

**MCB-24/50 (V2.10) - Revision Date: 12/06/04**

**MCB-24/50  
USERS MANUAL**



**Introduction**

CONGRATULATIONS! ..... 1

**Installation**

REQUIRED HARDWARE..... 3  
 HARDWARE SET-UP..... 3  
 SOFTWARE SET-UP (EASI DISKETTE)..... 4

**Communication**

SERIAL COMMAND, SINGLE MODE..... 7  
 RS-422/485 PARTY LINE PROTOCOL..... 7  
 PARTY LINE HARDWARE..... 7  
 SIN-8 ADAPTER MODULE ..... 8  
 SIN-8 SCHEMATIC..... 9  
 PARTY LINE STARTUP ..... 9  
 SERIAL PROTOCOL..... 10  
 PROGRAM CONSIDERATIONS..... 10  
 PARTY LINE COMMUNICATION FLOW CHART ..... 11  
 ASCII Baud Rates ..... 12

**Auxiliary I.O.**

I.O. CONNECTIONS..... 13  
 I.O. SCHEMATIC..... 14

**Operating Parameters**

RESET/INITIALIZE..... 15  
 INPUT COMMAND EXAMPLE..... 15  
 RESULT DATA..... 16  
 PARTY LINE COMMANDS..... 16  
 INSTRUCTION EXECUTION..... 16  
 INTERRUPT COMMANDS..... 17  
 SOFT STOP “@”..... 17  
 SOFT STOP INTERRUPT ..... 17  
 JOG SPEEDS, HOMING..... 17  
 HIGH SPEED CONSIDERATIONS..... 18  
 RELATED ITEMS: TRIP AND LOOP..... 18  
 NON-VOLATILE MEMORY..... 18

**Programming Notes**

INSTRUCTIONS..... 21  
 EDITING PROGRAMS..... 21  
 SINGLE STEP..... 22  
 TRIP FUNCTION..... 22

**Program Commands**

PROGRAM COMMAND DESCRIPTION..... 23  
 ESC (GLOBAL ABORT)..... 23  
 @ (SOFT STOP)..... 23  
 ^C (RESET)..... 24  
 A (PORT)..... 24  
 B (SET JOG SPEEDS)..... 25  
 C (CLEAR AND RESTORE)..... 25  
 D (DIVIDE RESOLUTION)..... 26  
 E (ENABLE CONTROL)..... 27

F (FIND HOME).....	27
G (GO) .....	28
H (STEP RESOLUTION).....	29
I (INITIAL VELOCITY).....	29
J (JUMP TO ADDRESS).....	30
K (RAMP SLOPE).....	30
L (LOOP ON PORT).....	31
M (MOVE AT CONSTANT VELOCITY) .....	31
O (SET ORIGIN).....	32
P (PROGRAM MODE).....	32
Q (QUERY PROGRAM).....	34
R (INDEX RELATIVE TO ORIGIN).....	34
S (STORE PARAMETERS).....	35
T (TRIP POINT).....	35
V (SLEW VELOCITY).....	36
W (WAIT).....	37
X (EXAMINE PARAMETERS).....	37
Y (HOLD AND RUN CURRENT).....	38
Z (READ POSITION).....	38
[ (READ NON-VOLATILE MEMORY).....	38
] (READ LIMITS/HARDWARE).....	39
+ (INDEX IN PLUS DIRECTION).....	39
- (INDEX IN MINUS DIRECTION).....	39
^ (READ MOVING STATUS) .....	40
\ (WRITE TO NON-VOLATILE MEMORY).....	40
 <b>Specifications</b>	
ELECTRICAL.....	41
PHYSICAL.....	42
 <b>Addendum</b>	
COMMAND SUMMARY.....	43
ASCII COMMAND TABLE.....	45
ACCELERATION RAMP STEP TABLE.....	46

***Congratulations!***

.....on your purchase of an MCB-24 (23,000 SPS) and/or an MCB-50 (50,000 SPS) Stepper Motor Control Board from ADVANCED MICRO SYSTEMS, INC. The MCB is designed to provide years of reliable, accurate and cost-effective motion control. As with all AMS products, the MCB is backed by nearly two decades of manufacturing excellence and a commitment to quality and support that guarantees your satisfaction.

This Technical Reference Guide will assist you in optimizing the performance of your MCB. Its purpose is to provide access to information that will facilitate a reliable and trouble-free installation. The Technical Reference Guide is organized into the following sections: Installation, Communication, Auxiliary I.O., Operating Parameters, Programming Notes, Commands, and Specifications. We recommend that each section be reviewed prior to installation.

In addition to the Technical Reference Guide, a PC/AT compatible systems integration diskette (EASI) is supplied. This powerful programming tool is an easy to use, menu driven utility file with on-line help screens and available source code (Microsoft "C"). Used together, the Technical Reference Guide and the EASI diskette will enable you to quickly take advantage of the advanced programming features and system capabilities inherent in the MCB design.

Although the MCB and supporting documentation were designed to simplify the installation and on-going operation of your equipment, we recognize that the integration of motion control often requires answers to many complex issues. Please feel free to take advantage of our technical expertise in this area by calling one of our support personnel to discuss your application; 603-882-1447.

Thank You!  
Your AMS Team



**Required Hardware**

Before installing the MCB, inspect the contents of your package to ensure the following items are included:

Qty	Unit	Model #	Description
1	Axis	MCB-24/50	Step Motor Control Board
1	Axis	SIN-7 or SIN-9	RS-232 serial adapter, 25 pin/9 pin (Single axis)
1	System	SIN-8	RS232/RS422 serial adapter and cable (Multi-axis)
1	System	SIN-10	Intelligent serial adapter and cable (Windows 2000/NT operating systems or complex programming)
1	Added axis	BLC-51-3	Interconnect cable, Cat5 (3 ft.)
1	System	TERM-2	Terminator plug (included with serial adapters)

**Hardware Set-Up**

Prepare for operation by following these simple instructions and referring to the MCB schematic and silkscreen drawing in the Specifications section of this manual:

- A. Attach the 25 Pin “D” connector end of the SIN-8/SIN-10 or SIN-7A cable assembly to either COM1 or COM2 of your PC/AT compatible computer. If using a SIN-8 or SIN-10 make sure that the switch is in the single (S) line mode.
- B. Connect the other end of the cable assembly (looks like a telephone connector), to the mating connector J1 “Party Line Serial Input.”
- C. Install a terminator plug (TERM-1) into J1A “Party Line Serial Output” of the last axis. The controller will work without a terminator plug, however reliability may be compromised, based on the environment (noise) and length of interconnect cables.

**Power Supply Input (J4)**

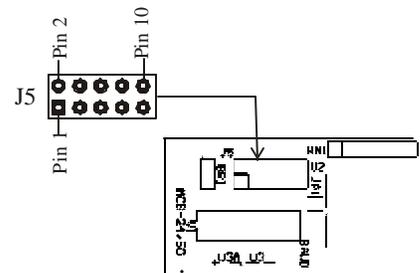
D. The MCB requires a 5 volt regulated power supply. Connectors J1, J2, and J5 may be used as alternate means to supply power to the board. The MCB does not use motor power Vmm. Rather it is fed across to the opposite edge of the board to mate with companion AMS’ Drivers.

Pin	Signal	Type
1	+5v	Logic power supply (+5v)
2	GND	Common/logic
3	Vmm	Motor Power (10-42 volts)

**Driver Connections (J5)**

E. Ten pin headers provide interface to motor power drivers. Buffered outputs have open collector (7407) circuits.

Pin	Signal	Type	Pin	Signal	Type
1	+5v	Power	2	Step	Output
3	Direction	Output	4	Full/half	Output
5	Quarter	NA	6	Enable	Output
7		Reserved	8	Gnd	Power Return
9		Reserved	10	Moving	Output



**Software Set-Up (EASI Diskette)**

EASI is a DOS program tailored to operate with AMS motion control products. It is not intended to operate as an end user application program, but rather to allow familiarization and evaluation of the products performance.

Almost any IBM PC/AT compatible computer will work with EASI. Although EASI will operate from a floppy disk drive, copying the contents to a subdirectory (call it "easi") will increase the speed and performance.

*Note: It is recommended that you review the "Getting Started" instructions on the EASI diskette by browsing the "README" file.*

A. Install the diskette and type: EASI<CR> at the prompt. At the opening screen enter "n" if you are operating from a monochrome terminal or hit the ENTER key for color.

B. Use the arrow keys to select COM1 or COM2. COM1 is the default setting. Follow the same procedure to select the correct Baud rate. 9600 BPS is the default setting.

**Sign-On**

C. Select the "DUMB TERMINAL" mode. A blank window will appear.

D. Strike the SPACE BAR key. The controller should sign on with the software version number V2.03. If not, enter a (^C) (reset) and strike the SPACE BAR key again.

Striking the ENTER <CR> key should result in an echo of "# " characters, further indicating communication is established.

**Name Axis RS-422 Party Line Mode**

Each axis should be assigned a unique "name" for proper operation when used in Party-Line mode.

A. Reset the controller (^C), then type a single, valid (upper or lower case) name character:

*Note: Make sure that the switch on the SIN-8 or SIN-10 is in the single (S) line mode.*

Recommended Names: Upper case A through Z Lower case a through z		Non-valid Names	
ASCII	HEX	ASCII	HEX
[	5B	^C	03
\	5C	CR	0D
]	5D	LF	0A
^	5E	@	40
-	5F		
`	60		

B. Follow the name with a SPACE BAR. The sign-on message will appear.

C. Verify the Name by entering the "X" command <CR>. The last item of the first line should contain the "Name" character.

