-volatile memory 2048 bytes

Components that can increase performance/function
1. Go inputs
2. Jog switches (auxiliary position control)
3. I.O. port buffers: read, write, increment, loop
4. Home and limit switches

Clock

Step rates, timing and Baud rates are derived from an external clock. The clock may be either a crystal or external TTL/CMOS compatible clock applied to X2. If non standard clock frequencies are used then the user must scale step rates and other clock dependent parameters to maintain calibration. Use of a pre-packaged oscillator is recommended for driving several SMC-25's on the same assembly.

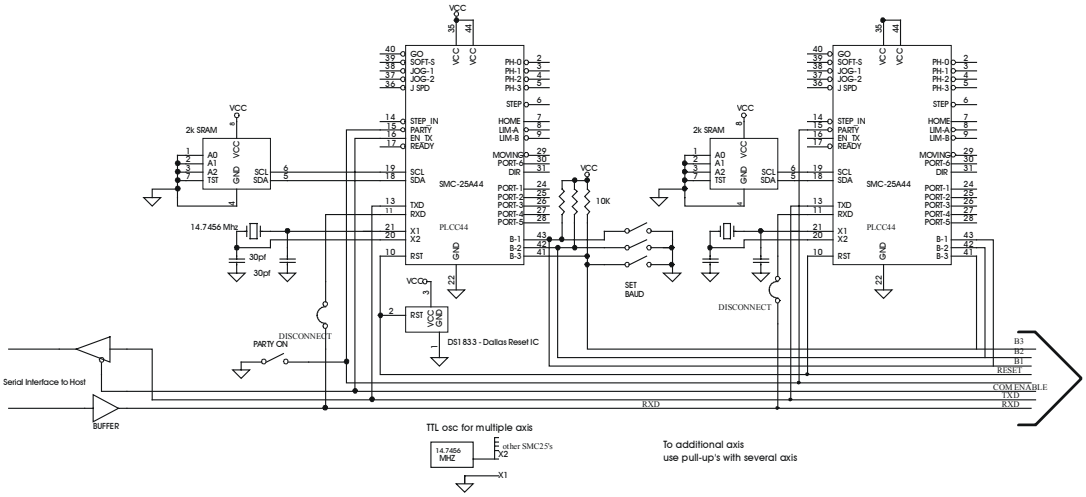
Hardware Reset

Hardware reset is accomplished by placing a high level signal on the reset pin of the SMC-25. The recommended reset circuit is a Dallas Semiconductor DS1833. We have found that critical NV data may be lost under certain power fail conditions, the Dallas Semiconductor Econo Reset IC avoids the data corruption. During reset, all inputs and outputs are at a high state. We define a high (>0.9v) as "off." Only during initialization are the baud rate input pins sampled and set.

The SMC-25 requires the presence of external NV memory. Parameters most recently stored by the "S" command are down-loaded into NV memory. Once initialization is complete the "JOG" and "GO" inputs become activated to allow jogging, or a low pulse on the GO input to execute a program previously stored in NV memory. A terminal or host is NOT required for these functions. If valid instructions are found at NV memory address "1600" they will be executed as a power up sequence. Reception of a "ESC" character is allowed to abort execution if desired.

Serial Interface

All communication to an external computer is done by a two signal TXD and RXD data line with ASCII characters. It is safe to say that the Com port interface must be used either to initialize and load sequences in the beginning or to communicate to a permanent host computer.



Systems with one (1) to fifty (50) axis can be universal. Assume two controllers named X and Y as shown in the above schematic. The open drain outputs are wire or'ed for the transmitted data. It is recommended that not more than 10 axis be connected this way. The TX enables are connected in a wire or'ed fashion to enable a line driver or 3 state buffer. The disconnected jumpers are all removed except for the single axis that you wish to initialize.

Instead of socketed jumpers, SMC-25's could be removed and one inserted into each axis. Another setup procedure would be to place the NV memory in sockets. The individual NV's could be programmed externally then installed. This method reduces the possibility of in-circuit debugging, however.

The SMC-25's may be interfaced to a terminal, host computer, or P.C. with an RS-232 level buffer. Suitable devices include Motorola or equivalent MC1488 and MC1489 (requires dual 12 volt supplies) or Maxim MAX232 (single I.C. 5 volt only)

Secondary logic I/O can provide specialized status bits in user-defined applications.

Motor Interface

The SMC-25 provides for several options of motor control. Four outputs generate the proper logic sequence for step motors. Built in tables permit full step, half step, wave mode or a user sequence is selectable. The external power drivers can be power transistors or FET circuits for low cost unipolar drivers, or power bridges for chopper designs. The User table is stored in NV memory and may be up to 8 states of output values. Possible sequences include 3 phase VR and 4 phase VR motors, as well as solenoid or lamp sequences.

These are the built in table choices:

Sequence	1	2	3	4	5	6	7	8
Full (H0)	1010	1001	0101	0110	1010	1001	0101	0110
Half (H1)	1010	1000	1001	0001	0101	0100	0110	0010
Wave (H2)	1000	0001	0100	0010	1000	0001	0100	0010
Bin. (H3)	0000	1000	1100	1100	0010	1010	0100	1110
Res 0 (H4)	0010	0010	0010	0010	0010	0010	0010	0010
Res 1 (H5)	0011	0011	0011	0011	0011	0011	0011	0011
Res 2 (H6)	1010	1010	1010	1010	1010	1010	1010	1010
Res 3 (H7)	0110	0110	0110	0110	0110	0110	0110	0110

You can enter a custom input table as follows:

NV Address	Content
147	# of sequences 2 thru 8
148	Step 1
149	Step 2
150	Step 3
151	Step 4
152	Step 5
153	Step 6
154	Step 7
155	Step 8

Assume we want a 4 step table with one phase on for each step. Location 147 contains the number of values that follow, in this case 4.

\147 4<cr>

Enter decimal numbers 0 to 15. 0 represents all off (5 volts). At 15 all 4 outputs are on (0 volts).

Step 1 \148 1<cr> Phase 0
Step 2 \149 2<cr> Phase 1
Step 3 \150 4<cr> Phase 2
Step 4 \151 8<cr> Phase 3

The 4 outputs start by loading the value in 148 for step 1 then step 2.

As we step in the plus direction:

Step1 > Step2 > Step3 >Step4 >Step1

In the negative direction:

Step1 >Step4>Step3>Step2>Step1

Each time the table is modified the SMC-25 must have a hard reset. The write to NV memory (\) is direct, and does not require use of the Store command.

Step and direction control outputs generate direction, enable and step pulses for external step and direction translators such as the L297/L298 (SGS). These outputs are rated to sink 1.6 ma with a weak internal pull-up to Vcc.

I/O Ports

Six general-purpose I/O ports are available on the SMC-25. To the SMC-25 they all look the same. They can be written to and be read back. In order to be used as inputs, the open drain portion of the output must be off, as in the power up condition.

An SMC-25 command that contains zeros in the binary mapped value will also set the desired ports "off." The command "A 0" will turn all the outputs to the off (all 1's) or open condition. Usually external hardware defines the port direction.

Most Advanced Micro Systems products buffer ports 1, 2 and 3 as inputs, port 4 as input and output, and port 5 and 6 as outputs. Resistors should be inserted on "inputs" driven by totem-pole drivers to prevent excessive current should the port be turned on as an output conflict.

The SMC-25 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, G2048, or A129 (read) command.

SECTION 3: COMMAND OVERVIEW

Serial Interface

Baud rates of 1200 through 38.4K are selectable by asserting combinations of pins B1, B2, and B3 to logic lows. Baud rates are for a clock frequency of 14.7456 MHz.

Baud	B1	B2	B3
9600	1	1	1
4800	0	1	1
2400	1	0	1
19.2k	0	0	1
1200	1	1	0
38.4k	0	1	0
9600	1	0	0
470k	0	0	0 (Special inter-processor mode)

(AMS' SIN-10, smart serial adapter supports 470k mode).

A "0" indicates a low or ground connection, a "1" represents a high or pull-up to +5v.

Serial data is 8 bits -1 start bit, 1 stop bit, and no parity.