D. Enter the Save command (S) <CR>. The axis Name is now stored in non-volatile memory.

**Name Axis RS-422 Party Line Mode (continued)**

The MCB is ready to operate in the current single axis mode or be switched over to Party Line mode. It is suggested that the operator use single mode first to become familiar with command input. The single axis mode can be used with any “dumb” terminal device and is not dependant on using the AMS software. For multi-axis operation refer to “Party Line Startup” in the Communication section of this manual.

*Note, single axis, RS-232 mode operation does not require the name axis procedure.*

**Examine Command**

The Examine command (X)<CR>, will display a set of parameter values that were last stored in non-volatile memory. These parameters may be modified using the appropriate commands, then stored in non-volatile memory as the new “defaults”.

Default Parameter	Value
C	10
K	5
I	400
V	5016

Where:

- C= Hold Current %
- K= Ramp Slope
- I= Initial Velocity
- V= Slew Velocity

The values shown assumes there are no input connections or special modes such as inverted limit switches.

Simple Program Examples:

1. Issue the command: “R - 1000”<cr>. The motor should move.
2. Issue the “Z”<cr> command. The position (-1000) should be displayed.

**Rules**

1. The command line may be edited using back-space as characters are typed.
2. The line may be canceled using <ESC>.
3. The command line is limited to 15 characters.
4. Only one command may be entered per line.
5. A space is optional between the command and first number.
6. A space or comma must be used to separate two parameter commands.

**A. Entering Programs**

The above examples were samples of immediate commands. The following is a sample of the sequences that are stored in non-volatile memory. Note that when programing the sequence is immediately written to non-volatile memory, without any additional action required to save it.

<u>Address</u>	<u>Enter</u>	<u>Remark</u>
	P0<CR>	Place in Program mode. Insert instructions at location 00.
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.

**A. Entering Programs (continued)**

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.
--	-------	----------------

*Note: An upper case "Q" displays one line at a time. A lower case "q" will display up to 25 lines at a time.*

**B. Verify the Program**

The MCB 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		

**C. Execute the Program**

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

*Note: The program can be terminated at any time by hitting the ESCape key.*

**D. Edit Program**

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

<u>Enter</u>	<u>Remark</u>
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. It is recommended that application programs be developed using the menu driven program (Party Line selection) in the EASI diskette which includes a sophisticated Editor and Compiler.*

### **Serial Command, Single Mode**

The words “host” and “terminal” are used interchangeably and refer to any source of commonly available standard RS-422 interface including; terminals, printer ports and UARTs (COM ports). A character is a stream or packet of 8 consecutive bits (1s and 0s) with a defined sequence. Each unique character represents a letter, number, punctuation or unreadable control character as defined by an international set of standards called ASCII. The complete ASCII Table may be found at the end of this chapter. The speed that characters are sent or received is defined by the Baud rate. The serial communication link refers to any two wire, full or half duplex communication link.

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. Motion commands with 2 numbers require at least one space between numbers. Motion commands can either be upper or lower case.

In the Command mode, the command is executed upon receipt of a Carriage Return (Single mode) or ^J, ^Enter (Party Line mode). The MCB will respond with a Carriage Return and Line Feed upon acceptance of the command.

### **RS-422/485 Party Line Protocol**

Communication options include RS-422/485 Party Line protocol, recommended for multi-axis applications. In this mode each receiver monitors the host and responds when receiving a matching Name character. Communication to each individual axis in Single mode is performed to assign the unique Name (s) for each slave axis.

The Single mode provides User friendly one axis communication for setup and debug functions. Setup usually involves optimizing operational parameters, writing and storing programs and assigning a unique Name for Party Line operation.

### **Party Line Hardware**

“Party Line” products have a 4 wire RS-422/485 interface. Differential line drivers and receivers are used to provide reliable communication in noisy environments. This design allows a single Master (or Host) computer and up to 32 Slave controllers. The hybrid design retains all the desirable characteristics of both EIA RS-422 and RS-485 specifications. In general, the hardware implementation follows the extended RS-485 standard with higher voltage and receiver capacity. Rather than half duplex protocol however, a full duplex Party Line communication is provided.

#### **Slaves**

Each Slave unit contains the following:

1. RS-485 line receiver; always active.
2. RS-485 tri-state line driver; activated on receipt of “address received.”

The line driver is always enabled when Single mode operation is selected.

#### **Master**

The Master interface, consists of:

1. RS-485 line receiver; always active
2. RS-485 line driver; always active

A Master interface, using the SIN-8 (RS-232 to RS-422) modular adapter, also contains a switch to select Single or Party Line mode.

*Note: The RS-485/422 is rated for a maximum cable length of 4,000 feet. It is recommended that a second terminator (120 ohms) be used if runs exceed 15 feet or if operated in an electrically noisy environment.*

*Note: DO NOT bundle signal wires and motor wires together. The high current and frequencies generated by chopper drivers will couple, even if shielded wire is used (unless 100% shielding can be guaranteed). Avoid proximity to relays, motors and other RFI sources.*

**Moving**

The Moving signal is an open collector buffered output on all AMS motion control products. The SIN-8 converts this signal to RS-232 level and supplies it to Pin 8 “Carrier Detect” of the 25 Pin “D” connector.

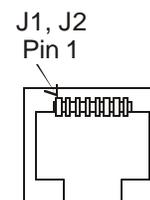
**SIN-8 Adapter Module**

(Instructions for using the SIN-10 serial adapter are covered in the SIN-10 Users Manual).

Once the unique Name has been assigned and stored in the non-volatile memory, multiple axis may be connected in parallel for Party Line mode operation. A switch on the SIN-8 (RS-232 to RS-422 adapter module) allows convenient selection between modes.

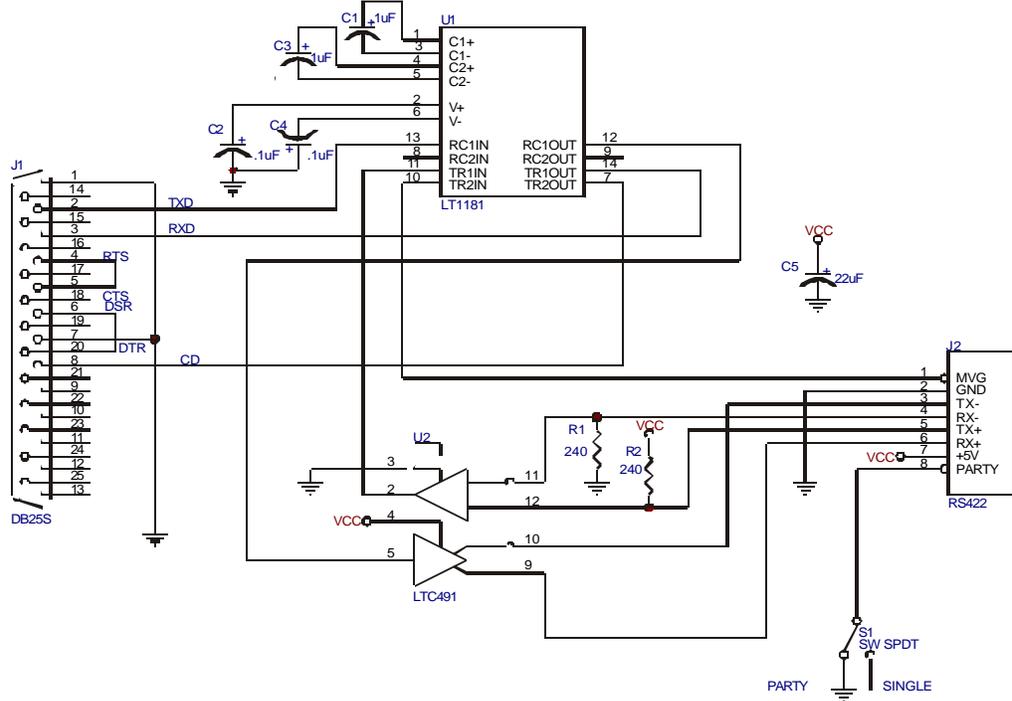
Single mode communication enables terminals or computers with serial communication capabilities to interface with the built-in command line editor. Party Line protocol requires the correct input character sequence to address an axis. It is recommended that a PC or other computer be used to simplify programming.

Serial Interface Connector (RJ45) on SIN-8			Serial Interface Connector (J1) on MCB		
Pin	Signal	Type	Pin	Signal	Type
1	Moving	Output	1	Moving	Output
2	GND	Logic	2	GND	Logic
3	TXD-	Output	3	RXD-	Input
4	RXD-	Input	4	TXD-	Output
5	RXD+	Input	5	TXD+	Output
6	TXD+	Output	6	RXD+	Input
7	+5v	Logic	7	+5v	Logic
8	Party Enable	Input	8	Party Enable	Input



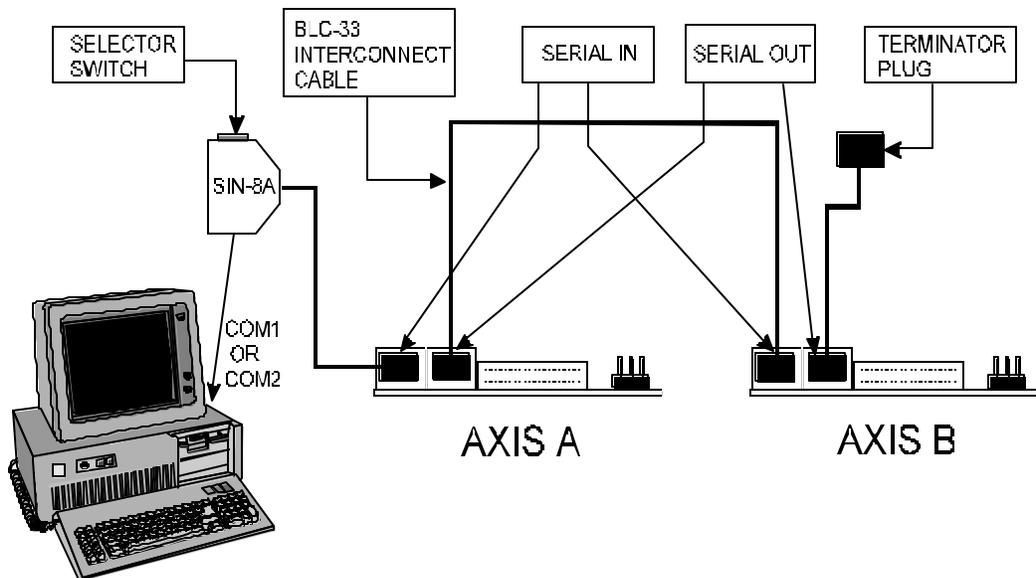
Party Line operation provides for parallel communication of up to 32 axis from a single serial communication port.

SIN-8 Schematic



Party Line Startup

After assignment of a unique "Name(s)" connect the axis (1 to 32) as shown. **Move the red selector switch on the SIN-8 to Party Line (P) mode.** When starting in Party Line the Name is read from non-volatile memory and then skips any sign-on procedure. At this point the sleep condition (serial outputs tri-stated) exists until a Name match occurs.



**Serial Protocol**

Each Slave unit residing on the Party Line acts as a listener, waiting for its personal address (name) character. Once the proper name is received, the Slave enables its RS-485 driver onto the TX bus. The activated Slave then echoes the start character and receives and echos the remainder of the command string until the terminator is received. The terminator must be a "line feed" (lf).

Once the line-feed character has been received, another axis may be commanded. In order for any axis to recognize a name, the name must be preceded with a line-feed (the previous terminator will do).

The echoed characters provide reliable handshaking and a means of receiving data from the Slave, i.e., a "Z" position register command.

Party Line communication using a dumb terminal might go like this (assume 2 axis with address "X" and "Y"):

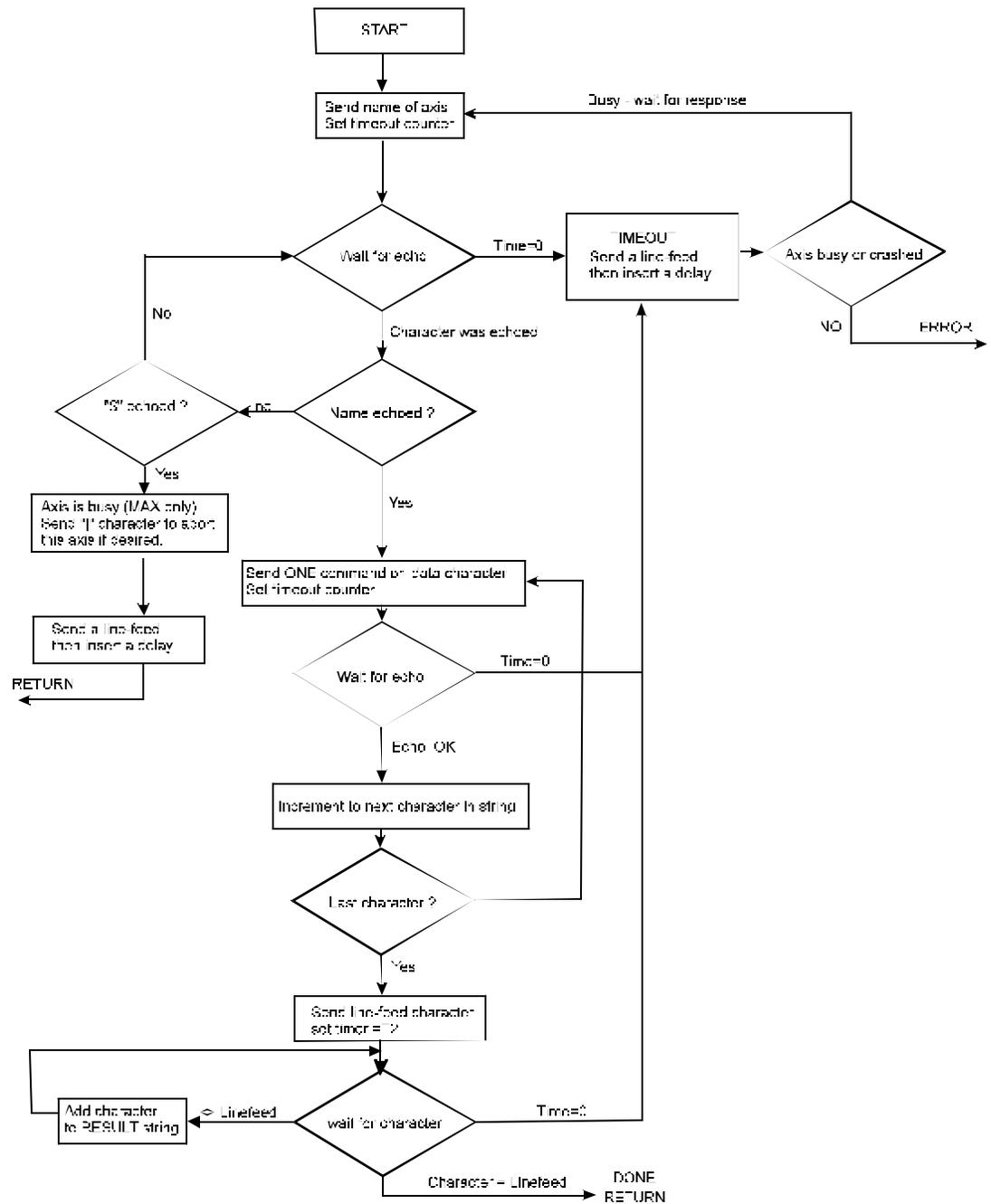
```
Master:  <lf> X+1000 <lf> Y-500 <lf> XZ <lf>  
Slave:   X+1000 <lf> Y-500 <lf> XZ1000 <lf>
```

**Program Considerations**

The MCB incorporates a buffered UART input. Because motion control is of the highest priority, processing of received information may be delayed if commands are sent while stepping at very fast rates. This condition may only occur at internal/external step rates exceeding 10,000 steps per second. In serial applications where commands are sent while motion is active, the User should monitor echoed data to avoid UART overrun.

***Note: IT MUST BE STRESSED THAT ANY HOST COMPUTER MUST VERIFY THE CHARACTER ECHO AS EVERY SINGLE CHARACTER IS SENT. Inserting a delay between character will inevitably fail and slow communications. Your application software must be capable of single character (as opposed to string) transmission without reception loss at 9600 Baud.***

Party Line Communication Flow Chart



**ASCII Baud Rates**

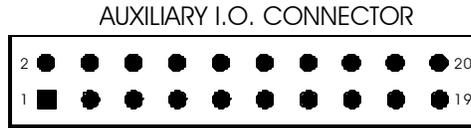
Baud rates for communication is fixed at 9600 BPS. Consult AMS if an alternate communication rate is required.

Host/terminal settings:

Start	Data	Stop	
<u>Bits</u>	<u>Length</u>	<u>Bits</u>	<u>Parity</u>
1	8	1	None

**I.O. Connections**

Twenty contact header connectors provide auxiliary inputs and outputs. Buffered signals use optical isolation or TTL buffers.



Pin	Signal	Type	Pin	Signal	Type
1	Port 2	Input	2	+5V	Power
3	Port 4	Output	4	Home	Input*
5	Moving	Output	6	Limit A	Input*
7	Port 5	Output	8	Limit B	Input*
9	Port 3	Input	10	Jog -	Input**
11	Port 1	Input	12	Jog +	Input**
13	Jog HS	Input *	14	Trip	Output
15	GND	Power	16	Soft Stop	Input
17	Go	Input*	18	Port 4	Input
19	+5V	Power	20	Opto Supply	Power-in

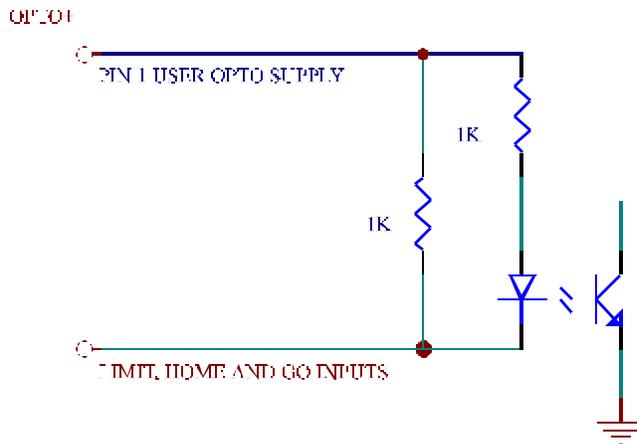
Inputs are TTL with 10k pull up resistors.

\* Optically isolated inputs with internal series resistor and LED power via J2.

\*\* Schmitt-Trigger inputs.