IMPORTANCE OF HANDSHAKE

The SMC-25 is a single chip microprocessor, incorporating a buffered UART input, capable of receiving and holding one character at a time. The microprocessor must read this character before another one is received, otherwise the Uart will be over-run, resulting in errors. The handshake method used is a simple "echo" of the received character. The host computer MUST ALWAYS wait for the echo. Fixes such as insertion of delays between characters may seem to work but will eventually fail.

Beware that many PLC manufacturers do not provide the serial software flexibility or are willing (or able) to furnish adequate information so that your application can do the proper communication. Example host software is included in the EASI software (PC) supplied on request.

To simplify the entire process, AMS' SIN-10 smart serial adapter will perform the handshake function as well as relieve many software requirements from your computer.

Single Axis

The Party Line mode must be a high (open circuit, pull up resistor optional).

The start sequence is initiated by hitting the space bar. The result is a pre-defined sign on message:

dddd Advanced Micro Systems vx.xx

Where:
dddd =Software compile date
x.xx = Version number

Different compile dates do not necessarily mean that the version or software has changed.

Entering Commands (Single Mode)

Command lines consist of an ASCII character followed by 0, 1, or 2 decimal ASCII numbers depending on command requirements. The user may edit the line, prior to entry by using either the BACKSPACE or DELete key. The command line may be up to 12 characters long, including spaces. Spaces are optional between the command character and first number. Commands with two numbers require at least one space between numbers. Command characters may be either upper or lower case. In the COMMAND mode, the command is executed upon receipt of a carriage RETURN. The SMC-25 will respond with a carriage RETURN, Line Feed on acceptance of the command.

Input Command Example

Command	Comment
+ 1000 (CR)	Step 1000 steps in + direction
+1000 (CR)	Same as #1
E0 (CR)	Disable motor current
E0 (CR)	Same as #3
E (CR)	Same as #3 (0 is used by default)
R -1000 (CR)	Move to position -1000

Certain commands such as jump and loop instructions are only valid when used in the "PROGRAM THEN EXECUTE" mode.

The following can only be executed from programs stored in optional NV memory:

J 0 5 (CR) Jump to location 0, 6 (n + 1) time
J0 5 (CR) Same as above

Some commands result in a numerical display. These consist of whole numbers that may have preceding spaces and are followed by a RETURN and Line Feed character. Negative numbers are preceded by the minus "-" sign.

Daisy Chain Start-Up (not recommended)

Daisy chain mode is useful with a single axis or multiple axes where communication delays can be tolerated. The advantage of daisy chain mode is that the "Name" is assigned by the host each time it is initialized, independent of any NV memory stored "Name."

Multiple SMC-25's can be interconnected directly (Daisy Chained) using the controller logic levels or buffered with line driver-receivers. All Baud settings MUST be the same. Characters are received by the first controller input, then echoed to the next controller in the serial link. The host terminal/computer receives characters from the last SMC-25 in the link. The closed loop communications assures the integrity of data.

Note: The Party Line mode input pin must be at a high (open circuit, pull up resistor optional).

The initial input sequence MUST be a Line Feed (LF) character followed by an ASCII "Name" character. The first controller will absorb the Name, then output the Line Feed character followed by the next higher ASCII character in the character set. This sequence continues until all SMC-25 controllers have assigned Names.

If the first valid usable "Name" character is an upper case "A" the controller will be assigned the prefix Name "A" then output a "B" to the next SMC-25 controller in line. Thus four Daisy Chained SMC-25's would assume the Names of "A", "B", "C" and "D".

Here, sign on messages are NOT generated. The last controller will respond with a Line Feed and character representing the last Name plus one ("E" in this example).

CAUTION should be exercised in selection of Names to avoid the generation of unprintable characters and to improve host source code readability.

Valid Names include:

1. Upper case A through Z
2. ASCII HEX
   [5Bh
   \ 5Ch
   ] 5Dh
   ^ 5Eh
   - 5Fh
   \ 60h
3. Lower case a through z

Numerical characters 0 thru 9 are also permissible.

Daisy Chain Line Input Commands

Some Rules:

1. The first character of a command MUST be the "Name" character assigned to the appropriate axis.
2. The command line terminator MUST be a Line Feed character
3. The Name must be preceded by a LF (presumably the terminator for the previous command), i.e., <LF>"n" xxxxxxx <LF>.

Note: An LF is generated using a Ctrl–Enter key combination on a PC.

The first Line Feed "resets" the command buffer for all units. The SMC-25 controller then tests the character immediately following a Line Feed. If this character matches the assigned "Name," the controller will interpret the following characters (up to 12) as an input command. If the controllers do NOT detect a proper Name and command, then the data is simply echoed back to the terminal. The designated controller re-issues the Line Feed after processing the command.

If the command is of the type that results in a data output (such as "Z"), then the data (result) will be inserted before the Line Feed. The Line Feed does NOT indicate that a move or other time consuming command is finished but only initiated. The terminal can interrogate the motion status using the appropriate command to determine if a function is complete. Editing features are NOT supported in Daisy Chain or Party Line operation.

Note: In "Daisy Chain" the commands "Control C" and "ESCape" do NOT require the use of, and will NOT be qualified by, a "Name" prefix. All devices in the chain will respond.

Daisy Chain Command Example

The following example assumes two controllers are serially connected with Name assignments of "X" and "Y". The characters shown in parentheses are echoed back to the host after passing through all of the SMC-25 controllers:

Index 1000 steps for axis X
Input from Host: X + 1 0 0 0 (LF)
Response from Named SMC-25: X + 1 0 0 0 (LF)

Index 500 steps for axis Y
Input from Host: Y - 5 0 0 (LF)
Response from Named SMC-25: Y - 5 0 0 (LF)

Read position
Input from Host: X Z (LF)
Response from Named SMC-25: X Z 1 0 0 0 (LF)

Note: Response is the position data requested from axis X.
The hand-shake must be character by character confirmation.

Example, the +1000 command
Host sends "X", host waits for "X" echo
Host sends "+", host waits for "+" echo
Host sends "1", host waits for "1" echo
Host sends "0", host waits for "0" echo
Host sends "0", host waits for "0" echo
Host sends "0", host waits for "0" echo
Host sends "LF", host waits for "L" echo

Example, Read Position
Host sends "X", host waits for "X" echo
Host sends "Z", host waits for "Z" echo
Host sends "LF", host waits for "LF" echo

While waiting for the LF the host receives the "1000" data and stores it into the position value.

Party Line (preferred)

The SMC-25 Party Line protocol should be implemented in systems where one or more axis is used. NV memory is used for Name storage. Each SMC-25 receiver monitors the host, and "wakes-up" on receiving a matching Name character.

The "Name" must be set and stored one axis at a tine, using the single line-dumb terminal mode. Once stored in NV memory it is protected from accidental erasure by most commands (accept direct NV memory write).

Naming Axis

Use a terminal or computer with standard, simple terminal software. The serial communication should be full duplex and allow control characters.

Each axis is assigned a unique Name by following a specific sequence:

1. A single controller is used.
2. The Party Line mode input pin must be at a high (open circuit, pull up resistor optional).

3. Reset the SMC-25
   a) Cycle power.
   b) Apply a high to the reset pin.
   c) Send a control-C character.
4. Enter the "Name" character: 0-9, A-Z, a-z.
5. Enter a "space" character – the SMC-25 will sign on.
6. Verify by entering the "X" command. The last n= value should show the Name.
7. Enter the "S" command.
8. Done. The Name is now stored in the NV memory.

Party Line Start-Up

After the Name is assigned in NV memory, the Party Line mode pin is set to a low level. Each time the controller is reset the Name is read from NV memory and used for communication.

When the SMC-25 is reset (power on) with the Party Line protocol enabled (low on Party Line input), the controller will operate in Party Line mode.

Party Line Commands

Party Line operation is essentially the same as Serial Daisy Chain operation except that no characters will be echoed to the host until the proper "Name" (proceeded by a line feed) is detected. All axes concurrently monitor the common TXD line from the host. Once the Name is received, the target axis will wake-up and start echoing as described above. The awakened axis will assert a low output on the enable TXD pin until the terminating line feed is re-transmitted. The enable output is used to enable tri-state buffers or line driver outputs.

Instruction Execution

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 SMC-25 continually tests for jog, 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 axis mode. ESCape aborts operation and returns to idle state. A carriage RETURN (Line Feed for Daisy Chain) 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, proceeded 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. The READY* (pin 15) output is available for hand-shake operation. READY* becomes false (high) on receipt of the command and remains high through the result cycle of the command.