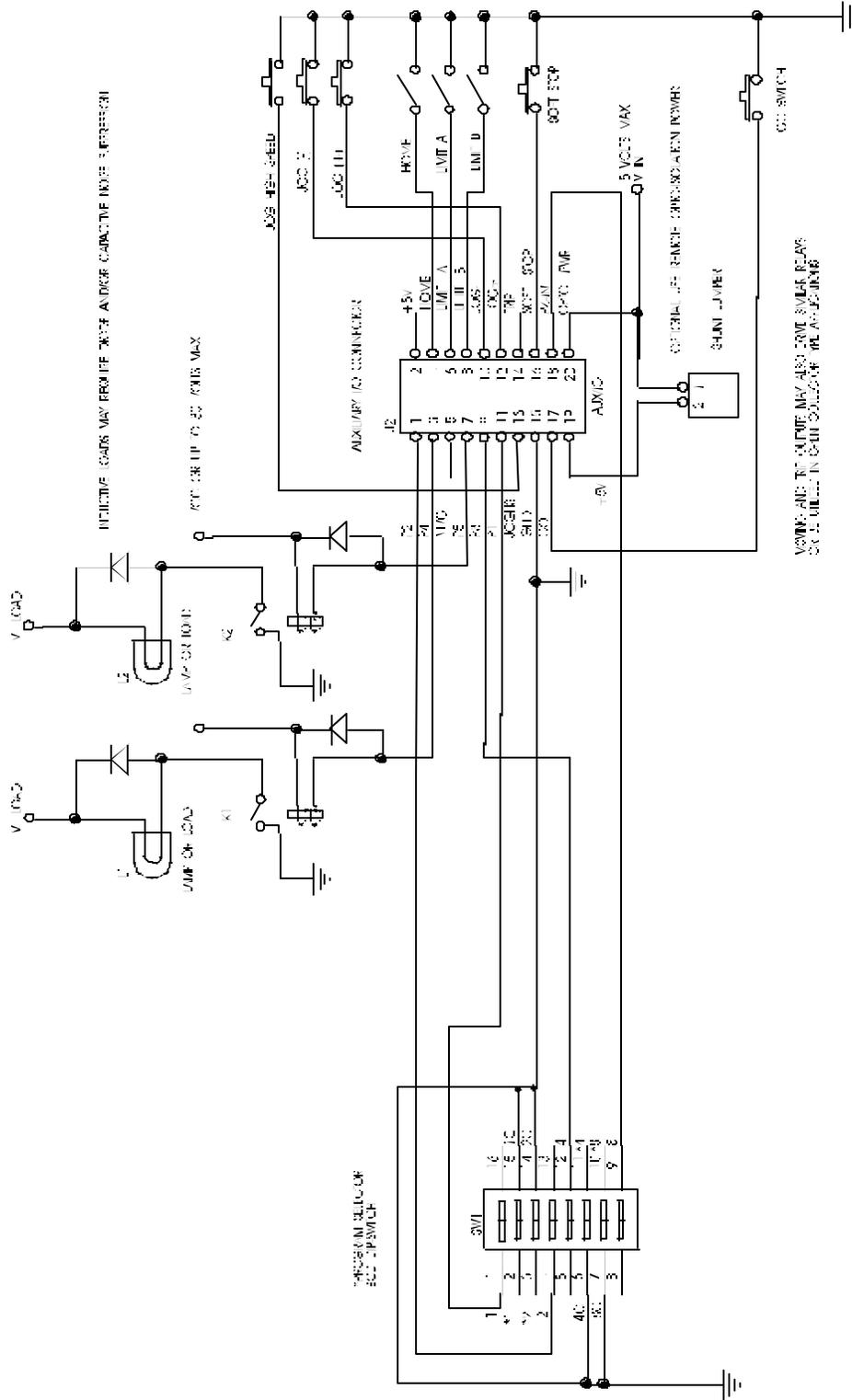
Applications requiring optical isolation may supply power (8V max) to pin 20 of the Opto supply input. Removal of the factory supplied jumper is required. Outputs are open collector TTL (7407) with 10k pull up resistors. The opto common signal supplies optical isolator power for limits, home and remote party line select (on J1).

**Typical Input**



**Note: NEVER attempt to draw current from the OPTO+ pin.**

I.O. Schematic



NOTE: RESISTORS INDICATED IN MEMORY FOR MULTIPLE BIT SCHEMATIC SHOULD FOLLOW THE COLUMN OF THE EXAMPLE PROGRAM BRANCHMENT ON PROGRAMMING POINT ON THE DATA SHEET.

**Reset/Initialize**

Automatic hardware reset occurs each time the power is turned on or a command “^C” is issued from serial communication. During reset all inputs and outputs will be at a high state. After hardware reset all parameters are initialized to factory set default values. The communication mode is established at 9600 BPS. Additionally, checks for Party Line operation are initiated.

Resident non-volatile memory is accessed, and the parameters most recently stored by the “S” command are downloaded, replacing the standard defaults in the working registers of the controller. The following block of parameters are stored and recovered as a set:

Parameter	Standard Defaults
Initial Velocity (I)	400 (steps per second)
Slew Velocity (V)	5016 (steps per second)
Divide Factor (D)	1
Ramp Slope (K)	5/5
Jog Speeds (B)	30, 200
Trip Point (T)	Off
Full/Half Step (H)	Full
Auto Power Down (E)	Yes
Hold/Run Current (Y)	10, 70
Limit Polarity	Low
Auto Position Readout (Z)	Off
Name (after reset)	Undefined

*Note: Commands that modify these parameters use the working registers inside the controller. Actual non-volatile memory storage is initiated by the “Save” command. Once initialization is complete, Jog and Go inputs are active to allow jogging or a low pulse on the Go input to execute a program previously stored in non-volatile memory. A terminal or host is NOT required for these functions and may be initiated from the Auxiliary Input/Output connector.*

**Input Command Example**

Command	Single Axis Mode Operation
+1000 (Carriage Return)	Step 1000 steps in + direction
+ 1000 (Carriage Return)	Same as #1
H 0 (Carriage Return)	Set fixed step resolution
H0 (Carriage Return)	Same as #3
H (Carriage Return)	Same as #3 (0 is used by default)
R -1000 (Carriage Return)	Move to position -1000

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 non-volatile memory:

- J 0 5 (Carriage Return) Jump to location 0, 6 (n + 1) times.
- J0 5 (Carriage Return) same as above.

**Result Data**

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

**Party Line Commands**

During Party Line operation characters will NOT be echoed to the host until the proper "Name" (preceded by a ^J or ^Enter) is detected. All axis monitor concurrently the common TXD line from the host. Once the Name is received, the target axis will wake-up and start echoing as described above.

**Instruction Execution**

For each Motion command there are four cycles:

1. Entry
2. Execution
3. Result
4. Completion

All other commands have three cycles:

1. Entry
2. Execution
3. Result

In the idle state the controller continually tests for Jog, Go or Command input. The following describes each sequence operation that takes place on receipt of a command:

**Cycle 1. Entry**

The input commands and data are loaded via RS-232 or RS-422/485 interface. Command and data information is placed in a command line buffer as received. Editing is permitted in Single mode. ESCape aborts operation and returns to an idle state. A Carriage Return (^J or ^Enter 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 are:

Action Command	Completion Cycle
Go	Until last instruction is complete
Half/Full Step	Until previous action is complete
Wait	Until previous action is 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. Another action command will be queued-up during the Completion cycle of a preceding Action command. The Execution and Result cycle of this Pending command is delayed until the Completion phase is finished. This interval is called the Pending Period. During this Pending Period the command accepted is the one character interrupt (abort) command, limit switches ( J2 pins 6, 8) soft stop input (pin 16) and home switch (J2 pin 4).

External indication of Pending Period end, Execution and Result cycle of the pending instruction is the Carriage Return. 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

*Note: All process(es) are aborted upon ESCape.*

### **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 and Constant Velocity commands (inter-speed ramping) and 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. A command can NOT be loaded while jogging and jogging can NOT be performed until the last command is complete. A command starts with the reception of the first command character.*

**High Speed Considerations**

Step rates are controlled with a high degree of accuracy. As a result, step control is given priority over other processes. At high step rates this will manifest itself as a slight latency. The execution time increases when high step rates are active during command cycles. An example might be reading positions while moving at a high speed. Usually this latency has little effect at step rates below 10,000 steps per second. At speeds approaching the maximum step rate the processing latency may have to be taken into account.

**Related Items: Trip and Loop**

The Trip point output (J2 pin 14) is activated on the exact step position specified. When running a program (from the GO command) several "fetches" from the non-volatile memory are required along with the service time. This latency may allow several motor steps to occur before the desired action takes place.

Loop on port may exhibit similar latency effects at high speeds. The port will require a longer "true" condition to be recognized. A faster method is to implement the "wait for port" condition using the "Go-Sub," (branch on port) condition.

**Non-Volatile Memory**

The MCB hosts a 2048 byte non-volatile memory for each axis. The non-volatile memory may be used to store User programs for future execution via the "Go" command. Any number of programs may coexist, limited only by the available memory space.

The following memory map indicates that address locations are segmented into 8 pages and are accessible through direct read/write commands or cleared using the appropriate "C" command:

Location	Description
(0-226)	User program or data storage
(256-2048)	User program or data storage
(227)	Configuration byte
(228)	Internal initial status byte. (Do NOT modify)
(229)	Divide factor (D)
(230,231)	Initial velocity low and high bytes
(232,233)	N/A
(234,235)	Slew speed (V) low and high bytes
(236,237)	N/A
(238)	Low speed jog value
(239)	High speed jog value
(240)	Ramp factor acceleration (K)
(241)	Ramp factor deceleration (K)
(242,243,244)	Trip
(245)	Hold current
(246)	Run current
(247)	Name for Party Line

A special memory "Scratch Pad" of 64 bytes is accessed in location 128 to 192 for use in the working registers of the controller.

The EEPROM has a finite life of approximately 400,000 "write" cycles. Care should be used when writing to non-volatile memory to exclude unnecessary write cycles. For example, the Restore command ("^C" from a terminal) will retrieve the parameters from the EEPROM without doing a write.