Cycle 4: Completion
The completion phase is required for any "ACTION" command cycle. The following are action commands:

Action Command	Completion Cycle
Go	Until last instruction is complete
Wait	Until any 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 SMC-25 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 command.

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.

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.

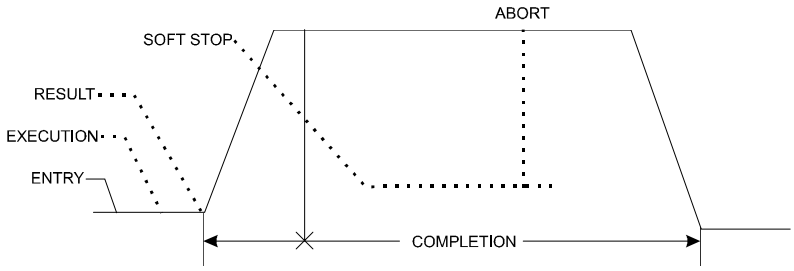
Jog Speeds, Homing

Jog input and home speed is a special case of the constant velocity command. Inter-speed ramping is used if the programmed jog speeds are above the initial velocity. Homing does NOT employ a deceleration ramp on reaching the home sensor.

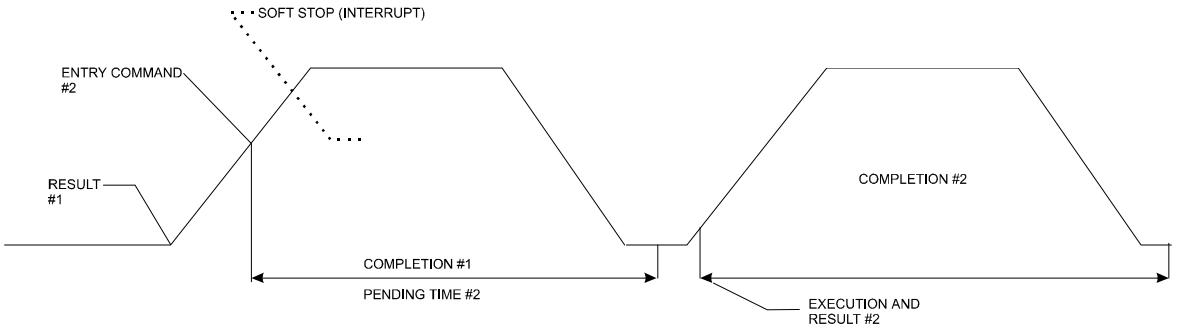
Note: In any mode, jogging and command reception are mutually exclusive. That is, a command canNOT be loaded while jogging and jogging canNOT be performed until the last command is complete. A command starts with the reception of the first command character.

Command Cycle Examples

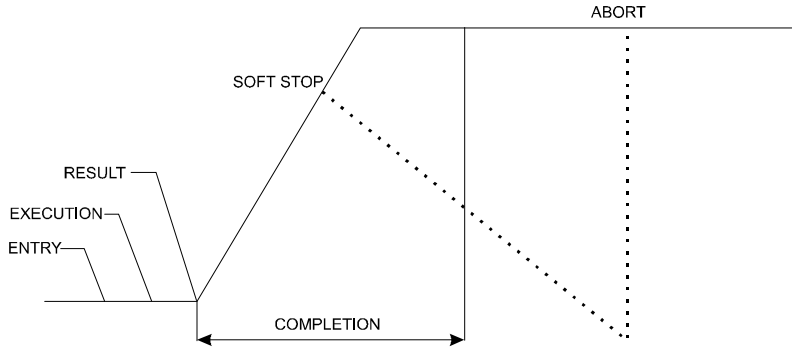
Index Cycle Resulting From +, -, R Commands



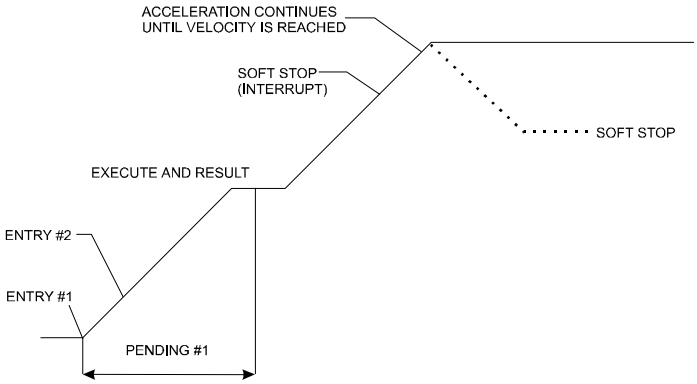
Queued Index Cycle Resulting From +, -, R Commands



Constant Velocity Cycle Resulting From M Command



Constant Velocity Cycle From 2nd M Command



Execution Times

The time for a complete cycle between command entry and result is variable, depending on number of data bytes, command type, and motion in process. The following times may be used as a start point for determining time requirements. Most commands execute in under 1 millisecond, the exceptions are:

Instruction	Execute Time
I,V,	3-4 ms
C0 (Reset defaults)	60
C (Clear memory block)	1500MS
S (Store)	60
/,] (Read, Write)	1.1ms

The timing requirements, in PROGRAM mode, where commands are directed to non-volatile memory is 0.1 + 1 Ms per byte.

SECTION 4: NON-VOLATILE MEMORY

Design Considerations

The NV memory may be used to store user programs for future execution via the "Go" command. Any number of programs may co-exist, limited only by the available memory. The NV memory stores power-up defaults, user programs and phase look-up tables. The NV memory is used to perform the following commands:

Command	Command
G: (Go from address)	J: (Jump)
L: (Loop)	P: (Program)
Q: (Query program)	S: (Store)
[: (Read NV Ram)	\: (Write NV Ram)
S:Save Party Line Names	

Memory Map

The following locations are accessible through the NV memory read/write commands:

Decimal	Description
0-127	User program or data storage
128-191	Shadow program area
160[1]	Trip routine
192-226	Unused
227	Configuration byte
228	Internal initial status byte (Do NOT modify)
229	Divide factor (D)
230-1	Initial velocity low and high bytes (I)
232-3	Pointer value (I)
234-5	Slew speed (V) low and high bytes
236-7	Pointer value (V)
238	Low speed jog value (B)
239	High-speed jog value (B)
240	Acceleration ramp factor (K)
241	Deceleration ramp factor (K)
242-244	Trip Point low, mid and high bytes
245	Port value for trip ("k" data)
246	Resolution
247	Name
248	
256-2047	User program or data storage
256-511[1]	Branch area power up commands
1600-2047[2]	User program power up commands

[1]Committed only when specific command is being used, otherwise used as general-purpose storage. Locations 247 thru 255 are protected from the "Clear" command. Most of the data contained in these locations is in binary and should not be tinkered with.

[2]If a valid command exists at location 1600 through 2047 it will be executed on power up.

Initializing NV Memory

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

Parameter	Value
Initial Velocity (I)	400 SPS
Slew Velocity (V)	5,016 SPS
Divide Factor (D)	1
Ramp Slope (K)	5
Jog Speeds (B)	90/600
Trip Point (T)	Off
Mode (H)	Half step
Auto Power Down (E)	Yes
Limit Polarity (H)	Low
Auto Position Readout (Z)	Off
Name (after reset)	Undefined
User Programs (0-191)	Cleared

The SMC-25 uses the X24C16, a 2048 byte EEPROM. A worst case of 4 bytes per instruction yields a capacity of 500 commands. These devices are rated to retain data for 100 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 the dielectric becomes less effective and the cell cannot retain its charge. The write life cycle endurance rating is constantly being improved. At this time a life in excess of 1 million cycles is available.

To extend the life of the EEPROM in your device it is necessary to be aware of which commands of the SMC-25 perform writes to the EEPROM, and eliminate those which are not needed. For example, the RESTORE command ("C 0") will retrieve the parameters from the EEPROM without doing a write. If the INITIALIZE command ("C 1") was chosen, the first 256 BYTES of EEPROM are written. If you require a sequence of motions to be done without host attention, break-up the motions into sub-groups rather than repeatedly programming the EEPROM. Then use the GO from address command to execute the sub-groups in the required sequence.

Note: Use the SAVE command sparingly. The SMC-25 parameters are set so quickly, even in SERIAL mode, that you should let the host down-load them.