## **MCB STEPPER MOTOR CONTROL BOARD      OPERATING PARAMETERS**

If the Initialize command ("C 1") was chosen, the first 256 bytes of EEPROM are written. Should you require a sequence of motions to be done without host attention, you may breakup the motions into subgroups rather than repeatedly programming the EEPROM. Use the Go from address command to execute the subgroups in the required sequence.

Use the Save command sparingly. Parameters are set so quickly that it is sufficient to just let the host download them.

Changing parameters should NOT be done by writing directly to EEPROM. The MCB will not know that it was changed and may initiate an overwrite. Use the commands available to set parameters. Unlike writing, reading is non-taxing on the EEPROM.



## **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 for automatic or repetitive operations by initiating a command or by strobing the Go (input J2-17) to a logic low. When in the Program mode, the entered commands (now called instructions) are directed into the non-volatile memory. After leaving Program mode, the stored program(s) may be subsequently executed by entering the "G" (Go) command.

The following procedure assumes a standard (RS-232) serial interface using a common terminal:

The Program mode is initiated by entering "P aa" (Carriage Return). 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" (Carriage Return) or by strobing the "Go" (input J2 pin 17).

Once in the Program mode the current memory location is displayed on the terminal and instructions may be entered. As each instruction is entered, the location is displayed. All instructions have the same format as in the Command mode.

Terminating the Program mode is done by entering a "P." This will cause the end of program flag to be inserted and the controller will echo the pound (#) character. It will then return to 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 may occupy the full 2k byte of memory space.

***Note: 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 an address defined by the trip point.***

## **Editing Programs**

Existing program(s) may be modified 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 than 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, entering ESCape 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 reenter the remaining portion of the program***

### **Single Step**

Instructions may be used with a "Single Step" attribute. Instruction/commands that have this attribute will wait until a low trigger level is input to port 1 (J2, pin 11). This feature is valuable in implementing a master/slave type system.

The "master" can generate trigger pulses to the slaves that are waiting. The trigger may be any source capable of generating pulses between 10 and 100 microseconds. Entering a period "." will also generate a "software" trigger. The trigger is DC level sensitive and is equivalent to having a "LO\$" command preceding a standard instruction. Consecutive instructions with this attribute will be executed as long as the port is held low, (for either a single input or previous set port (A1) command).

The "Single Step" attribute is implemented during terminal entry by terminating the instruction command line with a period ".", rather than a Carriage Return. The instructions maintain the attribute in both Program Storage and Execute Now modes. Instructions are listed (using the Query command) with a period "." immediately after the instruction letter.

### **Trip Function**

A position sensitive trip point may be set. When moves are made in Command mode the Trip Port changes state each time the specified position is reached. During program execution a User program located at address 200 will be automatically "called" when the trip point is passed. The program (at 200) will execute even if movement resulting from the previous instruction is still active. Setting trip position to +0 disables the trip function. Setting T to -0 will define position 0 as the trip point.

The logic state of the trip output may be toggled by use of the "A 64" command. Reading the trip status by a host can be accomplished by using the "A 128" command and testing bit 6 (64 decimal). Some delay (milliseconds) will be experienced, as a result of the above mentioned "call", but the trip output will still occur at the exact position specified.

**Program Command Description**

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
	<i>Mnemonic</i>	<i>Data 1</i> <i>(Range)</i>	<i>Data 2</i> <i>(Range)</i>	<i>Result</i>

Where:

<i>Command</i>	ASCII character (Keystroke)
<i>Function</i>	Functional description of command
<i>Type</i>	D= Default; initial parameter setting I= Immediate (direct execution) P= Program command; executable in stored program
<i>NV</i>	Non-volatile memory byte requirements in program
<i>Mnemonic</i>	Single character prefix used in multi-axis protocol; (prefixed by axis "Name" assignment in Party Line mode)
<i>Data/Range 1</i>	Affected parameters and valid numerical range of parameters
<i>Data/Range 2</i>	Same as Data Range 1 (as required)
<i>Result</i>	Information returned as a result of command execution or examination

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>ESC</b>	Terminate Operation	Immediate		0
	<i>Mnemonic</i> (Name) Esc. Character	<i>Data 1</i> None	<i>Data 2</i> None	<i>Result</i> Echo

**ESC (Global Abort)**

Global Abort terminates any active operation and forces the controller to revert to the idle state. Output drivers or ports are not effected. Stepping and position counter update will cease immediately without deceleration. Any program "running" will be terminated.

Any axis in the program mode will exit the program mode without creating the "end of program marker," therefore the escape character is useful in editing non-volatile program segments. In single mode a pound (#) sign is returned.

*Note: Because the deceleration is immediate (without ramping) mechanical overshoot may result, especially with high speeds and/or inertia loads.*

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>@</b>	Soft Stop	Immediate, Program		1
	<i>Mnemonic</i> (Name) @	<i>Data 1</i> None	<i>Data 2</i> None	<i>Result</i> #

**@ (Soft Stop)**

The Soft Stop command is useful as a gentle stop. It behaves differently, depending on how it is used. If the axis is moving it causes an immediate deceleration to a stop, based on the established deceleration K value. If one or more axis is running a program when this command is sent via the serial port, the running program(s) will terminate after deceleration. The soft stop may be embedded within a program (in Program mode). During program execution, the encountered soft stop will not cause termination and is functionally equivalent to the "M 0" command

<b>^C</b>	<b>Function</b> Reset Controller		<b>Type</b> Immediate, Program		<b>NV Bytes</b> 0
	<b>Mnemonic</b> (Name) ^C	<b>Data 1</b> None	<b>Data 2</b> None		<b>Result</b> None

**^C (Reset)**

Software Reset is a global RESET command. All axis ABORT immediately and a reset, equivalent to the power-up condition, is executed:

1. Down load default values from the non-volatile memory.
2. Set origin(s) to zero.
3. Calibrate motor current to hold value.
4. Assume an idle state waiting for GO pulse input, Jog input or serial command input.

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

**A (Port)**

Input data ranging between 0 and 31 is complemented then output to port 1 through port 5. Port 1 is the least significant bit. Binary combinations of bits will turn on more then one port. Example; "A7" 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. The command "A0" will reset all ports to their in-active state.

Port	Data	Port	Data
1	1	Trip	64 Inverts Output
2	2	Increment	128 Increment port 1-5
3	4	Read	129
4	8		
5	16		

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	High if moving
4	Low input present on port 3*	64	Trip Point passed
8	Low input present on port 4*	128	Direction level, high for CW rotation

\*Ports 1,2 and 3 are dedicated inputs. Port 4 is an input and output. Port 5 is a dedicated output. The ports are forced active low by the set port command. The "A0" command will reset all ports. Result: None/Read State.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>B</b>	Set Jog Speeds	Immediate, Program		3
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) B (n1,n2)	Slow Speed 0-255	High Speed 0-255	None

**B (Set Jog Speeds)**

Data range 1 and data range 2 represent the speeds to use when the jog inputs are utilized. The first is usually a slower speed. The second number is used when the high speed jog ( J2 pin 13) 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 inputs. Ramped acceleration using the "K" value is implemented. Ramped deceleration is used if the high speed input is released while maintaining one of the two jog inputs on (low). Both clockwise and counter clockwise inputs operate at the same speeds.

The Jog inputs are active:

1. After power-up.
2. When not executing a motion command.
3. When not running a program.

Jogging is inhibited during actual command line entry.

\*Step rate is multiplied by 60 for MCB-50 applications.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>C</b>	Clear and Restore	Immediate		0
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) C (n)	0-8 Page	None	Version

**C (Clear and Restore)**

The C1 command clears the entire non-volatile memory and restores the default parameter settings. This command should be used sparingly since the non-volatile memory has a finite life of approximately 400,000 write cycles.

Parameter	Values (after Non-volatile Clear (C1) Command)
Initial velocity (I)	400 steps per second
Slew velocity (V)	5,016 steps per second
Divide factor (D)	1
Ramp slope (K)	5
Jog speeds (B)	30, 200 ( x30 steps/sec)
Trip Point (T)	Off
Full/Half (H)	Full
Auto power down (E)	Yes
Power down polarity (E)	High
Hold/Run current (Y)	10, 75
Limit polarity (H)	Low
Auto position readout (Z)	Off
Name	Unchanged

All these parameters are saved as a block from the working registers. Frequent use of this command should be avoided, as memory longevity may be effected.

To complete the restoration of programmable current control for the motor load, the appropriate E command (typically E8) should be entered and saved (S) immediately following the use of the C1 command.

The following commands clear the corresponding page of non-volatile memory from the indicated address locations:

Command	Page#	Address
C2	2	256 to 511
C3	3	512 to 767
C4	4	768 to 1023
C5	5	1024 to 1279
C6	6	1280 to 1535
C7	7	1536 to 1291
C8	8	1792 to 2047

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>D</b>	Divide Resolution	Immediate, Program		2
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) D (n)	Resolution 1-255	None	None

**D (Divide Resolution)**

All speeds during ramping and slewing are divided by the specified number (n). The prescale number may range between 1 and 255. Speeds as low as 3 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 slower slew speeds. The "D" command should NOT be changed while moving. characteristics at slower slew speeds. The "D" command should NOT be changed while moving.

*Note: This command is not supported by the MCB-50.*

<b>E</b>	<b>Command</b>		<b>Function</b>	<b>Type</b>	<b>NV Bytes</b>
			Enable Current Control	Immediate, Program, Default	2
		<b>Mnemonic</b>	<b>Data 1</b>	<b>Data 2</b>	<b>Result</b>
		(Name) E (n)	0 - 13	None	

**E (Enable Control)**

There are several operational modes with the use of this command. The following is a table of values for programming auto phase power down and limit switch polarity:

"n" Value	Auto Power Down	Enable	Limit Switch (active)
0	Y	Low	Low Input
1	N	Low	Low Input
2	Y	High	Low Input
3	N	High	Low Input
4	Y	Low	High Input
5	N	Low	High Input
6	Y	High	High Input
7	N	High	High Input
8	Y	*	Low Input
9	N	*	Low Input
12	Y	*	High Input
13	N	*	High Input

\*Programmed current mode is used in conjunction with the "Y" command to individually program motor run and hold currents.

*Note: This command is only valid when the MCB is interfaced to an AMS Driver.*

<b>F</b>	<b>Command</b>		<b>Function</b>	<b>Type</b>	<b>NV Bytes</b>
			Find Home	Immediate, Program, Default	2
		<b>Mnemonic</b>	<b>Data 1</b>	<b>Data 2</b>	<b>Result</b>
		(Name) F (n,d)	SPS (18-23,000)	Direction (0-1)	None

**F (Find Home)**

The special Find Home algorithm is intended to eliminate mechanical hysteresis typically found in many switches and encoders. 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.

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; 18 to 23,000 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	Direction 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

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>G</b>	Execute Program	Immediate, Program		3
	<i>Mnemonic</i> (Name) G (a,t)	<i>Data 1</i> 0-2048	<i>Data 2</i> None	<i>Result</i> None

**G (Go)**

The Go command executes a User programmed sequence starting at a predefined address location. Although most programs will start at "0," the User can start at another address. That address however, MUST begin at a stored instruction address. The Trace option is useful in debugging single axis programs. If trace is a 1, 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. Address locations between 225 and 255 are reserved for parameter storage and may not be used in programs.

There is a special case for the Go instruction. If the address is specified as 2048 (above the last non-volatile address), the control signal will 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 at address 256, with 16 character (byte) intervals. This instruction is analogous to "on PORT go to."

Input Port State				Address Location
P1	P2	P3	P4	
1	1	1	1	256
0	1	1	1	272
1	0	1	1	288
0	0	1	1	304
1	1	0	1	320
0	1	0	1	336
1	0	0	1	352
0	0	0	1	368
1	1	1	0	384
0	1	1	0	400
1	0	1	0	416
0	0	1	0	432
1	1	0	0	448
0	1	0	0	464
1	0	0	0	480
0	0	0	0	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. Reference: BRANCH.SMC program structure on the EASI diskette.

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
<b>H</b>	Step Resolution		Immediate, Program, Default	2
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) H (n)	Full/Half (0-2)	None	None

**H (Step Resolution)**

This command selects step size resolution. To select a half step sequence enter H 1. To select full step sequence enter H 0.

Any User sequence in non-volatile memory is left intact; however a hardware reset or clear command (C1) MUST be used to access the User values. When changing between full and half step, position is maintained to the nearest full step. Default = 0 (full step).

H 16

Earlier versions of the SMC-C24 (microprocessor) used on the MCB had an 11 Mhz clock and the value used in the “W” (Wait) command was correct. New devices are specified at 14.7 Mhz and the calibration is off by 4/5. Normally, the designer has compensated for this error by adjusting the data value. Software V2.07 provides for calibration to 10 Ms, while being backward compatible with earlier versions (8 Ms).

Setting calibration on:

1. Enter the command “H 16”. This sets only a calibrate flag and does not change other “H” modes (0-15). H (0 thru 15) does not affect the calibrate flag.
2. Issue the S (save) command. The mode is saved in NV memory.

The CLEAR (“C1”) command will cause the calibrate flag to be reset, and steps 1 and 2 must be repeated.

To determine the calibrate mode, the EXAMINE (X) command may be used. When in dumb terminal mode, a lower case “w” will appear as the first character of the display response.

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
<b>I</b>	Initial Velocity		Immediate, Program	3
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) I (n)	SPS 18-23,000	None	None

**I (Initial Velocity)**

The initial velocity specifies the start and stop speed for the motor. The first velocity of the acceleration ramp is specified by the I parameter. To achieve an optimum value the following variables must be taken into account: motor size (rotor inertia), system inertia, starting torque to overcome friction (stiction), and motor or system resonance’s. Generally values between 50 and 1200 SPS are appropriate.

The initial velocity applies to:

1. All index commands (+, -, R).
2. Start speed used in constant velocity (M).
3. Decelerate to 0 in constant velocity or soft stop.
4. Final phase of home routine.

The Examine (X) command displays velocity information. This parameter is a user default. After setting the value and issuing the Save command, it will be stored for future power-ups. A value of 400 SPS is preset on issuance of the "C8" (clear) command.

<b>J</b>	<b>Function</b> Jump to Address		<b>Type</b> Program	<b>NV Bytes</b> 4
	<b>Mnemonic</b> (Name) J (a,n)	<b>Data 1</b> Address (2047)	<b>Data 2</b> N + 1 Times (0-255)	<b>Result</b> None

**J (Jump To Address)**

Example: 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 may be used only within a program. This instruction may NOT be nested. Only one jump counter is available for use at any given time.

<b>K</b>	<b>Function</b> Ramp Slope		<b>Type</b> Immediate, Program	<b>NV Bytes</b> 2
	<b>Mnemonic</b> (Name) K (n)	<b>Data 1</b> Accel (0-127)	<b>Data 2</b> Decel (128-255)	<b>Result</b> None

**K (Ramp Slope)**

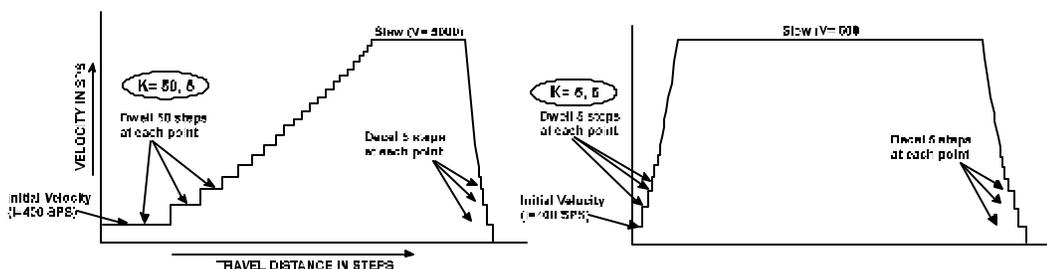
Specify the ramp acceleration and deceleration time.

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.



*Note: Values of "K" for deceleration mirrors the acceleration ramp. If a value of less than 127 is entered for "K", then both ramps assume the same slope. To alter the deceleration ramp, it is necessary to enter values between 128-255.*

Example:

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.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>L</b>	Loop on Port	Program		3
	<i>Mnemonic</i> (Name) L (a,c)	<i>Data 1</i> Address (0-2648)	<i>Data 2</i> Condition (0-9)	<i>Result</i> None

**L (Loop On Port)**

The Loop command will test the specified input port for the required condition . If the port is NOT at the required level, the program will jump to the specified address. If the address is to a previous instruction, the program will loop until it becomes the specified level. The program will then continue to the next step. Input ports are tested as follows:

Condition	Wait For
0	Port 1 Low
1	Port 1 High
2	Port 2 Low
3	Port 2 High
4	Port 3 Low
5	Port 3 High

There is also an additional feature for implementing a "wait until" function. The standard loop tests the condition every 2-3 Ms. If the unique address is 2048, the controller executes a tight loop at this instruction while 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 non-volatile program execution.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>M</b>	Move at Constant Velocity	Immediate, Program		3
	<i>Mnemonic</i> (Name) M (±)	<i>Data 1</i> SPS ± 23,000	<i>Data 2</i> None	<i>Result</i> None

**M (Move At Constant Velocity)**

Move at the specified velocity and direction "forever."

The motor will ramp up or down to a constant step velocity and motion will continue at the given speed until a new velocity is entered. The specified slew speed is in steps per second. Ramp parameters (K command) may be modified prior to each velocity command, allowing different ramp slopes.

The direction is specified by the sign preceding the velocity. Decelerating from full speed in one direction to full acceleration speed in the opposite direction can be accomplished with this single command.

Examples:

M 1000

Move in the + direction at 1000 SPS starting at the initial velocity then accelerating to the 1000 SPS slew speed.

M -4000

Decelerate to initial velocity (from 1000). Stop and change direction. Accelerate to 4000 SPS in the “-” direction.

Motion may be stopped by:

1. The "M 0" command.
2. Soft stop "@" command or interrupt.
3. ABORT (ESC) interrupt (without deceleration).

The default initial velocity is used at the first invocation of the command. Position Trip points may be used. If the encoder feedback option is implemented, stall supervision is functional.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>O</b>	Set Origin	Immediate, Program		4
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) O (±n)	Position (± 8,388,607)	None	None

**O (Set Origin)**

The Set Origin command resets the internal 24 bit position counter to the specified value. Base position for the Relative mode is zero. Signed numbers are used. Hardware reset clears to zero.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>P</b>	Program Mode On/Off	Immediate		0
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) P (a)	Address (256-2047)	None	None, #

**P (Program Mode)**

This Program command allows composition and storage of “program sequences” in non-volatile memory. Program mode allows entering commands for future execution by use of the Go command or External go signal. Existing programs are overwritten as new instructions are stored. Entering a second "P" command will terminate the Program mode, and insert an end of program marker in the stored program before returning to the Immediate Command mode. While in Program mode, commands and data are directed into the non-volatile memory. The address specifies the start point in non-volatile memory where the application program will reside. As instructions are entered, the address counter is updated and displayed. Any number of independent program segments can coexist. These can be accessed via Jump, Loop, Go or other special instructions.