Changing parameters should NOT be done by writing directly to EEPROM. The SMC-25 will not recognize that it was changed and may over-write them. Use the commands available to set parameters. Reading on the other hand is non-taxing on the EEPROM. The DIVIDE factor is readable at 229 (0E5 hex).

Trying to read and write Initial and Slew velocities from the EEPROM will be confusing, as they are stored as timer reload values. Use the EXAMINE command ("X") in SERIAL mode.

Turbo Ram

The SMC-25 has a small dedicated memory area called Turbo Ram. This 64 bytes resides between address location 128 and 192. Instructions written here during program mode use "real" internal RAM rather than EEPROM in order to achieve these advantages:

1. Very fast execution. EEPROM access time is 1 Ms. or more per byte.
2. No wear and tear on the EEPROM.
3. The trip service routine executes at address 160.

Macros may be downloaded directly into this area and executed as frequently as desired. Programs in this area are stored in corresponding NV memory and "down-loaded" at power up, making an effective shadow RAM.

Command behavior between address locations128 and 192	
Q	List from RAM
P	Program to RAM
S	Copy to EEPROM
]	Write to EEPROM
\	Read from EEPROM
C1	Clear EEPROM, reload register

SECTION 5: PROGRAMMING

Instructions

In the COMMAND mode, commands are normally executed as soon as they are entered. The use of non-volatile memory allows storage of a list of commands. These stored program(s) can be triggered at power up, by command or strobe, for automatic or repetitive operations. When in the PROGRAM mode, the entered commands (now called instructions) are directed into the user implemented NV memory. After leaving PROGRAM mode, the stored program(s) may be subsequently executed by entering the GO command.

Example:
The PROGRAM mode is initiated by entering "P aa" (CR). The desired start address "aa" is chosen by the user. Generally address 0 is a good choice for the main program because a program located at address "0" can be started with a simple "G" (CR) or strobing the "GO" input pin.

Once in the PROGRAM mode the memory address is displayed on the terminal and instructions are entered directly to NV memory (or RAM for addresses between 128 and 191). As each instruction is entered, the address is displayed. All instructions have the same format as in the COMMAND mode.

Terminating the PROGRAM mode is done by entering a second "P" command. This will cause the end of program flag (0ffh) to be inserted and the SMC-25 will echo the pound (#) character. The SMC-25 will now be back in the COMMAND mode.

Several programs may co-exist in memory. Each program may be executed independently by issuing a "GO" command with the appropriate address. The length and quantity of programs is limited by the available memory space. (Reference "Memory Map" in the NON-VOLATILE MEMORY Chapter). Remember, the end of program indicator occupies one additional byte. A program sequence that will be "called" when a Trip Point is passed may be located at address 160.

Editing Programs

Existing program(s) may be modified, as desired, at any time. The user can review the existing instructions by entering the "Q" command. This command produces a list of instructions along with their memory addresses. To edit an existing program, enter "P" along with the desired address and proceed to enter the new instruction(s) as in the PROGRAM mode.

The edit session may be terminated in two ways. If the edit results in a program that is longer then the previous program or if the user wishes to discard the old instructions (shorten program), enter "P" to terminate edit and cause an end of program marker to be inserted. If only one or several successive new instructions are to be altered, then entering "ESC"ape will terminate the edit. Any instructions outside of the edit area will NOT be altered.

Note: If any instructions are of different byte lengths than existed previously, the program could wind up with invalid instructions in the middle of the program. Keeping track of the byte count will avoid this condition. The user may insert redundant or "dummy" one byte instructions to fill the gap. If in doubt re-enter the remaining portion of the program.

SECTION 6: COMMANDS

Command Format Description

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 axis present Default = Initial parameter setting Hardware = Auxiliary I/O
NV Bytes:	Storage requirements in program
Mnemonic:	Single character prefix used in multi-axis protocol; (prefixed by axis "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: Two parameters are separated by a comma.

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)

Terminate any active operation and cause the controller to revert to the idle state waiting for a new command. Output drivers or 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.

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.

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

^C (Reset)

Resets controller to power-up condition, waiting for start sequence. 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.

Command	Function	Type	NV Bytes
A	Read/Write to Ports	Immediate, Program	2, 2
	Mnemonic	Data 1	Data 2	Result
	(Name) A (n)	0-129	None	Port Data

A (Port Read/Write)

Input data ranging between 0 and 63 is complemented then output to port 1 through port 6. Port 1 is the least significant bit. Binary combinations of bits will turn on more then one port. Example "A 7" will set ports 1, 2 and 3 to an ON condition. At hardware reset all outputs are set off (high). The command "A 128" will cause ports 1 through 5 to increment in a binary fashion. The command "A 129" will read and display the port data.

Port	Data
1	1
2	2
3	4
4	8
5	16
6	32
129	Read Port

Reading the port data provides the following result information:

Data	Cause	Data	Cause
1	Low input present on port 1	16	Low input present on port 5
2	Low input present on port 2	32	Low input present on port 6
4	Low input present on port 3	64	
8	Low input present on port 4	128	

Command	Function	Type	NV Bytes
B	Set Jog Speeds	Default, Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) B (n1, n2)	Slow Speed (0-255)	High Speed (0-255)	None

B (Set Jog Speeds)

These two numbers represent the speeds to use for jog inputs. The first is usually a lower speed. The second number is used when the high-speed jog (pin 32) is held low. The values are multiplied by 30 to determine the actual step rate in steps per second. Setting values of 0 will disable the jog. Speeds are divided by the "D" value. The power-up settings are stored in NV memory.

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

Mnemonic	Data 1	Data 2	Result
(Name) C (n)	0-8	None	Version

C (Clear and Restore NV Memory)

Previously stored programs are erased. Using a 1 forces complete NV memory initialization with factory default values with erasure of all previously stored programs. This MUST be done when new NV memory is installed or existing memory is corrupted. Frequent use of this command should be avoided, as memory longevity may be affected.

The "C 0" command simply reads the last stored values into the working registers.

Command	Page Cleared
C 1	(0, reset defaults)
C 2	2
C 3	3
C 4	4
C 5	5
C 6	6
C 7	7
C 8	8

Command	Function	Type	NV Bytes
D	Divide Speeds	Immediate, Program	2

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

D (Divide Speeds)

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 4 1/2 steps per minute may be obtained. As "n" is increased, other parameters (internal speeds) must be increased to obtain a given output step speed. Using a value of 2 or 3 may be helpful in producing smoother acceleration characteristics at lower speeds. D should not be changed while moving. The power-up settings are stored in NV memory.

Command	Function	Type	NV Bytes
E	Enable Control	Default, Immediate, Program	2

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

E (Enable Control)

The phase outputs are defined as off when at a "high" (5v) such as during power up reset. The enable output is also at a high to turn the outputs off (or reduce current to a holding value). The binary data values control the behavior of these modes and allow a programmable

Bit 0. On receipt of any "Move" command, enable output(s) are automatically energized while stepping. When stepping is complete, a settling period equivalent to 255 steps at the initial velocity is provided before the enable output(s) are automatically deenergized. If enable is set to a "1" then the enable output(s) will always be energized.

Bit 1. Invert Limit sense. The input levels on the travel Limit sensors are inverted, allowing source type sensors such as hall-effect devices to be used.

The following is a table of values for programming auto phase power down and limit switch polarity:

"n" Value	Auto Power Down	Limit Switch (active)
0	Yes	Low Input
1	No	Low Input
2	Yes	High Input
3	No	High Input

Default = 0
The power-up settings are stored in NV memory.

Command	Function	Type	NV Bytes
F	Find Home	Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) F (n,d)	SPS (19-25,000)	Direction (0,1)	None

F (Find Home)

The SMC-25 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. It is intended to eliminate mechanical hysteresis typically found in many switches, encoders and is generally present in the form of system mechanical backlash.

1. 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 19-25,000 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.

2. 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

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

Mnemonic	Data 1	Data 2	Result
(Name) G (a,t)	0-192, 256-2048	Trace (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 2047, depending on NV memory capacity. Address locations between 225 and 255 are reserved for parameter storage and may not be used in programs. The SMC-25 also features a special case for the "Go" instruction.

SPECIAL CASE "Go'

If the address is specified as 2048 (above the last NV memory address), the SMC-25 will read the Go input ports, then, branch to an address based on the state of input ports 1 through 4. The target address starts at the second page of program memory, starting at address 256 with 16 character (byte) intervals. This instruction is analogous to "on PORT go to".

Input Port State				Address of 'Go-to'	
P1	P2	P3	P4	HEX	Location
1	1	1	1	0	256
0	1	1	1	1	272
1	0	1	1	2	288
0	0	1	1	3	304
1	1	0	1	4	320
0	1	0	1	5	336
1	0	0	1	6	352
0	0	0	1	7	368
1	1	1	0	8	384
0	1	1	0	9	400
1	0	1	0	A	416

Input Port State				Address of 'Go-to'	
P1	P2	P3	P4	HEX	Location
0	0	1	0	B	432
1	1	0	0	C	448
0	1	0	0	D	464
1	0	0	0	E	480
0	0	0	0	F	496

The physical input ports are internally inverted as part of the address computation. State 1111 is defined as a high or +5v on port 1 through port 4. ***Note: Commands located between address 129-191 will execute much faster.***

Command	Function	Type	NV Bytes
H	Set Step Size Resolution	Default, Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) H (n)	0-8	None	None

H (Step Resolution)

This command selects step size resolution. The SMC-25 has an output signal (pin 1) that may be used to activate control lines to the external driver. A "1" input produces a low output on pin 1. Pin 4 is allocated for quarter step control. A value of 3 will assert a low on this pin and on pin 1.

Initial and final velocities may require appropriate changes. When changing between full and half step, position is maintained to the nearest full step.

Value	Sequence
0	Full step
1	Half step
2	Wave
3	Binary
4	Fixed 1
5	Fixed 2
6	Fixed 3
7	Fixed 4
8	Custom

Phase Pins Output States: Steps 1-8 (***Note: Phase sequence order is: Ph3, Ph2, Ph1, Ph0)***

Sequence	1	2	3	4	5	6	7	8
Full Step	0101	1001	1010	0110	0101	1001	1010	0110
Half Step	0101	0001	1001	1000	1010	0010	0110	0100
Wave	0001	1000	0010	0100	0001	1000	0010	0100
Binary	0000	0001	0010	0011	0100	0101	0110	0111
Fixed 1	0100	0100	0100	0100	0100	0100	0100	0100
Fixed 2	1100	1100	1100	1100	1100	1100	1100	1100
Fixed 3	0101	0101	0101	0101	0101	0101	0101	0101
Fixed 4	0110	0110	0110	0110	0110	0110	0110	0110

Default = 1 (Half Step)

Custom Table
Entering a custom sequence is accomplished via the "\" write to NV memory command. Location 247contains the number of output states to follow, up to 8 maximum.
Location 248 through 255 contain the 4 bit data sequence. For example, lets say we want to toggle phase 1 pin in a 4 step sequence:

Enter: "\ 247 4"
"\ 248 0"
"\ 249 1"
"\ 250 0"
"\ 251 1"
Enter: "H 8"
Enter: "S" (to store)
Enter: ^C (to reset)

Command	Function	Type	NV Bytes
I	Set Initial Velocity	Default, Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) I (n)	SPS (19-25,000)	None	None

I (Initial Velocity)

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:

A. All index commands (+, -, R)
B. First execute in constant velocity
C. Decelerate to 0 in constant velocity or soft stop
D. Final phase in home command if home speed is above initial velocity

Default = 400 steps per second.

Command	Function	Type	NV Bytes
i lower case	Restart Special 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) Restart Special Trip (See lower case "k" command)

*Actual values are determined by the hardware configuration.

Command	Function	Type	NV Bytes
J	Jump to Address	Program	4

Mnemonic	Data 1	Data 2	Result
(Name) J (a,n)	Address (0-2047)	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, because only one jump counter is available for use at any given time.

Command	Function	Type	NV Bytes
K	Set Ramp Slope Time	Default, Immediate, Program	3

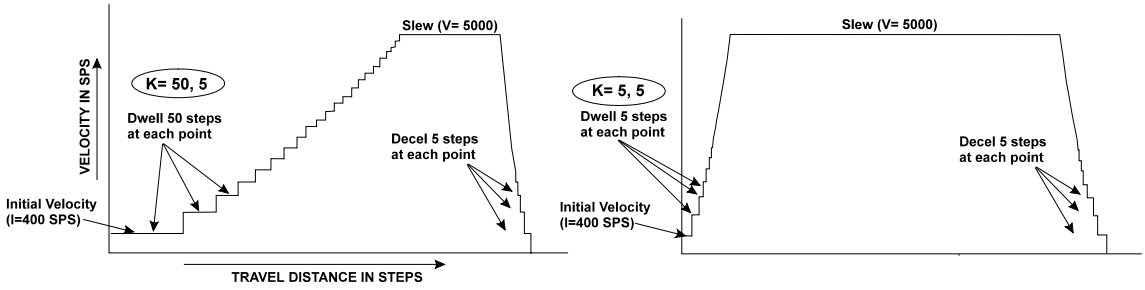
Mnemonic	Data 1	Data 2	Result
(Name) H (n1,n2)	Accel (0-255)	Decel (0-255)	None

K (Ramp Slope)

The "K" command is used to adjust the ramp slope during the motor acceleration or deceleration. The profile or shape of the acceleration/deceleration curve is defined by an internal lookup table. Depending on the values of initial and slew velocities, 0 to 157 discrete velocities may be attained to adjust the acceleration or deceleration of the motor armature rotation.

The "K" value determines how many steps are made at each step rate point 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 systems with fixed acceleration/deceleration times. Examples: (Two ramped indexes, each 2000 steps with I=400, V=5000, but different “K” values; K50 5 and K5 5:



Entering a value of "130" would change only the deceleration slope and would cause the deceleration ramp to have a value of "2" steps on each point of the deceleration portion of the ramp table.

The K command can be issued:
1. As part of a setup.
2. In an application program.
3. As user defined defaults at reset.

Command	Function	Type	NV Bytes

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

*Actual values are determined by the hardware configuration.

The latency described in use of the "T" 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:

Enter as follows:

P 0			Start programming mode
	0	O 0	Set position to zero
	4	k 200 8	Trip position 200, port 4 on, RAM=128
	9	+ 6000	Index 6000 steps
	13	W 0	Force wait till index complete
	16	P 0	End program
P 128			Program RAM commands
	128	k 400 0	Set new trip at 400 and turn ports off
	133	k 600 16	Set new trip at 600 and turn ports 4 and 5 on
	138	i 200 8	Reset origin, RAM=128, port 4 on
	143	P 0	End program
S			Save the shadow RAM program
G 0			Execute program

Failure to store the shadow memory will result in loss of all commands between 128 and 192. Once they are stored, they will automatically reload with every reset.

The following further describes the program sequence:

Address		Description
0	"O 0"	The position counter is reset to zero
4	"k 200 8"	The initialize command "k" is first used to initialize the real time sequence. Assume that the command "k 200 8" is executed at the beginning of the program. The following actions take place: 1. Port 4 is set on per data2 – see the "A" command. 2. The first trip position is set per Data1 (200). 3. A special trip program counter (PC) is set to 128.
9	"+6000"	Now the +6000 index command is started. When the position matches 200, the command located at 128 is checked for either a "i" or "k."
128	"k 200 0"	The "k" changes the trip position to 400 and turns all output ports off (high logic voltage) This is executed while at the exact 400 position. The program counter is advanced to 133.
133	"k 600 16"	This is executed like the previous 128, new trip=600, ports 4 and 5 are turned on and the program counter set to 138.

138	"i 200 8"	The restart command. here performs the same actions as the initial k 200 8 command. Port 4 is turned on, the trip is set to 200 and program counter is reset to 128.

ONE IMPORTANT ADDITIONAL ACTION is performed. The position counter is reset to ZERO and causes the repeat of trips 200, 400 and 600.

Notes:
1. The physical motor travel will be 6000 steps, even though the position counter has been reset 10 times.
2. The cycle will repeat 10 times.
3. The position counter ends up at zero

Command	Function		Type	NV Bytes
L	Loop on Port		Program	3
	Mnemonic	Data 1	Data 2	Result
	(Name) L (a,c)	0-2048	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.

Input ports are available as follows:

Port	Low	High
1	0	1
2	2	3
3	4	5
4	6	7
5	8	9

The SMC-25 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 2048, or A129 (read) command.