**Special Locations, "Shadow Memory"**

Sixty four bytes of the program are configured as FAST memory. Locations 128 through 192 are downloaded from the external EEprom to internal RAM at power-up.

Instructions executing in this segment run faster than other locations that fetch from relatively slow EEPROM (1 Ms. per byte). Programmers should reserve this segment for time critical code. The shadow RAM is not actually written to non-volatile memory until the Store command is issued. Host computers may download subroutines without concern about wearing out the EEPROM. Locations 256 through 511 are predefined if the "G 2048" command is used in an application.

The Trip command can jump to any location between 0 and 255. Editing of programs should be done WITH CAUTION. The general programming sequence is to (1) start programming at desired address, (2) enter new instructions, and (3) terminate programming with the ESC command. (This will cause a return to the Command mode without inserting the end of program marker).

*Note: Do not attempt to use locations above 1792 for programs (default storage).*

Example program in terminal mode (each command is terminated with Carriage return):

<u>Response</u>	<u>Command</u>	<u>Comment</u>
	P 0	Enter program mode.
0 (echoed)	+5000	Index 5000 steps.
5	-5000	Index -5000 steps.
10	P 0	End program, insert program marker One program has been entered and stored in non-volatile memory. You may now insert another routine.
	P 100	Start program at memory location 100.
100 (echoed)	O	Set origin to zero.
101	R 2000	Move to position 2000.
106	R - 500	Move to position -500.
111	J 101 9	Repeat sequence 10 times.
	P 0	End program. You may now list the programs.
	L 0 5	
	L 100 5	You may now execute the programs.
	G 100	You may now edit the programs.
101	R 3000	Change position from 2000 to 3000.
	P 101	
106	<ESC>	Terminate edit without "end" mark.

While most commands can be part of a program, some may not be necessary or impractical while some should just be avoided.

Command	Reason
B Jog Speeds	User default from non-volatile memory at power-up
C Clear	Never use - , it will erase the programs
D Resolution	User default from non-volatile memory at power-up
E Delay	User default from non-volatile memory at power-up
H Resolution Mode	User default from non-volatile memory at power-up
I Initial Velocity	User default from non-volatile memory at power-up
K Ramp Slope	User default from non-volatile memory at power-up
Q Query	N/A
S Store	No application, non-volatile memory wear
T Trip	User default from non-volatile memory at power-up
V Slew Speed	User default from non-volatile memory at power-up
X Examine	N/A
Y Current Control	User default from non-volatile memory at power-up

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>Q</b>	Query (List) Program	Immediate		0
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) Q (a)	Address (0-2047)	None	Listing

**Q (Query Program)**

This command will produce a list (disassembled) of the instructions stored in non-volatile memory using the format:

"ADDRESS" "INSTRUCTION" "DATA 1" "DATA 2"

1. The values will be displayed only if applicable to the particular instruction type.
2. One instruction will be listed at a time.
3. The space key will advance the address and list more instructions.
4. Listing is terminated when an "end of program" marker is found or the ESC character is received.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>R</b>	Index Relative to Origin	Immediate, Program		4
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) R (dd)	Position ( $\pm 8,388,607$ )	None	None

**R (Index Relative To Origin)**

Move, with ramping, relative to the "0" origin. The target position has a range of  $\pm 8,388,607$  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).

<b>S</b>	<b>Function</b> Store Parameters		<b>Type</b> Immediate, Program		<b>NV Bytes</b> 1
	<b>Mnemonic</b> (Name) S	<b>Data 1</b> None	<b>Data 2</b> None	<b>Result</b> None	

**S (Store Parameters)**

The S (Store) command will force a write sequence to non-volatile memory. This command should be used to save new operational parameters or to down-load motion programs. The S command should not be embedded internally as part of a program but rather used to save the program after loading.

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

Parameter	Standard Defaults
Initial velocity (I)	400 SPS
Slew velocity (V)	5,016 SPS
Divide factor (D)	1
Ramp slope (K)	5
Jog speeds (B)	30, 200
Trip Point (T)	Off
Full/Half (H)	Full
Auto power down (E)	Yes
Power down polarity (E)	High
Quarter step (H)	Off
Hold/Run current (Y)	10, 75
Limit polarity (H)	Low
Auto position readout (Z)	Off
Name (after reset)	Undefined

All of these parameters are saved as a block from the working registers in the SMC-C24. Frequent use of this command should be avoided, as memory longevity may be affected.

*Note: Observe the precaution of writing too frequently to non-volatile memory.*

<b>T</b>	<b>Function</b> Set and Enable Trip Point		<b>Type</b> Default, Program		<b>NV Bytes</b> 4
	<b>Mnemonic</b> (Name) T (n)	<b>Data 1</b> Position ( $\pm 8,388,607$ )	<b>Data 2</b> None	<b>Result</b> None	

**T (Trip Point)**

The MCB has a programmable "Trip Point" feature. During moves the current position is compared to the Trip position at each step. The Trip output will be alternated each time the Trip Point is reached or passed. When in the Program Run mode, a User program located at address 200 will be automatically executed (called). On completion of this sub program, a return to the call point will be 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 (minus zero) will set the Trip Point to 0. The Trip Point is displayed using the examine command. Default = 0 (off).

<b>V</b>	<b>Function</b> Set Final (Slew) Velocity in SPS		<b>Type</b> Default, Immediate, Program		<b>NV Bytes</b> 3
	<b>Mnemonic</b> (Name) V (n)	<b>Data 1</b> SPS (18-23,000)	<b>Data 2</b> None	<b>Result</b> None	

**V (Slew Velocity)**

This command sets final Slew Velocity after ramping up to speed. The final output velocity is divided by the value of "D." This value is independent of constant velocity (ramp completed), jog or home speeds and is used for indexing absolute or relative.

The following commands use this parameter:

- R (Relative index)
- + (Plus index)
- (Minus index)

The following functions do not use or affect this parameter:

- J (Jog)
- F (Find home)

The MCB also allows specifying speeds in step time values. The actual step clock period may be directly controlled, rather than in steps per second. This allows fine control of step rates, especially at lower speeds. The MCB is placed in this mode by using a negative value with the "V" command, and restored to the SPS mode with positive values. Switching between modes will not effect speeds previously specified. In the period mode, all subsequent commands that relate to speed must use period values for data (positive data).

These include:

- V (-65535 to -54) Slew velocity
- I (65535 to 54) Initial velocity
- M (65535 to 54) Constant velocity
- F (65535 to 54) Find home
- B Jog Speeds (minimum obtainable value is 160 SPS)

The data range is between 54 (23,000 SPS) and 65,535 (18.76 SPS).

The formula for determining step rate is:

$$N = 1228800 / SPS$$

As with all speed commands, the output rate is divided by the value specified with the "D" command.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>W</b>	Wait (n) Milliseconds	Immediate, Program		3
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) W (n)	0.0075 Sec. (0-65.535)	None	None

**W (Wait)**

WAIT n milliseconds. (See related command "H16")

The controller will remain in an idle state for the specified amount of time. The Wait command, if issued while indexing (as a result of a R,+,- or F command), will NOT start until all motion is complete. Using this command with 0 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 operations, executed during wait commands, will increase the delay time by as much as 14 times normal value. The result will NOT be available until the delay is complete. High speed step operations, executed during wait commands, will increase the delay time by as much as 14 times normal value. The result will NOT be available until the delay is complete.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>X</b>	Examine Settings	Immediate		0
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) X	None	None	Display Setting

**X (Examine Parameters)**

The Examine command displays the parameter settings. This command will produce two different responses, depending on the mode of Motion command operation. When in the non-Party Line (Single) mode the display is as follows:

(C=cc) K=kk, I=ii, V=vv, (T=tt) nn [Carriage Return, LF]"

Where:

cc = Phase current setting (if enabled and will display Hold or Run value depending on operational mode)

kk = Ramp factor

ii = Initial velocity divided by "D" (calculated)

vv = Slew velocity divided by "D" (calculated)

tt = Trip Point (if enabled)

nn = Name of Party Line

When in the multi-axis (Party Line) mode the data is returned in the following format:

cc kk ii vv tt (LF). It is assumed the host computer will interpret the data string for processing.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>Y</b>	Set Hold and Run Current	Immediate, Program		3
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) Y (h,r)	Hold (0-100)	Run (0-100)	None

**Y (Hold and Run Current)**

This command allows specifying the Hold and Run values of motor current (per phase) between 0 and 100% with a resolution of 1%. To utilize this feature the following commands must have been initiated:

1. The Current Control mode is enabled via a E8 or E12 Enable command.
2. The Y (xx) (xx) data values were entered.
3. A store command was issued (S).

Step 1 and step 2 need only be executed once to establish the Current Control mode, unless the C1 Clear and Restore command was used.

The two values may be saved in non-volatile memory by use of the "S" (Save) command. Current change does not take place until a subsequent "E" command is issued or a motion command is executed.

*Note: This command is only valid when the MCB is interfaced to an AMS driver.*

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>Z</b>	Read and Display Current Position	Immediate, Program		2
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) Z	Readout Mode (0-1)	None	None

**Z (Read Position)**

Read and display the current position. During motor move commands the value will change depending on the direction of travel. The position counter is reset by the "O" (Set Origin) command. The "Z1" command enables a continuous readout, via the serial interface. 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 "on" if a Save command is issued. This mode is only practical using a single axis protocol.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
<b>[</b>	Read NV Memory	Immediate		0
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) [	Address (0-2047)	Number (0-255)	Displayed Values

**[ (Read Non-Volatile Memory)**

The Read Non-Volatile Memory command allows the User to display any byte of the 2047 byte external non-volatile memory. The address specifies the desired location to access.

<b>J</b>	<b>Function</b> Read Limits, Hardware		<b>Type</b> Immediate, Program		<b>NV Bytes</b> 1
	<b>Mnemonic</b> (Name) J	<b>Data 1</b> 0-1	<b>Data 2</b> None		<b>Result</b> None

**J (Read Limits/Hardware)**

This command allows examination of 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
Controller	Lb	La	Hm	*	Qs	Fb	En	Half

Where:

- La= Limit "a" switch
- Lb= Limit "b" switch
- Hm= Home switch (32=low input)
- \*= Always low
- Qs= Quarter step output level (n/a: MCB)
- fb+,fb-= Encoder feedback input level (n/a: MCB)
- En= Enable output level
- Half= Half step output level

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

<b>+</b>	<b>Function</b> Index in the Plus Direction		<b>Type</b> Immediate, Program		<b>NV Bytes</b> 4
	<b>Mnemonic</b> (Name) + (n)	<b>Data 1</b> Steps (0-16,777,215)	<b>Data 2</b> None		<b>Result</b> None

**+ (Index In Plus Direction)**

Step in a positive direction for the specified distance. The motor will ramp up, slew, then ramp down per the previously set parameters. The range is 0 to 16,777,215 steps. The position counter will overflow at 8,388,607. A "+" direction index is defined as a clock-wise rotation of the motor as viewed looking toward the motor mounting flange.

<b>-</b>	<b>Function</b> Index in the Minus Direction		<b>Type</b> Immediate, Program		<b>NV Bytes</b> 4
	<b>Mnemonic</b> (Name) - (n)	<b>Data 1</b> Steps (0-16,777,215)	<b>Data 2</b> None		<b>Result</b> None

**- (Index in Minus Direction)**

Step in a negative direction. Except for the direction, this command behaves exactly as the "+" command above.

A "-" direction index is defined as a counter clock-wise rotation of the motor as viewed looking toward the motor mounting flange.

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

**^ (Read Moving Status)**

The Read Moving Status command is used to determine the current moving and mode status. These status bits are converted to a decimal number (0-255) in Terminal mode.

The status byte contains the current status of the controller as follows:

Bit	Decimal	Status
0	1	Moving - indicates axis moving
1	2	Constant - high in constant velocity
3	8	Homing - homing routine is active
4	16	Slewing - ramping complete

*Note: Other bits in this result should be ignored.*

<b>\</b>	<b>Function</b> Write to Non-Volatile Memory		<b>Type</b> Immediate	<b>NV Bytes</b> 0
	<b>Mnemonic</b> (Name) \ (a,d)	<b>Data 1</b> Address (0-2047)	<b>Data 2</b> Data (0-255)	<b>Result</b> None

**\ (Write To Non-Volatile Memory)**

The Write to Non-Volatile Memory command allows the programmer to directly modify any byte in the memory. The life expectancy of the non-volatile memory may be effected by excessive use of this command.

Note: Non-volatile memory has a finite life of approximately 10 years for data retention and 460,000 write cycles.

**Electrical**

**D.C. Characteristics**

Parameter	Min	Typ	Max	Units
Logic Power (VCC)	4.5..	+5	5.5.	Volts
Current (ICC)	300.			Ma.
Output Signals:				
High Level(Voh)		5..	15*	Volts
Low Level(Vol)			0.3	Volts
Low Current(Iil)			30.	Ma.
Input Signals:				
High Level(Vih).	3.6.		5.2.	Volts
Low Level (Vil)			0.8.	Volts
Low Current (Iil)			2.	Ma.
Optical Inputs:				
LED Current			5.	Ma.
LED Drop.			1.5.	Volts
Internal Series Resistor.			1k.	Ohm
RS-232 Input Level		5	25.	Volts
RS-232 Outputs		9.	12.	Volts
RS-422 Input Level.		5.	25.	Volts
RS-422 Outputs		5	25.	Volts
RS-232/422 Loads.	1.		32.	Axis

\* Open collector with internal pull-up to Vcc, over driven.

**Note: The RS-422 implements drivers rated for EIA RS-485 transmission and limits receiver count to 32 with a maximum cable length of 4,000 feet. RS-232 units are rated for 1 receiver and a maximum cable length of 50 feet.**

**A.C. Characteristics**

Parameter	Description	Min	Typ	Max	Units
Swl	Limit & home switch response.	1.	2		Step Clock

**Non-Volatile Memory**

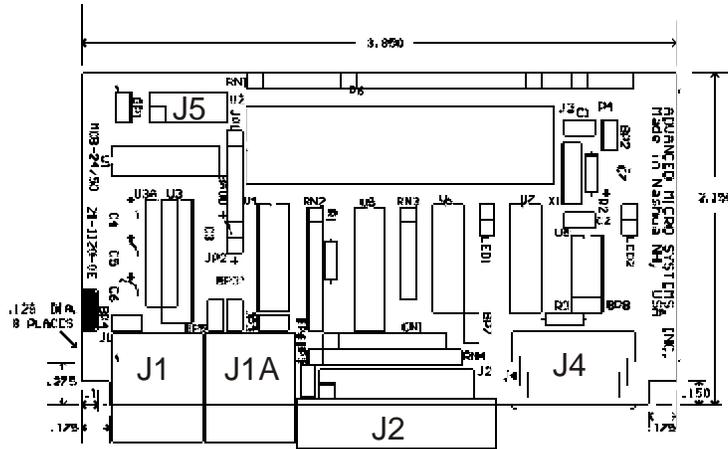
Instruction	Condition	Min	Typ	Max	Units
Fetch & execute cycle	Loop		1.7		Ms
Save parameters	Store.		63		Ms

**Step Clock**

Product	Min	Typ	Max	Units
MCB-24.	18.		23K.	SPS
Clock Period (Low Active)		5.	7.	us
System Clock.			14.745.	MHz
MCB-50	36		50K.	SPS
Clock Period (Low Active)..		5.	7..	us
Systems Clock			29.491.	MHz

**Physical**

Size:.....2.150" W x 3.850" L



- J1..... Serial Input
- J1A..... Serial Output or Terminate
- J2..... Auxiliary I/O
- J4..... Power Supply Input
- J5..... Driver Connection

***Addendum***

**Command Summary**

MNEMONIC / COMMAND	DATA 1	RANGE	DATA 2	RANGE	NV	D	M	P	U
+ INDEX IN "+" DIRECTION	STEPS	0-16777215			4		⊙	⊙	
- INDEX IN "-" DIRECTION	STEPS	0-16777215			4		⊙	⊙	
ESC ABORT/TERMINATE									⊙
@ SOFT STOP					1			⊙	⊙
^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			1				⊙
^ READ MOVING STATUS					1				⊙
A PORT	BINARY	0-129			2			⊙	⊙
B SET JOG SPEEDS	SLOW	0-255	HIGH	0-255	3	⊙		⊙	
C CLEAR AND RESTORE	PAGE	1-8				⊙			
D DIVIDE RESOLUTION	RES.	1-255			2	⊙		⊙	
E ENABLE CONTROL	N VALUE	0,8,12			2	⊙		⊙	
F FIND HOME	SPS	18-23000	DIRECTION	0-1	3		⊙	⊙	
G GO	ADDRESS	0-226,256-2048	TRACE	0-1	3			⊙	
H DECAY, RESOLUTION	FULL/HALF	0-3			2	⊙			
I INITIAL VELOCITY	SPS	18,23000			3	⊙		⊙	
J JUMP	ADDRESS	0-225,2047	N+1 TIMES	0-255	4			⊙	
K RAMP SLOPE	ACCEL	0-127	DECEL	128-255	2	⊙		⊙	
L LOOP ON PORT	ADDRESS	0-2048	CONDITION	0-9	3			⊙	
M MOVE AT CONST. VEL.	SPS	±18-23000			3		⊙	⊙	
O SET ORIGIN		±8388607			4			⊙	⊙
P PROGRAM MODE	ADDRESS	0-226,256-2048							
Q QUERY PROGRAM	ADDRESS	0-2047							
R INDEX TO POSITION	POSITION	±8388607			4		⊙	⊙	
S STORE PARAMETERS					1	⊙			
T TRIP POINT	POSITION	±8388607			4	⊙		⊙	
V SLEW VELOCITY	SPS	18-23000			3	⊙		⊙	
W WAIT, (DELAY)	0.0075 SEC	0-65535			3			⊙	⊙
X EXAMINE PARAMETERS									⊙
Y RUN AND HOLD CURRENT	HOLD	0-100	RUN	0-100	3	⊙		⊙	
Z READ POSITION	CONTINUOUS	0-1			2				⊙

ASCII Command Table

ASCII CHARACTER CODES

\* All products  
 1. SMC-C24  
 2. SMC25, 26  
 3. MAX  
 4. SMC-10  
 E= Eshood

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		

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	_	*

Dec	Hex	Char	Use
32	20		*
33	21	!	E3
34	22	"	
35	23	#	E
36	24	\$	E
37	25	%	
38	26	&	+
39	27	'	
40	28	(	
41	29	)	
42	2A	*	+
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	;	+
60	3C	<	
61	3D	=	
62	3E	>	
63	3F	?	E4

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	+
	^E	05	05	ENO	+
	^F	06	06	ACK	+
	^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	DC4	
	^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	*
	^L	28	1C	FS	
	^M	29	1D	GS	
	^N	30	1E	RS	
	^O	31	1F	US	

**Acceleration Ramp Step Table**

These 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:

10 steps at 400 SPS, then 10 steps at each value 874, 1277, 2831 SPS, then slew at 3000 SPS.

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		
50	8474												