The SMC-25 has an additional feature of implementing a "wait till" function. The standard loop tests the condition every 2-3ms. If the unique address is 2048, the controller executes a tight loop at this instruction, monitoring the specified condition. 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.

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

Mnemonic	Data 1	Data 2	Result
(Name) M	SPS (±19-25,000)	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-25 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:

A. The "M 0" command.
B. Soft stop command or interrupt.
C. Abort (ESC) interrupt (without deceleration).

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

D. Divide
E. Ramp factor
F. Step Resolution

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)

This command sets the internal 24 bit position counter to the specified value. Zero position for the RELATIVE mode is "0000". 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.

Command	Function	Type	NV Bytes

P	Program Mode On/Off		Immediate	N/A
	Mnemonic	Data 1	Data 2	Result
	(Name) P (a)	Address (0-2047)	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, 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
128-191	Fast "shadow" RAM
256-511	"G 2048" command
1600	Power-up routines

Command Q	Function List Program		Type Immediate	NV Bytes N/A
	Mnemonic	Data 1	Data 2	Result
	(Name) Q (a)	Address (0-2047)	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 a 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).

Command	Function	Type	NV Bytes
S	Save Parameters to NV Mem.	Immediate	1

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

S (Save)

The following parameters are saved in the NV memory and will be recalled as defaults during power-on reset:

1. NV memory addresses 128 through 191 (shadow RAM)
2. Initial velocity (I)
3. Slew velocity (V)
4. Divide factor (D)
5. Ramp slope (K)
6. Jog speeds (B)
7. Resolution mode (H)
8. Auto power down (E)
9. Limit polarity (H)
10. Name (for Party Line use)
11. Trip point settings

Note: All of these parameters are saved as a block from the working registers in the SMC-25. Frequent use of this command should be avoided, as memory longevity may be affected.

Command	Function	Type	NV Bytes
T	Set and Enable Trip Point	Default, Program	4

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

T (Trip Point)

The SMC-25 has a programmable "Trip Point" feature. During moves the position counter is compared to the trip position at each step. When the position matches, up to two events can be triggered.

This will always happen
The non-zero value set by the "k"(lower case k) will be sent to the ports – See the "A" command for description of operation. The port data is always set at the actual step, without delays.

This will only happen while a program is running
An internal trigger is set to cause instructions stored at location 160 (fast shadow RAM) to be executed then resume the main program. This is the only case where the equivalent of a "sub routine" is allowed. The last instruction of a motion sequence should be a "W 0", otherwise the program will be considered over when the motion command is executed

When the Trip Point is enabled, available top step speed capability is reduced by approximately 10%. A value of 0 will disable the Trip Point trigger. A value of -0 will set the Trip Point to 0. The Trip Point is displayed using the examine command.

Example: (terminal mode)

Command		Comment
T 1000		Set Trip Point
k 16		Set output to turn on port 5 on
S		Save to NV memory
P 0		Start programming mode
0	O 0	Set origin to zero
4	+ 5000	Do index
8	W 0	Force wait till index complete
11	P 0	End program
P 160		
160	T 2000	Set new trip
164	k 8	Change outputs next time
16	P 0	Done programming
S		Save the shadow program
G 0		Execute program

Default = 0 (off)

Command	Function	Type	NV Bytes
V	Set Final Slew Velocity in SPS	Default, Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) V (n)	SPS (19-25,000)	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), jog (B) or home (F) speeds and is used when indexing absolute or relative (+ ,-, R commands).

Default = 5,016

Command	Function	Type	NV Bytes
W	Wait (n) Milliseconds	Immediate, Program	3

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

W (Wait)

The controller will remain in an idle state for the specified time. The Wait command, if issued while indexing (as a result of a R,+,-,or F command), timing will NOT start until the motion has completed.

Wait until motion complete

Using this command with zero time can provide an alternate method of determining motion. If issued while running at constant velocity, the time-out will occur without waiting for motion to cease. High-speed step operation during Wait commands will increase the delay time by as much as 14 times the normal value. The result will NOT be available until the delay is complete.

Command	Function	Type	NV Bytes
X	Examine Settings	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
X	None	None	Display Setting

X (Examine)

The Examine command produces two different responses, depending on the mode of operation. When NOT in the multi-axis mode (non-Daisy Chain or Party Line) the display is as follows:

"K= aa/dd, I= ii, V= vv, N=nn, mm, (T= tt pp)[CR, LF]"

Where:
F = Full, or H = Half Step
aa/dd = Ramp slope accel/decel
ii = Initial velocity divided by "D"
vv = Final step rate divided by "D"
mm = Step mode (full, half, wave, etc)
tt = Trip Point (if enabled)
pp = Port data (trip enabled)
nn = Name for Party Line

When in the multi-axis (Daisy Chain or Party Line) mode the data is returned in the following format:
mm[LF]
mm= model (25)

Command	Function	Type	NV Bytes
Z	Read/Display Current Position	Immediate	1

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

Z (Read Position)

During motor move commands the value will change depending on the direction of travel. The counter is programmable by the "O" command. The SMC-25 has the option of continuous readout via the serial interface. The "Z 1" enables this operation. Any change in position causes the position data to be sent to the serial output. The readout is terminated by a carriage return only.

The readout mode will be defaulted as "On" if a SAVE command is issued. This mode is only practical using single axis protocol.

Command	Function	Type	NV Bytes
[	Read NV Memory	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
(Name) [	Address (0-2047)	Sequential Bytes (0-255)	Displayed Values

[(Read NV Memory)

The user may display any byte of the 2047 byte external NV memory. The address specifies the desired location to access. At addresses of 128-191 the NV memory is <u>always</u> Read (not the RAM). The data contained at the specified location is output as a decimal value.

Command	Function	Type	NV Bytes
]	Read Limits, Hardware	Immediate, Program	1

Mnemonic	Data 1	Data 2	Result
(Name)]		None	Status

] (Read Limits, Hardware)

This command allows the user to examine the status of the various switch inputs. The result will contain the state of the limit switch inputs and current phase outputs 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-25:	Lb	La	Hm	*	Ph3	Ph2	Ph1	Ph0

<u>Where:</u>
La = Limit "a" switch
Lb = Limit "b" switch
Hm = Home switch (32 = low input)
Ph 0-4 = Phase State outputs (high = off)
* = Always low

If the motor is disabled and limit switches are inactive then the result will be 0.

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, 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 a 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).

Command	Function	Type	NV Bytes
-	Index in Minus Direction	Immediate, Program	4
	Mnemonic	Data 1	Data 2	Result
	(Name) – (n)	SPS ±8,388,607	None	None

- (Index in Minus Direction

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

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-25. A non-zero value indicates moving.

Command	Function	Type	NV Bytes
\	Write to NV Memory	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
(Name) \ (a,d)	Address (0-2047)	Data (0-255)	None

\ *(Write to NV Memory)*

This command allows the programmer to modify any location in the memory.

Special step sequences may be entered, and all initialization constants may be changed. (Reference "Memory Map" in the NON-VOLATILE MEMORY section of this manual for specific locations).

The life expectancy of the NV memory may be affected by this command. This command complements the Read NV Memory ([) command. Addresses of 128-191 in the NV Memory is always written to (not the RAM).

SECTION 7: SPECIFICATIONS

DC Characteristics

Parameter	Description	Condition	Min	Max	Units
Icc	Power supply current		5.0	50	Ma
Vcc	Logic Power		4.5	5.5	V
Vil	Input low voltage		-.05	0.8	V
Iil	Input low current	Vil=0.45v		800	µa
Vih	Input high voltage		2.0	Vcc+0.5	V
Iih	Input high current	Vi=vcc-1.5		500	µa
Vol	Output low voltage	Iol=1.6ma		0.45	V
Voh	Output high voltage	Ioh=-80ua	2.4		V
X2h	High ext. oscillator voltage		2.5	Vcc+.5	V
X2l	Low ext oscillator voltage			0.8	V
X2il	Low ext. oscillator current			-3.2	Ma

AC Characteristics

Parameter	Description	Min.	Typ	Max.	Units
Clock	Oscillator frequency	3.6864	14.7	16	Mhz
Step Rate	Range (14Mhz crystal)	19		28k	SPS
Step Pulse Width	Output to driver	5		7	µs
X2tr,tf	Ext. oscillator rise/fall time			15	ns
X2th,tl	Ext. oscillator clock width			15	ns
Swl	Limit/home switch response	2			Step Clk

Non-Volatile Memory Operation

Instruction	Condition	Typ	Units
Fetch and execute cycle	Loop	1.7	Ms
Save parameters	Store	63	Ms

SECTION 8: APPENDIX

Appendix A: ASCII Command Table

ASCII CHARACTER CODES

* All products
1. SMC-C24
2. SMC25, 26
3. MAX
4. SIN-10

E= Echoed

Ctrl	Char	Dec	Hex	Code	Use
^@		00	00	NUL	
^A	☺	01	01	SOH	
^B	☻	02	02	STX	
^C	♥	03	03	ETX	*
^D	♦	04	04	EOT	4
^E	♣	05	05	ENQ	4
^F	♠	06	06	ACK	4
^G	•	07	07	BEL	
^H	◘	08	08	BS	*
^I	○	09	09	HT	
^J	◙	10	0A	LF	*
^K	♂	11	0B	VT	
^L	♀	12	0C	FF	
^M	♪	13	0D	CR	*
^N	♫	14	0E	SO	
^O	☼	15	0F	SI	
^P	►	16	10	DLE	
^Q	◄	17	11	DC1	
^R	↕	18	12	DC2	
^S	‼	19	13	DC3	
^T	¶	20	14	EC4	
^U	§	21	15	NAK	
^V	▬	22	16	SYN	
^W	↨	23	17	ETB	
^X	↑	24	18	CAN	
^Y	↓	25	19	EM	
^Z	→	26	1A	SUB	
^[	←	27	1B	ESC	*
^\	∟	28	1C	FS	
^]	↔	29	1D	GS	
^^	▲	30	1E	RS	
^_	▼	31	1F	US	

Dec	Hex	Char	Use
32	20		*
33	21	!	E3
34	22	"	
35	23	#	E
36	24	$	E
37	25	%	
38	26	&	4
39	27	'	
40	28	(	
41	29	)	
42	2A	*	4
43	2B	+	*
44	2C	,	
45	2D	-	*
46	2E	.	3
47	2F	/	
48	30	0	*
49	31	1	*
50	32	2	*
51	33	3	*
52	34	4	*
53	35	5	*
54	36	6	*
55	37	7	*
56	38	8	*
57	39	9	*
58	3A	:	
59	3B	;	4
60	3C	<	
61	3D	=	
62	3E	>	
63	3F	?	E4

Dec	Hex	Char	Use
64	40	@	*
65	41	A	*
66	42	B	*
67	43	C	*
68	44	D	*
69	45	E	*
70	46	F	*
71	47	G	*
72	48	H	*
73	49	I	*
74	4A	J	*
75	4B	K	*
76	4C	L	*
77	4D	M	*
78	4E	N	*
79	4F	O	*
80	50	P	*
81	51	Q	*
82	52	R	*
83	53	S	*
84	54	T	*
85	55	U	*
86	56	V	*
87	57	W	*
88	58	X	*
89	59	Y	*
90	5A	Z	*
91	5B	[	*
92	5C	\	*
93	5D	]	*
94	5E	^	*
95	5F	_	1

Dec	Hex	Char	Use
96	60	`	
97	61	a	2,3
98	62	b	2,3
99	63	c	2,3
100	64	d	2,3
101	65	e	2,3
102	66	f	2,3
103	67	g	2,3
104	68	h	2,3
105	69	i	2,3
106	6A	j	2,3
107	6B	k	2,3
108	6C	l	2,3
109	6D	m	2,3
110	6E	n	2,3
111	6F	o	2,3
112	70	p	2,3
113	71	q	2,3
114	72	r	2,3
115	73	s	2,3
116	74	t	2,3
117	75	u	2,3
118	76	v	2,3
119	77	w	2,3
120	78	x	2,3
121	79	y	2,3
122	7A	z	2,3
123	7B	{	
124	7C	|	2,3
125	7D	}	
126	7E	~	
127	7F		

Appendix B: Acceleration Ramp Step Table

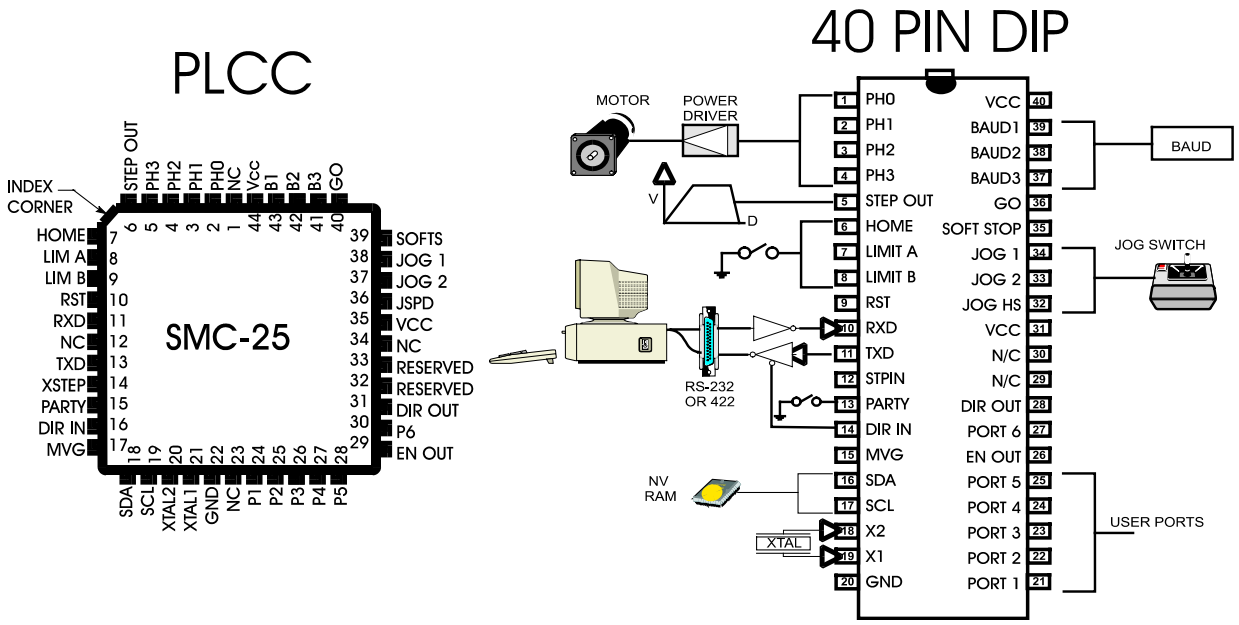
The following step rates are used internally to determine step rates during acceleration or deceleration ramping. Example: User values: I=400, V=3000, K=10. During an index execution the following actions take place:

1. 10 steps at 400 SPS, then 10 steps at each value 874, 1277, "…..," 2831 SPS, then Slew at 3000 SPS.
2. Decelerate using the reverse sequence.

Note: The actual initial and slew speeds are used with this table, "filling in" intermediate velocities during ramping

#	SPS	#	SPS	#	SPS	#	SPS	#	SPS	#	SPS	#	SPS
0	100	51	8593	101	13072	151	16605	201	19199	251	21557	301	23184
1	874	52	8714	102	13072	152	16605	202	19199	252	21557	302	23184
2	1277	53	8777	103	13212	153	16605	203	19504	253	21557	303	23184
3	1604	54	8904	104	13212	154	16605	204	19504	254	21557	304	23184
4	1890	55	8969	105	13356	155	16832	205	19504	255	21557	305	23184
5	2148	56	9102	106	13356	156	16832	206	19504	256	21557	306	23184
6	2390	57	9170	107	13503	157	16832	207	19504	257	21557	307	23184
7	2614	58	9309	108	13503	158	16832	208	19504	258	21557	308	23184
8	2831	59	9380	109	13653	159	17066	209	19819	259	21942	309	23184
9	3034	60	9452	110	13653	160	17066	210	19819	260	21942	310	23184
10	3225	61	9599	111	13806	161	17066	211	19819	261	21942	311	23184
11	3413	62	9675	112	13806	162	17066	212	19819	262	21942	312	23184
12	3592	63	9752	113	13963	163	17307	213	19819	263	21942	313	23184
13	3769	64	9830	114	13963	164	17307	214	19819	264	21942	314	23184
14	3938	65	9990	115	14124	165	17307	215	19819	265	21942	315	23184
15	4109	66	10072	116	14124	166	17307	216	20144	266	21942	316	23184
16	4266	67	10155	117	14288	167	17554	217	20144	267	21942	317	23184
17	4436	68	10239	118	14288	168	17554	218	20144	268	21942	318	23184
18	4585	69	10326	119	14288	169	17554	219	20144	269	21942	319	23630
19	4726	70	10413	120	14456	170	17554	220	20144	270	21942	320	23630
20	4856	71	10502	121	14456	171	17808	221	20144	271	22341	321	23630
21	5015	72	10593	122	14628	172	17808	222	20144	272	22341	322	23630
22	5141	73	10685	123	14628	173	17808	223	20479	273	22341	323	23630
23	5296	74	10778	124	14804	174	17808	224	20479	274	22341	324	23630
24	5437	75	10874	125	14804	175	17808	225	20479	275	22341	325	23630
25	5560	76	10971	126	14804	176	18070	226	20479	276	22341	326	23630
26	5688	77	11070	127	14985	177	18070	227	20479	277	22341	327	23630
27	5823	78	11170	128	14985	178	18070	228	20479	278	22341	328	23630
28	5965	79	11170	129	14985	179	18070	229	20479	279	22341	329	23630
29	6083	80	11273	130	15170	180	18070	230	20479	280	22341	330	23630
30	6206	81	11377	131	15170	181	18340	231	20827	281	22341	331	23630
31	6334	82	11484	132	15359	182	18340	232	20827	282	22341	332	23630
32	6467	83	11592	133	15359	183	18340	233	20827	283	22341	333	23630
33	6571	84	11592	134	15359	184	18340	234	20827	284	22755	334	23630
34	6714	85	11702	135	15554	185	18340	235	20827	285	22755	335	23630
35	6826	86	11815	136	15554	186	18618	236	20827	286	22755	336	23630
36	6942	87	11930	137	15554	187	18618	237	20827	287	22755	337	23630
37	7062	88	11930	138	15753	188	18618	238	20827	288	22755	338	23630
38	7185	89	12047	139	15753	189	18618	239	20827	289	22755	339	23630
39	7271	90	12166	140	15753	190	18618	240	21186	290	22755	340	23630
40	7402	91	12166	141	15958	191	18904	241	21186	291	22755	341	23630
41	7492	92	12287	142	15958	192	18904	242	21186	292	22755	342	23630
42	7632	93	12412	143	15958	193	18904	243	21186	293	22755	343	23630
43	7728	94	12412	144	15958	194	18904	244	21186	294	22755	344	23630
44	7826	95	12538	145	16168	195	18904	245	21186	295	22755	345	24094
45	7979	96	12668	146	16168	196	18904	246	21186	296	22755		
46	8084	97	12668	147	16168	197	19199	247	21186	297	22755		
47	8192	98	12800	148	16384	198	19199	248	21186	298	22755		
48	8246	99	12800	149	16384	199	19199	249	21557	299	22755		
49	8359	100	12934	150	16384	200	19199	250	21557	300	23184		

Appendix C: Signal Assignments



Pin # PLCC	Pin # Dip	Description	Pin Name	Pin # PLCC	Pin # Dip	Description
1		Reserved		23		Reserved
2	1	Phase 0 - Out	Port 1	24	21	User port
3	2	Phase 1 - Out	Port 2	25	22	User port
4	3	Phase 2 - Out	Port 3	26	23	User port
5	4	Phase 3 - Out	Port 4	27	24	User port
6	5	Step clock - Out	Port 5	28	25	User port
7	6	Home encoder - In	EN	29	26	Enable Output
8	7	Limit switch 1 - In	Port 6	30	27	User port
9	8	Limit switch 2 - In	Dir Out	31	28	Direction
10	9	Reset - In		32	29	Reserved
11	10	Serial data		33	30	Reserved
12		Reserved		34		Reserved
13	11	Serial data/clock - Out	VCC	35	31	
14	12	Step - Input	JSPD*	36	32	High speed jog - In
15	13	Input and output	Jog 2*	37	33	(+) Jog input - In
16	14	Direction Input	Jog 1*	38	34	(-) Jog input - In
17	15	Moving Output	Soft stop*	39	35	In
18	16	NV memory data	Go*	40	36	In
19	17	NV memory clock - Out	Baud 3*	41	37	B3 - In
20	18	Xtal 2 or Gnd - In	Baud 2*	42	38	B2 - In
21	19	Xtal 1 or ext. osc. - In	Baud 1*	43	39	B1 - In
22	20		VCC	44	40	+5v power - In

*External pull-up resistor required.

Note: Unused inputs MUST use pull-up resistors. I/O pins 36, 37, 38, 39, 40, 41, 42 and 43 require pull-up resistors. Other inputs and/or outputs have internal pull-ups or totem pole outputs.

Appendix D: Command Summary

MNEMONIC / COMMAND		DATA 1	RANGE 1	DATA 2	RANGE 2	NV	D	I	P
+	INDEX IN "+" DIRECTION	STEPS	1- 16,777,215			4		⊙	⊙
-	INDEX IN "-" DIRECTION	STEPS	1- 16,777,215			4		⊙	⊙
ESC	ABORT/TERMINATE							⊙	
@	SOFT STOP					2		⊙	⊙
^C	SOFTWARE RESET								
[	READ NV MEMORY	ADDRESS	0-2047*	NUMBER	0-255			⊙	
\	WRITE TO NV MEMORY	ADDRESS	0-2047*	DATA	0-255			⊙	
]	READ LIMITS/HARDWARE	LIM/HW	0-1					⊙	⊙
^	READ MOVING STATUS							⊙	⊙
|	SELECTIVE TERMINATE					2		⊙	⊙
A	PORT R/W	BINARY	0-128			2		⊙	⊙
B	SET JOG SPEEDS	SLOW	0-255	HIGH	0-255	3	⊙	⊙	⊙
C	CLEAR AND RESTORE	PAGE	0-9					⊙	
D	DIVIDE STEP RATE	DIVIDER	0-255			2	⊙	⊙	⊙
E	ENABLE/ INVERT LIMITS	MODE	0-3			2	⊙	⊙	⊙
F	FIND HOME	SPS	19-25000	DIRECTION	0-1	3		⊙	⊙
G	GO	ADDRESS	0-2048*	TRACE	0-1	3		⊙	⊙
H	RESOLUTION MODE	TABLE	0-8			2	⊙	⊙	⊙
I	INITIAL VELOCITY	SPS	19-25000			3	⊙	⊙	⊙
i	RESTART SPECIAL TRIP	NEXT TRIP	±8388607	PORT	0-63	5	⊙		⊙
J	JUMP	ADDRESS	0-2048*	N+1 TIMES	0-255	4			⊙
K	RAMP SLOPE	ACCEL	0-255	DECEL	0-255	3	⊙	⊙	⊙
k	TRIP OUTPUT VALUE	NEXT TRIP	±8388607	PORT	0-63	5	⊙		⊙
L	LOOP ON PORT	ADDRESS	0-2048*	CONDITION	0-8	4			⊙
M	MOVE AT CONST. VEL.	SPS	±25000			3		⊙	⊙
N									
O	SET ORIGIN	STEPS	±8388607			4		⊙	⊙
P	PROGRAM MODE	ADDRESS	0-2047*					⊙	
Q	QUERY PROGRAM	ADDRESS	0-2047*					⊙	
R	INDEX TO POSITION	POSITION	±8388607			4		⊙	⊙
S	STORE PARAMETERS							⊙	
T	TRIP POINT	POSITION	±8388607	VECTOR	0-255	4	⊙		⊙
U	RESERVED								
V	SLEW VELOCITY	SPS	19-25000			3	⊙	⊙	⊙
W	WAIT, (DELAY)	0.01 SEC	0-65535			3		⊙	⊙
X	EXAMINE PARAMETERS							⊙	
Y									
Z	DISPLAY POSITION	CONTINUE	0-1					⊙	

** Program address ranges are 0-192, 256-2047. Address location 2048 is used as a special case. location 128 – 192 is high speed "shadow" RAM.*

Appendix E: Minimum Circuit Requirement